

CENTRO UNIVERSITÁRIO UNIFACVEST
CURSO DE CIÊNCIA DA COMPUTAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO
ANDERSON DI DOMENICO

**SARPA: SISTEMA DE AUTOMAÇÃO RESIDENCIAL COM A
PLATAFORMA ARDUINO**

LAGES

2014

ANDERSON DI DOMENICO

**SARPA: SISTEMA DE AUTOMAÇÃO RESIDENCIAL COM A
PLATAFORMA ARDUINO**

Trabalho apresentado ao Centro Universitário
UNIFACVEST, como parte dos requisitos
para obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Prof. Msc. Márcio José Sembay

Co-orientador: Prof. Msc. João Francisco
Frank Gil

LAGES

2014

ANDERSON DI DOMENICO

**SARPA: SISTEMA DE AUTOMAÇÃO RESIDENCIAL COM A
PLATAFORMA ARDUINO**

Trabalho apresentado ao Centro Universitário
UNIFACVEST, como parte dos requisitos
para obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Prof. Msc. Márcio José Sembay

Co-orientador: Prof. Msc. João Francisco
Frank Gil

Lages, SC ____/____/2014. Nota _____

(data de aprovação)

(assinatura do orientador do trabalho)

(coordenador do curso de graduação, nome e assinatura)

LAGES

2014

AGRADECIMENTOS

Em primeiro lugar a Deus por ter me dado saúde e força para chegar até aqui.

Aos meus pais: Edeno Roque Di Domenico e Ilza de Fatima Padilha Di Domenico; ao meu irmão Robson Carlos Di Domenico; e a minha namorada Josielle Cristina Dias que me deram todo apoio e incentivo.

A todos os professores do curso pelos seus ensinamentos em especial ao professor Márcio José Sembay e João Francisco Frank Gil que me orientaram no desenvolvimento do meu TCC.

SUMÁRIO

| | |
|---|----|
| FOLHA DE APROVAÇÃO..... | 2 |
| AGRADECIMENTOS..... | 3 |
| SUMÁRIO..... | 4 |
| LISTA DE ABREVIATURAS E SIGLAS | 6 |
| LISTA DE FIGURAS | 8 |
| RESUMO | 9 |
| ABSTRACT | 10 |
| I. INTRODUÇÃO | 11 |
| 1.1 Apresentação..... | 11 |
| 1.2 Justificativa..... | 12 |
| 1.3 Importância..... | 13 |
| 1.4 Objetivo do Trabalho | 13 |
| 1.4.1 Objetivo Geral..... | 13 |
| 1.4.2 Objetivo Específico | 13 |
| II. FUNDAMENTAÇÃO TEÓRICA..... | 14 |
| 2.1 História da Domótica | 14 |
| 2.2 História do Arduino | 14 |
| 2.3 Hardware e Software..... | 16 |
| 2.4 Hardware do Arduino..... | 16 |
| 2.4.1 CPU do Arduino..... | 17 |
| 2.4.2 Entradas e Saídas | 18 |
| 2.4.3 Entradas e Saídas Digitais | 18 |
| 2.4.4 Entradas e Saídas Analógicas | 19 |
| 2.4.5 Porta de conexão USB do Arduino | 20 |
| 2.5 Software do Arduino | 21 |
| 2.6 Computação Física..... | 22 |
| 2.7 Shields | 22 |
| 2.7.1 Ethernet Shield..... | 22 |
| 2.7.2 Rele Shield..... | 23 |
| 2.8 Linguagens Web | 24 |
| 2.8.1 HTML..... | 24 |
| 2.8.2 CSS..... | 24 |

| | |
|--|----|
| 2.8.2.1 Bootstrap..... | 24 |
| 2.8.3 Java Script..... | 24 |
| 2.8.4 PHP..... | 25 |
| 2.8.5 Apache..... | 25 |
| 2.9 MySQL..... | 26 |
| 2.9.1 PhpMyAdmin..... | 26 |
| 2.10 Socket..... | 27 |
| 2.11 Acesso Externo..... | 27 |
| III METODOLOGIA..... | 30 |
| 3.1 Estudo de Caso..... | 30 |
| 3.2 Estudos Bibliográficos..... | 30 |
| 3.3 Cronograma..... | 31 |
| 3.4 Estrutura do trabalho..... | 31 |
| IV. PROJETO..... | 33 |
| 4.1 Diagrama de Caso de Uso..... | 34 |
| 4.2 Diagrama de Seqüencia..... | 34 |
| 4.3 Diagrama de Atividades..... | 35 |
| 4.4 Diagrama de Componentes e Implantação..... | 35 |
| 4.5 Utilizações do SARPA..... | 36 |
| V. CONSIDERAÇÕES FINAIS..... | 40 |
| VI. REFERÊNCIAS BIBLIOGRÁFICAS..... | 41 |
| VII. ANEXOS..... | 43 |

LISTA DE ABREVIATURAS E SIGLAS

BSD - *Berkeley Software Distribution*
CGI's - *Common Gateway Interface*
CSS - *Cascading Style Sheets*
DIs - *direct ignition system*
DNS - *Domain Name System*
ES - *Entrada e Saída*
EPROM – *Erasable Programmable Read-Only Memory*
FTDI - *Technology Devices International*
GND – *graduated neutral density filter*
HTML - *Hyper Text Markup Language*
HTTP – *Protocolo padrão de navegação na Web.*
IDE - *Integrated Development Environment*
IP – *Internet Protocol*
JS- *Java Script*
LAN – *Local Area Network*
LED – *Light Emitting Diode*
MAC – *Media Access Control*
MAN – *Metropolitan Area Network*
MPL - *Modulação por Largura de pulso*
NCSA - *National Center for Computer Applications*
PHP – *Personal Home Page ou Hypertext Preprocessor*
PWM - *Pulse-width modulation*
RAM - *Randomic Aritimetic Memory*
Rj45 - *Registered Jack Type 45*
ROM - *Read Only Memory*
S.O - *Sistema Operacional*
SSI - *Server Side Includes*
SSL - *Secured Socket Layer*
TCP - *Transmission Control Protocol*
UDP - *User Datagram Protocol*
USB - *Universal Serial Bus*
WAN - *Wide Area Network*

WEB – *World Wide WEB*

W3C – *World Wide Web Consortium*

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Placa Arduino..... | 17 |
| Figura 2 - Microcontrolador do Arduino..... | 18 |
| Figura 3 - PWM..... | 20 |
| Figura 4 - IDE do Arduino..... | 21 |
| Figura 5 - Ethernet Shield..... | 23 |
| Figura 6 - Rele Shield..... | 23 |
| Figura 7 - phpMyAdmin..... | 27 |
| Figura 9 – Liberação de Portas..... | 28 |
| Figura 9 - Configuração do DNS..... | 29 |
| Figura 10 - Protoboard..... | 32 |
| Figura 11 - LED..... | 33 |
| Figura 12 - Diagrama de Caso de Uso..... | 34 |
| Figura 13 - Diagrama de Seqüência..... | 35 |
| Figura 14 - Diagrama de Atividades..... | 35 |
| Figura 15 - Diagrama de Componentes e Implantação..... | 36 |
| Figura 16 - Tela de Login..... | 36 |
| Figura 17 - Tela Inicial..... | 37 |
| Figura 18 - Tela de Configuração de Horários..... | 38 |
| Figura 19 - Tela de Acionamento de Tomadas..... | 38 |
| Figura 20 - Pagina de Usuários..... | 39 |
| Figura 21 – Maquete..... | 39 |

RESUMO

Este trabalho apresenta um sistema de automação residencial nas linguagens de programação: PHP, C, Java Script, HTML e CSS, desenvolvida com a plataforma de computação física e de fonte aberta arduino na versão uno, juntamente com seu modulo ethernet shield, que permite a conexão da placa arduino com a internet, o objetivo é possibilitar com que as pessoas possam controlar e programar tomadas elétricas de sua residência de qualquer computador ou celular com acesso a internet, dispondo ao usuário um maior conforto e segurança para sua residência, e com um menor custo.

Palavras chaves: Linguagens de Programação, Arduino, Ethernet Shield.

ABSTRACT

This paper presents a home automation system programming languages: PHP, C, Java Script, HTML and CSS, developed with the physical computing platform and open-source Arduino in one version, along with your ethernet module shield, which allows connection arduino board with the internet, the goal is to enable people to be able to control and schedule electrical outlets of your home from any computer or mobile phone with internet access, offering the user greater comfort and security to your home, and with a lower cost.

Keywords: Programming Languages, Arduino, Ethernet Shield.

I. INTRODUÇÃO

1.1 Apresentação

A capacidade de acessar aparelhos elétricos da residência através da internet traz maior comodidade, segurança e economia de energia elétrica para o morador, com esta tecnologia é possível automatizar ações efetuadas no dia-dia, consultar os estados das tomadas e interruptores antes ou após ter saído de sua residência, programar horários para ligar e desligar as tomadas elétricas ou lâmpadas, ou até mesmo receber mensagens de alerta quando os sensores de movimento forem acionados.

“Cada tecnologia traz acoplado um novo vocabulário. Quando o assunto é residência inteligente, não é diferente: casa automática, casa inteligente, automação residencial, retrofitting, domótica, etc. Mas tudo pode ser resumido em uma só palavra: conforto.” (BOLZANI, 2004, p.51).

Com a adição de inteligência em todos os equipamentos espalhados pela casa e não centralizado no PC como acontece atualmente, consegue-se uma transformação radical na vida do morador de uma casa inteligente, pois ela coopera com a sua própria gestão, desafogando o usuário. Imagine uma casa de campo ou de praia onde houvesse a possibilidade de verificação e controle dos equipamentos remotamente. Por exemplo, abrir as janelas para arejar os cômodos e fechá-las mediante a possibilidade de chuva ou executada pelos próprios DIs e atuadores. Estes são atualmente denominados sistemas inteligentes (BOLZANI, 2004, p.39).

Atualmente as pessoas estão cada vez mais ligadas à tecnologia, até pessoas mais velhas estão se adaptando as novas tendências, mas quando o assunto é automação residencial muitos ficam desmotivados, pois o custo para implantação deste tipo de sistema ainda é muito caro, a solução encontrada para este problema é usar a placa de prototipagem arduino, este hardware foi projetado tanto para finalidades educacionais, como para desenvolvimento de projetos de pequenos e grandes portes, sua capacidade de interagir com ambiente externo possibilita o desenvolvimento de diversos tipos de aplicações e por ser uma plataforma de hardware livre e de código aberto, seu valor se torna muito mais acessível.

“Uma residência inteligente contém um sistema para gerenciar todo o tráfego de informações, bem como um controle dos equipamentos, permitindo um maior conforto com um menor gasto de energia.” (BOLZANI, 2004, p.69).

Segundo BANZI (2011), o arduino é a principal ferramenta de prototipagem, onde podemos criar e desenvolver objetos que interagem com outros objetos, pessoas e até mesmo

redes, desenvolvedores estão sempre tentando encontrar formas de produzir protótipos mais simples e com prazos e custos menores.

1.2 Justificativa

As pessoas sempre estão em busca de segurança, conforto e praticidade para suas vidas, formas de aperfeiçoar serviços do dia-dia para ter o menor esforço físico e em menos tempo. Uma casa automatizada trás vários benefícios aos seus moradores, como exemplo as tarefas rotineiras que teriam de ser realizada manualmente, pode ser programadas para serem executadas automaticamente, apenas com um clique de botão.

“Desde os tempos mais remotos o homem vem tentando fazer com que os utensílios e ferramentas substituam-no no trabalho, sendo o seu maior sonho criar um autômato que realize todas as funções operárias.” (ROSÁRIO, 2009, p.15).

O mercado de AUTOMAÇÃO RESIDENCIAL no Brasil aos poucos está adquirindo características muito próximas às de mercados mais evoluídos. A incorporação de um novo profissional, intitulado Integrador de Sistemas Residenciais, define uma situação específica desse mercado: em função das diferentes tecnologias, da sua complexidade de projeto, instalação e programação, ainda não existem soluções plug-and-play, exigindo, assim, uma especialização do profissional que nele atua. (MURATORI,DAL BÓ,2013, p.21).

A implantação de um sistema deste porte exige grandes conhecimentos e um bom planejamento antes, para não haver falhas. Muratori e Dal Bo (2013), afirma que não existe nenhuma solução plug-and-play, segundo Oliveira, Carissimi e Toscani (2010, p.289) “por meio do plug-and-play um usuário pode inserir e remover periféricos do seu sistema sem se preocupar com sua configuração ou interferência com os demais componentes do sistema”, ou seja, antes de fazer a instalação é necessário verificar com o cliente suas necessidades, o que vai lhe beneficiar e se ajustar no seu estilo de vida.

Atualmente já existem vários dispositivos para automação residencial, mas poucas pessoas têm acesso devido ao seu custo ser muito elevado, a solução proposta neste projeto, consiste na utilização da placa de prototipagem arduino, juntamente com seu shield de ethernet, desta forma é possível acionar relés instalados nas tomadas elétricas e motores, através de um browser, onde o usuário poderá controlar remotamente os aparelhos elétricos de qualquer dispositivo conectado a internet, e com um valor bem mais acessível.

1.3 Importância

Pessoas necessitam cada vez mais de praticidade ao fazer tarefas cotidianas, com a correria do dia-a-dia sobra pouco tempo para executá-las, reduzindo seu tempo de lazer e descanso.

Segundo Bolzani (2004), com uma casa automatizada podemos realizar tarefas cotidianas que tomam muito tempo para ser efetuadas de forma mais prazerosa, podendo distribuir melhor o nosso escasso tempo, proteger a residência com sistemas de monitoramento, economizar energia elétrica, entre outras várias possibilidades que uma residência inteligente proporciona ao morador.

1.4 Objetivo do Trabalho

1.4.1 Objetivo Geral

Desenvolvimento de um sistema de automação residencial com acesso realizado através de um navegador, com qualquer dispositivo conectado a uma rede LAN, MAN ou WAN, utilizando a plataforma de prototipagem arduino uno.

1.4.2 Objetivo Específico

O objetivo específico do trabalho é possibilitar que mais pessoas possam automatizar suas residências trazendo mais segurança, praticidade e conforto para suas vidas. As principais funções do sistema são:

- a) Página de configuração de horários para acionar tomadas elétricas;
- b) Página de controle de tomadas elétricas;

Na página de configuração de horários, o usuário do sistema pode determinar um horário para ligar e desligar algum aparelho elétrico, como exemplo: ligar a cafeteira minutos antes de se levantar da cama.

Na página de controle, o usuário pode ligar ou desligar as tomadas elétricas da residência em tempo real com qualquer dispositivo conectado a internet através de um navegador, ao clicar em algum botão relacionado à tomada da residência o arduíno envia um sinal para o relé onde o aparelho elétrico conectado a ele é acionado.

II. FUNDAMENTAÇÃO TEÓRICA

2.1 História da Domótica

Domótica é a junção do latim “Domus” que significa casa com “Robótica”, ou seja, a domótica é o uso da tecnologia para controlar e programar funções de uma casa, onde o controle pode ser realizado localmente ou remotamente.

Segundo Rosário (2009), as primeiras formas de automação foram realizadas em indústrias de processo com o desenvolvimento de equipamentos de medição elétrica e pneumática, mas a palavra automação foi reconhecida somente após o surgimento da máquina de comando numérico em 1949/50, essa máquina tinha a capacidade de realizar operações programadas sem a necessidade de um operador, onde se abriu vários caminhos para grandes mudanças na produção industrial.

