

UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO DE GRADUAÇÃO EM TECNOLOGIA EM REDES DE
COMPUTADORES

Rogério Pauli

**DOMÓTICA: AUTOMAÇÃO RESIDENCIAL USANDO PLATAFORMA
ARDUÍNO**

Santa Maria, RS
2016

STRC/UFSM, RS PAULI, Rogério

Tecnólogo em Redes de Computadores

2016

Rogério Pauli

DOMÓTICA: AUTOMAÇÃO RESIDENCIAL USANDO PLATAFORMA ARDUÍNO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Tecnólogo em Tecnologia em Redes de Computadores.**

ORIENTADOR: Prof. Tarcisio Ceolin Junior

Santa Maria, RS
2016

Pauli, Rogério

DOMÓTICA: AUTOMAÇÃO RESIDENCIAL UTILIZANDO
PLATAFORMA ARDUÍNO / por Rogério Pauli. – 2016.

93 f.: il.; 30 cm.

Orientador: Tarcisio Ceolin Junior

Trabalho de Conclusão de Curso - Universidade Federal de Santa
Maria, Colégio Técnico Industrial de Santa Maria, Tecnologia em Re-
des de Computadores, RS, 2016.

1. Domótica. 2. Automação Residencial. 3. Arduino. 4. Segu-
rança. 5. Open Source. I. Ceolin Junior, Tarcisio.

© 2016

Todos os direitos autorais reservados a Rogério Pauli. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: rogeriopauli@gmail.com

Rogério Pauli

DOMÓTICA: AUTOMAÇÃO RESIDENCIAL USANDO PLATAFORMA ARDUÍNO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Tecnólogo em Tecnologia em Redes de Computadores.**

Aprovado em 12 de julho de 2016:

Tarcisio Ceolin Junior, Ms. (CTISM/UFSM)
(Presidente/Orientador)

Tiago Antonio Rizzetti, Ms. (CTISM/UFSM)

Renato Preigschadt de Azevedo, Ms. (CTISM/UFSM)

Santa Maria, RS
2016

AGRADECIMENTOS

Dedico meus sinceros agradecimentos a Deus primeiramente. Obrigado ao \LaTeX por facilitar a digitação do trabalho. A minha família que sempre acreditou em mim, ao Prof. Tarcisio por ter aceito ser meu orientador. Aos professores Tiago Antonio Rizzetti e Renato Preigschadt de Azevedo por aceitarem de última hora ser a banca avaliadora. Aos amigos Daniel Biasoli e Ricardo de Moura Zago pela ajuda na finalização do trabalho. Aos desenvolvedores da plataforma Arduino, que facilitaram a vida dos entusiastas por novas tecnologias e desenvolvedores e, principalmente a minha mãe, que além de acreditar em mim financiou todo o projeto.

“Os nossos pais amam-nos porque somos seus filhos, é um fato inalterável. Nos momentos de sucesso, isso pode parecer irrelevante, mas nas ocasiões de fracasso, oferecem um consolo e uma segurança que não se encontram em qualquer outro lugar.”

(Bertrand Russell)

RESUMO

DOMÓTICA: AUTOMAÇÃO RESIDENCIAL USANDO PLATAFORMA ARDUÍNO

AUTOR: Rogério Pauli

ORIENTADOR: Tarcisio Ceolin Junior

Este trabalho descreve o desenvolvimento de um sistema destinado ao controle de acesso de baixo custo para implementação em uma residência denominado Sistema de Automação Residencial (SAR). É apresentado de forma detalhada a montagem, implementação e utilização em um laboratório de testes, utilizando tanto hardware de baixo custo através da Plataforma Arduino, quanto software aberto, como PHP e MySQL utilizados na implementação do sistema Web. Os experimentos realizados no laboratório de testes provaram que o sistema é eficaz para ser implementado em residências. Com relação aos custos, o projeto apresentou-se mais acessível com relação aos sistemas pagos existentes no mercado.

Palavras-chave: Domótica. Automação Residencial. Arduino. Código aberto

ABSTRACT

HOME AUTOMATION SYSTEM: RESIDENTIAL AUTOMATION USING ARDUINO PLATFORM

AUTHOR: Rogério Pauli

ADVISOR: Tarcisio Ceolin Junior

This paper describes the development of a system for the low cost access control for implementation in a residence called Home Automation System (SAR). It is presented in detail the assembly, implementation and use in a test lab, using both low-cost hardware through the Arduino platform, as open source software such as PHP and MySQL used to implement the Web system. The experiments performed in laboratory tests They proved that the system is effective to be implemented in homes. Regarding costs, the project had become more accessible with regard to existing pay systems on the market.

Keywords: Home Automation. Arduino. Open Source

LISTA DE FIGURAS

Figura 1.1 – Estimativa de Casas Inteligentes na América do Norte e Europa entre 2012 e 2017	11
Figura 1.2 – Trabalho proposto	12
Figura 2.1 – Sinal analógico e sinal digital	15
Figura 2.2 – Tag RFID	16
Figura 2.3 – Funcionamento básico do protocolo HTTP	19
Figura 3.1 – Plataforma Arduino Mega 2560	23
Figura 3.2 – Ethernet shield W5100 Arduino	23
Figura 3.3 – Arduino MEGA 2560 juntamente com o módulo Shield Ethernet W5100 .	24
Figura 3.4 – Módulo RFID Mifare RC522	25
Figura 3.5 – Imagem do Sensor DYP-ME003	25
Figura 3.6 – Reprodução do funcionamento do sensor PIR	26
Figura 3.7 – Relay Board	27
Figura 3.8 – Display 20 x 4	28
Figura 3.9 – Adaptador I2C	28
Figura 3.10 – Interface IDE Arduino	29
Figura 4.1 – Laboratório de testes	31
Figura 4.2 – Detalhe Arduino Mega 2560 + <i>Shield Ethernet W5100</i>	32
Figura 4.3 – Display indicando entrada autorizada	33
Figura 4.4 – Simulação de acionamento de cargas pelo SAR	34
Figura 4.5 – Simulação de acionamento do sensor de presença	34
Figura 4.6 – Modelo entidade relacionamento do Bando de Dados	35
Figura 4.7 – Método GET com base no trabalho	36
Figura 4.8 – Fluxograma interação do usuário com o SAR através da leitura do cartão de acesso.	36
Figura 4.9 – Fluxograma interação do usuário com o SAR através do Sistema Web. .	37
Figura 4.10 – Tela de login da interface web	37
Figura 4.11 – Tela de listagem dos usuários	38
Figura 4.12 – Tela de cadastro dos usuários	38
Figura 4.13 – Tela da parte do usuário	39
Figura 4.14 – Tela de comandos	39
Figura 4.15 – Ambiente de Testes	41

LISTA DE TABELAS

Tabela 2.1 – Faixas de frequência RFID	17
Tabela 2.2 – Principais características Servidor Web Apache	18
Tabela 2.3 – Características dos métodos mais utilizados no protocolo HTTP	19
Tabela 2.4 – Custos para montagem do laboratório de testes	21
Tabela 3.1 – Comparativo entre os sensores DHT11 e DHT22	26
Tabela 3.2 – Características Relay Board	27

LISTA DE ABREVIATURAS E SIGLAS

<i>SAR</i>	Sistema de Automação Residencial
<i>CI</i>	Circuito Integrado
<i>RSSF</i>	Rede de Sensores Sem Fio
<i>PIR</i>	Sensor Infravermelho Passivo
<i>RFID</i>	Identificação por Rádio Frequência
<i>HTTP</i>	Protocolo de Transferência de Hipertexto
<i>LCD</i>	Display de Cristal Líquido
<i>PHP</i>	<i>Personal Home Page</i>
<i>SGBD</i>	Sistema de Gerenciamento de Banco de Dados
<i>RO</i>	<i>Read Only</i>
<i>WORM</i>	<i>Write Once - Read Many</i>
<i>RW</i>	<i>Read-Write</i>
<i>HTML</i>	Linguagem de Marcação de Hipertexto
<i>URL</i>	Localizador Padrão de Recursos - <i>Uniform Resource Locator</i>
<i>NCSA</i>	<i>National Center for Supercomputing Applications</i>
<i>BD</i>	Bando de Dados
<i>IDE</i>	Interface de Desenvolvimento
<i>SO</i>	Sistema Operacional
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>LED</i>	<i>Light Emitting Diode</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>PC</i>	<i>Personal Computer</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivos Específicos	12
1.2	JUSTIFICATIVA	13
1.3	ESTRUTURA DO TRABALHO	13
2	REVISÃO BIBLIOGRÁFICA	14
2.1	MICRO CONTROLADOR	14
2.2	SENSOR	14
2.2.1	Sensor infravermelho passivo	15
2.3	DISPLAY DE CRISTAL LÍQUIDO	15
2.4	IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA	15
2.4.1	Tipo de operação	16
2.4.2	Frequência de operação	17
2.4.3	Memória	17
2.5	DESENVOLVIMENTO WEB	17
2.5.1	Servidor de páginas Web	18
2.5.2	Sistema de gerenciamento de banco de dados	18
2.5.3	Protocolo HTTP	18
2.5.4	PHP - Hypertext Preprocessor	20
2.6	SISTEMAS EXISTENTES NO MERCADO	20
2.6.1	Comparação entre os sistemas	21
3	MATERIAIS E MÉTODOS	22
3.1	METODOLOGIA	22
3.2	HARDWARE UTILIZADO	22
3.2.1	Plataforma Arduino Mega 2560	23
3.2.2	Adaptador ethernet shield W5100 Arduino	23
3.2.3	Leitor RFID Mifare modelo RC522	24
3.2.4	Sensor PIR modelo DYP-ME003	25
3.2.5	Sensor de umidade relativa do ar e temperatura ambiente DHT 22	26
3.2.6	Relay Board	26
3.2.7	Display LCD 20x4	27
3.3	SOFTWARE UTILIZADO	28
3.3.1	IDE de desenvolvimento do algoritmo para o Arduino	28
3.4	DESCRIÇÃO DO FUNCIONAMENTO DO PROJETO	29
4	RESULTADOS	31
4.1	LABORATÓRIO DE TESTES	31
4.1.1	Plataforma Arduino com o módulo <i>Shield Ethernet</i>	32
4.1.2	Leitor RFID	32
4.1.3	Sensor de temperatura e umidade	32
4.1.4	Display 20x4	33
4.1.5	Relay Board	33
4.1.6	Sensor de presença	33
4.1.7	Servo motor	35
4.1.8	Fonte auxiliar	35
4.2	MODELAGEM DO BANCO DE DADOS	35

4.3	INTERFACE WEB	35
4.4	TESTES DE ESTABILIDADE	40
4.4.1	Equipamentos utilizados	40
4.4.2	Softwares utilizados	40
4.4.3	Ambiente de Testes	40
4.4.4	Resultado dos testes	41
5	CONCLUSÃO	42
5.1	TRABALHOS FUTUROS	42
6	REFERÊNCIAS BIBLIOGRÁFICAS	43
	APÊNDICE A – CÓDIGO FONTE ARDUINO	44
	APÊNDICE B – CÓDIGO SQL PARA CRIAÇÃO DO BANCO DE DADOS	53
	APÊNDICE C – CÓDIGO FONTE ARQUIVO "ADMINISTRATIVO.PHP" ...	55
	APÊNDICE D – CÓDIGO FONTE ARQUIVO "BEM_VINDO.PHP"	57
	APÊNDICE E – CÓDIGO FONTE ARQUIVO "VISUAL_USUARIO.PHP" ..	58
	APÊNDICE F – CÓDIGO FONTE ARQUIVO "VALIDA_LOGIN.PHP"	61
	APÊNDICE G – CÓDIGO FONTE ARQUIVO "USER.PHP"	63
	APÊNDICE H – CÓDIGO FONTE ARQUIVO "SEGURANCA.PHP"	65
	APÊNDICE I – CÓDIGO FONTE ARQUIVO "SAIR.PHP"	66
	APÊNDICE J – CÓDIGO FONTE ARQUIVO "MENU_ADMIN.PHP"	67
	APÊNDICE K – CÓDIGO FONTE ARQUIVO "LOGIN.PHP"	69
	APÊNDICE L – CÓDIGO FONTE ARQUIVO "LISTAR_USUARIO.PHP" ...	71
	APÊNDICE M – CÓDIGO FONTE ARQUIVO "INDEX.PHP"	74
	APÊNDICE N – CÓDIGO FONTE ARQUIVO "EDITAR_USUARIO.PHP" ..	75
	APÊNDICE O – CÓDIGO FONTE ARQUIVO "CONEXAO.PHP"	78
	APÊNDICE P – CÓDIGO FONTE ARQUIVO "CAD_USUARIO.PHP"	79
	APÊNDICE Q – CÓDIGO FONTE ARQUIVO "CARTAO.PHP"	82
	APÊNDICE R – CÓDIGO FONTE ARQUIVO	
	"PROC_EDIT_USUARIO.PHP"	83
	APÊNDICE S – CÓDIGO FONTE ARQUIVO	
	"PROC_CAD_USUARIO.PHP"	85
	APÊNDICE T – CÓDIGO FONTE ARQUIVO	
	"PROC_APAGAR_USUARIO.PHP"	87
	APÊNDICE U – ARQUIVOS JAVASCRIPT	89

1 INTRODUÇÃO

Domótica, como também é conhecida a automação residencial, corresponde à integração de serviços e tecnologias, com o propósito de automatizar as residências possibilitando segurança e adicionar conforto ao usuário. A palavra domótica tem origem do latim “Domus”, que significa casa, que junto com a palavra “Robótica” significa automação e controle de qualquer processo. Auxiliada pelos avanços da tecnologia, a área da domótica está em constante evolução (BOTELHO, 2005).

Nos últimos anos ocorreu um aumento das chamadas "Casas Inteligentes" tendo um aumento significativo. Conforme Figura 1.1, em um estudo realizado pela Berg Insight em 2013, na América do Norte equipamentos que transformam casas em Casas Inteligentes tiveram um destaque com 2,3 milhões de instalações no ano, com expectativa de crescimento em 66% ao ano.

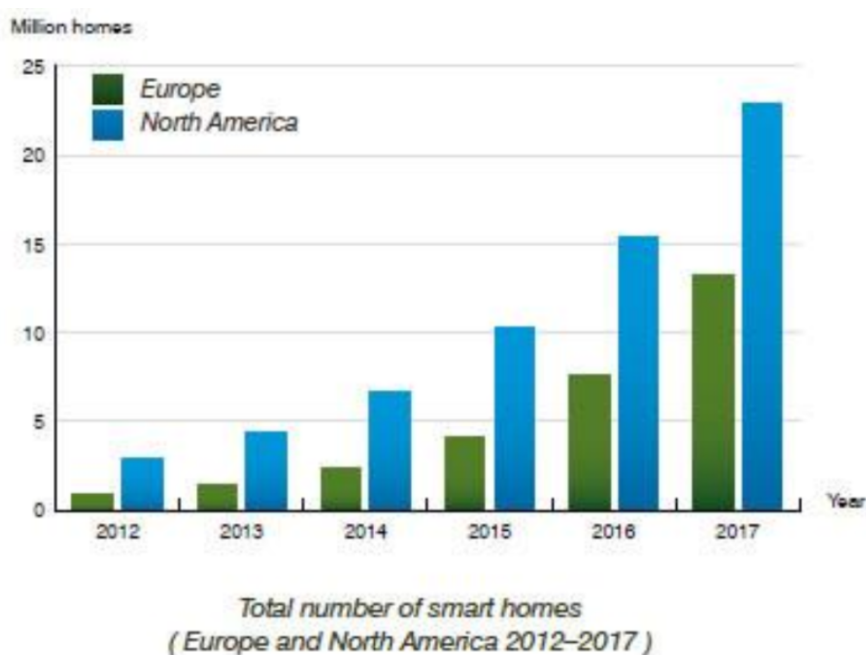


Figura 1.1 – Estimativa de Casas Inteligentes na América do Norte e Europa entre 2012 e 2017

Fonte: Berg Insight

Os SARs (Sistemas de Automação Residenciais), são compostos por micro controladores, sensores/atuadores e uma interface que possibilita controle sobre o ambiente. Este trabalho, voltado para o tema de automação residencial, busca trazer o equilíbrio entre segurança e conforto aos usuários, de forma prática e simples através de um sistema que possa ser acessado de múltiplas plataformas

1.1 OBJETIVOS

O principal objetivo deste trabalho foi desenvolver um sistema de controle de acesso residencial tendo como base *software* e *hardware* livre, de baixo custo e com gerenciamento através de uma interface Web, também desenvolver um laboratório de testes para ter uma idéia mais abrangente das dificuldade que podem ser encontradas durante a implementação do sistema em uma residência.

Conforme Figura 1.2, pode-se ter um entendimento do funcionamento do projeto, observa-se na parte da direita da figura, os sensores que serão responsáveis pela aquisição de informações e uma placa atuadora que será responsável por fechar os circuitos das lâmpadas da residência. Estes estão conectados a um micro controlador, que será responsável por interpretar e gerenciar as ações dos atuadores e/ou sensores. Este micro controlador comunica-se com o computador através de uma rede, onde encontra-se o SAR (Sistema de Automação Residencial), responsável pela interação do usuário com o sistema e gerenciar os usuários bem como enviar comandos para o Arduino. O SAR, quando recebe a informação de um cartão lido ele consulta este cartão no BD (Banco de Dados), caso o cartão esteja cadastrado, é enviado um comando para o Arduino de abertura da porta, caso não localize, o SAR armazena este cartão em uma tabela separada, com a *tag* do cartão, data e hora que foram passados e retorna com a mensagem de cartão não autorizado e quando o SAR recebe uma informação de interação com o sistema via *tablet*, celular ou computador, é enviado ao Arduino o comando solicitado.