No mercado brasileiro, a definição de automação residencial (AR) surgiu com herança de Home Automation utilizado no mercado americano. Isso ocorreu pelo fato de, no Brasil, os primeiros sistemas voltados para automação de residências serem oriundos de fabricas americanos. Na Europa, o termo mais utilizados é Domótica – junção da palavra latina Domus (casa) com robótica (controle automatizado). (MURATORI, 2013, p.39).

De acordo com Moya e Tejedor (2010), a domótica iniciou nos anos 70, com o surgimento de dispositivos desenvolvidos para automatizar edifícios. Nos anos seguintes cresceu o interesse pela busca da "casa perfeita", onde foram realizados testes em eletrodomésticos e dispositivos automáticos para o lar.

Atualmente a domótica pode ser instalada em residências, controlando tarefas e funções através de um software, porém este recurso ainda não é acessível a todas as pessoas devido ao custo muito elevado.

2.2 História do Arduino

O arduino é uma tecnologia de hardware livre e de código aberto, que possibilita o desenvolvimento de projetos interativos de maneira simples, esta placa inicialmente foi desenvolvida para fins educacionais, atualmente é utilizado em projetos de pequenos e grandes portes.

Segundo Evans, Noble e Hochenbaum (2013), O Arduino deu inicio em 2005 no Interection Desgin Institute na cidade de Ivrea, Italia, Massimo Banzi era um dos professores da instituição e estava decidido em achar um meio barato para facilitar o aprendizado em eletrônica dos alunos de design, discutiu seu problema com David Cuartielles onde fecharam

um acordo, pois Cuartielles estava com um problema parecido, o projeto foi iniciado, Banzi e Cuartielles desenvolveram um microcontrolador barato e de fácil utilização, Cuartielles e o engenheiro Gianluca Martino contrato por Massimo desenharam a placa onde um aluno de Massimo programou o software executado no arduino. Gianluca concordou em desenvolver 200 placas de inicio.

O nome arduino foi dado em homenagem a um bar onde os alunos da instituição passavam suas horas vagas, esta placa foi produzida para fins educacionais, para que os alunos pudessem apreender eletrônica mais facilmente e fazer seu próprios projetos com um menor custo, mas o sucesso foi muito grande, pessoas optaram em usar o arduino para desenvolver sistemas de pequeno e grande porte.

As placas eram vendidas em forma de kit para que os alunos fizessem seus próprios projetos. A tiragem inicial foi rapidamente vendida, e mais unidades foram produzidas para manter a demanda. Designers e artistas de outras áreas ouviram falar do Arduino e quiserem usá-lo em seus projetos. Sua popularidade cresceu rapidamente quando o grande público percebeu que o Arduino era um sistema de fácil utilização, de baixo custo e que poderia ser usado em seus próprios projetos, bem como era uma excelente introdução para programação de microcontroladores. O projeto original foi melhorado e novas versões foram introduzidas. As vendas dos Arduinos oficiais alcançaram agora a marca de 300 mil unidades, e eles são vendidos em todo o mundo por intermédio de uma série de distribuidores. (EVANS; NOBLE; HOCHENBAUM, 2013, p.25).

Atualmente existe varias versões do arduino no mercado, algumas delas são:

- a) Arduino Uno, possui um microcontrolador ATmega328 com 14 pinos, onde 6 podem ser utilizados como PWM e 6 analógicas. Esta placa foi utilizada para o desenvolvimento do projeto abordado neste trabalho, devido sua compatibilidade com os Shields disponíveis no mercado e a facilidade de encontrar material teórico;
- b) Arduino Duemilanove possui um microcontrolador ATmega168 com 14 pinos, onde 6 podem ser utilizados como PWM e 6 analógicas, é similar ao Arduino Uno em termos de compatibilidade;
- c) Arduino Mega 2560 possui um microcontrolador ATmega2560 com 54 pinos digitas, onde 14 podem ser utilizadas como PWM e 16 como analógicas, ideal para projetos maiores.
- d) Arduino Mega ADK, suas configurações são parecidas com a do Arduino Mega 2560, porem é utilizado no desenvolvimento de projetos com o uso da plataforma Android.

- e) Arduino Leonardo possui um microcontrolador ATmega 32u4 com 20 pinos digitais, onde 7 podem ser utilizadas como PWM e 12 como analógicas, já vem com a comunicação USB embutida no microcontrolador;

2.3 Hardware e Software

Hardware é toda parte física, formada por todos os componentes eletrônicos, ou seja, o próprio Arduino, ou seus Shields.

Software é toda parte lógica, é o programa que vai fornecer as instruções ao hardware, ou seja, é uma seqüência de instruções a serem seguidas que vai ser interpretada por um processador.

“Princípio de Equivalência de Hardware e software: Qualquer coisa que possa ser feita com software pode ser feita com hardware, e qualquer coisa que possa ser feita com hardware também pode ser feita com software.” (NULL; LOBUR, 2006, p.37).

2.4 Hardware do Arduino

O arduino é uma plataforma física da computação de hardware aberto, baseada em uma placa microcontroladora com entradas e saídas (E/S), sua placa possui um microcontrolador Atmel AVR de 8 bits, entradas analógicas e saídas digitais, pinos de alimentação 5v - 3,3v e terra (GND) 0v, cristal oscilador 16 MHz, porta de conexão USB, fonte de alimentação e um botão de reset.

Segundo Mcroberts (2011), a placa Arduino é desenvolvida com um microprocessador Atmel AVR, um cristal ou oscilador (relógio simples que envia pulsos de tempo em uma frequência especificada, para permitir sua operação na velocidade correta) e um regulador linear de 5 volts. Isso depende muito do modelo utilizado, A placa tem uma saída que permite conectá-lo a um PC ou MAC para upload ou recuperação dos dados. É possível conectar outros circuitos ou sensores nos pinos de entrada e saída do microcontrolador localizada na parte de cima da placa.

“O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou ate mesmo a Internet para recuperar e enviar dados do Arduino e atuar sobre eles.” (MCROBERTS, 2011, p.23).



Figura 1 - Placa Arduino

Fonte: < http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg>

2.4.1 CPU do Arduino

O microcontrolador ATMEL é a CPU do arduino sendo responsável por todo o processamento, é considerado um computador completo, com memória RAM, memória de programa (ROM), dispositivos de entrada e saída, mas diferente de computadores desktop ou notebook que além do sistema operacional (S.O), a muitos outros programas que são indispensáveis para ele, como o antivírus, isso reduz seu processamento e tempo, pois nunca se sabe quando haverá uma interrupção no sistema.

O microcontrolador é o hardware responsável para obter dados externos, processar esses dados e devolver para o mundo externo.

Na realidade, um microcontrolador é um pequeno computador dentro de um chip. Ele tem tudo o que havia nos primeiros computadores domésticos e ainda outras coisas. Ele contém um processador, um ou dois quilobytes de memória RAM* para guardar dados, uns poucos quilobytes de memória EPROM**(memória flash) para armazenar os programas, e ainda pinos de entrada e saída. Esses pinos de entrada/saída (ES) ligam o microcontrolador aos de mais componentes eletrônicos. (MONK,2013, p.6).

Existem placas Arduino oficiais com vários tipos de microcontroladores, todos eles da empresa ATMEL. Os mais usados são: ATmega8, ATmega162 e ATmega328p.

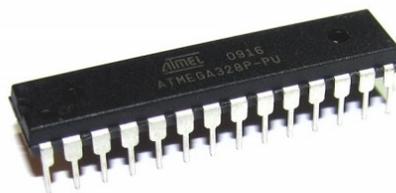


Figura 2 – Microcontrolador do Arduino

Fonte: <http://www.revista.espiritolivre.org/wp-content/uploads/2013/03/19-03-2013_php-logo.jpg>

2.4.2 Entradas e Saídas

O Arduino Uno como exemplo, possui 14 pinos numerados de 0 a 13, que podem ser configurados como entrada e saídas digitais dependendo da finalidade de seu projeto e também possui seis entradas analógicas que vai de A0 até A5.

Em termos práticos, um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software. (MCROBERTS, 2011, p.22).

Por padrão os pinos do arduino são configurados como entrada digital, para mudar este padrão, tem que ser especificado no esboço (sketch) do arduino o numero do pino e se vai ser do tipo entrada (input), para entrada de dados ou saída (output), para acionar dispositivos externos.

2.4.3 Entradas e Saídas Digitais

Entradas digitais têm apenas dois estados possíveis, 0 e 1. As entradas digitais permitem ler diversos tipos de sensores, já suas saídas pode ser utilizada para controlar desde um simples LED, ou até mesmo aparelhos ligados a rede elétrica, dispondo de um circuito adequado para a finalidade do projeto que esta sendo desenvolvido.

Evans, Noble e Hochenbaum (2013), afirmam que no mundo digital existem somente dois estados, ligado ou desligado, como exemplo um LED, ele vai acender quando seu estado

for 1 e apagar quando for 0. O pino do arduino funciona com 0 v e 5 v, sendo 0 v para desligado e 5 v para ligado.

2.4.4 Entradas e Saídas Analógicas

Um exemplo de usar o pino analógico é utilizando um potenciômetro que nada mais é do que um resistor onde é possível regular a voltagem manualmente. O potenciômetro tem três terminais e se conectarmos todos eles, se torna um divisor de tensão, dependendo de sua aplicação ao ajustar o botão uma voltagem de 0 a 5 V será aplicada ao pino central, podendo ler o valor dessa voltagem no pino analógico.

Segundo Mcroberts (2011), O arduino uno possui seis entradas e saídas analógicas, onde é feita a conversão do sinal analógico para o digital com um conversor de 10 bits integrado na placa. Com este conversor é possível ler números inteiros entre 0 (0 V) e 1.023 (5 V).

Para enviar um valor analógico para um pino digital podemos utilizar os pinos 3, 5, 6, 9, 10 e 11 que se notarmos tem especificado a sigla PWM ao lado. Mcroberts (2011), afirma que os pinos analógicos são diferentes dos pinos digitais, pois com eles é possível enviar um sinal PWM, A sigla PWM significa Pulse Width Modulation, ou Modulação por Largura de Pulso (MLP), com esta técnica é possível conseguir resultados analógicos através de sinais digitais. Esta técnica funciona alterando o tempo em que a saída de um pino permanece ligada ou desligada.

A imagem representa representada pela figura 3 mostra o ciclo de trabalho de um sinal usando a função do arduino analogWrite(), onde são passados diferentes valores em seu parâmetro.

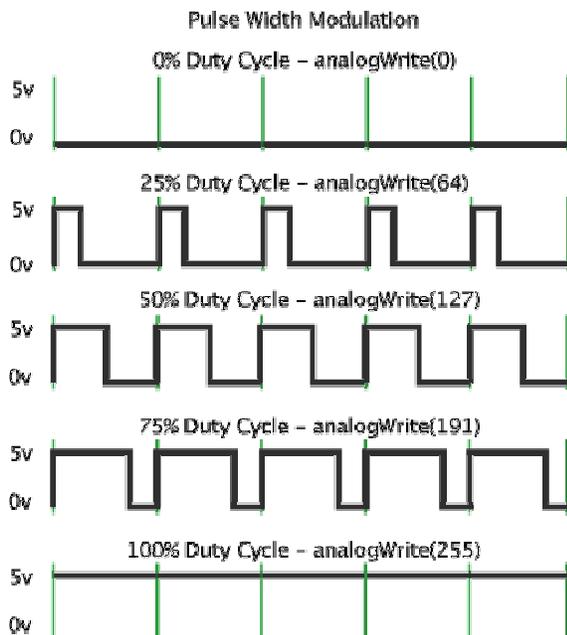


Figura 3 – PWM

Fonte: < <http://arduino.cc/en/uploads/Tutorial/pwm.gif>>

2.4.5 Porta de conexão USB do Arduino

A porta Serial USB possibilita o arduino se comunicar através de um cabo com um computador ou outro dispositivo que tenha interface USB. Por meio desta comunicação é possível enviar os esboços (sketch) do computador para serem gravadas no microcontrolador do arduino, e através dela também o arduino recebe 5 v diretamente da fonte de alimentação do computador.

A mais recente placa do Arduino, a Uno, difere das versões prévias por não utilizar o chip FTDI, que conduz a USB para a serial. Em vez disso, ela utiliza um Atmega8U2, programado como um conversor USB para serial. Isso confere à placa muitas vantagens quando comparada à seu predecessor, a Duemilanove. Primeiro, o chip Atmega é muito mais barato que o chip FTDI, diminuindo o preço das placas. Segundo, e mais importante, ele permite que o chip USB tenha seu firmware atualizado, para que o Arduino seja exibido em seu PC como outro dispositivo, tal como um mouse ou joystick de jogos. Isso abre uma série de novas possibilidades para o Arduino. Infelizmente, a mudança para esse tipo de novo chip USB tornou muito mais difícil para fabricantes de clones criarem clones do Arduino Uno. (MCROBERTS, 2011, p.24).

FTDI significa Future Technology Devices International é uma empresa que desenvolve tecnologias USB, o chip FTDI é um conversor USB para serial utilizado em algumas versões do arduino, Banzi (2012) afirma que o arduino “pode ser programado usando um cabo USB, sem necessidade de porta serial. Esse recurso é útil, uma vez que muitos computadores modernos não têm portas seriais.

2.5 Software do Arduino

A IDE do arduino é uma aplicação multiplataforma, ou seja, é compatível com qualquer sistema operacional, foi desenvolvida na linguagem de programação Java e baseado em Processing, avr-gcc e outros softwares open source.

Para programar o Arduino (fazer com que ele faça o que você deseja) você utiliza o IDE do Arduino, um software livre no qual você escreve o código na linguagem que o Arduino compreende (baseada na linguagem C). O IDE permite que você escreva um programa de computador, que é um conjunto de instruções passo a passo, das quais você faz o upload para o Arduino. Seu Arduino, então, executará essas instruções, interagindo com o que estiver conectado a ele. No mundo do Arduino, programas são conhecidos como sketches (rascunho, ou esboço). (MCROBERTS, 2011, p.24).

Segundo Warren, Adams e Molle, (2011), o arduino veio com uma linguagem fácil de aprender, ótimo para iniciantes, a linguagem utilizada é derivada do C ++. Esta linguagem já vem com varias funções complexas prontas e muito mais fáceis para ser implementado no código.

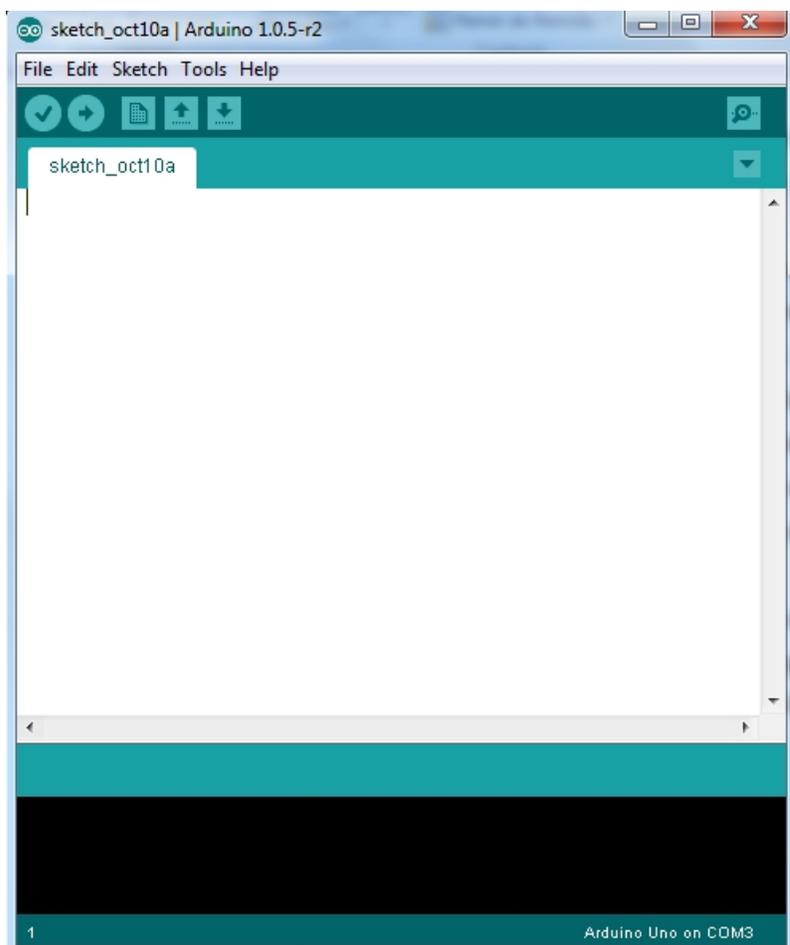


Figura 4 – IDE do Arduino
Fonte: Próprio Autor

2.6 Computação Física

A computação física é uso da eletrônica juntamente com a computação para desenvolver novas matérias e produtos capazes de interagir com o ambiente através de sensores de temperatura, sensores de movimento, sensores de distancia, LEDs entre outros.

Banzi (2011) descreve a computação física como objetos interativos que podem se comunicar com humanos, essa interação é realizada através de sensores e atuadores que são controlados por um software executado dentro de um microcontrolador.

2.7 Shields

Shields significa escudo, como o próprio nome diz, é um modulo que vai na parte de cima do Arduino como se fosse um escudo, serve para estender a capacidade do Arduino. Há diferentes Shields desenvolvidos para o Arduino, cada um com funções e finalidades diferentes facilitando o desenvolvimento de projetos.

Evans, Noble e Hochenbaum (2013), afirma que quase todos os Shields seguem o mesmo padrão de desenvolvimento do kit de ferramentas original, ou seja, são placas fáceis de montar e bem amigáveis para se trabalhar.

Para o desenvolvimento do projeto abordado neste trabalho foram utilizadas as placas Ethernet Shield e Rele Shield, que serão abordados nos próximos capítulos.

2.7.1 Ethernet Shield

A função do Ethernet shield é conectar o Arduino à internet ou em uma rede ethernet local, usando um cabo Rj45. Ele é baseado no chip Wiznet W5100 que fornece uma biblioteca de rede IP, que suporta tanto TCP como UDP.

A capacidade de conectar seu Arduino a uma rede ou a Internet torna possível muitos projetos diferentes. Você pode enviar dados para sites, como atualizações no Twitter. Também poderá controlar o Arduino pela Internet, ou utilizá-lo como um servidor web para disponibilizar paginas simples contendo dados de sensores, e assim por diante. (MCROBERTS, 2011, p.385).

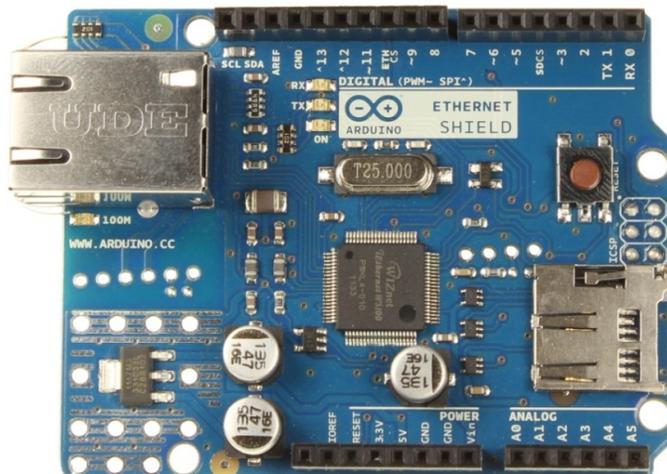


Figura 5 – Ethernet Shield

Fonte: < http://arduino.cc/en/uploads/Main/ArduinoEthernetShield_R3_Front.jpg>

2.7.2 Rele Shield

Este shield serve para controlar tensões de 110 v ou 220 v, como lâmpadas ou equipamentos ligados a rede elétrica, sendo acionados com pequenos sinais de 5 v enviados pelo Arduino.

Para Petruzella (2014), O relé é um dispositivo eletromecânico, que pode ser usado para controlar um circuito de alta tensão com outro de baixa tensão. Os contatos do relé são eletricamente isolados isso traz mais segurança para operador ao manipular o dispositivo. Um exemplo que pode ser citado é ligar ou desligar uma lâmpada partir de um sinal de 12 v a operação da chave energizaria ou desenenergizaria a bobina. Isso, por sua vez, fecharia ou abriria os contatos do relé para ligar e desligar a lâmpada

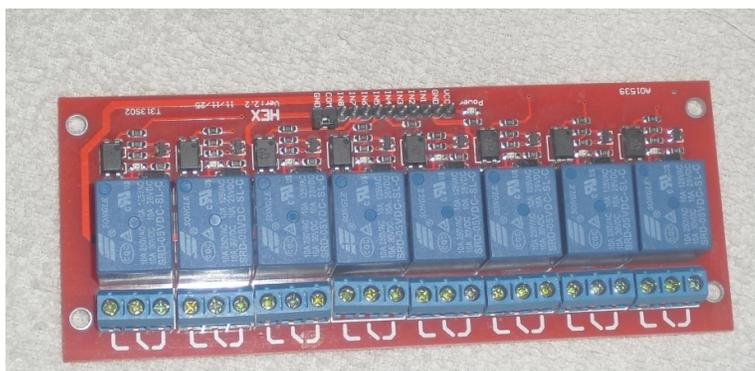


Figura 6 – Rele Shield

Fonte: Próprio Autor

2.8 Linguagens Web

Para construções de sites, sistemas web, é necessário conhecer linguagens de programação para web, há vários tipos de linguagens e cada uma delas tem suas peculiaridades. Nos próximos capítulos serão apresentadas as linguagens e tecnologias web utilizadas para o desenvolvimento do sistema abordado neste trabalho.

2.8.1 HTML

HTML é a sigla de Hyper Text Markup Language (Linguagem de Marcação de hiper Texto), esta linguagem é utilizada para criação de paginas Web.

“Um documento HTML é uma arquivo de texto contento tags de marcação que informa ao browser como exibir a pagina.” (SILVEIRA; PRATES, 2001, p.5).

2.8.2 CSS

Cascadingn Style Sheets é uma linguagem de estilo, trabalham junto com linguagens de Marcão de texto como XML, HTML ou XHTML, tem como finalidade definir estilos as páginas web, ou seja, como os elementos de marcação serão apresentados no navegador.

Meyer (2011), afirma que o CSS esta em todos os lugares dos navegadores WEB, e esta sempre em constante crescimento sem demonstrar qualquer sinal de enfraquecimento, à medida que o usuário da linguagem aumenta, aparece novas soluções para facilitar o desenvolvimento de layouts para WEB.

2.8.2.1 Bootstrap

O Bootstrap é um framework CSS e JS que ajuda no desenvolvimento de layouts das paginas HTML, dentro dos arquivos do bootstrap existem varias classes CSS e funções Java Script que podem ser escolhidas pelos desenvolvedores e inclusas no HTML, sem se preocupar com conflitos, pois este framework segue a risca os padrões WEB da W3C.

2.8.3 Java Script

O Java Script é um é uma linguagem de programação que roda no navegador, tem a função de interagir com o usuário, fazendo validações de campos, alterando conteúdo do documento ou até criar elementos HTML.

Atualmente, falar de Java Script tornou-se um assunto de escopo muito limitado. A linguagem foi criada em 1995 pela Netscape, com o objetivo de permitir a manipulação dinâmica dos elementos de uma página HTML que fosse apresentado no seu navegador, o Netscape Navigator. (DAMIANI, 2006, p.5).

Apesar de ser uma das linguagens mais populares da WEB, ainda a muitas incompatibilidades por parte dos navegadores, muitas vezes ao escrever um código, o mesmo não funciona em certos browsers, tendo que buscar outra solução que seja compatível com todos os navegadores.

2.8.4 PHP

O PHP é uma linguagem utilizada para desenvolver paginas web gerenciáveis com suporte a banco de dados, o PHP poder considerada uma linguagem segura, pois os dados são processados no lado do servidor, ou seja, o usuário final poderá visualizar apenas linguagens do lado cliente como HTML, Java Script ou CSS.

PHP (Hiptertext PreProcessor) é uma linguagem de programação utilizada para criar sites dinâmicos. Esse sites são os que permitem uma interação com o usuário mediante formulários, parâmetros de URL, links etc., e que retornam paginas criadas em tempo real. Aplicações em PHP são geradas em excelentes performance, e automaticamente, pelo servidor. O usuário não tem acesso ao código PHP, somente ao HTML. Esse tipo de detalhe é muito importante quando estamos lidando, por exemplo, com senhas e acesso a banco de dados. (MORAZ, 2005, p.14).

A linguagem PHP encontra-se na versão 5, mas já suporta orientação a objetos desde a sua versão 3, desta forma podemos reaproveitar códigos e agilizar o processo de desenvolvimento de sites e sistemas para web e com maior segurança, outra a vantagem é poder trabalhar com banco de dados, podendo assim ter um controle total sobre os dados inseridos.

2.8.5 Apache

O Apache é um servidor open source utilizado para interpretar linguagens de programação web, sua função é ler os códigos da linguagem lado servidor e converter para linguagem lado cliente para ser interpretado pelo browser.

A história do Apache e de seu desenvolvimento começa em 1995 quando a NCSA (National Center for Computer Applications) criou o antigo NCSA Web Server, que naqueles tempos tornou-se o mais popular servidor http existente. Contudo, a NCSA não desenvolveu muito o projeto, tornando-o até mesmo estagnado. Uma parte da equipe dos desenvolvedores não gostou muito desta atitude e desligou a NCSA e, por conta própria, juntamente com um grupo de desenvolvedores começou a criar uma série de inovações/ patches em cima do código básico do Web Server da NCSA e assim nasceu o Apache (derivação de Apatchy, um trocadilho devido ao enorme número de patches que foram criados pela equipe), que teve seu debut em abril de 1995 sob a versão 0.62. (MARCELO, 2005, p.3).

Marcelo (2005), ainda cita algumas vantagens do Apache. São elas:

- a) Suporte a HTTP 1.1 para criação de virtual hosts baseados em DNS;

- b) Suporte a Secured Socket Layer (SSL) para transações seguras;
- c) Suporte a CGI's, Perl e PHP;
- d) Suporte a autenticação baseada em HTTP;
- e) Suporte a Server Side Includes (SSI);
- f) Suporte a Servelets Java;
- g) Logs customizáveis;
- h) Configuração rápida e simples.

2.9 MySQL

O MySQL é um banco de dados gratuito para uso não comercial, compatível com varias linguagens de programação e de fácil instalação.

Segundo Jobstraibizer (2010), o MySQL suporta vários tipos de sistemas operacionais disponíveis no mercado, são eles: Windows, Linux, FreeBSD (e outros sistemas BSD), Solaris e até mesmo MAC OS.

Jobstraibizer (2010), ainda cita alguns dos recursos mais importantes do MySQL:

- a) Suporte completo para integridade referencial e chaves estrangeiras.
- b) Triggers, Views, Cursors e Stored Procedures.
- c) Sub selects (consultas secundarias realizadas dentro de uma consulta principal).
- d) Compatibilidade total com diversas linguagens de programação, como Java, C, Python, Perl, PHP, ASP, Delphi, Ruby e outras.
- e) Grande número de ferramentas gráficas que podem ser utilizadas para auxiliar a construção de queries e manutenção no banco de dados.
- f) Portabilidade entre diversos sistemas operacionais, além de ser open source.
- g) Não exige grandes recursos de hardware para ter um excelente desempenho.

2.9.1 PhpMyAdmin

É um programa Web desenvolvido na linguagem PHP, serve para gerenciar o MySQL, possui uma interface amigável, nela é possível criar, deletar, inserir banco de dados, tabelas e muitas outras opções pela parte gráfica, ou se preferir via código SQL.

Jobstraibizer (2010) destaca que o phpMyAdmin é conhecido por 10 entre 10 usuários de MySQL. Os desenvolvedores preferem usar este programa pelo fato de seu acesso ser via browser, ou seja, pode ser acessado de qualquer lugar do mundo.

O phpMyAdmin facilita o trabalho do programador, com este aplicativo pode ser criada todas as tabelas do banco de dados necessárias para o desenvolvimento do sistema usando a parte gráfica, agilizando a entrega do projeto.

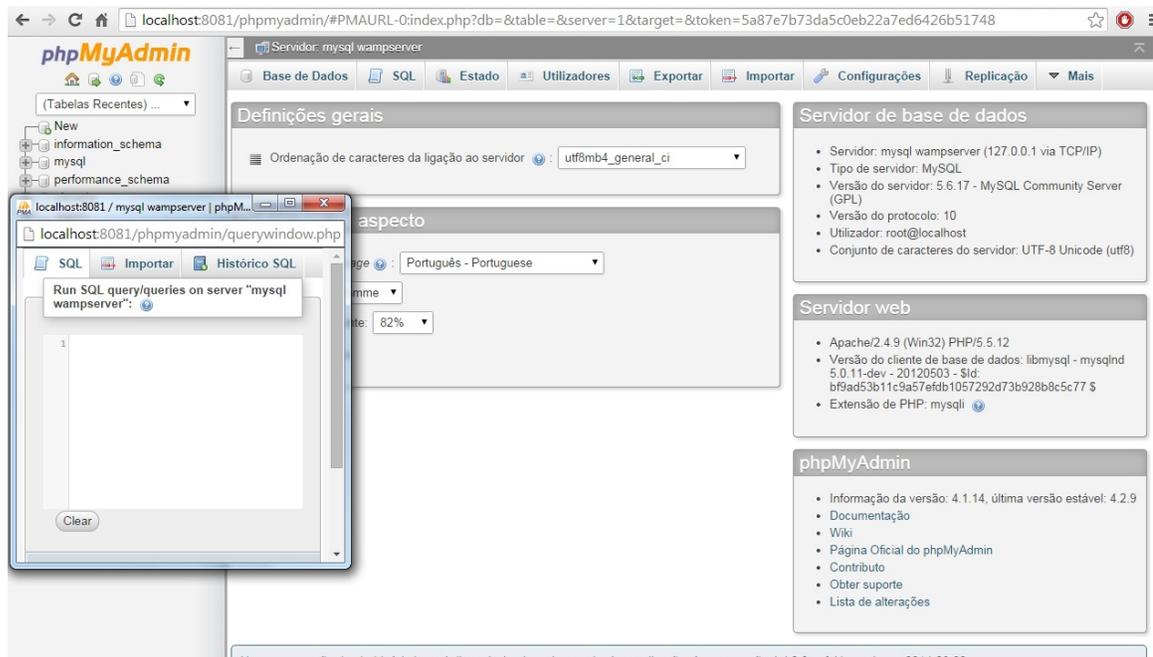


Figura 7 – phpMyAdmin
Fonte: Próprio Autor

2.10 Socket

Para realizar a conexão via socket do arduino com o servidor PHP, é necessário informar o IP e a porta que será realizada a comunicação, depois de realizada a comunicação é possível enviar e receber dados, podendo manipulá-los conforme as necessidades do projeto.