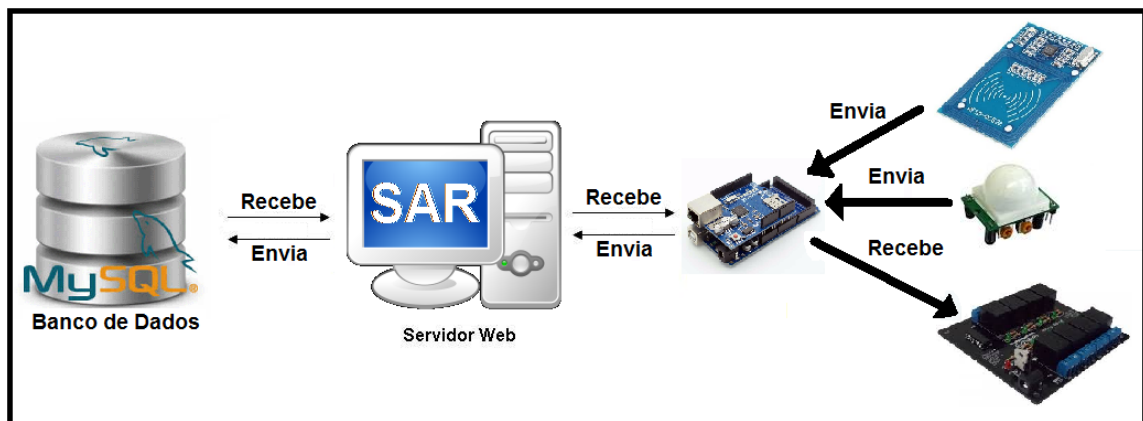


Figura 1.2 – Trabalho proposto

FONTE: Autor

1.1.1 Objetivos Específicos

- Desenvolver um sistema para a leitura e controle de sensores e/ou atuadores externos, responsáveis pelo acendimento de lâmpadas e abertura de portas;

- Desenvolver uma interface que interaja com o micro controlador, possibilitando sua consulta a uma base de dados onde serão armazenados os dados de acesso dos usuários;
- Desenvolver um sistema que possibilite ao utilizador o gerenciamento de todo projeto através de uma interface Web, possibilitando o acesso tanto por dispositivos móveis quanto de seu computador pessoal.

1.2 JUSTIFICATIVA

O Instituto Brasileiro de Geografia e Estatística (IBGE), realizou um estudo em 2009 evidenciando que 47,2% da população brasileira com 10 anos ou mais de idade não se sente segura na cidade em que vive, 21,4% da população se sente insegura no domicílio onde reside (IBGE, 2010). Estas porcentagens retratam uma tendência das famílias na busca por alternativas que reforcem a segurança de seu domicílio.

Neste sentido, um dos motivos que levou ao desenvolvimento deste trabalho é oferecer os subsídios necessários ao pesquisador entusiasta para que o mesmo tenha condições de utilizar estas novas tecnologias e implementar uma solução SAR de baixo custo, utilizando-se de apenas tecnologias *open source* em sua residência.

1.3 ESTRUTURA DO TRABALHO

Os capítulos seguintes estão estruturados da seguinte forma: O Capítulo 2 a revisão bibliográfica desde trabalho. No Capítulo 3 materiais e métodos para a confecção do trabalho. No Capítulo 4 são apresentados os testes de estabilidade do sistema, no Capítulo 5 é apresentado o laboratório de testes e resultados. Por fim, no Capítulo 6 serão apresentados as conclusões como também algumas sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta alguns conceitos importantes e definições relacionadas para melhor compreensão do sistema proposto. São apresentados aspectos tanto de *hardware* quanto de *software* que são utilizados na implementação do trabalho, como: o micro controlador; sensor; atuador e as ferramentas de desenvolvimento *web* utilizada.

2.1 MICRO CONTROLADOR

Um micro controlador combina *hardware* constituído de um circuito integrado (CI) com processador, memória e portas de entrada e saída, é o responsável por interpretar as instruções, via *software* (geralmente, em C ou *Assembly*), inseridas em sua memória.

A maioria dos CIs encontrados no mercado, possuem uma programação específica, executando rigorosamente as funções as para as quais foram programadas quando produzidos. Além dos CIs, temos os micro controladores PIC (*Programmable Interface Controller - PIC*) como também os ARM (*Advanced RISC Machine*) que também são programáveis, entretanto para sua programação são necessários conhecimento em uma destas linguagens, Pascal, Visual Basic, C ou *Assembly*, e também possuir um *hardware* gravador. A plataforma Arduino utilizada neste trabalho diferencia-se dos demais por ser de programação simples, usando a linguagem C, e de fácil expansão.

2.2 SENSOR

Um sensor é a forma como os equipamentos eletrônicos tem em capturar informações do mundo exterior. São responsáveis por alimentar o sistema, convertendo uma grandeza física como pressão, temperatura e movimento em um sinal elétrico capaz de ser interpretado pelo sistema ao qual está ligado.

Conforme Figura 2.1, os sinais gerados pelos sensores podem ser analógicos ou digitais. O valor da saída dos sensores analógicos oscila conforme o meio externo que atua sobre ele. Sensores digitais normalmente utilizam digital binário, assumindo dois valores distintos, zero ou um.

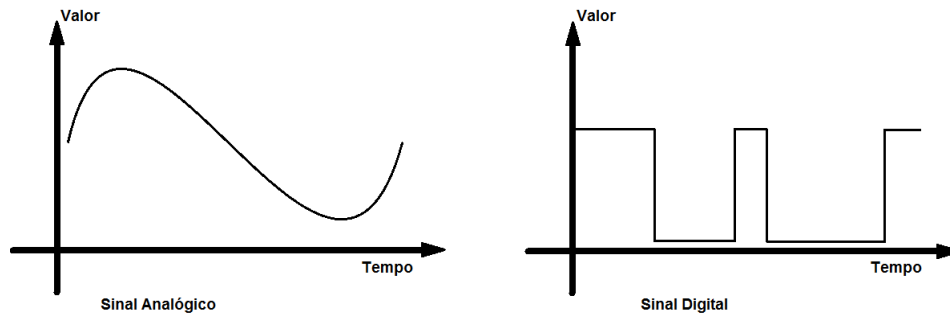


Figura 2.1 – Sinal analógico e sinal digital

FONTE: Autor

2.2.1 Sensor infravermelho passivo

Um sensor PIR (*Passive Infrared Sensor*), funciona por detecção de reflexão de irradiação infravermelha, para detectar movimento é necessário alterações dentro da faixa de operação do feixe infravermelho emitido pelo sensor que ocorre por exemplo, quando uma pessoa atinja seu raio de alcance.

2.3 DISPLAY DE CRISTAL LÍQUIDO

Utilizado para interação com os usuários, um *display* LCD (no português *Display* de Cristal Líquido) consiste de um polarizador líquido, eletricamente controlado, inserido dentro de duas lâminas, normalmente de vidro, polarizadas. Neste trabalho é utilizado para exibir informações de temperatura ambiente e umidade relativa do ar e, informar ao usuário a liberação ou não da porta de acesso a sua residência.

2.4 IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA

Considerada uma tecnologia promissora que é utilizada desde a segunda guerra mundial, a Identificação por Rádio Frequência (RFID *Radio-Frequency IDentification*) é uma propagação de dados sem fio. Seu primeiro registro de patente ocorreu no ano de 1973, porém, apenas recentemente que esta tecnologia está se popularizando, possibilitando um grande potencial na utilização em diversos segmentos, como automação residencial, industrial e também no setor hospitalar.

O identificador é representado por uma etiqueta (ou dispositivo) RFID onde são armazenados os dados de identificação ao qual aquela etiqueta está associada. Possui um *microchip* em que ficam armazenadas as informações e uma antena para o envio dos

dados, quando solicitado. Na Figura 2.2 é exibida uma *tag* RFID, tendo com destaque ao centro seu *microchip* e nas linhas ao seu entorno sua antena.

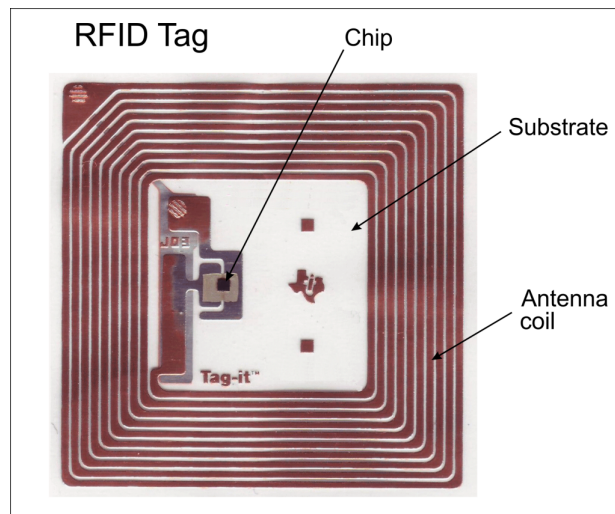


Figura 2.2 – Tag RFID

FONTE: <http://endtimetruth.com/mark-of-the-beast/rfid/>

2.4.1 Tipo de operação

Atualmente existem 4 tipos de sistema *RFID*. Esta divisão ocorre de acordo com o tipo de fonte de energia e como as *tags* fazem a comunicação com a antena do leitor, podendo ser classificadas em: identificador passivo, identificador semi-passivo, identificador ativo e identificador de duas vias (Glover:2007).

Para o projeto optou-se pelo identificador passivo pelo fato de não utilizar bateria, obtendo sua energia de ondas eletromagnéticas enviadas pela antena do leitor. As *tags*, como são chamados os identificadores RFID passivos, apenas refletem o sinal emitido pela antena do leitor. Considerado o tipo mais comum e por não possuir bateria, sua produção se torna barata, e sua durabilidade de grande prazo. Já os identificadores semi-passivos utilizam-se de uma bateria para auxiliar na transmissão dos dados armazenadas nos mesmos. Os identificadores ativos possuem uma bateria que auxilia na transmissão dos dados como também alimenta o circuito interno. Os identificadores de duas vias tem as mesmas características dos identificadores ativos, porém elas também podem trocar informações com outros identificadores do mesmo tipo de operação.

2.4.2 Frequência de operação

O que define a taxa de transferência de dados entre a etiqueta e o leitor, é a frequência, mas a velocidade não é o único aspecto a ser analisado em uma solução RFID.

Segundo Glover:2007, as frequências estão contígua as bandas denominadas ISM (*Industrial-Scientific-Medical*). Tendo como propriedades diferentes Conforme Tabela 2.1, as de frequência mais baixas podem viajar pela água, já as mais altas podem carregar mais informações tendo sua leitura facilitada.

Tabela 2.1 – Faixas de frequência RFID

Nome	Faixa de frequência	Frequência ISM
LF	30.0 - 300.0 kHz	< 135.0 kHz
HF	3.0 - 30.0 MHz	6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz
UHF	300.0 MHz - 3.0 GHz	433.920 MHz, 869.0 MHz, 915.0 MHz
Micro-ondas	> 3.0 GHz	2.45 GHz, 5.8 GHz, 24.125 GHz

2.4.3 Memória

Como última característica a ser considerada em uma *tag* RFID, a memória pode ser classificada em três tipos: *Read Only* (RO); *Write Once - Read Many* (WORM) e *Read-Write* (RW): Respectivamente, segundo Gomes:2007, a primeira, *tag* apenas permite leitura, os dados são gravados apenas uma vez, normalmente pelo fabricante e são muito utilizadas para a identificação de roupas ou livros em bibliotecas. No segundo tipo de memória, os dados são gravados apenas uma vez e pode-se efetuar várias leituras, mas na prática, as *tags* WORM podem efetuar a gravação de dados mais de uma vez, porém se o número de gravações for muito elevado, pode-se causar danos a memória. Já o último tipo, as *tags* RW são do tipo mais versátil, pois aceitam inúmeras gravações. As vantagens são imensas, pois permitem receber atualizações constantes nos dados sempre que necessário.

2.5 DESENVOLVIMENTO WEB

Nesta parte do trabalho são apresentados os recursos utilizados na implementação *web* e também as informações que poderão ser acessadas pela internet.

2.5.1 Servidor de páginas Web

Segundo Klabunde:2007, um servidor *web*, é um serviço capaz de responder as solicitações de um cliente HTTP (*Hyper Text Transfer Protocol*) com a página solicitada. Segundo o site Netcraft, atualmente o servidor *web* mais utilizado no mundo é o servidor Apache, o qual será utilizado neste trabalho, conforme características descritas na tabela a seguir:

Tabela 2.2 – Principais características Servidor Web Apache

Principais características
Suporte para plataformas <i>open source</i> e proprietárias.
Sem custos para sua utilização.
Estruturação em módulos.
Suporte a várias linguagens: PHP, PERL, HTML, PYTHON.
Controle de acesso e encriptação utilizando certificados digitais, SSL, entre outros.

FONTE:<http://www.apache.org/>

2.5.2 Sistema de gerenciamento de banco de dados

Sistema de Gerenciamento de Banco de Dados (SGBD) é um sistema que tem como principal característica gerenciar e otimizar o acesso a os dados nele armazenados.

No final de 1960 surgiu o primeiro Sistema Gerenciador de Banco de Dados (SGBD) baseando-se nos sistemas primitivos existentes na época. Com o passar do tempo, os SGBDs utilizaram-se de diferentes formas de representação, ou modelos de dados para descrever as estruturas das informações contidas em seus banco de dados (BD).

Segundo Takai:2005, na atualidade os SGBDs usam estes modelos de dados: modelo relacional (largamente usado), modelo em redes, modelo hierárquico e o modelo orientado a objetos. Optou-se por usar o MySQL pelo fato de possuir uma extensa documentação disponível na internet e também pelas características a seguir: Suporte para plataformas *open source* e proprietárias, Sem custos para sua utilização, Controle de acesso e encriptação utilizando certificados digitais, SSL, entre outros, Compatibilidade com várias linguagens, entre elas, Java, PHP, Python, Ruby, e C/C++, Base de dados de tamanho ilimitado, Tabelas com tamanho de até 32 TB, Quantidade de linhas de até 1.6 TB.

2.5.3 Protocolo HTTP

A troca de informações entre o cliente e o servidor caracteriza uma comunicação e, da mesma forma que na comunicação interpessoal, é essencial a cordialidade entre

ambas as partes. Nesse sentido, o protocolo HTTP estabelece regra de procedimento tanto para as requisições como para as respostas. O cabeçalho utilizado no protocolo HTTP contém informações complementares de uso exclusivo do servidor e navegador. Através dos métodos é possível realizar requisições como transmitir informações.

O protocolo HTTP (*Hypertext Transfer Protocol*) é o protocolo utilizado para a transferência de páginas HTML (*HyperText Markup Language*). O protocolo trabalha com requisições e respostas entre cliente e servidor. As requisições que o navegador envia são chamadas de métodos, segundo Emer:2014, os métodos mais utilizados no protocolo são:

Tabela 2.3 – Características dos métodos mais utilizados no protocolo HTTP

Método	Característica
GET	Método utilizado pelo navegador ao servidor para fazer uma solicitação de recurso (página HTML, imagem, gif, documento de texto ...) (Figura 4.7);
POST	Envia dados ao servidor (dados de um formulário HTML, <i>upload</i> de um arquivo
DELETE	Método utilizado para remove um recurso;
PUT	Método utilizado para atualizar um recurso na URL especificada
HEAD	Método utilizado para retornar informações sobre um recurso, funciona semelhante ao método GET, porém este não retorna o recurso no corpo da requisição.

O navegador realiza uma requisição enviando um pacote de informações que contém cabeçalhos (*headers*) com algumas informações e como a URL (*Uniform Resource Locator*). A figura 2.3 é demonstrado o funcionamento básico do protocolo HTTP.

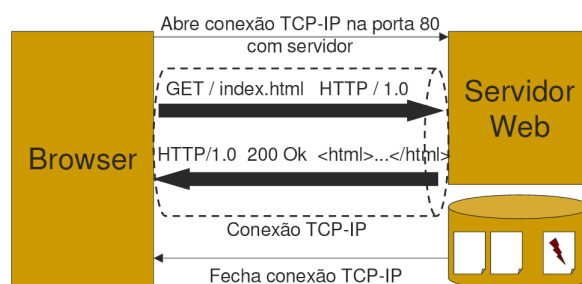


Figura 2.3 – Funcionamento básico do protocolo HTTP

FONTE: http://cin.ufpe.br/erp/DesenvWeb/aulas/http_servlet/http.pdf

Para toda a requisição feita pelo cliente, existe uma resposta (*status*), na RFC 2616 publicada em junho de 1999, elas estão mais detalhadas.

2.5.4 PHP - Hypertext Preprocessor

Optou-se pela linguagem PHP por ela ser de código aberto, versátil e estar recebendo atualizações constantes. A construção de conteúdo dinâmico é simplificada, tem fácil integração com a maioria dos banco de dados no mercado.

Juntamente com a linguagem PHP, utilizou-se do *framework* Bootstrap versão 3.3.6, por ser o mais popular *framework* HTML, CSS e JS para desenvolvimento de projetos responsivos com ênfase para dispositivos móveis, toda a parte gráfica do sistema *web* foi baseada nele. É uma ótima ferramenta para quem não tem muito conhecimento nesta parte, ele é um *framework* de aparência elegante, poderoso intuitivo e o principal, responsivo.