Para Franca (2005), socket são canais onde é possível realizar a comunicação de dois scripts independentes na rede.

Um socket é definido como uma extremidade para comunicação. Um par de processos comunicando por uma rede emprega um par de sockets – um para cada processo. Um socket é identificado por um endereço IP conectado a um número de porta. (Por exemplo, a representação de socket para um host no endereço IP 121.10.14.56 executado na porta 15 é 121.10.14.56:15). (SILBERSHATZ; GALVIN; GAGNE, 2008, p.83).

2.11 Acesso Externo

O acesso externo do sistema, somente é possível se a porta especificada no arduino for liberada no roteador, desta forma ao passar o endereço de IP externo no parâmetro da função

socket do PHP, é possível realizar a comunicação do servidor PHP com o arduino de qualquer lugar do mundo, desde que esteja conectado à internet.

Para liberar a porta definida para o arduino é necessário acessar o firmware do modem digitando o ip definido para ele na url, no caso deste projeto o modem utilizado é o D-Link DSL - 2640B, para configurá-lo deve se seguir os seguintes passos:

1. Digite o endereço IP 10.1.1.1 na url do navegador, esse endereço varia de acordo com o modelo e marca do equipamento;
2. Acessar o firmware do modem digitando o usuário e senha, para o modelo utilizado neste projeto o padrão é:
 - a. Usuário: admin;
 - b. Senha: admin;
3. Depois que estiver logado, no menu localizado na lateral esquerda clique em, Advance Setup, depois em Nat e em seguida Virtual Servers;
4. Dentro da opção Virtual Servers:
 - a) Clique no botão Add;
 - b) Selecione a opção Custom Server, no campo de texto de um nome para o servidor;
 - c) Na opção Server IP Address, digite o IP definido para o arduino;
 - d) Em External Port Start, External Port End, Internal Port Start e Internal Port End digite a porta definida para o arduino e no campo de seleção Protocol selecione a opção TCP/UDP;

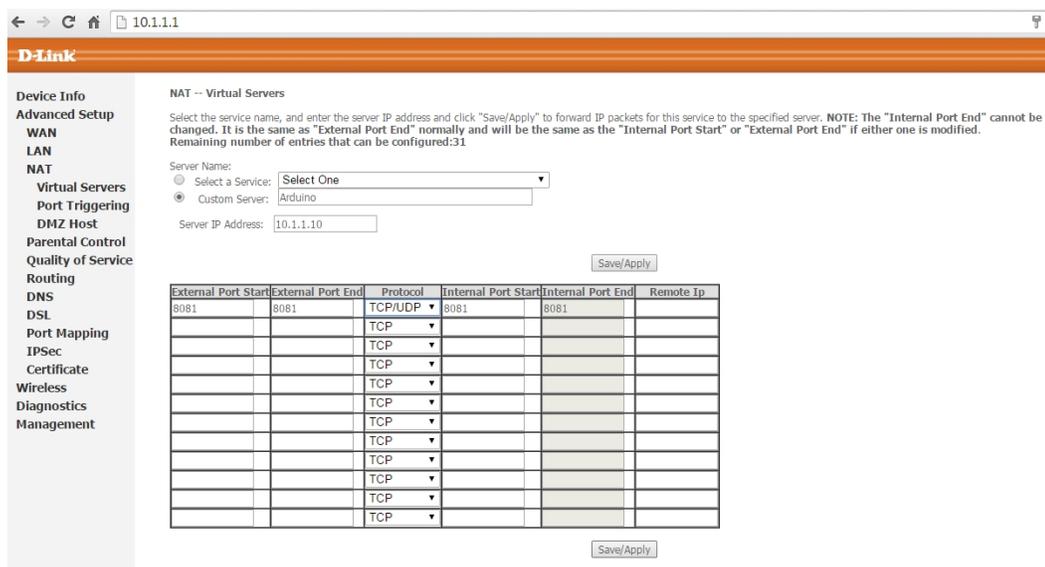


Figura 8 – Liberação de Portas
Fonte: Próprio Autor

Forouzan (2010, p.151) define “um roteador como uma caixa preta que aceita os pacotes recebidos por uma das portas de entrada (interfaces), usa uma tabela de roteamento para descobrir a porta de saída da qual parte o pacote e o envia a partir dessa porta de saída”.

A maioria das pessoas tem contrato de IP dinâmico, ou seja, cada vez que o modem é reiniciado o IP externo é modificado, isso acaba se tornando um problema para realizar a conexão com o arduino, pois sempre tem que estar verificando o novo número de IP e modificá-lo manualmente, a solução para este problema é realizar um cadastro em um serviço de DNS, há vários serviços disponíveis na internet, como por exemplo, o NO-IP e o DynDNS, o registro básico do NO-IP ainda é gratuito. Ao efetuar o cadastro em um desses sites é possível tornar seu endereço de IP um nome de sua preferência, outro passo é baixar o programa disponível no site para atualizar o nome do host com o endereço de IP correto, a cada restart do modem, alguns modems já vem com a opção de DNS, sendo necessário apenas informar o nome do domínio, seguido do usuário e senha cadastrado no site.

The image shows the D-Link router's web interface for configuring Dynamic DNS. The page title is "Add dynamic DDNS". Below the title, there is a descriptive text: "This page allows you to add a Dynamic DNS address from DynDNS.org or TZO." The configuration fields are as follows:

- D-DNS provider:** A dropdown menu with "DynDNS.org" selected.
- Hostname:** A text input field containing "TZO".
- Interface:** A dropdown menu with "pppoe_0_0_35_2/ppp_0_0_35_1" selected.
- DynDNS Settings:** Two empty text input fields for "Username" and "Password".

A "Save/Apply" button is located at the bottom right of the configuration area. On the left side, there is a navigation menu with the following items: Device Info, Advanced Setup, WAN, LAN, NAT, Parental Control, Quality of Service, Routing, DNS (highlighted), DNS Server, Dynamic DNS, DSL, Port Mapping, IPsec, Certificate, Wireless, Diagnostics, and Management.

Figura 9 – Configuração do DNS

Fonte: Próprio Autor

III METODOLOGIA

3.1 Estudo de Caso

Para execução deste trabalho, foram consideradas quatro etapas:

1. Estudos;
2. Desenvolvimento;
3. Maquete;
4. Testes;

Na etapa de estudos, foi realizado um levantamento bibliográfico a fim de conhecer as funcionalidades da plataforma arduino e seus Shields.

Na etapa de desenvolvimento, foi efetuada a codificação do sistema, desde a elaboração do layout, a estruturação com as linguagens de programação lado cliente HTML e CSS, e algumas validações com Java script, ainda nesta fase foi feita a parte de programação na linguagem C para o arduino, e por fim foi realizada a comunicação do arduino e do banco de dados com a linguagem lado servidor PHP.

Na terceira etapa foi construída uma maquete, com o objetivo de demonstrar as funcionalidades do sistema ao implantar em uma residência real.

Na quarta etapa, vários testes foram realizados a fim de localizar erros no software e hardware e assim corrigi-los.

3.2 Estudos Bibliográficos

Para a realização deste trabalho foram pesquisados em livros de domótica, arduino, programação, C/C++, PHP, HTML, CSS e Java Script, podendo assim adquirir um bom conhecimento nesta área, resultando no desenvolvimento do sistema de domótica inteligente de forma correta, segura e com uma interface amigável, dispondo ao usuário final a interação do arduino com o ambiente externo.

3.3 Cronograma

O seguinte cronograma foi utilizado para o desenvolvimento deste trabalho.

| Atividades | 2014 | | | | | | |
|--|------|-----|-----|-----|-----|-----|-----|
| | Jun | Jul | Ago | Set | Out | Nov | Dez |
| Determinação do tema | | | | | | | |
| Revisão Bibliográfica | | | | | | | |
| Testes | | | | | | | |
| Codificação | | | | | | | |
| Desenvolvimento do TCC | | | | | | | |
| Entrega do TCC a Banca Avaliadora | | | | | | | |
| Apresentação do TCC a Banca Avaliadora | | | | | | | |

Tabela 1. Cronograma do TCC.

3.4 Estrutura do trabalho

Para cumprir com o desenvolvimento deste trabalho foram realizadas as seguintes etapas: pesquisa de material bibliográfico, revisão bibliográfica, modelagem da estrutura do projeto através de UML (Unified Modeling Language), componentes físicos do protótipo, codificação do sistema, implementação e testes.

As pesquisas bibliográficas foram realizadas em livros, artigos, e no próprio site do arduino, visando assim adquirir informações necessárias para dar início ao trabalho e justificar os tópicos abordados no TCC.

Para auxiliar nos levantamentos dos requisitos necessários para o desenvolvimento do projeto gerando diagramas UML, foi usada uma ferramenta gratuita, chamada startUML.

Na fase de componentes físicos do protótipo, foram utilizadas as placas arduino uno ethernet shield e rele Shield, com a finalidade de fazer a conexão com a rede, processar os dados recebidos de uma pagina web e enviar sinais para acionar os reles.

Na fase de codificação e implementação, foi necessário a instalação do wampserver que é uma plataforma web desenvolvida pra sistema operacional Windows, sua função é rodar como um servidor local na máquina, o wampserver trata-se de um pacote que instala os seguintes programas: servidor apache, MYSQL database, PHP e phpMyAdmin para gerenciar facilmente o banco de dados. A IDE Dreamweaver, foi utilizada para a escrita dos códigos nas linguagens web PHP lado servidor, HTML, CSS e Java Script lado cliente, para escrever e enviar os códigos para o microcontrolador da placa arduino foi usada a sua própria IDE, este ambiente foi desenvolvido na linguagem de programação Java e pode ser instalada e executada em qualquer sistema operacional.

Para efetuar os testes do projeto, foi utilizado uma protoboard, esta placa contém pequenos furos, também chamada de matriz de contatos, que possibilita a inserção de componentes sem a necessidade de solda, agilizando o trabalho. Para verificar se o sistema estava enviando os sinais para acionar as lâmpadas foi utilizados LEDs.

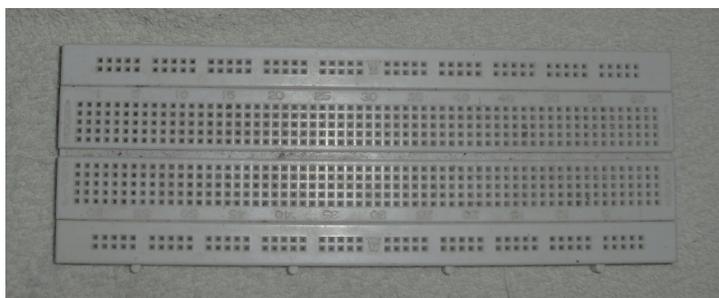


Figura 10 – Protoboard
Fonte: Próprio Autor



Figura 11 – LED
Fonte: Próprio Autor

Hardware utilizado:

- Notebook para implementação e documentação;
- Processador Intel Core™ 2 Duo CPU T6600, 2,20GHz;
- 4 GB de memória RAM;
- HD 320 GB.
- Arduino Uno.
- Ethernet Shield.
- Rele Shield.

IV. PROJETO

O protótipo desenvolvido neste trabalho tem como finalidade controlar remotamente aparelhos elétricos de uma residência. Este projeto foi montado em uma maquete para simular uma casa real, a fim de mostrar a interação do software elaborado com a parte física.

A interface web roda em um servidor PHP, podendo ser acessado de qualquer sistema operacional através de um navegador, o site é responsável de enviar comandos via socket para o arduino e receber de volta o estado que os pinos se encontram, ligado ou desligado.

O arduino é a parte da computação física responsável de acionar os aparelhos elétricos, a placa foi programada para verificar em qual condição ela vai entrar ao receber os comandos enviados pela pagina PHP, desta forma podemos controlar e acionar os pinos do arduino. Para conseguir ligar e desligar aparelhos elétricos é necessário o uso de relês, por se tratar de tensões mais altas. O relê ao receber a voltagem enviada pelo pino do arduino é ligado ou

desligado, dependendo seu estado, por consequência o aparelho conectado a ele, fará o mesmo.

Os usuários cadastrados no banco de dados interligado com o sistema, ao entrar com seu login e senha terá acesso a pagina responsável de controlar o acionamento dos aparelhos elétricos da residência, mas algumas restrições serão aplicadas dependendo o nível do acesso do usuário, as restrições definem quais tomadas o usuário terá acesso.

4.1 Diagrama de Caso de Uso

Para facilitar o entendimento dos processos realizado no sistema, o diagrama de caso de uso representado na figura 12 mostra as principais interações entre usuário e sistema.

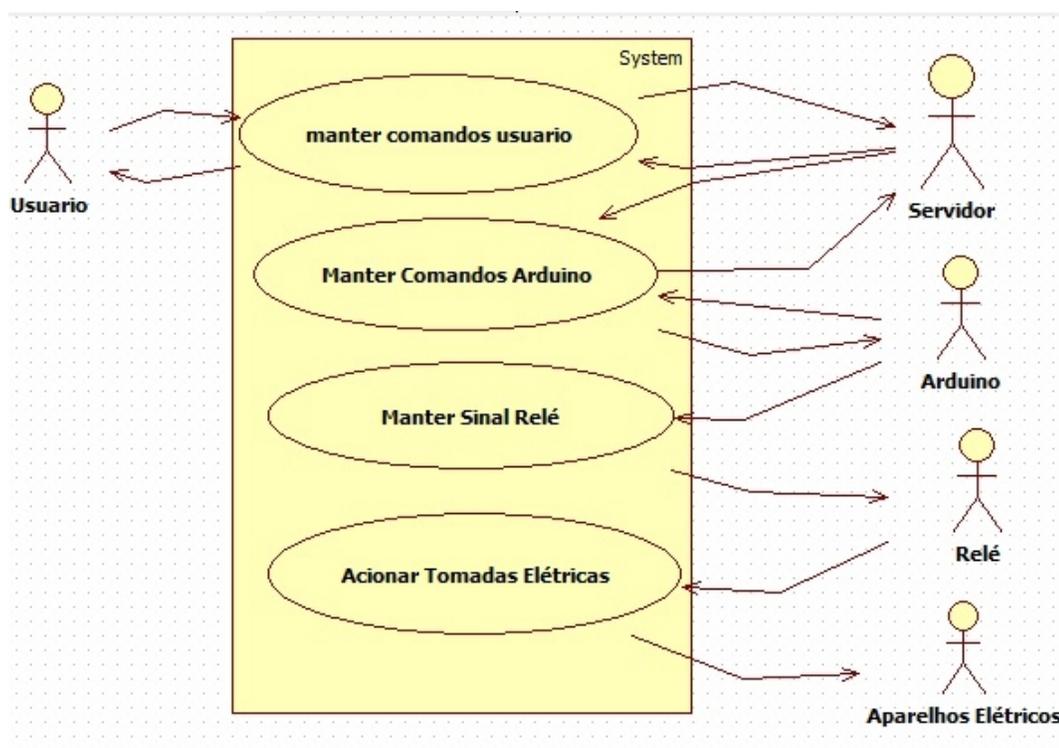


Figura 12 – Diagrama de caso de uso
Fonte: Próprio Autor

4.2 Diagrama de Seqüencia

As interações também podem ser observadas no diagrama de seqüência representada na figura 13.

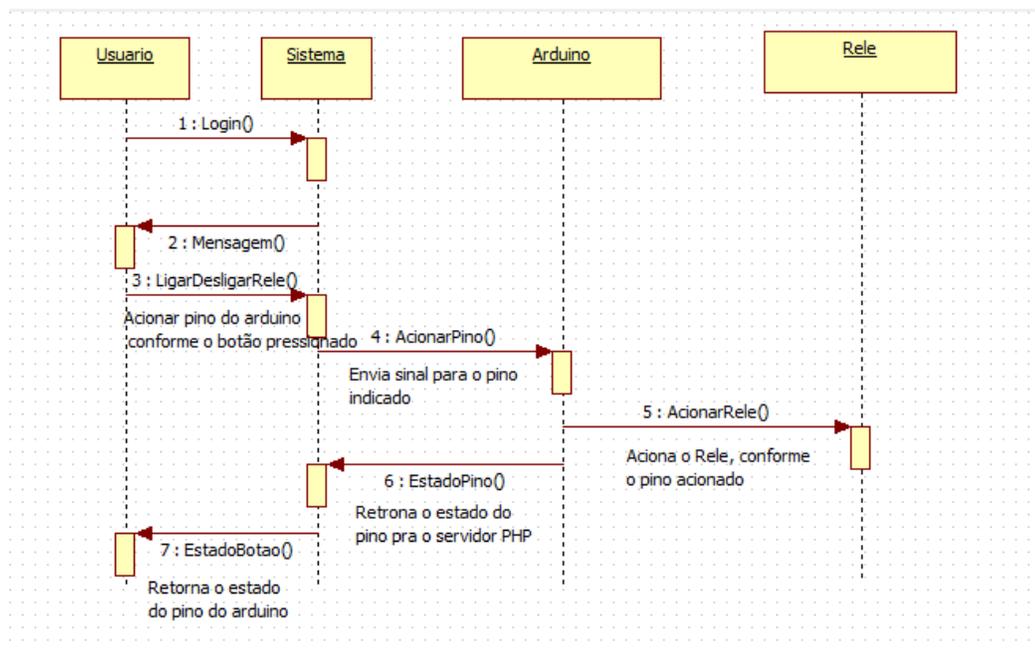


Figura 13 – Diagrama de seqüência
Fonte: Próprio Autor

4.3 Diagrama de Atividades

As atividades realizadas entre o usuário e o sistema esta sendo demonstrada no diagrama de atividades representada pela figura 14.

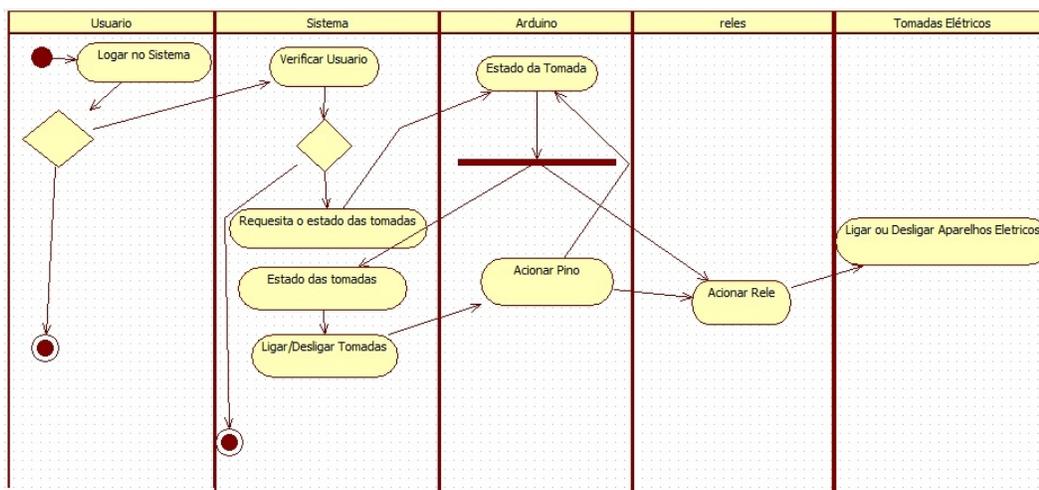


Figura 14 – Diagrama de Atividades
Fonte: Próprio Autor

4.4 Diagrama de Componentes e Implantação

Na figura 15, poder ser visualizado o diagrama com os componentes necessários e os tipos de comunicação realizados entre ambos utilizados para implantação do projeto.

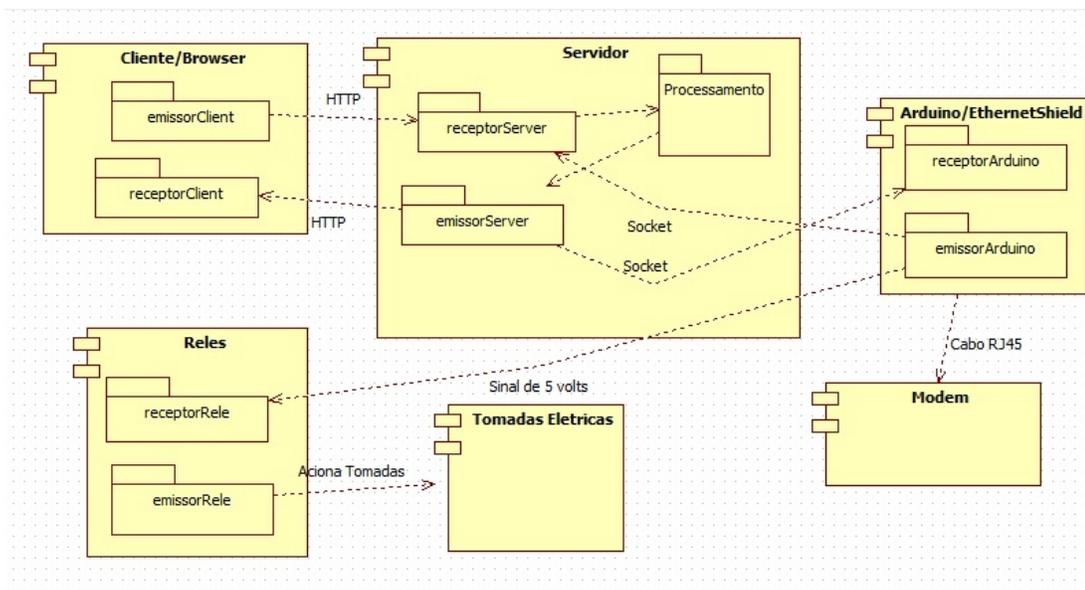


Figura 15 – Diagrama de Componentes e Implantação
Fonte: Próprio Autor

4.5 Utilizações do SARPA

Para demonstrar as funcionalidades e o funcionamento do sistema de automação residencial, a seguir, serão demonstradas algumas figuras com as telas do sistema.

Pensando na segurança e privacidade, o sistema desenvolvido consiste de um controle de usuários com níveis de acesso, desta forma, somente pessoas autorizadas podem acessar o sistema.

A figura 16 representa a tela de login, onde para acessar o sistema o usuário deve informar seu usuário e senha.

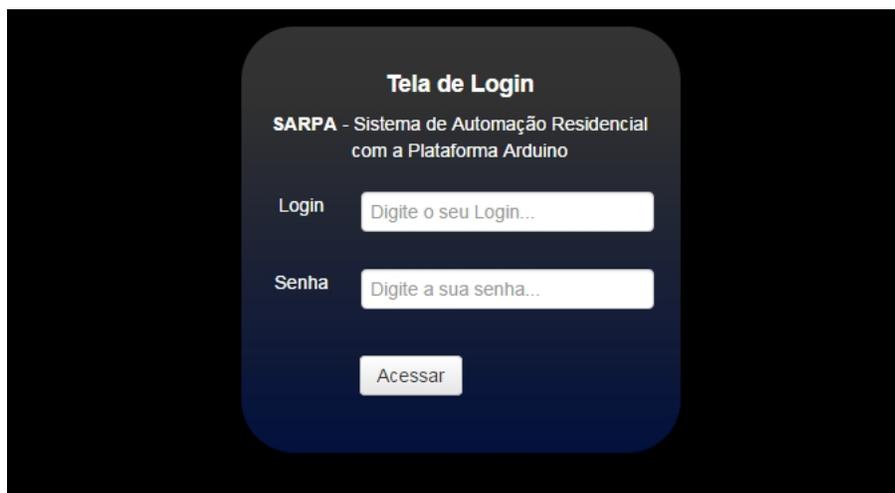


Figura 16 - Tela de Login
Fonte: Autoria própria.

Ao logar no sistema, o usuário vai se deparar com a tela principal, onde terá uma mensagem de boas vindas juntamente com a data de seu ultimo login, abaixo da mensagem há vários botões que facilita a navegação no sistema.

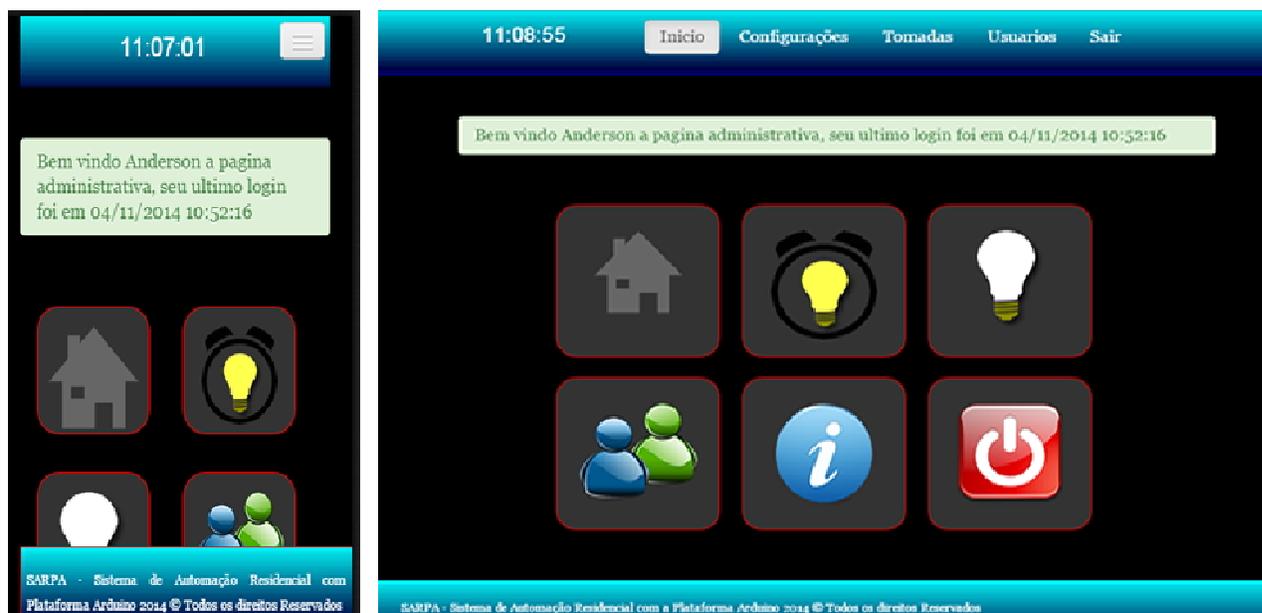


Figura 17- Tela inicial
Fonte: Autoria própria.

O usuário logado terá acesso a varias opções de controle, na figura 18 esta sendo mostrado à tela de configurações de horários, onde o usuário informara a data, hora e a tomada desejada para ser ligada ou desligada.



Figura 18 - Tela de Configuração de Horários
Fonte: Autoria própria.

Na tela principal do sistema representada pela figura 19, o usuário pode controlar o acionamento das tomadas elétricas em tempo real.

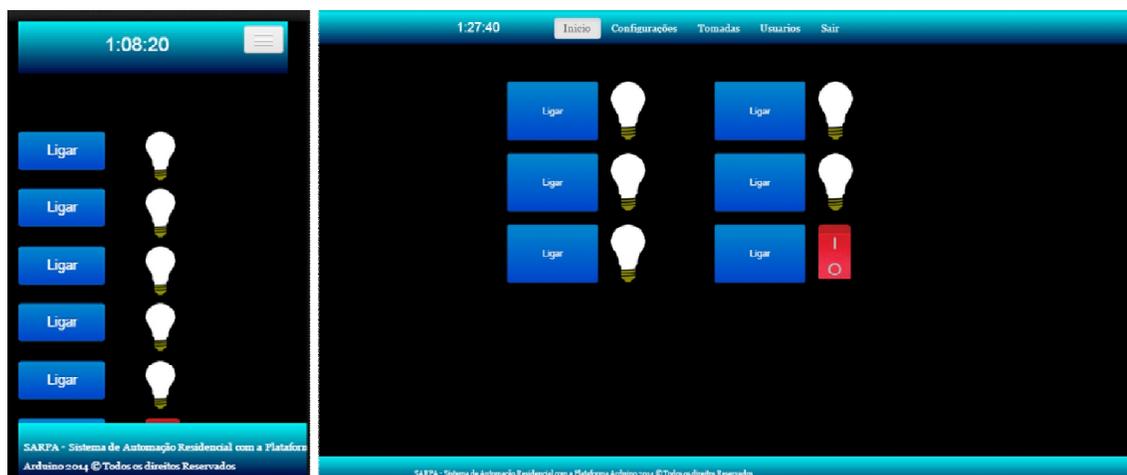


Figura 19 – Tela de acionamento de tomadas
Fonte: Autoria própria.

Para acessar o sistema é necessário que o usuário esteja cadastrado, o cadastro pode ser efetuado somente pelo administrador do sistema onde será necessário informar nome, login, senha e nível de acesso. A figura 20 demonstra a tela responsável de listar, cadastrar, alterar e excluir usuários do sistema.



Figura 20– Pagina de Usuários
Fonte: Autoria própria

A figura 21 apresenta em uma maquete a instalação dos componentes necessários para realizar a comunicação com os aparelhos elétricos, com o intuito de realizar testes e verificar as funcionalidades antes de instalar em uma residência real.



Figura 21– Maquete
Fonte: Autoria própria

V. CONSIDERAÇÕES FINAIS

Este trabalho apresentou a elaboração de um sistema residencial de baixo custo com a plataforma arduino com o objetivo de trazer conforto, praticidade e segurança para as pessoas, possibilitando o acesso de tomadas elétricas de qualquer lugar do mundo via software para o gerenciamento de energia residencial, e atingir todos os níveis sociais com um menor custo para implantação do sistema.

Para a elaboração deste trabalho foi necessário fazer um estudo aprofundado sobre computação física, tecnologias web e desenvolvimentos de sistemas embarcados com a plataforma arduino, com o objetivo de realizar a comunicação do software com o hardware, a fim de desenvolver uma tecnologia de controle de residência com um menor custo, nota-se que as pessoas estão se adaptando as novidades do mercado tecnológico, em busca de conforto, segurança e praticidade para suas vidas, com o avanço da tecnologia, varias tarefas que eram efetuadas manualmente, estão sendo automatizadas, mas muitos desses avanços se tornam inacessíveis para pessoas com uma renda mais baixa.

Atualmente o valor de um sistema de automação residencial é muito alto. O projeto desenvolvido neste trabalho denominado SARPA, mostrou uma forma barata de adquirir esta tecnologia, com o uso de ferramentas open source, a fim de possibilitar que mais pessoas possam obter este tipo de tecnologia.