2.6 SISTEMAS EXISTENTES NO MERCADO

Durante a pesquisa, foram encontrados alguns sistemas já existentes como por exemplo o sistema da empresa Iluflex¹, que trabalha tanto com soluções residencias como também soluções corporativas. A empresa trabalha com uma linha de produtos própria, com comunicação por RF (Rádio Frequência) ou cabeada. Como a comunicação por RF é utilizada por outros equipamentos, podem haver interferências entre os sistemas. Suas centrais de comandos funcionam somente com equipamentos da própria empresa, sendo assim, ao usuário que adquirir o produto, caso queira efetuar alguma alteração ou ampliação do seu sistema, estará cativo a adquirir somente produtos desta marca. Por utilizar RF em sua comunicação entre os dispositivos atuadores, é necessária a instalação de alguns repetidores de sinal para que seu alcance amplie. Caso o usuário queira automatizar a abertura da porta de entrada será necessário a instalação de uma interface de comunicação com um controle remoto semelhante aos controles de portões de garagem ou alarmes.

Da mesma maneira, durante a pesquisa, foi localizada a empresa iHouse² que também trabalha no ramo de automação residencial, possui uma linha de produtos própria, apesar disso, o sistema é mais flexível, pois permite a integração de câmeras IP em seu sistema, assim, quando acessado remotamente pode-se visualizar as imagens em tempo real mas não faz a gravação das imagens. Seus módulos utilizam o padrão de comunicação ZigBee³, facilitando assim a instalação dos mesmos. Os dispositivos quando instalados criam de uma rede *mesh* para a comunicação entre eles e, como a cobertura de atuação deste padrão é bastante amplo, sua instalação é mais simplificada, não necessitando de replicadores de sinal para atender uma área maior, porém demanda a atenção de alguém especializado para a configuração dos equipamentos com *software* próprio e

¹Iluflex - <http://www.iluflex.com.br/>

²iHouse - <http://www.ihouse.com.br/>

³ZigBee - Padrão de comunicação sem fio que utiliza baixa potência, bastante utilizado na industria de energia

treinamento específico na área para a configuração dos mesmos. Para a entrada da residência, o sistema possui um módulo de leitura biométrica, simplificando a entrada da residência sem a necessidade de chave.

Outro sistema de automação é o da empresa SIMPLIFIES SimpleHome⁴, dentre os pesquisados é o sistema mais simples, como os citados acima, também usa plataforma fechada, a comunicação entre os módulos é feita através do protocolo *ethernet*, suas funções não são muito abrangentes, pois seus módulos são limitados a 3 ou 6 atuadores. O de 6 atuadores ainda podem controlar a intensidade de uma lâmpada, desde que adquirido juntamente com um módulo adicional. Para a entrada da residência, este sistema não possui nenhum adaptador específico como leitor de biometria ou RFID.

2.6.1 Comparação entre os sistemas

Com base na pesquisa realizada, observa-se que a instalação de um sistema de automação residencial por uma empresa onde a plataforma não é aberta ainda está fora da nossa realidade, por exemplo, o sistema da empresa iHouse, para automatizar somente a sala de estar juntamente com o sistema da porta de entrada da residência, o sistema parte de R\$ 6000,00 podendo chegar ao valor de R\$ 100.000,00 de acordo com os dispositivos desejados. A empresa SIMPLIFIES SimpleHome não respondeu ao e-mail solicitando valores para a instalação de um sistema semelhante ao do projeto. A empresa Iluflex respondeu ao e-mail enviado e o orçamento para um sistema semelhante sai em torno de R\$ 7000,00, e esta empresa possui um representante aqui em Santa Maria, já o projeto proposto, teve um custo de R\$ 455,00, conforme Tabela 2.4, em equipamentos gastos para montar o laboratório de testes

Tabela 2.4 – Custos para montagem do laboratório de testes

Material	Valor	Material	Valor
Arduino Mega 2560	R\$ 89,90.	Leitor RFID	R\$ 44,90
Shield Ethernet	R\$ 59,90	Display 20 x 4	R\$ 59,90
Fonte Protoboard	R\$ 14,90	Sensor Temperatura	R\$ 39,90
Fonte 12V x 1A	R\$ 19,90	Relay Board	R\$ 79,90
Sensor de movimento	R\$ 16,90	Servo Motor	R\$ 18,90
Protoboard	R\$ 21,90	Cabos	R\$ 10,00
		TOTAL	R\$ 455,00

⁴SIMPLIFIES SimpleHome - <http://www.simplifies.com.br>

3 MATERIAIS E MÉTODOS

Neste capítulo é apresentada a metodologia utilizada neste trabalho, bem como uma descrição dos procedimentos. É apresentado o *hardware* utilizado, com suas características de funcionamento, e também os recursos escolhidas para o desenvolvimento *web*.

3.1 METODOLOGIA

Para o desenvolvimento deste trabalho, optou-se pela utilização dos seguintes componentes: Plataforma de desenvolvimento Arduino MEGA 2560; *Ethernet Shield* modelo W5100; Leitor RFID da marca Mifare, modelo RC522; Relay Board incubado pelo Laboratório de Garagem com 8 relés; Sensor PIR modelo DYP-ME003; Sensor de temperatura DHT-22; Display LCD 20x4; Banco de dados MySQL 4.1.14 Linguagem de programação *web*: PHP versão 5.6.14.

Após a definição das tecnologias, na segunda fase do trabalho, desenvolveu-se o algoritmo para o funcionamento do *hardware* de acordo com o projeto proposto. Em seguida, foi desenvolvido um sistema *web*, permitindo que o usuário interaja com sua residência. Por fim, na quarta e última fase do trabalho, realizou-se os testes necessários em laboratório para verificar seu funcionamento.

3.2 HARDWARE UTILIZADO

Como observado na pesquisa, a maioria dos micro controladores disponíveis no mercado não possuem uma plataforma de desenvolvimento, sendo somente o micro controlador sem nenhum outro componente agregado a si, sendo necessário a confecção de uma plataforma para a sua utilização com fonte de alimentação, suporte para outros componentes e suas respectivas entradas e saídas. Como o objetivo deste trabalho não é montar uma plataforma de desenvolvimento e sim, elaborar um sistema de automação, de baixo custo e de tecnologia aberta, optou-se em utilizar uma plataforma que não é necessária a confecção de uma placa para que possamos utilizá-la e consolidada no mercado.

Com isto, definiu-se pela plataforma a ser utilizada será a Arduino por apresentar todos os componentes necessários para a sua utilização, bem como uma plataforma onde pode-se desenvolver o algoritmo para o seu funcionamento, com uma grande comunidade, recursos disponíveis sem qualquer custo.

3.2.1 Plataforma Arduino Mega 2560

Conforme figura 3.1, a plataforma Arduino Mega 2560 é bem compacta com suas dimensões de 10 cm x 5,25 cm, podendo ser instalada em diversos lugares.

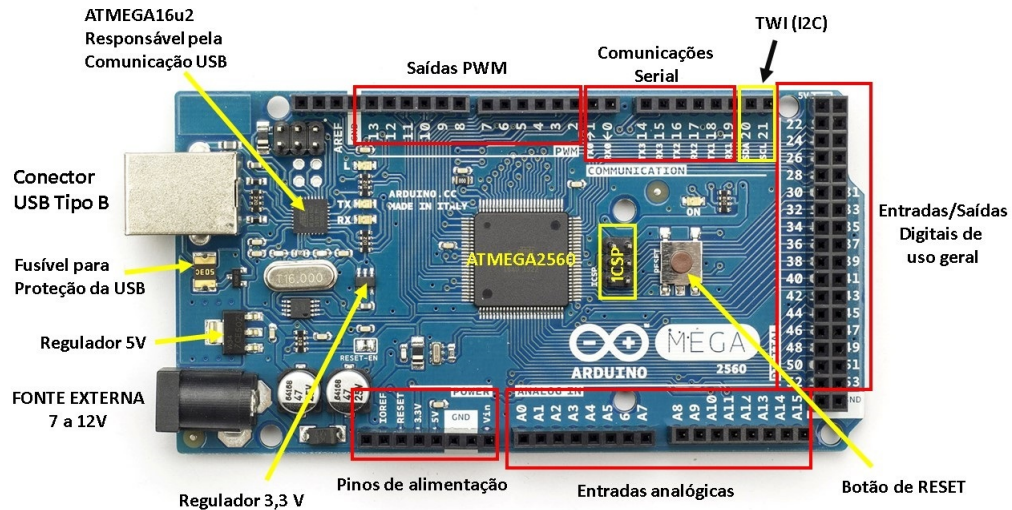


Figura 3.1 – Plataforma Arduino Mega 2560

FONTE: <http://www.embarcados.com.br/arduino-mega-2560/>

Dentre os modelos disponíveis no mercado da mesma plataforma, foi definido que o modelo a ser utilizado no projeto será o Arduino Mega 2560 por apresentar o maior número de portas entre os modelos, sendo assim maior a possibilidade de expansão.

3.2.2 Adaptador ethernet shield W5100 Arduino

Além dos recursos que estão disponíveis originalmente no Arduino Mega 2560, existem inúmeros adaptadores e acessórios que podem ser acoplados a placa denominados *shields*. Para o desenvolvimento deste trabalho, utilizou-se a *shield ethernet W5100* (Figura 3.2) responsável pelo envio e recebimento de informações para aplicação servidor.



Figura 3.2 – Ethernet shield W5100 Arduino

FONTE: <http://www.filipeflop.com/pd-6b62c-ethernet-shield-w5100-para-arduino.html>

A plataforma Arduino MEGA 2560 é compatível com todas as *shields* disponíveis no mercado, não sendo necessário nenhum tipo de adaptação para a comunicação com a *shield ethernet* (Figura 3.3 mostra o detalhe do barramento SPI) proposta, como é necessário em outras versões da plataforma.

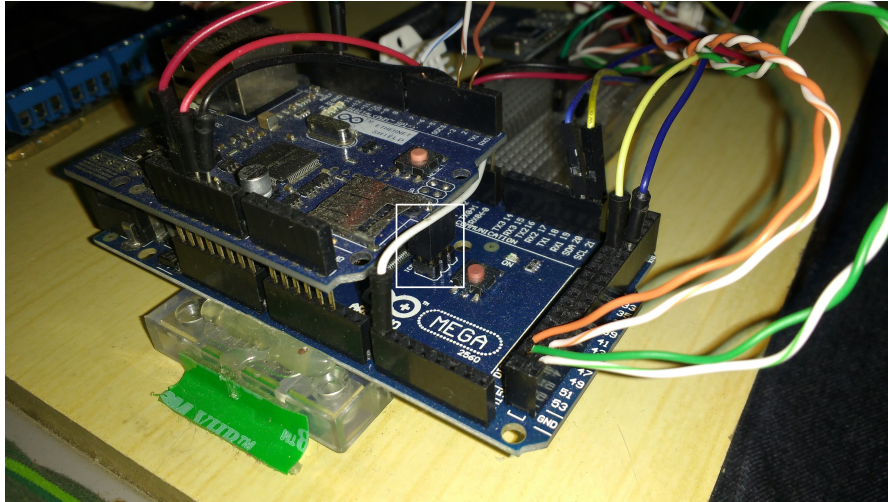


Figura 3.3 – Arduino MEGA 2560 juntamente com o módulo Shield Ethernet W5100
 FONTE: Autor

Esta *shield* utiliza-se da interface SPI (*Serial Peripheral Interface*) para a comunicação com a plataforma.

Desenvolvido originalmente pela Motorola, devido a sua simplicidade e popularidade do barramento SPI, que outras empresas adotaram este mesmo módulo, é utilizado principalmente na intercomunicação entre os micro controladores e seus periféricos.

O barramento SPI é *full-duplex*, isto é, transmite e recebe informações ao mesmo tempo. Consiste em três linhas de transmissão de informações de 8bits e uma quarta linha que seleciona qual dos dispositivos *slave* será ativado.

3.2.3 Leitor RFID Mifare modelo RC522

O leitor que foi utilizado neste trabalho é o fabricado pela MIFARE, modelo RC522 (Figura 3.4), que trabalha na frequência 13.56 Mhz. Foi escolhido este leitor por se tratar de um leitor bastante utilizado comercialmente, como em bilhetes de passagem por exemplo (O Sistema Integrado Municipal (SIM) de Santa Maria utiliza esta mesma frequência, podendo assim, usar o seu cartão de passagem para entrar em casa), fácil localização no mercado e ser bem maleável quanto a sua utilização, podendo usar protocolos criptografados.



Figura 3.4 – Módulo RFID Mifare RC522

FONTE: <http://www.filipeflop.com/pd-6b883-kit-modulo-leitor-rfid-mfrc522-mifare.html?ct=41d98p=1s=1>

3.2.4 Sensor PIR modelo DYP-ME003

Utilizou-se o sensor PIR (*Passive Infrared Sensor*) (Figura 3.5), que segundo a pesquisa realizada a respeito dos dispositivos que seriam utilizados nele, atende as necessidades do trabalho bem como proporciona segurança e comodidade para o usuário, além de ser utilizado na maioria dos projetos Arduino.



Figura 3.5 – Imagem do Sensor DYP-ME003

FONTE:

<http://www.arduinoocia.com.br/2014/06/sensor-presenca-modulo-pir-dyp-me003.html>

Este tipo de sensor detecta o movimento quando ocorrem alterações nas duas faixas do feixe infravermelho em que o sensor transmite, acionando assim sua porta de saída. Para aumentar seu campo de atuação é utilizado uma lente *fresnel*, onde a luz captada é direcionada diretamente ao sensor, conforme figura 3.6.

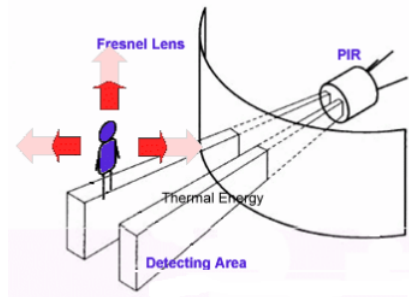


Figura 3.6 – Reprodução do funcionamento do sensor PIR
 FONTE: <http://electronicsgurukulam.blogspot.com.br/>

3.2.5 Sensor de umidade relativa do ar e temperatura ambiente DHT 22

Optou-se em adicionar informações que exibam no *display* LCD a temperatura ambiente e a umidade relativa do ar em tempo real. O sensor DHT-22 foi escolhido devido a sua maior precisão e possuir uma faixa de operação maior em relação a outros sensores, conforme Tabela 3.1. Em seus componentes internos consistem um sensor capacitivo de umidade e um termistor utilizado para medir a temperatura ambiente do ar que circula dentro do sensor, transformando essas informações em um sinal digital, enviadas para a porta de saída.

Tabela 3.1 – Comparativo entre os sensores DHT11 e DHT22

	DHT11	DHT22
Alimentação	3 -5,5V	3,3 - 6V
Faixa de leitura - Umidade	20-80%	0-100%
Precisão - Umidade	5%	5%
Faixa de Leitura - Temperatura	0-50°C	-40-125°C
Precisão - Temperatura	+/- 2°C	+/- 0,5°C
Intervalo entre medições	1s	2s

Fonte: <http://raspi.br.com.br/utilizando-sensor-de-temperatura/>.

3.2.6 Relay Board

Conforme figura Figura 3.7, utilizou-se uma Relay Board de fabricação nacional elaborada por Gustavo Barcelo Suim¹. Diferente das relés *shields* que utilizam 1 porta para cada relé, esta placa utiliza somente 3 portas para controle, permitindo controlar os 8 relés que acompanham a placa, além disso ela permite que sejam ligados mais de uma

¹Integrante do site Laboratório de Garagem, Desenvolvedor independente de ciência e tecnologia, projetou a placa Relay Board, a qual foi incubada pela Incubadora LdG. <<http://labdegaragem.com/profile/GustavoBarceloSui>>

placa, conjunto permitindo assim o número total de 80 relés, controlados com os mesmos 3 pinos.

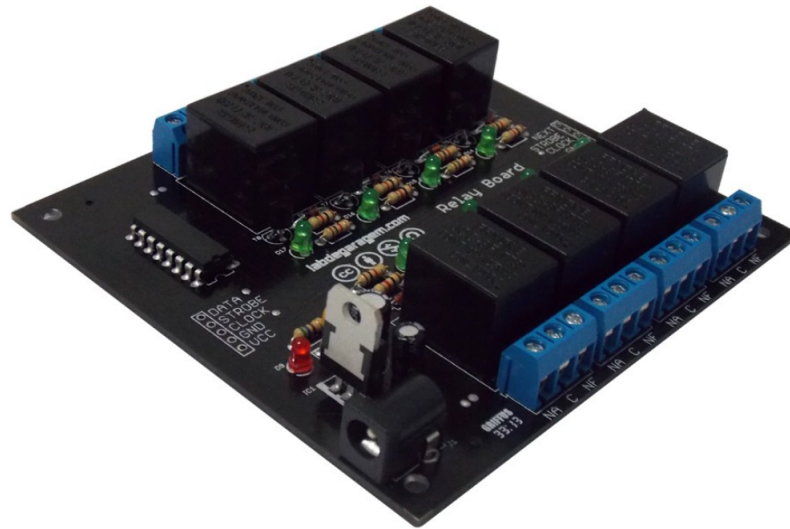


Figura 3.7 – Relay Board

FONTE: <http://labdegaragem.com/profiles/blogs/como-utilizar-a-Relay Board>

Conforme características descritas na Tabela 3.2, a placa Relay Board é responsável pelo acionamento das cargas descritas no trabalho, como iluminação e abertura de portas

Tabela 3.2 – Características Relay Board

Alimentação 12 VDC
Somente 3 portas de controle
Biblioteca de fácil utilização, com apenas 4 funções
Corrente nominal (contato do relé: 125VAC/10A - 220VAC/20A)

3.2.7 Display LCD 20x4

Conforme Figura 3.8, neste trabalho utilizou-se o *display* LCD 20x4, possibilitando 20 colunas (caracteres por linha) por 4 linhas. Sua instalação utiliza 4 portas para dados e 2 portas para controle.