Os valores gastos nos dispositivos necessários para o desenvolvimento e implantação deste projeto pode ser analisado no quadro abaixo:

| Dispositivo | Valor |
|--------------------|-------------------|
| Arduino Uno | R\$ 48,89 |
| Ethernet Shield | R\$ 59,90 |
| Relé Shield | R\$ 42,89 |
| Total: | R\$ 151,68 |

Tabela 2. Orçamento.

VI. REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, José Augusto; MOTA, José. **Casas Inteligentes**. Centro Altântico , 2003.
- ARDUINO. **Ethernet Shield**. Disponível em: < <http://arduino.cc/en/Main/ArduinoEthernetShield>>. Acesso em 04 out. 2014.
- ARDUINO. **Ethernet Shield**. Disponível em: <<http://arduino.cc/en/Main/arduinoBoardUno> >. Acesso em 05 ago. 2014.
- BANZI, Massimo. **Primeiros Passos Com o Arduino**. Novatec Editora Ltda,2012.
- BOLZANI, Caio Augustus Morais, **Residências Inteligentes** : Editora Livraria da Física, 2004.
- DAMIANI, Edgard B. **Java Script – Guia de Consulta Rapida 2º edição**. Editora Novatec, 2006.
- DNS. **Dynamic DNS**. Disponível em: < <http://pt.dyn.com/dns> >. Acesso em 24 out. 2014.
- DNS. **Dynamic DNS**. Disponível em: < <http://www.noip.com>>. Acesso em 24 out. 2014.
- Editora, 2011.
- EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em Ação**. São Paulo: Novatec, 2013.
- FOROUZAN, Behrouz A. **Protocolo TCP/IP [recurso eletrônico]**. Porto Alegre: AMGH, 2010.
- FRANCA, Alexander. **Tcl/Tk: Programação Linux**. Brasport Livros e Multimídia Ltda, 2005.
- JOBSTRAIBIZER, Flávia. **Criação de Banco de Dados Com MySQL**. São Paulo: Digerati Books, 2010.
- MARCELO, Antonio. **Apache, Configurando o servidor WEB para linux**. Brasport Livros e Multimídia Ltda, 2005.
- MCROBERTS, Michael; tradução Rafael Zanolli. **Arduino básico**.São Paulo: Novatec
- MEYER, Eric A. **SMASHING CSS – Técnicas Profissionais Para um Layout Moderno**. Artmed* Editora S.A , 2011.
- MONK, Simon. **Programação com Arduino – Começando Com Sketches – Série Tekne**. Editora BOOKMAN, 2013.
- MORAZ, Eduardo. **Treinamento Avançado em PHP 5.0**. Digerati Books, 2005.
- MOYA, José Manuel Huidobro; TEJEDOR, Ramón Jesús Millán. **Manual de Domótica**. Editora Creaciones Copyright, SL, 2010.
- MURATORI, José Roberto, DAL BÓ, Paulo Henrique, **Automação Residencial – Conceitos e aplicações**: Belo Horizonte: Editora Educere Ltda., 2013.
- NULL, Linda; LOBUR , Julia. **Arquitetura e Organizações de Computadores**. São Paulo: BOOKMAN COMPANHIA EDITORA LTDA, 2006.

OLIVEIRA, Roumo Silva; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo; **Sistemas Operacionais 4º edição**. Porto Alegre: Bookman: Instituto de Informática da UFRGS, 2010.

PETRUZELLA, Frank D. **Eleto-Técnica I**. AMGH Editora LTDA, 2014.

ROSÁRIO, João Mauricio. **Automação Industrial**.: Editora Baúrana, 2009.

SILBERSHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg; [trdução de Daniel Vieira] **Sistemas Operacionais com Java**. Rio de Janeiro: Editora Elsevier, 2008.

SILVEIRA, Marcelo; MOTA; PRATES, Rubens. **HTML 4 – Guia de Consulta Rápida**. Editora Novatec , 2001.

WARREN, John-David; ADAMS, Josh; MOLLE, Harald. **Arduino Robotics**. Editora Apress, 2011.

VII. ANEXOS

Código Fonte Das Páginas PHP

```

<?php
    if(!isset($_SESSION['usuario']) and !isset($_SESSION['logado_user'])){
        unset($_SESSION['usuario']);
        unset($_SESSION['nivel_acesso']);
        unset($_SESSION['logado_user']);
        header("Location: index.php");
    }

// inicializa variáveis
$nivel = $_SESSION['nivel_acesso'];
$status = "";
$cor1 = "";
$cor2 = "";
$cor3 = "";
$cor4 = "";

// inicializa socket
$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
//se conecta com o ip e porta do arduino
socket_connect($sock,"10.1.1.10", 80);

if(isset($_POST['stateArduino'])) {
    $stateLuz = $_POST['stateArduino'];
    //echo $stateLuz ;
    for($i=1;$i<=6;$i++){

        if(isset($_POST['pino'.$i])){
            //echo "pino".$i;
            if($stateLuz[$i-1]=='1') {
                $stateLuz[$i-1]='0';
            }else {
                $stateLuz[$i-1]='1'; }
        }
        socket_write($sock,$stateLuz,strlen($stateLuz));
    }
}

socket_write($sock,'S',2); //envia comando para o arduino , para pegar o estado dos
pinos.

$status = socket_read($sock,8);

```



```

    }

    socket_write($sock,$estadoAlarme,strlen($estadoAlarme));
}

if(isset($_POST['cadastrar'])){
    // cadastra estado dos pinos no banco de dados
    $data = obrigatorio("data",$_POST['data']);
    $hora = obrigatorio("hora",$_POST['hora']);
    $numero_pino = obrigatorio("num_pino",$_POST['pino']);
    $ligar = obrigatorio("ligar",$_POST['ligar']);
    if(isset($_POST['ativar'])){
        $ativar = 1;
    }else{
        $ativar = 0;
    }
    //$ativar = obrigatorio("ativar",$_POST['ativar']);
    $id = $numero_pino;
    global $msg;

    if(empty($msg)){

        $dados = pegarPeloId("time","id_time",$numero_pino);

        if(alterarAlarme($dadosUser = Array("data"=>$data,
            "hora"=> $hora,

            "num_pino"=> $numero_pino,

            "ligar"=> $ligar,

            "ativar"=> $ativar,

            "id"=>$numero_pino))){

                $mensagem = "";
                $mensagem = "Alarme Alterado Com Sucesso";
                //echo $mensagem;

            }else{

                $erro = "Erro ao Alterar Alarme";

            }

        }else{
            $erro = $msg;
        }
    }
}

```

```

}

//$dadosUsuarios = listar('time');
?>

<br>
<div class="container-responsive">

    <div class="span8" style="overflow:auto;">
        <?php include_once "inc/listar/alarme.php"; ?>
    </div>
    <div class="span1"></div>
    <div class="spa3">
        // formulario de cadastro do alarme no banco de dados
        <form method="post" action="">
            <table border="1" class="tbl_conf" style="margin-top:30px;margin-
bottom:70px;">
                <tr>
                    <td><input type="date" name="data"></td>
                </tr><tr>
                    <td><input type="time" name="hora" id="hora"></td>
                </tr><tr>
                    <td><select id="pino" placeholder="Digite o numero da Tomada"
name="pino">
                        <option value="1">Tomada 1</option>
                        <option value="2">Tomada 2</option>
                        <option value="3">Tomada 3</option>
                        <option value="4">Tomada 4</option>
                        <option value="4">Tomada 5</option>
                        <option value="4">Tomada 6</option>
                    </select></td>
                </tr><tr>
                    <td><input type="radio" name="ligar" value="1">Ligar&nbsp;&nbsp;&nbsp;<input
type="radio" name="ligar" value="0">Desligar<br /></td>
                </tr><tr>
                    <td><input type="checkbox" name="ativar" value="1">Ativar</td>
                </tr><tr>
                    <td><button type="submit" class="btn btn-primary btn-lg"
name="cadastrar">Enviar</button></td></tr>
            </table>
        </form>

    </div>

```

```

<?php
    echo isset($mensagem) ? '<div class="mensagem">'.$mensagem.'</div>': "";
    echo isset($erro) ? '<div class="erro">'.$erro.'</div>': "";

?>

```

Código Fonte do Arduino

```

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 10,1,1,10 }; // ip do arduino
byte gateway[] = { 10,1,1,1 }; // gateway
byte subnet[] = { 255, 0, 0, 0 }; // mascara

char retornoEstado[9] ; // recebe estado dos pinos
char caracterSocket[9]; // recebe caracteres via socket
int val =0;
int tamanhoVetor = 0;
EthernetServer server(80); // Cria o servidor na porta 80

void preencheVet(){

for(int j=0;j<8;j++){
    retornoEstado[j]='0';
    caracterSocket[j]= '0';
    if(j==6){
        retornoEstado[j]='L';
        caracterSocket[j]='L';
    }
    if(j==7){
        retornoEstado[j] ='/';
        caracterSocket[j]='/';
    }
}
}

```

```
}  
}  
  
void setup() {  
  Ethernet.begin(mac, ip, gateway, subnet);  
  server.begin();  
  
  preencheVet();  
  Serial.begin(9600);  
  pinMode(2,OUTPUT);  
  pinMode(3,OUTPUT);  
  pinMode(4,OUTPUT);  
  pinMode(5,OUTPUT);  
  pinMode(6,OUTPUT);  
  pinMode(7,OUTPUT);  
  
}  
  
int i=1;  
  
void loop() {  
  EthernetClient client = server.available();  
  // SE receber um caracter...  
  
  if (client) {  
  
    caracterSocket[i] = client.read(); // le os caracter recebido aramazena no vetor  
  
    // Serial.println(i);  
    if(caracterSocket[i] == 'S'){  
  
      if(caracterSocket[i-1] == 'S'){ //retorna estado dos pinos  
        i=0;
```

```
    client.write(retornoEstado);
}
else if(caracterSocket[i-1] == 'L'){
    i=0;
    for(int j=0;j<6;j++){
        retornoEstado[j]=caracterSocket[j+1];
        if (retornoEstado[j]=='1'){
            digitalWrite(j+2,HIGH);
        }
        else digitalWrite(j+2,LOW);
    }
}
}
}

i++;

}
}
```

SARPA: SISTEMA DE AUTOMAÇÃO RESIDENCIAL COM A PLATAFORMA ARDUINO

SARPA: AUTOMATION SYSTEM RESIDENTIAL DECK WITH ARDUINO

Anderson Di Domenico

Departamento de Ciência da Computação
Centro Universitário Unifacvest, Lages, SC
anderson.di.domenico@hotmail.com

Márcio José Sembay

Departamento de Ciência da Computação
Centro Universitário Unifacvest, Lages, SC
m_sembay@yahoo.com.br

Resumo. Este trabalho apresenta um sistema de automação residencial nas linguagens de programação: PHP, C, Java Script, HTML e CSS, desenvolvida com a plataforma de computação física e de fonte aberta arduino na versão uno, juntamente com seu modulo ethernet shield, que permite com que a placa arduino possa se conectar a internet, o objetivo é controlar tomadas elétricas de qualquer computador ou celular com acesso a internet, dispondo ao usuário um maior conforto e segurança para sua residência, e com um menor custo.

Palavras Chaves: Linguagens de Programação; PHP; C; Java Script; CSS; Plataforma de Computação Física; Arduino; Ethernet Shield.

Abstract. This work presents a system of home automation in programming languages: PHP, C, Java Script, HTML and CSS, designed with physical computing platform Arduino and open source in one version, along with your ethernet shield module, which allows to that the arduino board to connect to the Internet, the goal is to control electrical outlets from any computer or phone with internet access, offering the user greater comfort and safety for your home, and at a lower cost.

Key words: Programming Languages; PHP; C; Java Script; CSS; Physical Computing Platform; Arduino; Ethernet Shield.

1 Introdução

De acordo com Bolzani [1] cada tecnologia traz acoplado um novo vocabulário. Quando o assunto é residência inteligente, não é diferente: casa automática, casa inteligente, automação residencial, retrofitting, domótica, etc. – mas tudo pode ser resumido em uma só palavra: conforto.

Atualmente para automatizar uma residência é extremamente caro, mas com o avanço da tecnologia já esta disponível varias soluções open source que facilita o desenvolvimento e implantação destes sistemas com um custo muito mais acessível. A solução proposta neste trabalho é o uso da plataforma Arduino, juntamente com seu modulo de ethernet, Ethernet

Shield, desta forma torna-se possível o desenvolvimento de um sistema com acesso remoto através de um browser de qualquer dispositivo conectado a rede.

Desde os tempos mais remotos o homem vem tentando fazer com que os utensílios e ferramentas substituam-no no trabalho, sendo o seu maior sonho criar um autômato que realize todas as funções operárias [2].

As pessoas sempre estão em busca de segurança e conforto para suas vidas, formas de aperfeiçoar serviços do dia-dia para ter o menor esforço físico e em menos tempo. Uma casa automatizada trás vários benefícios aos seus moradores, como exemplo as tarefas rotineiras que teria que ser realizada manualmente, pode ser programadas para serem executadas automaticamente, apenas com um clique de botão, proporcionando maior praticidade, conforto, segurança e economia para o usuário.

O mercado de AUTOMAÇÃO RESIDENCIAL no Brasil aos poucos está adquirindo características muito próximas às de mercados mais evoluídos. A incorporação de um novo profissional, intitulado Integrador de Sistemas Residenciais, define uma situação específica desse mercado: em função das diferentes tecnologias, da sua complexidade de projeto, instalação e programação, ainda não existem soluções plug-and-play, exigindo, assim, uma especialização do profissional que nele atua [3].

A implantação de um sistema deste porte exige grandes conhecimentos e um bom planejamento antes, para não haver falhas. Não existe nenhuma solução plug-and-play, ou seja, antes de fazer a instalação é necessário verificar com o cliente suas necessidades, o que vai lhe beneficiar e se ajustar no seu estilo de vida.

Atualmente já existem vários dispositivos para automação residencial, mas poucas pessoas têm acesso devido ao seu custo ser muito elevado, a solução proposta neste artigo, consiste na utilização da placa de prototipagem Arduino, juntamente com seu shield de ethernet, desta forma é possível acionar relés instalados nas tomadas elétricas e motores, através de um browser, onde o usuário poderá controlar remotamente os aparelhos elétricos de qualquer dispositivo conectado a internet, e com um valor bem mais acessível.

2 Métodos e Materiais

2.1 Automação Residencial

Segundo Rosário [2], a introdução das primeiras formas de automação deu-se nas indústrias de processo, por meio do desenvolvimento de equipamentos de controle e de medição elétrica e pneumática. Porém a palavra automação ganhou relevância com o surgimento da máquina de comando numérico em 1949/50. Criada com a capacidade para realizar certas operações previamente programadas sem a intervenção direta de um operador, essa máquina abriu perspectivas para mudanças profundas na produção industrial.

No mercado brasileiro, a definição de automação residencial (AR) surgiu com herança de Home Automation utilizado no mercado americano. Isso ocorreu pelo fato de, no Brasil, os primeiros sistemas voltados para automação de residências serem oriundos de fabricas americanas. Na Europa, o termo mais utilizado é Domótica – junção da palavra latina Domus (casa) com robótica (controle automatizado) [3].