Além disso, optou-se pela utilização de um componente adicional, o adaptador I2C (3.9) que pode ser utilizado para simplificar a instalação do LCD, convertendo as 6 portas originais em apenas 2, ligadas diretamente ao Arduino.

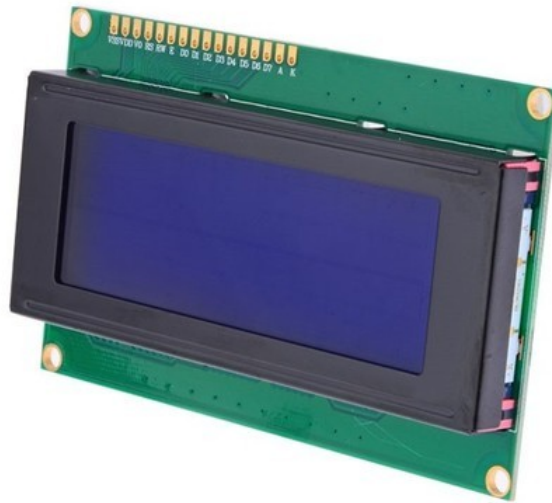


Figura 3.8 – Display 20 x 4

FONTE: <http://www.filipeflop.com/pd-6b7e6-display-lcd-20x4.html>



Figura 3.9 – Adaptador I2C

FONTE: <https://arduino-info.wikispaces.com/LCD-Blue-I2C>

3.3 SOFTWARE UTILIZADO

Nesta parte do trabalho, são apresentados os *softwares* utilizados para o desenvolvimento do mesmo.

3.3.1 IDE de desenvolvimento do algoritmo para o Arduino

O fundamental motivo da escolha pela plataforma Arduino, é a disponibilidade de uma Interface de Desenvolvimento - IDE (Figura 3.10) multiplataforma e *open source*, isto é, pode-se utilizar esta IDE em vários sistemas operacionais (SOs) distintos. Sendo ela multiplataforma, pode-se escrever os algoritmos, alterar e enviar ao Arduino de qualquer SO.

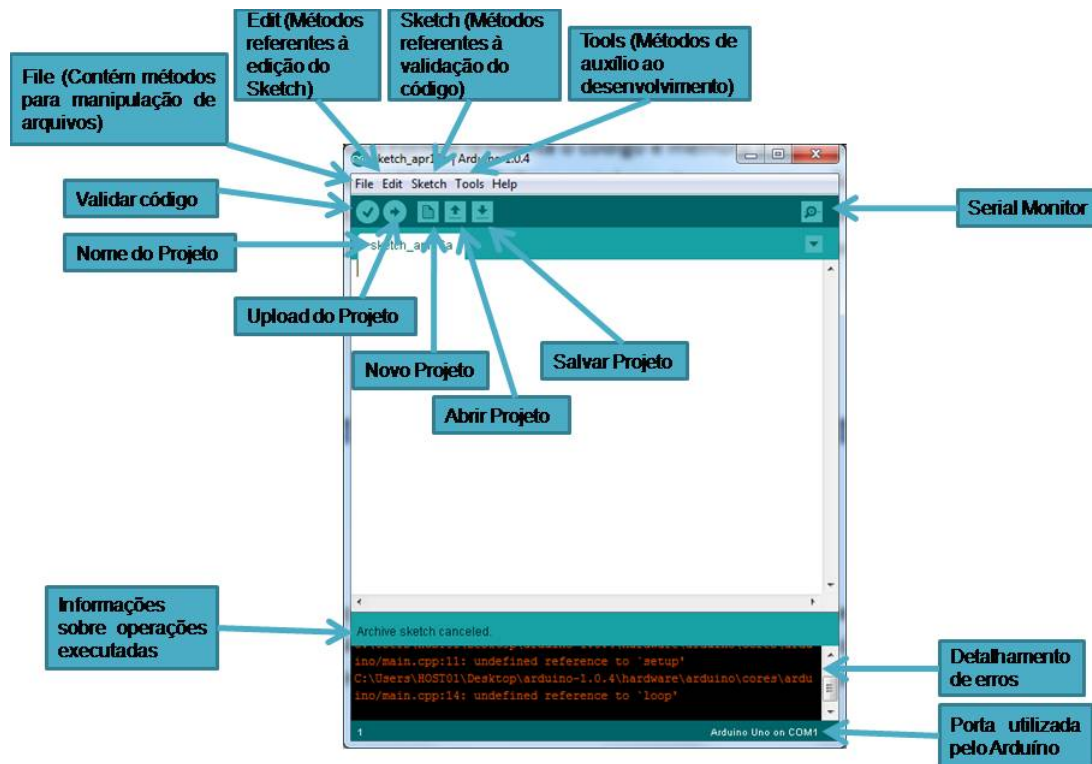


Figura 3.10 – Interface IDE Arduino

FONTE: <https://programmingelectronics.com/wp-content/uploads/2012/08/Figure-1.Arduino-Sketch-with-shorcut-keys.jpg>

3.4 DESCRIÇÃO DO FUNCIONAMENTO DO PROJETO

O projeto foi primeiramente desenvolvido para o controle de acesso residencial e uma quantidade limitada de acionamento de cargas limitadas a 8, pelo motivo de ter sido adquirido somente uma Relay Board e esta possui somente 8 relés, sendo assim, seria possível automatizar somente um ambiente da residência ao instalar o projeto proposto. Já que não foi instalado definitivamente em uma residência, as informações do sensor de temperatura são apresentadas somente no *display* de entrada da residência a título de conhecimento do funcionamento do mesmo.

A Relay Board foi escolhida por se mostrar menos onerosa a sua aquisição do que adquirir 8 *shields* relé como costumam instalar a maioria dos admiradores desta tecnologia. Para cada *shields* relé instalada, é utiliza uma porta, e com a Relay Board são utilizadas apenas 3 portas para o comando de até 80 relés, pois é possível o cascadeamento de até 10 placas iguais a ela, sendo necessário apenas informar no algoritmo a quantia de placas e suas respectivas programações.

O sistema *web* foi desenvolvido para ser o mais simples possível, mas não deixando

a desejar em estabilidade e portabilidade, já que ele pode ser acessado de praticamente qualquer dispositivo que possa visualizar páginas *web*. O SAR, como comentado anteriormente é composto de duas seções e estas estão descritas na seção 5.3 deste trabalho.

As funções exercidas por este projeto estão limitadas as características do *hardware* utilizado, podendo ser ampliado ou modificado por qualquer pessoa que queira aplicar o mesmo projeto em uma circunstância diferente, como o acesso dos funcionários de uma empresa ao seu estoque.

Como este trabalho tem fins acadêmicos e sua posterior instalação na residência do autor do projeto, optou-se em limitar o número de cartões RFID em três para cada usuário, pois caso aconteça de um dos usuários extrair um de seus cartões, fica mais fácil a identificação do mesmo e a sua consequente exclusão do SAR.

4 RESULTADOS

Este capítulo apresenta os resultados dos testes realizados em laboratório com todos os dispositivos descritos no decorrer do texto. Na primeira parte será apresentado o laboratório de testes, em seguida, o sistema *web*. Por último a união do *hardware* e *software*, formando assim o sistema SAR.

4.1 LABORATÓRIO DE TESTES

Na Figura 4.1, pode-se observar os componentes utilizados no trabalho. Para uma melhor identificação, cada componente é identificado e descrito nas sub-sessões a seguir: 1- Plataforma Arduino com o módulo *Shield Ethernet W5100*, 2- Leitor RFID, 3- Sensor de Temperatura, 4- Display 20x4, 5- Relay Board, 6- Sensor de Presença, 7- Servo motor, utilizado para representar uma porta, 8- Fonte auxiliar para alimentar o servo motor, sensor de presença e *display*.

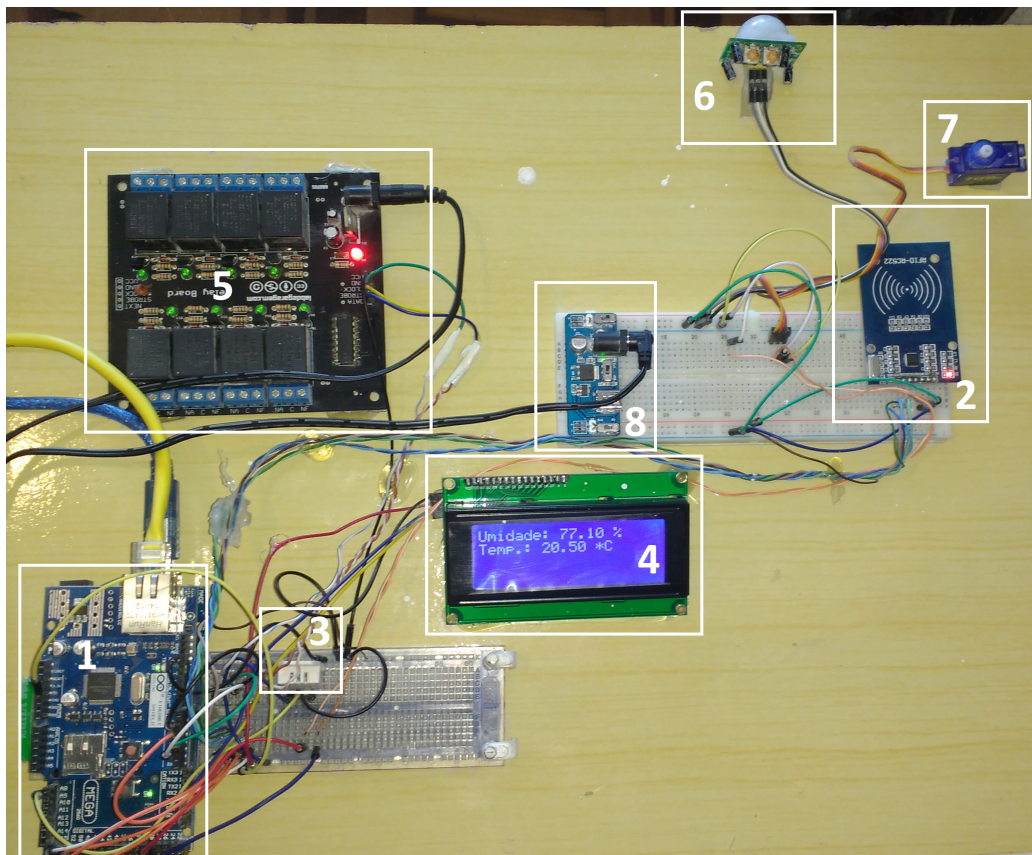


Figura 4.1 – Laboratório de testes

FONTE: Autor

4.1.1 Plataforma Arduino com o módulo *Shield Ethernet*

Utilizado como ponto central da arquitetura, a plataforma Arduino Mega 2560 é conectada com o *Shield Ethernet W5100* (Figura 4.2 que desempenha a função de servidor *web* fazendo a comunicação com o SAR. Monitorando os todos os sensores, verificando a temperatura ambiente e umidade relativa do ar, e aguarda uma mensagem na porta serial a qual a o adaptador *ethernet* está conectado.

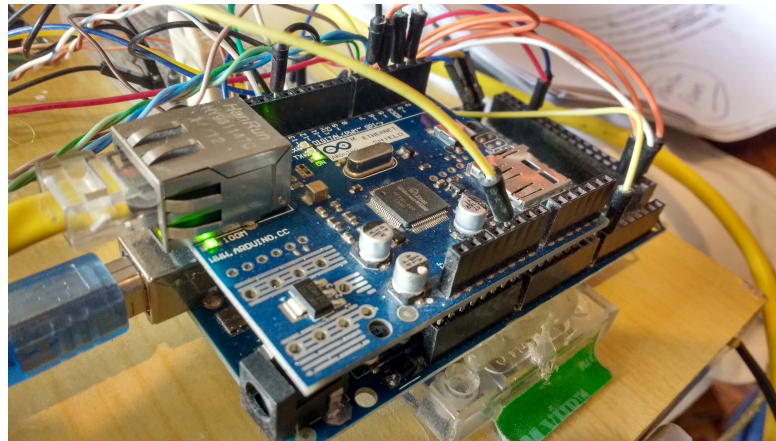


Figura 4.2 – Detalhe Arduino Mega 2560 + *Shield Ethernet W5100*
FONTE: Autor

Com o objetivo de auxiliar a instalação e suporte aos sensores e servo motor (que simula uma porta abrindo e fechando), foi utilizado uma *protoboard* ou matriz de contato, onde todos os demais componentes do sistema estão conectados.

4.1.2 Leitor RFID

Conectado a plataforma Arduino através da *protoboard*, o leitor RFID Mifare RC522 é utilizado para desempenhar a função de leitura do cartão de acesso do usuário, permitindo através deste, a abertura caso o cartão esteja cadastrado, caso não se encontre no BD dos cartões liberados para acesso, o mesmo é salvo em um BD em separado para que fique registrado a tentativa de abertura da porta por um cartão não autorizado.

4.1.3 Sensor de temperatura e umidade

Também conectado à *protoboard*, o sensor captura informações de umidade relativa do ar e temperatura ambiente AM2302 DHT22, permitindo realizar a leitura de temperaturas entre -40 a +80 graus Celsius, e umidade entre 0 a 100%, que serão apresentados no *display*, descrito a seguir.

4.1.4 Display 20x4

Conforme pode ser observado na Figura 4.3, um *display* lcd é utilizado para a interação com o usuário. É através dele que são exibidas as informações de umidade relativa do ar e a temperatura ambiente. Também retorna a informação ao utilizador quando efetuada a leitura do cartão de acesso, ou seja, a liberação ou não da porta de entrada da residência conforme resposta do servidor. Quando o utilizador encontra-se cadastrado o sistema informa que a porta será liberada, como pode-se verificar na Figura 4.3.

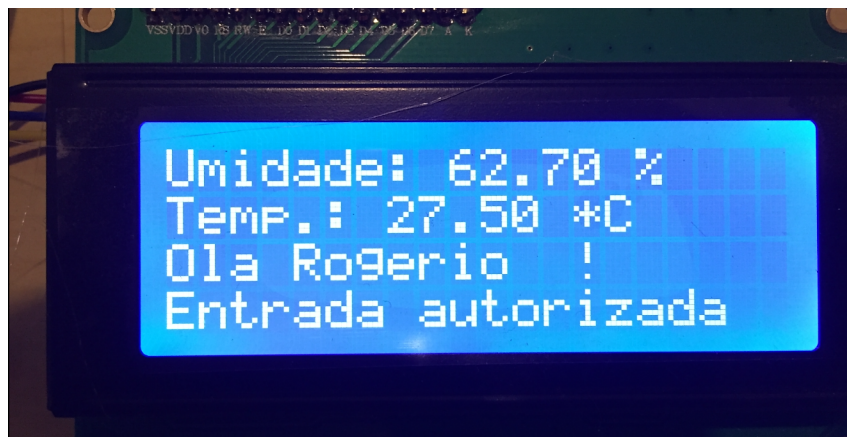


Figura 4.3 – Display indicando entrada autorizada
FONTE: Autor

4.1.5 Relay Board

Através da placa Relay Board que pode-se ativar ou desativar as lâmpadas ligadas ao sistema, assim como abrir ou fechar o portão da garagem, entre outras cargas ligadas ao mesmo. Pode-se verificar o funcionamento do sistema SAR através da Figura 4.4. Neste exemplo foram acionados os relés 6 e 8 através da interface web e os LEDs correspondentes estão ativos na placa Relay Board.

4.1.6 Sensor de presença

Quando o sensor de presença detecta movimento, ele aciona um dos relés da Relay Board, simulando assim o acionamento da lâmpada de entrada da residência, proporcionando maior segurança e conforto ao utilizador pois, quando o local instalado é pouco iluminado possibilita a melhor visualização de onde deve-se passar o cartão RFID, necessário para para abrir a porta de sua residência. Conforme Figura 4.5, pode-se visualizar melhor o funcionamento desta etapa do sistema.