De acordo com Moya e Tejedor [4], A origem da domótica remonta-se aos anos 70, quando depois de muitas investigações, apareceram os primeiros dispositivos de automação de edifícios baseados na ainda exitosa tecnologia X-10. Durante os anos seguintes se manifestou um crescente interesse pela busca da "casa ideal", começando diversos ensaios com avançados eletrodomésticos e dispositivos automáticos para o lar. Os primeiros sistemas comerciais foram instalados principalmente nos Estados Unidos e limitavam-se à regulação da temperatura

ambiente dos edifícios de escritórios e pouco mais. Mais tarde, depois do auge dos computadores pessoais, no final da década dos 80 e princípios dos 90, começaram-se a incorporar nestes edifícios os SCE (sistema de cabeamento estruturado) para facilitar a conexão de todo o tipo de terminais e periféricos entre si, usando um padrão de fiação e tomadas localizadas em todo o edifício. Além dos dados, estes sistemas de cabeamento permitiu o transporte da voz e a conexão de alguns dispositivos de controle e de segurança, de modo que esses edifícios foram equipados com um SCE, ele começou a chamá-los de edifícios inteligentes.

Atualmente esta tecnologia esta sendo usada em residências comuns, controlando tarefas e funções que facilita o cotidiano do morador, isso através de um software que interage com equipamentos que acionam tomadas e aparelhos elétricos da casa.

2.2 Arduino

Segundo Mcroberts [5], a placa Arduino é composta de um microprocessador Atmel AVR, um cristal ou oscilador (relógio simples que envia pulsos de tempo em uma frequência especificada, para permitir sua operação na velocidade correta) e um regulador linear de 5 volts. Dependendo do tipo do Arduino que você utiliza, ele também pode ter saída USB, que permite conectá-lo a um PC ou MAC para upload ou recuperação dos dados. A placa expõe os pinos de entrada/saída do microcontrolador, para que você possa conectá-los a outros circuitos ou sensores.

O Arduino é uma plataforma física da computação de hardware aberto, baseada em uma placa microcontroladora com entradas e saídas (E/S), possui um microcontrolador Atmel AVR de 8 bits, entradas analógicas e saídas digitais, pinos de alimentação 5v - 3,3v e terra (GND) 0v, cristal oscilador 16 MHz, porta de conexão USB, fonte de alimentação e um botão de reset.

O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou ate mesmo a Internet para recuperar e enviar dados do Arduino e atuar sobre eles [5].

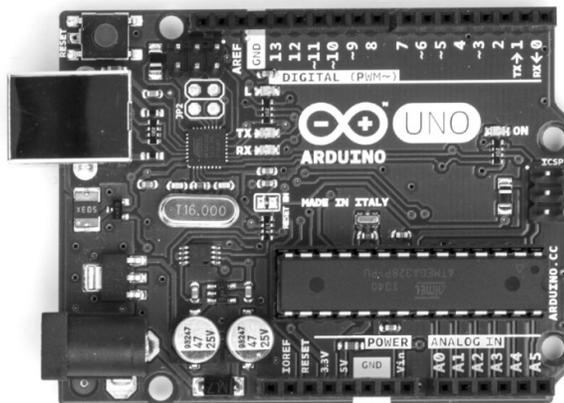


Figura 1. Placa Arduino [8]

Arduino veio com uma linguagem de programação fácil de aprender (derivado do C + +), que incorpora diversas funções de programação complexas em simples comandos que são muito mais fácil para um iniciante aprender [6].

A IDE do Arduino é uma aplicação multiplataforma, ou seja, é compatível com qualquer sistema operacional, foi desenvolvida na linguagem de programação Java e baseado em Processing, avr-gcc e outros softwares open source.

Para programar o Arduino (fazer com que ele faça o que você deseja) você utiliza o IDE do Arduino, um software livre no qual você escreve o código na linguagem que o Arduino

compreende (baseada na linguagem C). O IDE permite que você escreva um programa de computador, que é um conjunto de instruções passo a passo, das quais você faz o upload para o Arduino. Seu Arduino, então, executará essas instruções, interagindo com o que estiver conectado a ele. No mundo do Arduino, programas são conhecidos como sketches (rascunho, ou esboço) [5].

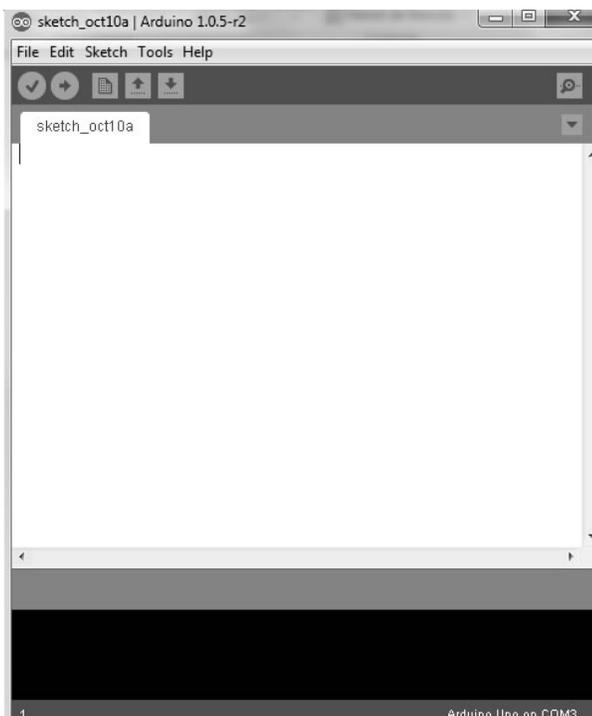


Figura 2. IDE do Arduino

2.2.1 Shields

Shields significa escudo, como o próprio nome diz, é um módulo que é encaixado na parte de cima do Arduino como fosse um escudo, serve para estender a capacidade do Arduino. Há diferentes Shields desenvolvidos para o Arduino, cada um com funções e finalidades diferentes facilitando o desenvolvimento de projetos.

Evans, Noble e Hochenbaum [7], afirma que quase todos os Shields Arduino seguem a mesma filosofia do kit de ferramentas original: eles são fáceis de montar e amigáveis para se trabalhar.

Para o desenvolvimento do projeto abordado neste trabalho foi utilizadas duas placas de shield: Ethernet Shield e Rele Shield.

2.2.1.1 Ethernet Shield

A função do Ethernet Shield é conectar o Arduino à internet ou em uma rede ethernet local, usando um cabo Rj45. Ele é baseado no chip Wiznet W5100 que fornece uma biblioteca de rede IP, que suporta tanto TCP como UDP. O Arduino Ethernet shield suporta até quatro conexões simultâneas.

2.2.1.2 Rele Shield

Este shield serve para acionar cargas altas, como ligar e controlar lâmpadas e equipamentos ligados a rede elétrica.

2.3 Linguagens WEB

Para construções de sites, sistemas WEB, é necessário conhecer linguagens de programação para WEB, há vários tipos de linguagens e cada uma delas tem suas peculiaridades. Nos próximos capítulos será apresentado as linguagens e tecnologias WEB utilizadas para o desenvolvimento do sistema abordado neste trabalho.

2.3.1 HTML

HTML é a sigla de Hyper Text Markup Language (Linguagem de Marcação de hiper Texto) utilizada para criação de paginas WEB.

Um documento HTML é um arquivo de texto contento tags de marcação que informa ao browser como exibir a pagina [9].

2.3.2 CSS

A CSS se tornou tão bem-sucedida quase tanto quanto o próprio HTML – que às vezes é difícil entender, Atualmente esta em todos os lugares dos navegadores Web às lojas de aplicativos e clientes de bate-papo, e não demonstra qualquer sinal de enfraquecimento. À medida que o uso da linguagem aumenta, seus recursos continuam a progredir [10].

Cascadingn Style Sheets é uma linguagem de estilo, trabalha junto com linguagens de Marcação de texto como XML, HTML ou XHTML, tem como finalidade definir estilos as páginas web, ou seja, como os elementos de marcação serão apresentados.

A função da linguagem HTML é definir a estrutura e a semântica de documentos. Entretanto, o HTML cruzou esta linha e passou a conter muitos elementos e atributos que dizem a respeito ao aspecto visual (apresentação) das informações [11].

2.3.2.1 Bootstrap

O Bootstrap é um framework CSS e JS que ajuda no desenvolvimento de layouts das paginas HTML, dentro dos arquivos do bootstrap existem varias classes CSS e funções Java Script que podem ser escolhidas pelos desenvolvedores e inclusas no HTML, sem se preocupar com conflitos, pois este framework segue a risca os padrões WEB da W3C.

2.3.3 Java Script

O Java Script é um é uma linguagem de programação que roda no navegador, tem a função de interagir com o usuário, fazendo validações de campos, alterando conteúdo do documento ou até criar elementos HTML.

Atualmente, falar de Java Script tornou-se um assunto de escopo muito limitado. A linguagem foi criada em 1995 pela Netscape, com o objetivo de permitir a manipulação dinâmica dos elementos de uma página HTML que fosse apresentado no seu navegador, o Netscape Navigator [12].

Apesar de ser uma das linguagens mais populares da Web, ainda a muitas incompatibilidades por parte dos navegadores, muitas vezes ao escrever um código, o mesmo não funciona em certos browsers, tendo que buscar outra solução que seja compatível com todos os navegadores.

2.3.4 PHP

PHP (Hiptertext PreProcessor) é uma linguagem de programação utilizada para criar sites dinâmicos. Esses sites são os que permitem uma interação com o usuário mediante formulários, parâmetros de URL, links etc., e que retornam páginas criadas em tempo real. Aplicações em PHP são geradas em excelente performance, e automaticamente, pelo servidor. O usuário não tem acesso ao código PHP, somente ao HTML. Esse tipo de detalhe é muito importante quando estamos lidando, por exemplo, com senhas e acesso a banco de dados [13].

O PHP pode ser considerada uma linguagem segura, pois os dados são processados no lado do servidor, ou seja, o usuário final poderá visualizar apenas linguagens do lado cliente como HTML, Java Script ou CSS.

A linguagem PHP, atualmente está na versão 5, mas já suporta orientação a objetos desde a sua versão 3, desta forma podemos reaproveitar códigos e agilizar o processo de desenvolvimento de sites e sistemas para web e com maior segurança, outra vantagem é poder trabalhar com banco de dados, podendo assim ter um controle total sobre os dados inseridos.

2.4 Apache

O Apache é um servidor open source utilizado para interpretar linguagens de programação web, sua função é ler os códigos da linguagem lado servidor e converter para linguagem lado cliente para ser interpretado pelo browser.

A história do Apache e de seu desenvolvimento começa em 1995 quando a NCSA (National Center for Computer Applications) criou o antigo NCSA Web Server, que naqueles tempos tornou-se o mais popular servidor http existente. Contudo, a NCSA não desenvolveu muito o projeto, tornando-o até mesmo estagnado. Uma parte da equipe dos desenvolvedores não gostou muito desta atitude e desligou a NCSA e, por conta própria, juntamente com um grupo de desenvolvedores começou a criar uma série de inovações/ patches em cima do código básico do Web Server da NCSA e assim nasceu o Apache (derivação de Apatchy, um trocadilho devido ao enorme número de patches que foram criados pela equipe), que teve seu debut em abril de 1995 sob a versão 0.62 [14].

Marcelo [14], ainda cita algumas vantagens do Apache. São elas:

- a) Suporte a HTTP 1.1 para criação de virtual hosts baseados em DNS;
- b) Suporte a Secured Socket Layer (SSL) para transações seguras;
- c) Suporte a CGI's, Perl e PHP;
- d) Suporte a autenticação baseada em HTTP;
- e) Suporte a Server Side Includes (SSI);
- f) Suporte a Servlets Java;
- g) Logs customizáveis;
- h) Configuração rápida e simples.

2.5 MySQL

O MySQL é um banco de dados gratuito para uso não comercial, compatível com várias linguagens de programação e de fácil instalação.

Segundo Jobstraibizer [15], podemos destacar ainda como vantagens, a fácil portabilidade do MySQL entre os diferentes sistemas operacionais, sendo atualmente suportado por Windows, Linux, FreeBSD (e outros sistemas BSD), Solaris e até mesmo MAC OS.

Jobstraibizer [15], ainda cita alguns dos recursos mais importantes do MySQL:

- a) Suporte completo para integridade referencial e chaves estrangeiras.
- b) Triggers, Views, Cursors e Stored Procedures.
- c) Sub selects (consultas secundárias realizadas dentro de uma consulta principal).
- d) Compatibilidade total com diversas linguagens de programação, como Java, C, Python, Perl, PHP, ASP, Delphi, Ruby e outras.

- e) Grande número de ferramentas gráficas que podem ser utilizadas para auxiliar a construção de queries e manutenção no banco de dados.
- f) Portabilidade entre diversos sistemas operacionais, além de ser open source.
- g) Não exige grandes recursos de hardware para ter um excelente desempenho.

2.5.1 phpMyAdmin

É um programa Web desenvolvido na linguagem PHP, serve para gerenciar o MySQL, possui uma interface amigável, nela é possível criar,deletar, inserir banco de dados, tabelas e muitas outra opções pela parte gráfica, ou se preferir via código SQL.

Segundo Jobstraibizer [15], Popular, o PHPMyAdmin é conhecido por 10 entre 10 usuários de MYSQL. Isso se deve ao fato de que, por tratar-se de um aplicativo 100% baseado em web, ou seja, instalado no servidor e acessado via browser, fica acessível para o desenvolvedor de qualquer local em que este esteja.

O PHPMyAdmin facilita o trabalho do programador, com este aplicativo pode ser desenvolvido todas as tabelas de banco de dados necessárias para o desenvolvimento do sistema usando a parte gráfica, agilizando a entrega do projeto.

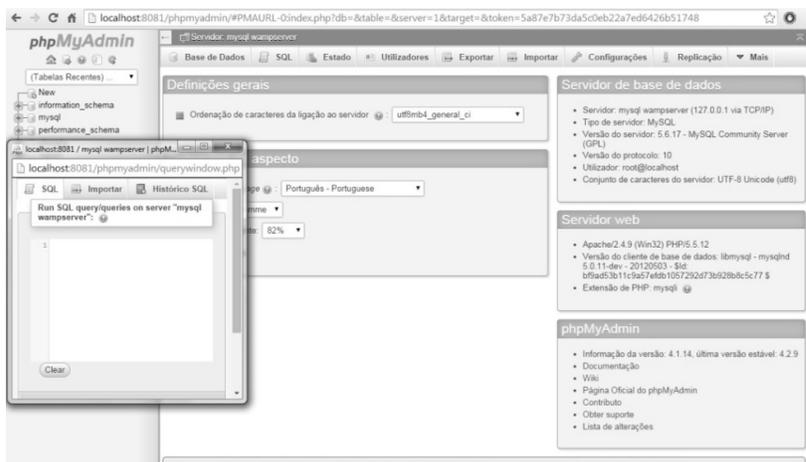


Figura 3. Tela do phpMyAdmin

2.6 Socket

Sockets são canais de comunicação de rede. Em outras palavras, se você quer fazer um script que se comunique em rede com outro script, então é um socket que você precisa [16].

Para realizar a conexão via socket do Arduino com o servidor PHP, é necessário informar o IP e a porta que será realizada a comunicação, depois de realizada a comunicação é possível enviar e receber dados, podendo manipulá-los conforme as necessidades do projeto.