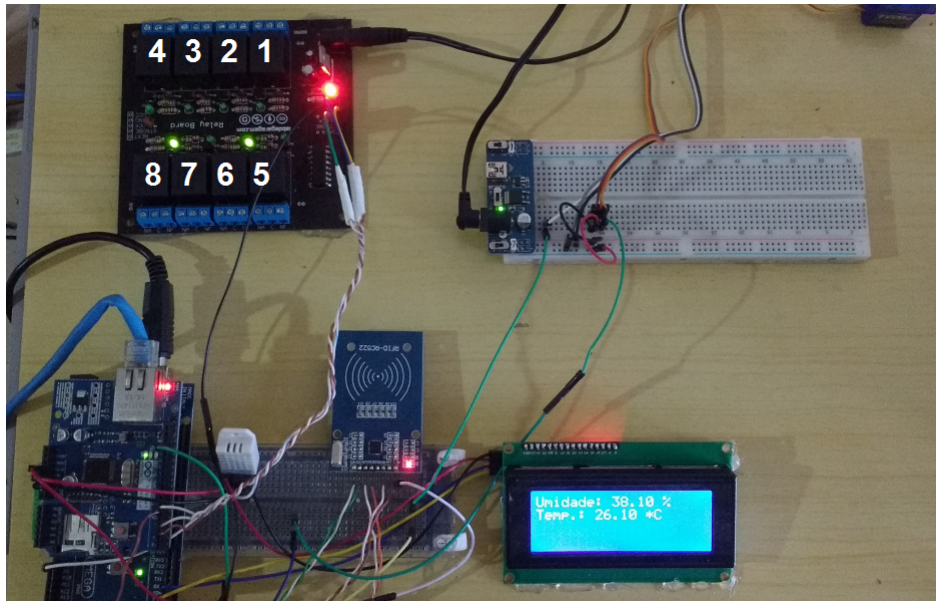


Figura 4.4 – Simulação de acionamento de cargas pelo SAR
FONTE: Autor

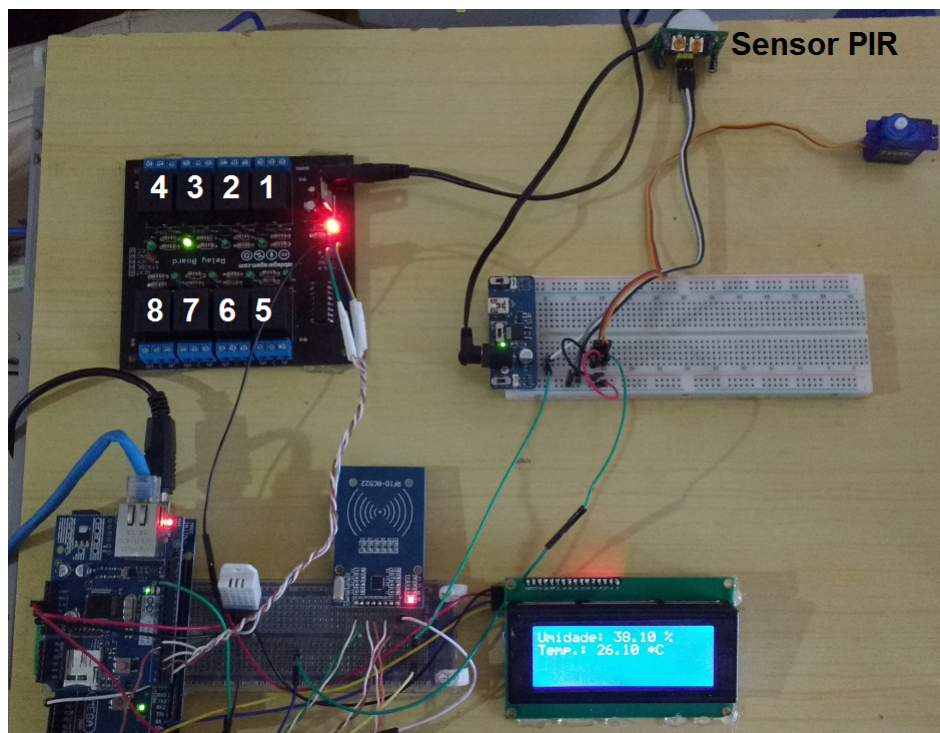


Figura 4.5 – Simulação de acionamento do sensor de presença
FONTE: Autor

4.1.7 Servo motor

Utilizado para representar a abertura e fechamento da porta de entrada da residência.

4.1.8 Fonte auxiliar

Utilizada para alimentar o servo motor, sensor de presença e *display* pois a porta USB (*Universal Serial Bus*) disponível não fornece energia suficiente para a utilização de todos os dispositivos de forma concomitante.

4.2 MODELAGEM DO BANCO DE DADOS

Para este trabalho criou-se um banco de dados para cadastro dos usuários que terão acesso ao sistema bem como os cartões de acesso a sua residência. Através do modelo entidade-relacionamento representado pela Figura 4.6, pode-se observar sua estrutura.

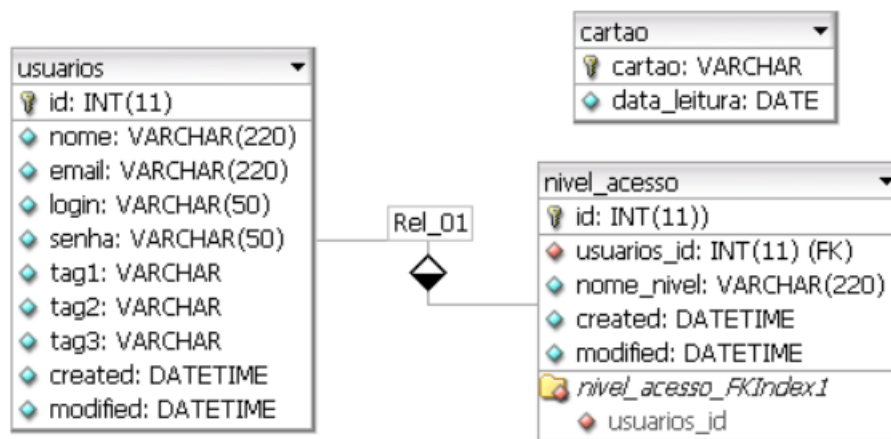


Figura 4.6 – Modelo entidade relacionamento do Banco de Dados

FONTE: Autor

4.3 INTERFACE WEB

Para a comunicação entre o SAR e o Arduino optou-se pela utilização do método GET do protocolo HTTP. Responsável pelo envio e recebimento das informações do Arduino até o SAR, tanto no envio da *tag* quando é efetuada a leitura, quanto o envio das

informações de temperatura ambiente e umidade relativa do ar. Na Figura 4.7 pode-se verificar como o Arduino envia as informações ao sistema.

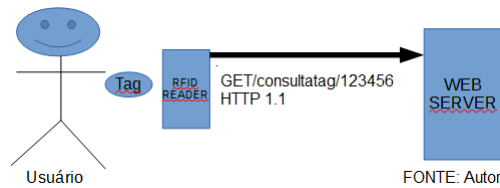


Figura 4.7 – Método GET com base no trabalho
FONTE: Autor

Existem duas opções de interação com o SAR, a primeira é quando (Figura 4.8) o usuário aproximando-se da porta de entrada da residência, o sensor de presença detecta o movimento, aciona a lâmpada da entrada aguardando a leitura do cartão, após a leitura, o Arduino comporta-se como cliente HTTP, passando a informação para o Servidor Web, que ao receber a informação busca no banco de dados, se o cartão está cadastrado na base de dados, caso positivo, ele retorna com o comando correspondente para abrir a porta, caso não encontre o cartão, ele retorna com uma mensagem informando que a entrada não está autorizada e ainda salva no banco de dados o número do cartão e o horário que o mesmo foi passado no leitor, assim pode-se saber se houve uma tentativa de entrada não autorizada na residência ou usamos esta leitura para capturar a numeração de um cartão não cadastrado para adicionar ao cadastro de um dos usuários.

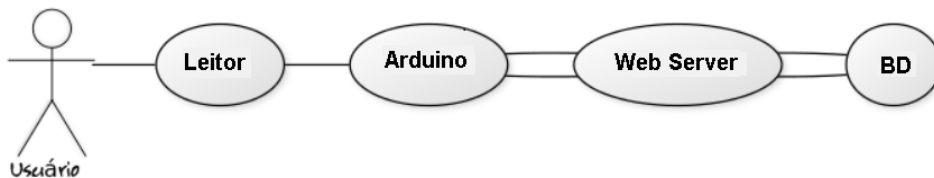


Figura 4.8 – Fluxograma interação do usuário com o SAR através da leitura do cartão de acesso.

FONTE: Autor

A segunda opção (Figura 4.9) de interação com o SAR, o usuário acessa o sistema através do celular, *tablet* ou computador. Caso o usuário que está acessando o SAR tenha acesso administrativo

Como descrito anteriormente, foi desenvolvido um sistema *web* utilizando a linguagem de programação PHP juntamente com o banco de dados MySQL, necessários para o controle e acesso ao sistema pelo usuário. Como o sistema foi desenvolvido com o objetivo de ser responsivo, o usuário pode acessá-lo tanto de seu Computador Pessoal quanto de seu *smartphone* ou *tablet*, possibilitando assim maior mobilidade, como pode ser observado na Figura 4.10.

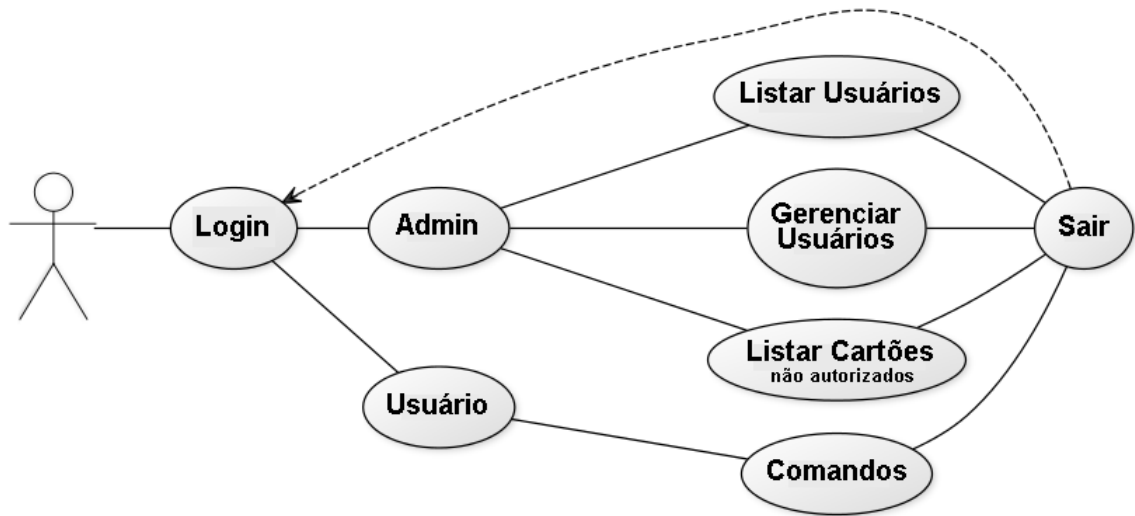


Figura 4.9 – Fluxograma interação do usuário com o SAR através do Sistema Web.
 FONTE: Autor

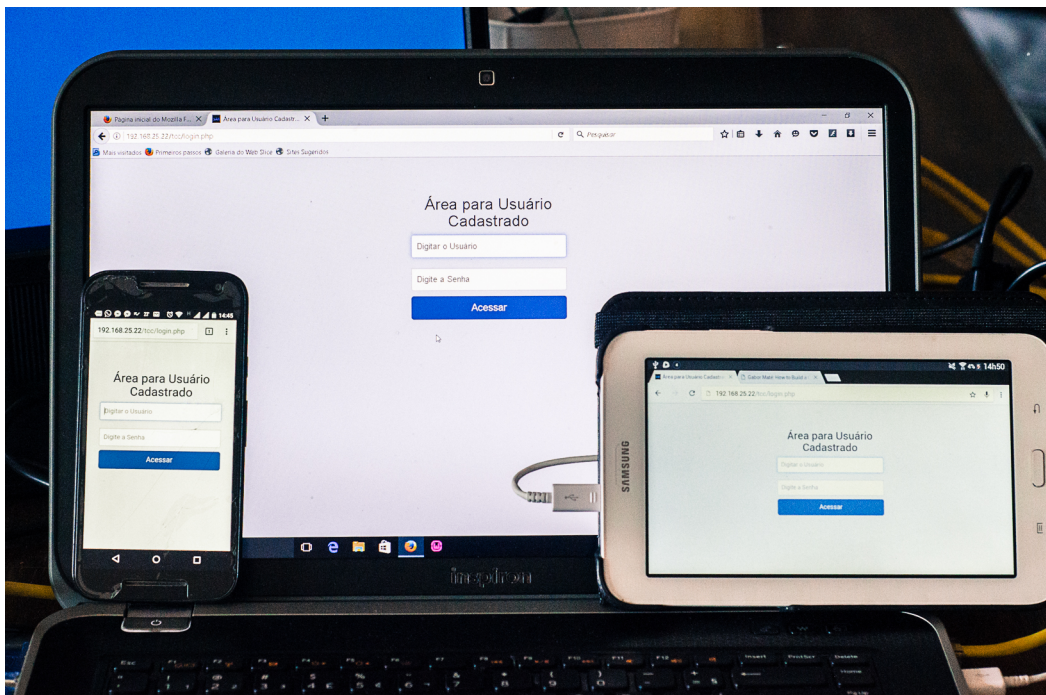


Figura 4.10 – Tela de login da interface web
 FONTE: Autor

A aplicação web proposta na implementação deste sistema SAR é dividida em duas seções, a seção administrativa e outra do usuário.

Na primeira, a seção administrativa, contém um menu onde é possível acessar a listagem dos usuários cadastrados, como pode ser verificado na Figura 4.11. Nesta tela são apresentados o número de identificação único (ID) no sistema, o nome, e-mail e o nível de acesso do usuário (quando apresenta o número 1, informa que o usuário tem acesso administrativo no sistema e, quando apresenta o número 2, informa que o usuário tem

acesso somente a parte dos comandos do sistema). Apresenta também o botão para a visualização dos dados, o botão para a edição dos dados bem como o botão de exclusão do usuário. Também foi optado em inserir um botão que acessa a parte de cadastro dos usuários.

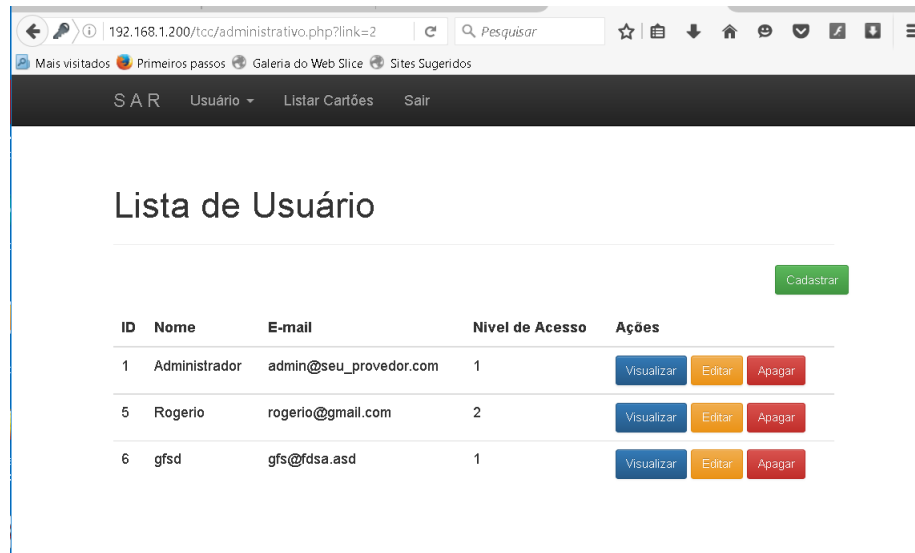


Figura 4.11 – Tela de listagem dos usuários

FONTE: Autor

Ainda na parte administrativa tem-se acesso a parte de cadastramento dos usuários (Figura 4.12), nesta parte é obrigatório o preenchimento dos seguintes campos: Nome, E-mail, Usuário, Senha e selecionar o Nível de acesso, já os cartões não são campos de preenchimento obrigatório.

Cadastrar Usuário

Nome:

E-mail:

Usuário:

Senha:

Nivel de Acesso:

Cartão 1:

Cartão 2:

Cartão 3:

Figura 4.12 – Tela de cadastro dos usuários

FONTE: Autor

Na segunda, a parte do usuário (Figura 4.13), tem-se acesso através do menu Comandos (Figura 4.14), nesta página estão disponíveis os *links* onde pode-se acionar as cargas (ligar ou desligar as lâmpadas) e também abrir a porta de entrada da residência.

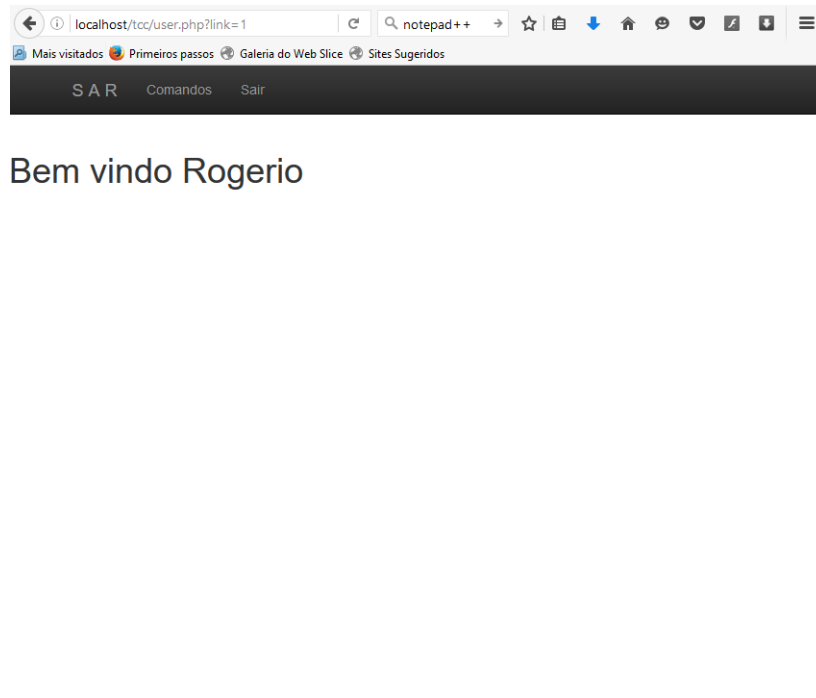


Figura 4.13 – Tela da parte do usuário

FONTE: Autor

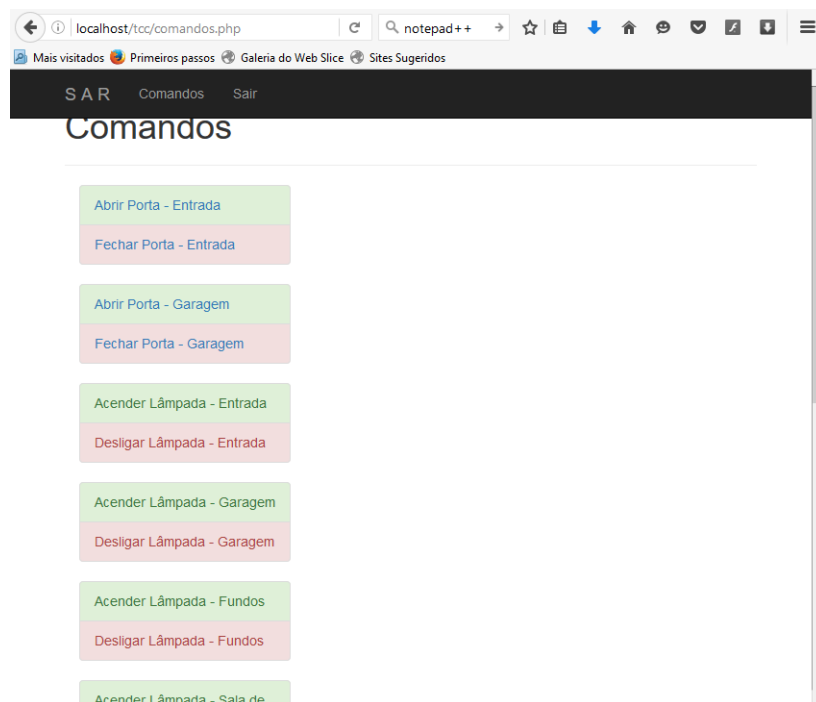


Figura 4.14 – Tela de comandos

FONTE: Autor

4.4 TESTES DE ESTABILIDADE

Convenientemente com o intuito de verificar a estabilidade do sistema, foram aplicados alguns testes que possibilite verificarmos se o sistema funcionará satisfatoriamente, segue abaixo etapas, materiais e equipamentos utilizados para efetuar os testes.

4.4.1 Equipamentos utilizados

Abaixo estão descritos os equipamentos utilizados nos testes:

- Laboratório de testes desenvolvido pelo próprio autor do projeto;
- Microcomputador DELL Inspiron 15R SE 7520, Core i5 3rd 3.1GHz, 8Gb RAM, interface wireless e interface Ethernet;
- Roteador Poweb BOX GVT;
- *Smartphone* iPhone 6S Plus;
- *Smartphone* Samsung 5S mini.

4.4.2 Softwares utilizados

Os *Softwares* instalados no microcomputador são: Sistema Operacional Windows 10, Suite de aplicativos WAMPServer 2.5 responsável pela hospedagem do SAR. Nos *Smartphones* foram utilizados os seguintes aplicativos de acesso a internet: Safari no iPhone e Google Chrome no Samsung.

4.4.3 Ambiente de Testes

Na Figura 4.15 podemos observar como ficou o ambiente de testes. Os testes foram realizados durante dois dias, foram feitos testes de aproximadamente 60 minutos, uma vez durante a parte da manhã e duas vezes durante a parte da tarde. Os *smartphones* acessaram o sistema simultaneamente fazendo o acionamento das cargas por diversas vezes, em um determinado tempo, além dos *smartphones* estarem interagindo com o sistema, foi acionado o sensor de presença e passado o cartão no leitor, nesses momentos, por um curto espaço de tempo, o sistema deixava de acionar os relés da Relay Board. Também foi

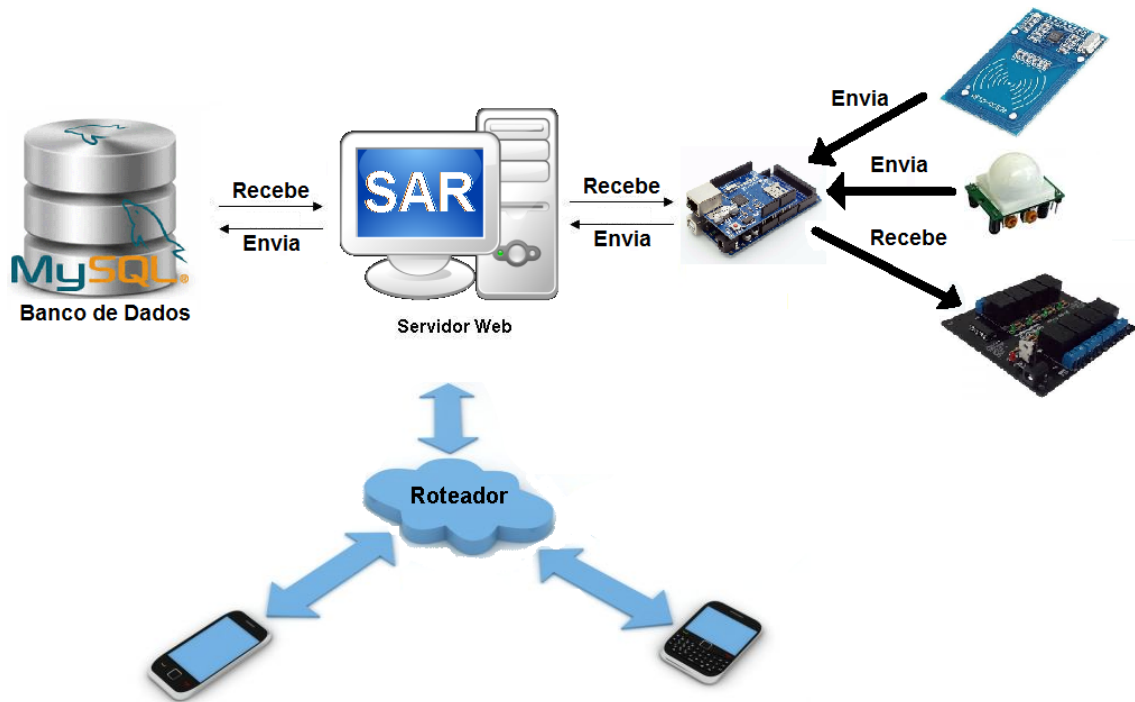


Figura 4.15 – Ambiente de Testes

FONTE: Autor

testado o desligamento do Arduíno, ao religá-lo, o sistema voltou ao normal, podendo ser acessado sem problemas.

4.4.4 Resultado dos testes

Durante os testes, enquanto somente os *smartphones* estavam interagindo com o sistema, não houve atraso algum perceptível no acionamento dos relés, o atraso que foi percebido foi no momento em que foi feito o teste de simulação de entrada na residência, que foi acionando o sensor de presença e passando o cartão no leitor, tirando esse pequeno atraso causado pela leitura do cartão, o sistema comportou-se satisfatoriamente bem.

5 CONCLUSÃO

Este trabalho surgiu após algumas aulas de Comunicação de Dados, com a constatação de que a aplicação dos ensinamentos adquiridos no decorrer do curso é mais descomplicado do que se imagina, também pelo autor se considerar um entusiasta na área e optar por automatizar a própria residência.

A Domótica, apresenta-se como uma revolução em ambientes domésticos atualmente. A concepção de integração entre dispositivos da uma residência, em uma central inteligente de comando vem aumentando com o tempo, a demanda por soluções em automação, visando segurança e comodidade, vem acompanhando.

Com o presente trabalho foi possível verificar que a utilização de tecnologias abertas, tanto em *hardware* quanto em *software*, por serem bastante disseminadas, consegue-se uma usabilidade bastante abrangente pelos desenvolvedores e até por usuários comuns, tornando possível a instalação em qualquer residência e por qualquer usuário.

Por se tratar da utilização de tecnologias abertas, o algoritmo deste trabalho (códigos Arduino e códigos PHP), serão disponibilizado nos apêndices, sendo assim, factível sua customização conforme a necessidade de cada usuário.

5.1 TRABALHOS FUTUROS

Como sugestão de trabalhos futuros, pode-se, no ambiente de segurança, incluir criptografia nos cartões de acesso evitando assim a cópia indevida. Também no contexto de segurança, implementar no sistema, a possibilidade de registrar em banco de dados o *log* de acesso de todas utilizações. Já na composição do sistema web, alterar a estrutura do banco de dados para que seja possível cadastrar infinitos cartões RFID para os usuários, similarmente efetuar modificações necessárias para que seja possível ter acesso ao sistema fora da residência através da internet. Também como sugestão, a implementação do Servidor Web em um dispositivo embarcado, como por exemplo Raspberry Pi.

Como neste projeto utilizamos a plataforma Arduino para ser a central de comandos e, sendo ela muito versátil, ainda pode-se, no contexto da comodidade, instalar a *shield* com reconhecimento de voz, para que em um determinado ambiente, quando gostaríamos de acender uma lâmpada por exemplo, não seria necessário acessar o sistema e simplesmente falar o comando.

REFERÊNCIAS

- BILL GLOVER, H. B. **FUNDAMENTOS DE RFID**. Rio de Janeiro, RJ, Brasil: Alta Books, 2007.
- BOTELHO, W. T. Um sistema de identificação e adaptação pervasivo para a casa inteligente utilizando sistemas multiagentes. **Dissertação Instituto Militar de Engenharia**, INSTITUTO MILITAR DE ENGENHARIA, Rio de Janeiro, RJ, p.194, 2005.
- EMER, J. C. **O grande desencontro do HTTP com o HTML**. Acessado em Dezembro/2015, <http://tableless.com.br/o-grande-desencontro-http-com-o-html/>.
- GOMES, H. M. C. Construção de um sistema de RFID com fins de localização especiais. **Dissertação (Mestrado em engenharia eletrônica e telecomunicações)**, Universidade de Aveiro, p.91, 2007.
- IBGE. **Pesquisa Nacional por Amostra de Domicílios**. Acessado em Janeiro/2014, ftp://ftp.ibge.gov.br/Indicadores_Sociais/Sintese_de_Indicadores_Sociais_2010/SIS_2010.pdf.
- KLABUNDE, F. SOFTWARE PARA MONITORAMENTO DE SERVIDORES WEB APACHE. **Trabalho de Conclusão de Curso (Ciências da Computação — Bacharelado)**, UNIVERSIDADE REGIONAL DE BLUMENAU, p.61, 2007.
- NETCRAFT. **November 2015 Web Server Survey**. Acessado em 10 Novembro 2015.
- TAKAI, O. K. INTRODUÇÃO A BANCO DE DADOS. **Departamento de Ciência da Computação**, INSTITUTO DE MATEMÁTICA E ESTATÍSTICA, Universidade de São Paulo, p.124, 2005.

APÊNDICE A – CÓDIGO FONTE ARDUINO

```
1 #include <DHT.h>
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h> // Display e Adaptador I2C
4 #include "DHT.h" //Sensor Temp e Humidade DHT22
5 #include <RelayBoard.h> //RelayBoard
6 #include <SPI.h> //Ethernet Shield
7 #include <Ethernet.h>
8 #include <MFRC522.h>
9 #include "Limits.h"
10 #include <Servo.h>
11
12 Servo myservo;
13
14 #define LCD_ADDR 0x3f // I2C address
15 #define EN 2 // LCD En (Enable)
16 #define RW 1 // LCD Rw (Read/write)
17 #define RS 0 // LCD Rs (Reset)
18 #define D4 4 // LCD data 0
19 #define D5 5 // LCD data 1
20 #define D6 6 // LCD data 2
21 #define D7 7 // LCD data 3
22 #define BACKLIGHT_PIN 3
23 #define BACKLIGHT_POL POSITIVE //ligar ou nao o backlight (luz de
    fundo). Valor pode ser POSITIVE ou NEGATIVE
24 #define COLUNAS 20 // quantidade de colunas do LCD
25 #define LINHAS 4 // quantidade de linhas do LCD
26
27 #define DHTPIN A8 // pino que estamos conectado o sensor de
    temperatura
28 #define DHTTYPE DHT22 // DHT 22
29
30 #define date 6 //config relay board
31 #define strobe 3 //config relay board
32 #define clock 2 //config relay board
33 #define numberboards 2 //config relay board
34
35 #define SS_PIN 53
36 #define RST_PIN 5
```

```

37
38 DHT dht(DHTPIN, DHTTYPE); //Inicializando sensor de temp.
39
40 LiquidCrystal_I2C lcd(LCD_ADDR, EN, RW, RS, D4, D5, D6, D7,
    BACKLIGHT_PIN, BACKLIGHT_POL); //Inicializa adaptador Display
41
42 RelayBoard relay(date, strobe, clock, numberboards);
43
44 /*
    -----
45 * This is a MFRC522 library example; see https://github.com/
    miguelbalboa/rfid
46 * for further details and other examples.
47 *
48 * NOTE: The library file MFRC522.h has a lot of useful info.
    Please read it.
49 *
50 * Released into the public domain.
51 *
    -----
52 * This sample shows how to read and write data blocks on a MIFARE
    Classic PICC
53 * (= card/tag).
54 *
55 * BEWARE: Data will be written to the PICC, in sector #1 (blocks
    #4 to #7).
56 *
57 *
58 * Typical pin layout used:
59 *
    -----
60 *
    MFRC522      Arduino      Arduino      Arduino
    Arduino      Arduino
61 *
    Reader/PCD  Uno          Mega          Nano v3
    Leonardo/Micro  Pro Micro
62 * Signal      Pin          Pin          Pin          Pin          Pin
    Pin
63 *

```

```

-----
64 * RST/Reset   RST           9           5           D9
      RESET/ICSP-5   RST
65 * SPI SS      SDA(SS)      10          53          D10          10
      10
66 * SPI MOSI    MOSI         11 / ICSP-4  51          D11          ICSP
      -4           16
67 * SPI MISO    MISO         12 / ICSP-1  50          D12          ICSP
      -1           14
68 * SPI SCK     SCK          13 / ICSP-3  52          D13          ICSP
      -3           15
69 *
70 */
71
72 MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
73
74 String readString;
75
76 int pinopir = 28; //Pino ligado ao sensor PIR
77 int acionamento;
78
79 //Definicoes de IP, mascara de rede e gateway
80 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
81 IPAddress ip(192,168,25,201); //Define o endereco IP
82 IPAddress gateway(192,168,25,1); //Define o gateway
83 IPAddress subnet(255, 255, 255, 0); //Define a mascara de rede
84
85 //Inicializa o servidor web na porta 80
86 EthernetServer server(80);
87 EthernetClient client;
88
89 void setup() {
90     //Inicializa a interface de rede
91     Ethernet.begin(mac, ip, gateway, subnet);
92     server.begin();
93
94     myservo.attach(29);
95
96     Serial.begin(9600);
97     SPI.begin(); // Inicia SPI bus

```

```
98   mfr522.PCD_Init();    // Inicia MFRC522
99   lcd.begin(COLUNAS,LINHAS);
100  lcd.setCursor(0,0);
101  lcd.print("Iniciando Sistema");
102  lcd.setCursor(0,1);
103  lcd.print("Autom. Residencial");
104  lcd.setCursor(0,2);
105  lcd.print("Rogerio Pauli");
106  delay(3000);
107  dht.begin();
108 }
109
110 void temp(){
111     // A leitura da temperatura e umidade pode levar 250ms!
112     // O atraso do sensor pode chegar a 2 segundos.
113     float h = dht.readHumidity();
114     float t = dht.readTemperature();
115     // testa se retorno valido, caso contrario algo est errado.
116     if (isnan(t) || isnan(h))
117     {
118         Serial.println("Failed to read from DHT");
119     }
120     else
121     {
122         lcd.clear();
123         lcd.setCursor(0,0);
124         lcd.print("Umidade: ");
125         lcd.setCursor(9,0);
126         lcd.print(h);
127         lcd.setCursor(15,0);
128         lcd.print("%");
129         lcd.setCursor(0,1);
130         lcd.print("Temp.: ");
131         lcd.setCursor(7,1);
132         lcd.print(t);
133         lcd.setCursor(13,1);
134         lcd.print("*C");
135         delay(500);
136
137     }
138 }
```

```

139
140 void leitor()
141 {
142     // Look for new cards
143     if ( ! mfrc522.PICC_IsNewCardPresent() )
144     {
145         return;
146     }
147     // Select one of the cards
148     if ( ! mfrc522.PICC_ReadCardSerial() )
149     {
150         return;
151     }
152     //Mostra UID na serial
153     Serial.print("UID da tag :");
154     String conteudo= "";
155     byte letra;
156     for (byte i = 0; i < mfrc522.uid.size; i++)
157     {
158         Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
159         Serial.print(mfrc522.uid.uidByte[i], HEX);
160         conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" :
161             " "));
162         conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
163     }
164     conteudo.toUpperCase();
165     Serial.println();
166     Serial.println("Mensagem : ");
167     lcd.setCursor(0,2);
168     lcd.print("Enviando dados");
169     lcd.setCursor(0,3);
170     lcd.print(".....Aguarde.....");
171     delay (1000);
172
173     if (client.connect("192.168.25.200", 80)) {
174         Serial.print("Enviando dados do cart o ... ");
175         conteudo.replace(' ', '_');
176         client.println("GET /tcc/cartao.php?cartao="+conteudo+" HTTP
177             /1.0");
178         client.println("HOST: 192.168.25.200");
179         client.println();