De acordo com SILBERSHATZ, GALVIN e GAGNE [17], um socket é definido como uma extremidade para comunicação. Um par de processos comunicando por uma rede emprega um par de sockets – um para cada processo. Um socket é identificado por um endereço IP conectado a um numero de porta. (Por exemplo, a representação de socket para um host no endereço IP 121.10.14.56 executado na porta 15 é 121.10.14.56:15).

2.7 Acesso externo

O acesso externo do sistema, somente é possível se a porta especificada no Arduino for liberada no roteador, desta forma ao passar o endereço de IP externo no parâmetro da função socket do PHP, é possível realizar a comunicação do servidor PHP com o Arduino de qualquer lugar do mundo, desde que esteja conectado à internet.

Forouzan [18], Define um roteador como uma caixa preta que aceita os pacotes recebidos por uma das portas de entrada (interfaces), usa uma tabela de roteamento para descobrir a porta de saída da qual parte o pacote e o envia a partir dessa porta de saída.

A maioria das pessoas tem contrato de IP dinâmico, ou seja, cada vez que o modem é reiniciado o IP externo é modificado, isso acaba se tornando um problema para realizar a conexão com o Arduino, pois sempre tem que estar verificando o novo número de IP e modificá-lo manualmente, a solução para este problema é realizar um cadastro em um serviço de DNS, há vários serviços disponíveis na internet, como por exemplo o NO-IP e o DynDNS, o registro básico do NO-IP ainda é gratuito. ao efetuar o cadastro em um desses sites é possível tornar seu endereço de IP um nome de sua preferência, outro passo é baixar o programa disponível no site para atualizar o nome do host com o endereço de IP correto, a cada restart do modem, alguns modems já vem com a opção de DNS, sendo necessário apenas informar o nome do domínio, seguido do usuário e senha cadastrado no site.

Figura 4. Tela de Configuração do DNS

3 Projeto

O protótipo desenvolvido neste trabalho denominado SARPA, tem como finalidade controlar remotamente aparelhos elétricos de uma residência. Este projeto foi desenvolvido com tecnologias open source, com a finalidade de possibilitar que mais pessoas possam automatizar suas residências com um custo menor.

A interface web roda em um servidor PHP, podendo ser acessado de qualquer sistema operacional através de um navegador, a página gerada é responsável de enviar comandos via socket para o Arduino e receber de volta o estado que os pinos se encontram, ligado ou desligado.

O Arduino é a parte da computação física responsável de acionar os aparelhos elétricos, a placa foi programada para verificar em qual condição entrara ao receber os comandos enviados

pela pagina PHP, desta forma podemos controlar e acionar os pinos do Arduino. Para conseguir ligar e desligar aparelhos elétricos é necessário o uso de relês, por se tratar de tensões mais altas. O relê ao receber a voltagem enviada pelo pino do Arduino é ligado ou desligado, dependendo seu estado, por consequência o aparelho conectado a ele, fará o mesmo.

Os usuários cadastrados no banco de dados interligado com o sistema, ao entrar com seu login e senha terá acesso a pagina responsável de controlar o acionamento dos aparelhos elétricos da residência, mas algumas restrições serão aplicadas dependendo o nível do acesso do usuário, essas restrições vai informar quais tomadas o usuário terá acesso.

3.1 Diagrama de Caso de Uso

Para facilitar o entendimento dos processos realizado no sistema, o diagrama de caso de uso representado na figura 5 mostra as principais interações entre usuário e sistema.

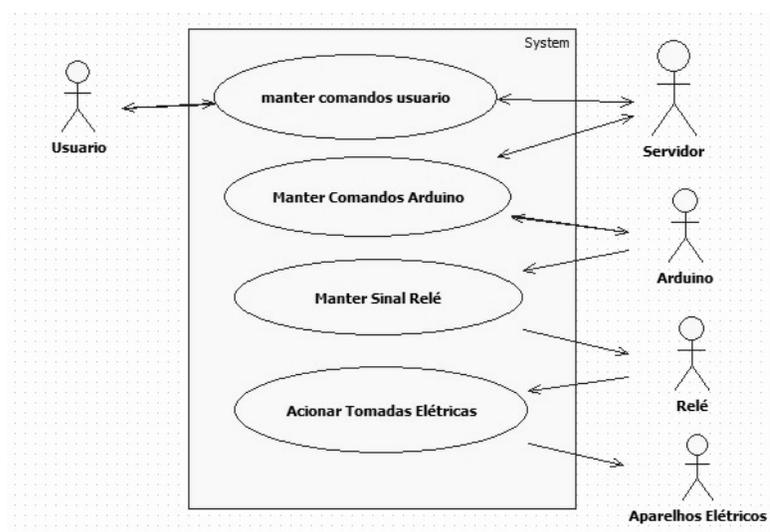


Figura 5. Diagrama de caso de uso

3.2 Diagrama de Seqüência

As interações também podem ser observadas no diagrama de seqüência representada na figura 6.

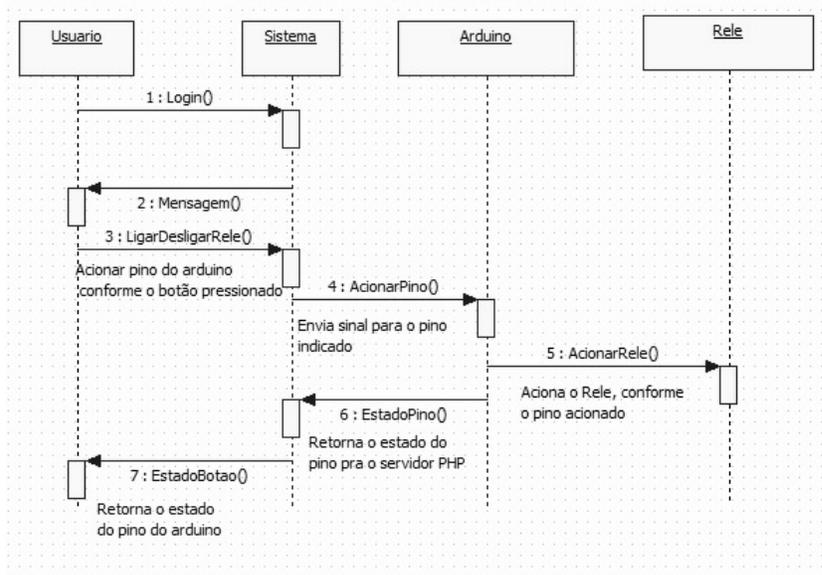


Figura 6. Diagrama de seqüência

3.3 Utilização do SARPA

Para demonstrar as funcionalidades e o funcionamento do sistema de automação residencial, a seguir, serão demonstradas algumas figuras com as telas do sistema.

Pensando na segurança e privacidade, o sistema desenvolvido consiste de um controle de usuários com níveis de acesso, desta forma, somente pessoas autorizadas podem acessar o sistema.

A figura 7 representa a tela de login, onde para acessar o sistema o usuário deve informar seu usuário e senha.



Figura 7. Tela de Login

Ao logar no sistema, o usuário vai se deparar com a tela principal, onde terá uma mensagem de boas vindas juntamente com a data de seu ultimo login, abaixo da mensagem há vários botões que facilita a navegação no sistema.

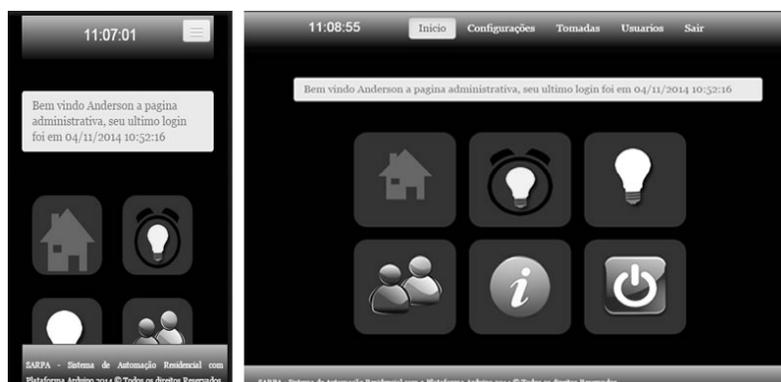


Figura 8. Tela de Inicial

O usuário logado terá acesso a varias opções de controle, na figura 9 esta sendo mostrado à tela de configurações de horários, onde o usuário informara a data, hora e a tomada desejada para ser ligada ou desligada.

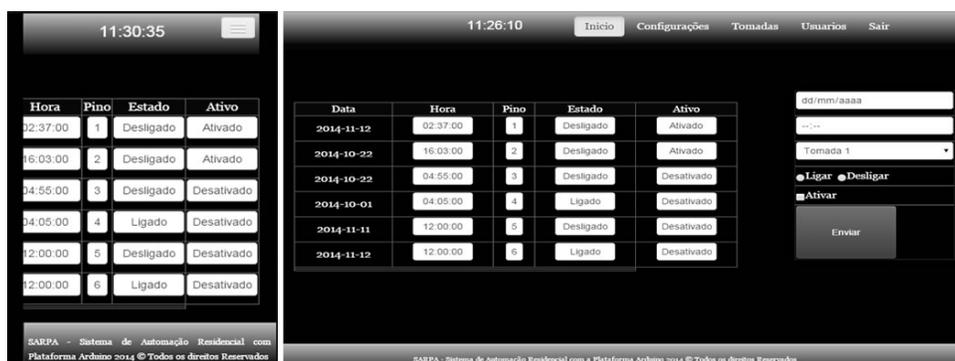


Figura 9. Tela de Configuração de Horários

Na tela principal do sistema representada pela figura 10, o usuário pode controlar o acionamento das tomadas elétricas em tempo real.

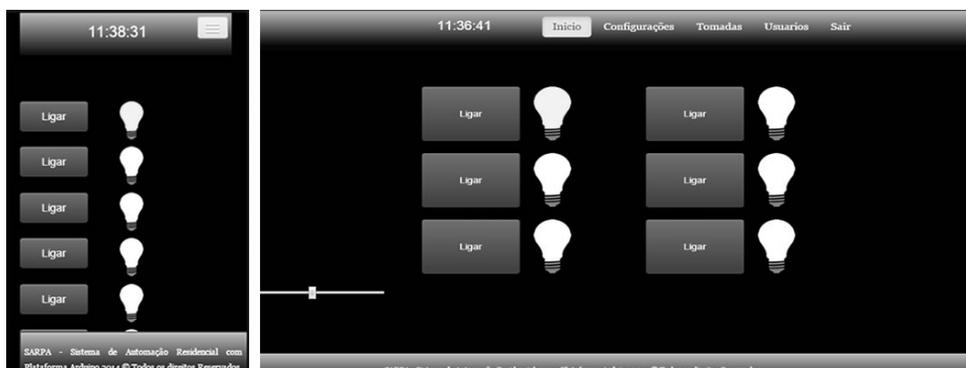


Figura 10 – Tela de acionamento de tomadas

Para acessar o sistema é necessário que o usuário esteja cadastrado, o cadastro pode ser efetuado somente pelo administrador do sistema onde será necessário informar nome, login, senha e nível de acesso. A figura 11 demonstra a tela responsável de listar, cadastrar, alterar e excluir usuários do sistema.



Figura 11. Tela de Usuários

4 Conclusão

Para a elaboração deste trabalho foi necessário fazer um estudo aprofundado sobre computação física, tecnologias web e desenvolvimentos de sistemas embarcados com a plataforma Arduino, com o objetivo de realizar a comunicação do software com o hardware, a fim de desenvolver uma tecnologia de controle de residência com um menor custo, nota-se que as pessoas estão se adaptando as novidades do mercado tecnológico, em busca de conforto, segurança e praticidade para suas vidas, com o avanço da tecnologia, varias tarefas que eram efetuadas manualmente, estão sendo automatizadas, mas muitos desses avanços se tornam inacessíveis para pessoas com uma renda mais baixa.

Atualmente o valor de um sistema de automação residencial é muito alto. O projeto desenvolvido neste trabalho denominado SARPA, mostrou uma forma barata de adquirir esta tecnologia, com o uso de ferramentas open source, a fim de possibilitar que mais pessoas possam obter este tipo de tecnologia.

Referências

- [1] BOLZANI, Caio Augustus Moraes, Residências Inteligentes: Editora Livraria da Física, 2004.
- [2] ROSÁRIO, João Mauricio. Automação Industrial.: Editora Baúrana, 2009.
- [3] MURATORI, José Roberto, DAL BÓ, Paulo Henrique, Automação Residencial – Conceitos e aplicações: Belo Horizonte: Editora Educere Ltda., 2013.
- [4] MOYA, José Manuel Huidobro; TEJEDOR, Ramón Jesús Millán. Manual de Domótica. Editora Creaciones Copyright, SL, 2010.
- [5] MCROBERTS, Michael; tradução Rafael Zanolli. Arduino básico. São Paulo: Novatec Editora, 2011.
- [6] WARREN, John-David; ADAMS, Josh; MOLLE, Harald. Arduino Robotics. Editora Apress, 2011.
- [7] EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. Arduino em Ação. São Paulo: Novatec, 2013.

- [8] ARDUINO. Imagem da Placa de Prototipagem Arduino. Disponível em: http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg Acesso em Set/2014.
- [9] SILVEIRA, Marcelo; MOTA; PRATES, Rubens. HTML 4 – Guia de Consulta Rápida. Editora Novatec, 2001.
- [10] MEYER, Eric A. SMASHING CSS – Técnicas Profissionais Para um Layout Moderno. Artmed* Editora S.A , 2011.
- [11] AMARAL, Luiz Gustavo. CSS Cascading Style Sheets – Guia de Consulta Rápida. Editora Novatec, 2009.
- [12] DAMIANI, Edgard B. Java Script – Guia de Consulta Rapida 2º edição. Editora Novatec, 2006.
- [13] MORAZ, Eduardo. Treinamento Avançado em PHP 5.0. Digerati Books, 2005.
- [14] MARCELO, Antonio. Apache, Configurando o servidor WEB para linux. Brasport Livros e Multimídia Ltda, 2005.
- [15] JOBSTRAIBIZER, Flávia. Criação de Banco de Dados Com MySQL. São Paulo: Digerati Books, 2010.
- [16] FRANCA, Alexander. Tcl/Tk: Programação Linux. Brasport Livros e Multimídia Ltda, 2005.
- [17] SILBERSHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg; [trdução de Daniel Vieira] Arduino Robotics. Rio de Janeiro: Editora Elsevier, 2008.
- [18] FOROUZAN, Behrouz A. Protocolo TCP/IP [recurso eletrônicos]. Porto Alegre: AMGH, 2010.

To: anderson.di.domenico@hotmail.com
Subject: [RECEN] Agradecimento pela Submissão
Date: Wed, 12 Nov 2014 03:37:55 -0200
From: zeza@unicentro.br

Anderson Di Domenico,

Agradecemos a submissão do seu manuscrito "SARPA: Sistema de Automação Residencial com a Plataforma Arduino" para RECEN - Revista Ciências Exatas e Naturais. Através da interface de administração do sistema, utilizado para a submissão, será possível acompanhar o progresso do documento dentro do processo editorial, bastando logar no sistema localizado em:

URL do Manuscrito:

<http://revistas.unicentro.br/index.php/RECEN/author/submission/3203>

Login: anderson_di_domenico

Em caso de dúvidas, envie suas questões para este email. Agradecemos mais uma vez considerar nossa revista como meio de transmitir ao público seu trabalho.

Maria José de Paula Castanho
RECEN - Revista Ciências Exatas e Naturais

Revista Ciências Exatas e Naturais

<http://http://web01.unicentro.br/revistas/index.php>