```



```
178     delay(200);
179     client.stop();
180 }
181 else {
182     // kf you didn't get a connection to the server:
183     Serial.println("connection failed");
184 }
185 Serial.print("Dados enviados");
186 lcd.setCursor(0,3);
187 lcd.print("...DADOS ENVIADOS...");
188 Serial.println();
189 client.stop();
190 }
191
192 void sensor(){
193     acionamento = digitalRead(pinopir); //Le o valor do sensor PIR
194
195     if (acionamento == HIGH) //Caso seja detectado um movimento, aciona
196         o rele
197     {
198         relay.set(0,2,1);
199         lcd.setCursor(0,3);
200         lcd.print("Aproxime o cartao:");
201         delay(500);
202         leitor();
203     }
204     else //Sem movimento, mantem rele desligado
205     {
206         relay.set(0,2,0);
207     }
208 }
209 void abreporta(){
210     lcd.setCursor(0,2);
211     lcd.print("Cart o Autorizado");
212     lcd.setCursor(0,3);
213     lcd.print("Seja Bem Vindo");
214     myservo.write(180);
215     delay (2000);
216     myservo.write(0);
217 }
```

```
218
219 void naoabreporta(){
220     lcd.setCursor(0,2);
221     lcd.print(".....Entrada.....");
222     lcd.setCursor(0,3);
223     lcd.print("...Nao Autorizada...");
224     delay(3000);
225
226 }
227
228 void loop(){
229     temp();
230     sensor();
231
232     //Aguarda conexao do browser
233     EthernetClient client = server.available();
234     if (client) {
235         while (client.connected()) {
236             if (client.available()) {
237                 char c = client.read();
238
239                 if (readString.length() < 100) {
240                     readString += c;
241                 }
242                 if (c == '\n') {
243                     client.println("HTTP/1.1 200 OK");
244                     client.println("Content-Type: text/html");
245                     client.println();
246
247                     client.println("<HTML>");
248                     client.println("<head>");
249                     client.println("<META HTTP-EQUIV='refresh' CONTENT='1;
250                                 URL=http://192.168.25.200/tcc/comandos.php'>");
251                     client.println("</head>");
252                     client.println("<BODY>");
253                     client.println("<H1>SAR</H1>");
254                     client.println("<hr />");
255                     client.println("<br />");
256
257                     client.println("Comando recebido com Sucesso!!!");
258                     client.println("</BODY>");
```

```
258     client.println("</HTML>");
259
260     delay(1);
261     client.stop();
262
263     if(readString.indexOf("?rele0on") > 0) { relay.set
264         (0,0,1); }
265     if(readString.indexOf("?rele0off") > 0) { relay.set
266         (0,0,0); }
267     if(readString.indexOf("?rele1on") > 0) { relay.set
268         (0,1,1); }
269     if(readString.indexOf("?rele1off") > 0) { relay.set
270         (0,1,0); }
271     if(readString.indexOf("?rele2on") > 0) { relay.set
272         (0,2,1); }
273     if(readString.indexOf("?rele2off") > 0) { relay.set
274         (0,2,0); }
275     if(readString.indexOf("?rele3on") > 0) { relay.set
276         (0,3,1); }
277     if(readString.indexOf("?rele3off") > 0) { relay.set
278         (0,3,0); }
279     if(readString.indexOf("?rele4on") > 0) { relay.set
280         (0,4,1); }
281     if(readString.indexOf("?rele4off") > 0) { relay.set
282         (0,4,0); }
283     if(readString.indexOf("?rele5on") > 0) { relay.set
284         (0,5,1); }
285     if(readString.indexOf("?rele5off") > 0) { relay.set
286         (0,5,0); }
287     if(readString.indexOf("?rele6on") > 0) { relay.set
288         (0,6,1); }
289     if(readString.indexOf("?rele6off") > 0) { relay.set
290         (0,6,0); }
291     if(readString.indexOf("?rele7on") > 0) { relay.set
292         (0,7,1); }
293     if(readString.indexOf("?rele7off") > 0) { relay.set
294         (0,7,0); }
295     if(readString.indexOf("?abreporta") > 0){ abreporta(); }
296     if(readString.indexOf("?naoabreporta") > 0){ naoabreporta
297         (); }
298     readString="";
```

```
282     }  
283   }  
284 }  
285 }  
286 }
```

APÊNDICE B – CÓDIGO SQL PARA CRIAÇÃO DO BANCO DE DADOS

Neste projeto usamos como nome do bando de dados o nome "tcc".

```
1 -- phpMyAdmin SQL Dump
2 -- version 4.1.14
3 -- http://www.phpmyadmin.net
4 --
5 -- Host: 127.0.0.1
6 -- Generation Time: 27-Set-2015    s    03:25
7 -- Vers  o do servidor: 5.6.17
8 -- PHP Version: 5.5.12
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8 */;
18
19
20 --
21 -- Estrutura da tabela 'nivel_acessos'
22 --
23
24 CREATE TABLE IF NOT EXISTS 'nivel_acessos' (
25   'id' int(11) NOT NULL AUTO_INCREMENT,
26   'nome_nivel' varchar(220) NOT NULL,
27   'created' datetime NOT NULL,
28   'modified' datetime DEFAULT NULL,
29   PRIMARY KEY ('id')
30 ) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
31
32 --
33 -- Inserindo dados da tabela 'nivel_acessos'
34 --
35
36 INSERT INTO 'nivel_acessos' ('id', 'nome_nivel', 'created', '
modified') VALUES
```

```

37 (1, 'Administrador', '2016-04-07 00:00:00', NULL),
38 (2, 'Usuario', '2016-04-07 00:00:00', NULL);
39
40 -----
41
42 --
43 -- Estrutura da tabela 'usuarios'
44 --
45
46 CREATE TABLE IF NOT EXISTS 'usuarios' (
47   'id' int(11) NOT NULL AUTO_INCREMENT,
48   'nome' varchar(220) NOT NULL,
49   'email' varchar(220) NOT NULL,
50   'login' varchar(50) NOT NULL,
51   'senha' varchar(50) NOT NULL,
52   'nivel_acesso_id' int(11) NOT NULL,
53   'tag1' varchar(100) DEFAULT NULL,
54   'tag2' varchar(100) DEFAULT NULL,
55   'tag3' varchar(100) DEFAULT NULL,
56   'created' datetime DEFAULT NULL,
57   'modified' datetime DEFAULT NULL,
58   PRIMARY KEY ('id')
59 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
60
61 --
62 -- Inserindo dados do admin na tabela 'usuarios'
63 --
64
65 INSERT INTO 'usuarios' ('id', 'nome', 'email', 'login', 'senha', '
66   tag1', 'tag2', 'tag3', 'nivel_acesso_id', 'created', 'modified')
67   VALUES
68 (1, 'Administrador', 'admin@seu_provedor.com', 'admin', '123', NULL
69   , NULL, NULL, 1, '2015-09-19 00:00:00', NULL);
70
71 CREATE TABLE IF NOT EXISTS 'cartao' (
72   'cartao' varchar(20) NOT NULL,
73   'data_leitura' datetime NOT NULL,
74   }

```

APÊNDICE C – CÓDIGO FONTE ARQUIVO "ADMINISTRATIVO.PHP"

```
1 <?php
2 session_start();
3 include_once("seguranca.php");
4 include_once("conexao.php");
5 ?>
6 <!DOCTYPE html>
7 <html lang="pt-br">
8     <head>
9         <meta charset="utf-8">
10        <meta http-equiv="X-UA-Compatible" content="IE=edge">
11        <meta name="viewport" content="width=device-width, initial-
12            scale=1">
13        <meta name="description" content="P gina Administrativa">
14        <meta name="author" content="Rogerio Pauli">
15        <link rel="icon" href="imagens/favicon.ico">
16
17        <title>Administrativo</title>
18        <link href="css/bootstrap.min.css" rel="stylesheet">
19        <link href="css/bootstrap-theme.min.css" rel="stylesheet">
20        <link href="css/theme.css" rel="stylesheet">
21        <script src="js/ie-emulation-modes-warning.js"></script>
22    </head>
23
24    <body role="document">
25        <?php
26
27            include_once("menu_admin.php");
28            $link = $_GET["link"];
29
30
31            $pag[1] = "bem_vindo.php";
32            $pag[2] = "listar_usuario.php";
33            $pag[3] = "cad_usuario.php";
34            $pag[4] = "editar_usuario.php";
35            $pag[5] = "visual_usuario.php";
36            $pag[6] = "lista_cartao.php";
37
```

```
38
39         if(!empty($link)){
40             if(file_exists($pag[$link])){
41                 include $pag[$link];
42             }else{
43                 include "bem_vindo.php";
44             }
45         }else{
46             include "bem_vindo.php";
47         }
48
49     ?>
50
51
52
53
54     <!-- Bootstrap core JavaScript
55     ===== -->
56     <!-- Placed at the end of the document so the pages load faster
57     -->
58     <script src="js/jquery.min.js"></script>
59     <script src="js/bootstrap.min.js"></script>
60     <script src="js/docs.min.js"></script>
61     <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
62     <script src="js/ie10-viewport-bug-workaround.js"></script>
63 </body>
</html>
```


APÊNDICE D – CÓDIGO FONTE ARQUIVO "BEM_VINDO.PHP

```
1 <h1><?php echo "Bem vindo ".$_SESSION ['usuarioNome'];?></h1>
```

APÊNDICE E – CÓDIGO FONTE ARQUIVO "VISUAL_USUARIO.PHP

```
1 <?php
2     $id = $_GET['id'];
3     //Executa consulta
4     $result = mysql_query("SELECT * FROM usuarios WHERE id = '
5         $id' LIMIT 1");
6     $resultado = mysql_fetch_assoc($result);
7 ?>
8 <div class="container theme-showcase" role="main">
9     <div class="page-header">
10         <h1>Visualizar Usu rio </h1>
11     </div>
12
13     <div class="row">
14         <div class="pull-right">
15             <a href='administrativo.php?link=2&id=?php
16                 echo $resultado['id']; ?>'><button type
17                 ='button' class='btn btn-sm btn-info'>
18                 Listar</button></a>
19
20             <a href='administrativo.php?link=4&id=?php
21                 echo $resultado['id']; ?>'><button type
22                 ='button' class='btn btn-sm btn-warning'
23                 >Editar</button></a>
24
25             <a href='processa/proc_apagar_usuario.php?
26                 id=?php echo $resultado['id']; ?>'><
27                 button type='button' class='btn btn-sm
28                 btn-danger'>Apagar</button></a>
29         </div>
30     </div>
31
32     <div class="row">
33         <div class="col-md-12">
34             <div class=" col-sm-3 col-md-1">
35                 <b>Id:</b>
36             </div>
37             <div class=" col-sm-9 col-md-11">
38                 <?php echo $resultado['id']; ?>
```

```
29         </div>
30
31         <div class="col-sm-3 col-md-1">
32             <b>Nome:</b>
33         </div>
34         <div class="col-sm-9 col-md-11">
35             <?php echo $resultado['nome']; ?>
36         </div>
37
38         <div class="col-sm-3 col-md-1">
39             <b>E-mail:</b>
40         </div>
41         <div class="col-sm-9 col-md-11">
42             <?php echo $resultado['email']; ?>
43         </div>
44
45         <div class="col-sm-3 col-md-1">
46             <b>Usu rio:</b>
47         </div>
48         <div class="col-sm-9 col-md-11">
49             <?php echo $resultado['login']; ?>
50         </div>
51
52         <div class="col-sm-3 col-md-1">
53             <b>Nivel de Acesso:</b>
54         </div>
55         <div class="col-sm-9 col-md-11">
56             <?php echo $resultado['
57                 nivel_acesso_id']; ?>
58         </div>
59
60         <div class="col-sm-3 col-md-1">
61             <b>Cart o 1:</b>
62         </div>
63         <div class="col-sm-9 col-md-11">
64             :<?php echo $resultado['tag1']; ?>
65         </div>
66
67         <div class="col-sm-3 col-md-1">
68             <b>Cart o 2:</b>
69         </div>
```

```
69         <div class="col-sm-9 col-md-11">
70             :<?php echo $resultado['tag2']; ?>
71         </div>
72
73         <div class="col-sm-3 col-md-1">
74             <b>Cart o 3:</b>
75         </div>
76         <div class="col-sm-9 col-md-11">
77             :<?php echo $resultado['tag3']; ?>
78         </div>
79     </div>
80 </div>
81 </div> <!-- /container -->
```

APÊNDICE F – CÓDIGO FONTE ARQUIVO "VALIDA_LOGIN.PHP"

```
1
2 <?php
3 session_start();
4 $usuariot = $_POST['usuario'];
5 $senha = $_POST['senha'];
6 include_once("conexao.php");
7
8 $result = mysql_query("SELECT * FROM usuarios
9     WHERE login='$usuariot' AND senha='$senha' LIMIT 1");
10 $resultado = mysql_fetch_assoc($result);
11 //echo "Usuario: ".$resultado['nome'];
12 if(empty($resultado)){
13     //Mensagem de Erro
14     $_SESSION['loginErro'] = "Usu rio ou senha Inv lido";
15
16     //Manda o usuario para a tela de login
17     header("Location: login.php");
18 }else{
19     //Define os valores atribuidos na sessao do usuario
20     $_SESSION['usuarioId'] = $resultado['id'];
21     $_SESSION['usuarioNome'] = $resultado['nome'];
22     $_SESSION['usuarioNivelAcesso'] = $resultado['nivel_acesso_id'];
23     $_SESSION['usuarioLogin'] = $resultado['login'];
24     $_SESSION['usuarioSenha'] = $resultado['senha'];
25     $_SESSION['usuarioTag1'] = $resultado['tag1'];
26     $_SESSION['usuarioTag2'] = $resultado['tag2'];
27     $_SESSION['usuarioTag3'] = $resultado['tag3'];
28
29     if($_SESSION['usuarioNivelAcesso'] == 1){
30         header("Location: administrativo.php?link=1");
31     }else{
```

```
32         header("Location: usuario.php");
33     }
34 }
35 ?>
```

APÊNDICE G – CÓDIGO FONTE ARQUIVO "USER.PHP"

```
1 <?php
2 session_start();
3 include_once("seguranca.php");
4 include_once("conexao.php");
5 ?>
6 <!DOCTYPE html>
7 <html lang="pt-br">
8     <head>
9         <meta charset="utf-8">
10        <meta http-equiv="X-UA-Compatible" content="IE=edge">
11        <meta name="viewport" content="width=device-width, initial-
12            scale=1">
13        <meta name="description" content="P gina Administrativa">
14        <meta name="author" content="Rogerio Pauli">
15        <link rel="icon" href="imagens/favicon.ico">
16
17        <title>Usu rio </title>
18        <link href="css/bootstrap.min.css" rel="stylesheet">
19        <link href="css/bootstrap-theme.min.css" rel="stylesheet">
20        <link href="css/theme.css" rel="stylesheet">
21        <script src="js/ie-emulation-modes-warning.js"></script>
22    </head>
23
24    <body role="document">
25        <?php
26
27            include_once("menu_user.php");
28            $link = $_GET["link"];
29
30
31            $pag[1] = "bem_vindo.php";
32            $pag[2] = "comandos.php";
33
34
35            if(!empty($link)){
36                if(file_exists($pag[$link])){
37                    include $pag[$link];
```

```
38         }else{
39             include "bem_vindo.php";
40         }
41     }else{
42         include "bem_vindo.php";
43     }
44
45     ?>
46
47
48     <!-- Bootstrap core JavaScript
49     ===== -->
50     <!-- Placed at the end of the document so the pages load faster
51     -->
52     <script src="js/jquery.min.js"></script>
53     <script src="js/bootstrap.min.js"></script>
54     <script src="js/docs.min.js"></script>
55     <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
56     <script src="js/ie10-viewport-bug-workaround.js"></script>
57 </body>
</html>
```


APÊNDICE H – CÓDIGO FONTE ARQUIVO "SEGURANCA.PHP"

```
1 <?php
2 ob_start();
3 if(($_SESSION['usuarioId'] == "") || ($_SESSION['usuarioNome'] == ""
4     ") || ($_SESSION['usuarioNivelAcesso'] == "") || ($_SESSION['
5     usuarioLogin'] == "") || ($_SESSION['usuarioSenha'] == "")){
6         unset($_SESSION['usuarioId'],
7             $_SESSION['usuarioNome'],
8             $_SESSION['usuarioNivelAcesso'],
9             $_SESSION['usuarioLogin'],
10            $_SESSION['usuarioSenha']);
11
12         //Mensagem de Erro
13         $_SESSION['loginErro'] = " rea restrita para usu rios
14         cadastrados";
15
16         //Manda o usu rio para a tela de login
17         header("Location: login.php");
18     }
```

APÊNDICE I – CÓDIGO FONTE ARQUIVO "SAIR.PHP"

```
1 <?php
2 session_start();
3 session_destroy();
4
5 //Remove todas as informa es contidas na
6     variaveis globais
7 unset($_SESSION['usuarioId'],
8 $_SESSION['usuarioNome'],
9 $_SESSION['usuarioNivelAcesso'],
10 $_SESSION['usuarioLogin'],
11 $_SESSION['usuarioSenha']);
12
13 //redirecionar o usu rio para a p gina de login
14 header("Location: login.php");
15 ?>
```

APÊNDICE J – CÓDIGO FONTE ARQUIVO "MENU_ADMIN.PHP"

```
1 <!-- Inicio navbar -->
2 <nav class="navbar navbar-inverse navbar-fixed-top">
3   <div class="container">
4     <div class="navbar-header">
5       <button type="button" class="navbar-toggle collapsed"
6         data-toggle="collapse" data-target="#navbar" aria-
7         expanded="false" aria-controls="navbar">
8         <span class="sr-only">Toggle navigation</span>
9         <span class="icon-bar"></span>
10        <span class="icon-bar"></span>
11        <span class="icon-bar"></span>
12      </button>
13      <a class="navbar-brand" href="administrativo.php">S A R</
14      a>
15    </div>
16    <div id="navbar" class="navbar-collapse collapse">
17      <ul class="nav navbar-nav">
18        <li class="dropdown">
19          <a href="#" class="dropdown-toggle" data-toggle="
20            dropdown" role="button" aria-haspopup="true"
21            aria-expanded="false">Usu rio <span class="
22            caret"></span></a>
23          <ul class="dropdown-menu">
24            <li><a href="administrativo.php?link=2">
25              Listar</a></li>
26            <li><a href="administrativo.php?link=3">
27              Cadastrar</a></li>
28          </ul>
29          <li><a href="administrativo.php?link=6">Listar
30            Cart es</a></li> </li>
31          <li><a href="sair.php">Sair</a></li>
32        </ul>
33      </div><!--/.nav-collapse -->
34    </div>
```

```
30 </nav>  
31 <!-- Fim navbar -->
```

APÊNDICE K – CÓDIGO FONTE ARQUIVO "LOGIN.PHP

```
1 <?php
2     session_start();
3 ?>
4 <!DOCTYPE html >
5 <html lang="pt-br" >
6     <head >
7         <meta charset="utf-8" >
8         <meta http-equiv="X-UA-Compatible" content="IE=edge" >
9         <meta name="viewport" content="width=device-width, initial-
10             scale=1" >
11         <meta name="description" content="Página de login" >
12         <meta name="author" content="Rogério Pauli" >
13         <link rel="icon" href="/imagens/favicon.ico" >
14
15         <title > reá para Usu rio Cadastrado </title >
16         <link href="css/bootstrap.min.css" rel="stylesheet" >
17         <link href="css/signin.css" rel="stylesheet" >
18         <script src="js/ie-emulation-modes-warning.js" ></script >
19
20     </head >
21
22     <body >
23         <?php
24             unset($_SESSION['usuarioId'],
25                 $_SESSION['usuarioNome'],
26                 $_SESSION['usuarioNivelAcesso'],
27                 $_SESSION['usuarioLogin'],
28                 $_SESSION['usuarioSenha']);
29         ?>
30         <div class="container" >
31             <form class="form-signin" method="POST" action="valida_login.
32                 php" >
33                 <h2 class="form-signin-heading text-center" > reá para
34                     Usu rio Cadastrado </h2 >
35                 <label for="inputEmail" class="sr-only" >Usu rio </label >
36
37                 <input type="text" name="usuario" class="form-control"
38                     placeholder="Digitar o Usu rio" required autofocus ><br
```

```
35     />
36     <label for="inputPassword" class="sr-only">Senha</label>
37     <input type="password" name="senha" class="form-control"
38         placeholder="Digite a Senha" required >
39
40     <button class="btn btn-lg btn-primary btn-block" type="
41         submit">Acessar</button>
42 </form>
43     <p class="text-center text-danger">
44         <?php
45             if(isset($_SESSION['loginErro'])){
46                 echo $_SESSION['loginErro'
47                     ];
48                 unset($_SESSION['loginErro'
49                     ]);
50             }
51         ?>
52     </p>
53 </div> <!-- /container -->
54
55 <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
56 <script src="js/ie10-viewport-bug-workaround.js"></script>
57 </body>
58 </html>
```

APÊNDICE L – CÓDIGO FONTE ARQUIVO "LISTAR_USUARIO.PHP"

```
1 <?php
2
3     $resultado=@mysql_query("SELECT * FROM usuarios ORDER BY '
4         id'");
5     $linhas=@mysql_num_rows($resultado);
6 >?
7 <div class="container theme-showcase" role="main">
8     <div class="page-header">
9         <h1>Lista de Usu rio </h1>
10    </div>
11    <div class="row espaco">
12        <div class="pull-right">
13            <a href="administrativo.php?link=3"><button
14                type='button' class='btn btn-sm btn-
15                success'>Cadastrar</button></a>
16        </div>
17    </div>
18    <div class="row">
19        <div class="col-md-12">
20            <table class="table">
21                <thead>
22                    <tr>
23                        <th>ID</th>
24                        <th>Nome</th>
25                        <th>E-mail</th>
26                        <th>Nivel de Acesso</th>
27                        <th>A     es </th>
28                    </tr>
29                </thead>
30                <tbody>
31                    <?php
32                        while($linhas = @mysql_fetch_array(
33                            $resultado)){
34                            echo "<tr>";
35                            echo "<td>".$linhas
36                                ['id'].</td>";
37                            echo "<td>".$linhas
38                                ['nome'].</td>";
```

```
33     ;
34     echo "<td>".$linhas
        ['email']. "</td>";
35     echo "<td>".$linhas
        ['
        nivel_acesso_id'
        ]. "</td>";
36     ?>
37     <td>
        <a href='
            administrativo.
            php?link=5&id=?
            php echo $linhas
            ['id']; ?>'><
            button type='
            button' class='
            btn btn-sm btn-
            primary'>
            Visualizar</
            button></a>
38
39     <a href='
            administrativo.
            php?link=4&id=?
            php echo $linhas
            ['id']; ?>'><
            button type='
            button' class='
            btn btn-sm btn-
            warning'>Editar
            </button></a>
40
41     <a href='processa/
            proc_apagar_usuario
            .php?id=?php
            echo $linhas['id
            ']; ?>'><button
            type='button'
            class='btn btn-
            sm btn-danger'>
```



```
42                                     Apagar</button
43                                     ></a>
44                                     <?php
45                                     echo "</tr>";
46                                     }
47                                     ?>
48                                     </tbody>
49                                     </table>
50                                     </div>
51 </div> <!-- /container -->
```

APÊNDICE M – CÓDIGO FONTE ARQUIVO "INDEX.PHP

```
1 <?php header ("location: http://192.168.25.22/tcc/login.php"); ?>
```

APÊNDICE N – CÓDIGO FONTE ARQUIVO "EDITAR_USUARIO.PHP"

```
1 <?php
2     $id = $_GET['id'];
3     //Executa consulta
4     $result = mysql_query("SELECT * FROM usuarios WHERE id = '
5         $id' LIMIT 1");
6     $resultado = mysql_fetch_assoc($result);
7 ?>
8 <div class="container theme-showcase" role="main">
9     <div class="page-header">
10         <h1>Editar Usu rio </h1>
11     </div>
12     <div class="row espacio">
13         <div class="pull-right">
14             <a href='administrativo.php?link=2&id=?php
15                 echo $resultado['id']; ?>'><button type
16                 ='button' class='btn btn-sm btn-info'>
17                 Listar</button></a>
18
19             <a href='processa/proc_apagar_usuario.php?
20                 id=?php echo $resultado['id']; ?>'><
21                 button type='button' class='btn btn-sm
22                 btn-danger'>Apagar</button></a>
23
24         </div>
25     </div>
26     <div class="row">
27         <div class="col-md-12">
28             <form class="form-horizontal" method="POST" action="
29                 processa/proc_edit_usuario.php">
30
31                 <div class="form-group">
32                     <label for="inputEmail3" class="col-sm-2
33                         control-label">Nome</label>
34
35                     <div class="col-sm-10">
36                         <input type="text" class="form-control"
37                             name="nome" placeholder="Nome Completo
38                             " value="<?php echo $resultado['nome
39                                 ']; ?>">
40
41                     </div>
42                 </div>
43             </form>
44         </div>
45     </div>
46 </div>
```



```

55         <option value="1"
56             <?php
57                 if( $resultado['
                    nivel_acesso_id'
                    ] == 1){
58                     echo '
                        selected
                        ';
59                 }
60             ?>
61         >Administrativo</option>
62         <option value="2"
63             <?php
64                 if( $resultado['
                    nivel_acesso_id'
                    ] == 2){
65                     echo '
                        selected
                        ';
66                 }
67             ?>
68         >Usu rio </option>
69     </select>
70     </div>
71 </div>
72
73     <input type="hidden" name="id" value="<?php echo
74         $resultado['id']; ?>">
75     <div class="form-group">
76         <div class="col-sm-offset-2 col-sm-10">
77             <button type="submit" class="btn btn-
78                 success">Editar</button>
79         </div>
80     </div>
81 </form>
82 </div> <!-- /container -->

```

APÊNDICE O – CÓDIGO FONTE ARQUIVO "CONEXAO.PHP

```
1 <?php
2 $conectar = @mysql_connect("localhost","root","") or die ("Erro na
   conex o");
3
4 mysql_select_db("tcc")or die ("Base n o encontrada");
5 ?>
```

APÊNDICE P – CÓDIGO FONTE ARQUIVO "CAD_USUARIO.PHP"

```
1 <div class="container theme-showcase" role="main">
2   <div class="page-header">
3     <h1>Cadastrar Usu rio </h1>
4   </div>
5   <div class="row espacio">
6
7     </div>
8   <div class="row">
9     <div class="col-md-12">
10      <form class="form-horizontal" method="POST" action="
11        processa/proc_cad_usuario.php">
12
13        <div class="form-group">
14          <label for="inputEmail3" class="col-sm-2
15            control-label">Nome</label>
16          <div class="col-sm-10">
17            <input type="text" class="form-control"
18              name="nome" placeholder="Nome Completo
19              ">
20          </div>
21        </div>
22
23        <div class="form-group">
24          <label for="inputEmail3" class="col-sm-2
25            control-label">E-mail</label>
26          <div class="col-sm-10">
27            <input type="email" class="form-control"
28              name="email" placeholder="E-mail">
29          </div>
30        </div>
31
32        <div class="form-group">
33          <label for="inputEmail3" class="col-sm-2
34            control-label">Usu rio </label>
35          <div class="col-sm-10">
36            <input type="text" class="form-control"
37              name="usuario" placeholder="Usu rio">
38          </div>
39        </div>
40      </form>
41    </div>
42  </div>
43 </div>
```

```
31     </div>
32
33     <div class="form-group">
34         <label for="inputPassword3" class="col-sm-2
35             control-label">Senha</label>
36         <div class="col-sm-10">
37             <input type="password" class="form-
38                 control" name="senha" placeholder="
39                 Senha">
40         </div>
41     </div>
42
43     <div class="form-group">
44         <label for="inputPassword3" class="col-sm-2
45             control-label">Nivel de Acesso</label>
46         <div class="col-sm-10">
47             <select class="form-control" name="
48                 nivel_de_acesso">
49                 <option value="1">Administrativo
50                 </option>
51                 <option value="2">Usu rio </
52                 option>
53             </select>
54         </div>
55     </div>
56
57     <div class="form-group">
58         <label for="inputEmail3" class="col-sm-2
59             control-label">Cart o 1</label>
60         <div class="col-sm-10">
61             <input type="text" class="form-control"
62                 name="tag1" placeholder="Cart o 1">
63         </div>
64     </div>
65
66     <div class="form-group">
67         <label for="inputEmail3" class="col-sm-2
68             control-label">Cart o 2</label>
69         <div class="col-sm-10">
70             <input type="text" class="form-control"
71                 name="tag2" placeholder="Cart o 2">
72         </div>
```



```
61         </div>
62         <div class="form-group">
63             <label for="inputEmail3" class="col-sm-2
64                 control-label">Cart o 3</label>
65             <div class="col-sm-10">
66                 <input type="text" class="form-control"
67                     name="tag3" placeholder="Cart o 3">
68             </div>
69         </div>
70
71         <div class="form-group">
72             <div class="col-sm-offset-2 col-sm-10">
73                 <button type="submit" class="btn btn-
74                     success">Cadastrar</button>
75             </div>
76         </div>
77     </form>
78 </div>
79 </div>
80 </div> <!-- /container -->
```

APÊNDICE Q – CÓDIGO FONTE ARQUIVO "CARTAO.PHP

```
1 <?php
2
3 $cartao = $_GET['cartao'];
4
5 include_once("conexao.php");
6
7 $busca = mysql_query("SELECT * FROM usuarios WHERE (tag1 = '$cartao
      ' OR tag2 = '$cartao' OR tag3 = '$cartao')");
8
9 if(mysql_num_rows($busca) > 0){
10     $ch = curl_init();
11     curl_setopt($ch, CURLOPT_URL, "http://192.168.1.201/?
      abreporta");
12     curl_exec($ch);
13     curl_close($ch);
14
15 } else {
16
17 mysql_query("INSERT INTO cartao (cartao, data_leitura) VALUES ('
      $cartao', NOW())");
18     $ch = curl_init();
19     curl_setopt($ch, CURLOPT_URL, "http://192.168.1.201/?
      naoabreporta");
20     curl_exec($ch);
21     curl_close($ch);
22
23 }
24
25
26 ?>
```



```
32         /administrativo.php?link=2'>
33         <script type=\"text/javascript\">
34             alert(\"Usu rio n o foi
35                 editado com Sucesso.\");
36         </script>
37     ";
38 }
39 ?>
40 </body>
41 </html>
```

APÊNDICE S – CÓDIGO FONTE ARQUIVO "PROC_CAD_USUARIO.PHP"

Este arquivo deve ser criado dentro do diretório processa.

```
1 <?php
2 session_start();
3 include_once("../seguranca.php");
4 include_once("../conexao.php");
5 $nome = $_POST["nome"];
6 $email = $_POST["email"];
7 $usuario = $_POST["usuario"];
8 $senha = $_POST["senha"];
9 $nivel_de_acesso = $_POST["nivel_de_acesso"];
10 $tag1 = $_POST["tag1"];
11 $tag2 = $_POST["tag2"];
12 $tag3 = $_POST["tag3"];
13 $query = mysql_query("INSERT INTO usuarios (nome, email, login,
    senha, nivel_acesso_id, tag1, tag2, tag3, created) VALUES ('
    $nome', '$email', '$usuario', '$senha', '$nivel_de_acesso', '
    $tag1', '$tag2', '$tag2', NOW())");
14 ?>
15 <!DOCTYPE html>
16 <html lang="pt-br">
17 <head>
18 <meta charset="utf-8">
19 </head>
20
21 <body>
22 <?php
23 if (mysql_affected_rows() != 0 ){
24     echo "
25         <META HTTP-EQUIV=REFRESH CONTENT =
26             '0;URL=http://localhost/tcc/
27                 administrativo.php?link=2'>
28         <script type=\"text/javascript\">
29             alert(\"Usu rio cadastrado
30                 com Sucesso.\");
31         </script>
32     ";
33 }
34 else{
```

```
32         echo "
33         <META HTTP-EQUIV=REFRESH CONTENT =
           '0;URL=http://localhost/tcc/
           administrativo.php?link=2'>
34         <script type=\"text/javascript\">
35             alert(\"Usu rio n o foi
           cadastrado com Sucesso
           .\");
36         </script>
37         ";
38
39     }
40
41     ?>
42 </body>
43 </html>
```

APÊNDICE T – CÓDIGO FONTE ARQUIVO "PROC_APAGAR_USUARIO.PHP"

Este arquivo deve ser criado dentro do diretório "processa".

```
1 <?php
2 session_start();
3 include_once("../seguranca.php");
4 include_once("../conexao.php");
5 $id = $_GET["id"];
6
7 $query = "DELETE FROM usuarios WHERE id=$id";
8 $resultado = mysql_query($query);
9 $linhas = mysql_affected_rows();
10
11 ?>
12 <!DOCTYPE html>
13 <html lang="pt-br">
14 <head>
15 <meta charset="utf-8">
16 </head>
17
18 <body>
19 <?php
20 if (mysql_affected_rows() != 0 ){
21     echo "
22         <META HTTP-EQUIV=REFRESH CONTENT =
23             '0;URL=http://localhost/tcc/
24                 administrativo.php?link=2'>
25         <script type=\"text/javascript\">
26             alert(\"Usu rio apagado
27                 com Sucesso.\");
28         </script>
29     ";
30 }
31 else{
32     echo "
33         <META HTTP-EQUIV=REFRESH CONTENT =
34             '0;URL=http://localhost/tcc/
35                 administrativo.php?link=2'>
36         <script type=\"text/javascript\">
```

```
32         alert(\"Usu rio n o foi
33             apagado com Sucesso.\");
34     </script>
35     ";
36     }
37
38     ?>
39 </body>
40 </html>
```


APÊNDICE U – ARQUIVOS JAVASCRIPT

Em virtude dos arquivos de configuração JavaScript possuírem um volume expressivo de código, apenas indicaremos os arquivos, pois são configurações padrão e os mesmos são encontrados com grande facilidade na internet. Estes arquivos devem ser colocados dentro do diretório "js".

Segue abaixo relação dos arquivos utilizados:

- bootstrap.min.js
- docs.min.js
- ie10-viewport-bug-workaround.js
- ie-emulation-modes-warning.js
- jquery.min.js