IFRN-CAMPUS NATAL/ZONA NORTE
CURSO TÉCNICO INTEGRADO EM ELETRÔNICA

EVA ARAÚJO
LARYSSA DA SILVA FREITAS
VICENTE MEIRA NETO

FECHADURA AUTOMATIZADA COM CONTROLE VIA WEB

NATAL-RN
2018

EVA ARAÚJO
LARYSSA DA SILVA FREITAS
VICENTE MEIRA NETO

FECHADURA AUTOMATIZADA COM CONTROLE VIA WEB

Trabalho usado como um dos requisitos necessários para a consumação dos autores discentes como técnicos em Eletrônica pelo IFRN.
Orientador: Pedro Ivo de Araújo do Nascimento;
Coorientador: Rodolfo da Silva Costa.

NATAL-RN
2018

AGRADECIMENTOS

Em meio a uma rotina contínua de dedicação, estudo e desenvolvimento do projeto, colocando-o em primeiro plano, contamos com o apoio e incentivo de figuras muito importantes na nossa caminhada. Nesse círculo, além dos nossos orientadores Pedro Ivo de Araújo do Nascimento e Rodolfo da Silva Costa, destacam-se os apoiadores de campo afetivo e motivacional de todos os dias: pais, família, amigos e colaboradores do nosso campus. Para vocês, nosso muito obrigado.

RESUMO

Atualmente, o investimento em tecnologias promotoras de segurança em imóveis, residências e estabelecimentos comerciais diversos é cada vez mais assistido pelos cidadãos brasileiros. De tal maneira que se é pertinente, também, nessa perspectiva, a oferta de elementos facilitadores dessa condição essencial humana, sobretudo, em um ambiente escolar de potencial circulação de estudantes e servidores, como o MARIA, laboratório de robótica e automação do campus IFRN Natal/Zona Norte. Com esse intuito, e almejando a praticidade atingida por um acesso, via Web ou tags, de usuários autorizados ao local, o projeto Fechadura Automatizada Com Controle Via Web pleiteou, então, uma conexão entre um conjunto dispositivo-software, isto é, a transmissão de dados entre uma rede de hardware (com um Arduino Mega, RFID e ethernet) e uma página de acesso exclusiva do administrador.

Palavras-chave: Segurança. Elementos facilitadores. Web ou tags. Arduino. Fechadura Automatizada Com Controle Via Web.

ABSTRACT

Currently, the investment in technologies promoting security in real estate, residences and various commercial establishments is increasingly supported by Brazilian citizens. In such a way that it is also pertinent to follow this perspective, the provision of facilitating elements of this essential human condition, especially in a school environment of great circulation of student and workers, such as MARIA, which is a robotics and automation laboratory of IFRN in Natal North Zone campus. With this purpose and aiming the practicality achieved by an access via the Web or tags of users authorized to the workplace, the project named Automated Locking Through Via Web Control, pleaded a connection between a device-software set, that is, the data transmission between a hardware network (with a Mega Arduino, RFID and ethernet technology) and an administrator-only access page.

Keywords: Security. Facilitating elements. Web or tags. Arduino. Automated Locking through Via Web control.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ABESE	Associação Brasileira das Empresas de Sist. Eletrônicos de Segurança
CMD	Prompt de Comando
CPU	Unidade Central de Processamento
DC	Corrente Contínua
EEPROM	*Electrically Erasable Programmable Read-Only Memory*
FTP	Protocolo de Transferência de Arquivos
HTML	Linguagem de Marcação de Hipertexto
HTTP	Protocolo de Transferência de Hipertexto
IDE	Ambiente de Desenvolvimento Integrado
IFRN	Instituto Federal do Rio Grande do Norte
IP	Protocolo de Internet
IRQ	*Interrupt Request Line*
I2C	*Inter-Integrated Circuit*
GND	Filtro Graduado de Densidade Neutra
KB	*Kilobyte*
LAN	Local Area Network
Led	Diodo Emissor de Luz
MARIA	Movimento Aberto de Robótica, Inovação e Automação
MAC	*Media Access Control*
MISO	*Master IN Slave OUT*
MOSI	*Master OUT Slave IN*
NO	*Normally open*
PWM	Modulação por Largura de Pulso
QR	*Quick Response*
RAM	Memória de Acesso Aleatório
RFID	*Radio-Frequency IDentification*
RST	*Reset*
SCL	*Clock Line*
SDA	*Data Line*
SD	*Standard Definition*
SMTP	*Simple Mail Transfer Protocol*
SS	*Slave Select*

TCP	Protocolo de Controle de Transmissão
UART	Receptor/Transmissor Universal Assíncrono
UID	Identificador de Usuário
URL	Localizador Universal de Recurso
USB	Porta Universal
V	Volt

SUMÁRIO

1 INTRODUÇÃO

Segundo dados da revista Segurança Eletrônica, em 2017, o mercado nacional de segurança eletrônica reunia mais de 26 mil empresas dentro dos segmentos de sistemas de alarmes, circuitos fechados de TV, portas e portões automáticos, rastreamento de veículos, dispositivos de identificação por biometria, entre outros. Seu faturamento, no ano antecedente, foi de R$ 5,7 bilhões, com crescimento de 5% em relação a 2015. Além disso, há um estimativa pela Associação Brasileira das Empresas de Sistemas Eletrônicos de Segurança (ABESE) de quase 745 mil imóveis monitorados por sistemas eletrônicos de segurança no país em 2017.

O crescimento dessa área se baseia, essencialmente, nos alarmantes índices de violência e criminalidade no Brasil. Por isso, o uso de sistemas de controle de acesso é primordial, visto que garante que somente pessoas autorizadas terão acesso a áreas restritas de uma empresa, como almoxarifados e tesourarias. Hospitais também têm se beneficiado desses sistemas para controle de visitantes com catracas na portaria e controle de acesso a ambientes críticos como Unidades de Tratamento Intensivo e farmácias (FECHADURAS... 2018).

A empresa DIPREL Segurança Integrada elenca, dentre os principais componentes desse gênero, as fechaduras digitais, as quais são responsáveis pelo controle dos usuários que têm acesso aos ambientes mediante o uso de senhas numéricas e/ou cartões magnéticos. Nesse cenário, encontra-se o presente projeto desenvolvido ao longo do corrente ano pelos concluintes de Eletrônica do IFRN Campus Natal/Zona Norte.

1.1 Finalidades do projeto

A Fechadura Automatizada Com Controle Via Web é um produto que congrega um sistema físico (sensores de radiofrequência, rede ethernet e um Arduino Mega) a uma página web que permite a autenticação de um usuário com base no uso de cartões (como no exemplo supracitado), mas também confere o acesso por meio do site e chaveiros. Dessa forma, visa a segurança e praticidade para o ambiente em que será empregada: o laboratório do MARIA (Movimento Aberto de Robótica, Inovação e Automação) do IFRN Natal/Zona Norte.

1.2 Justificativa

O emprego de *tags* com um vasto espaço para armazenamento de informações (em contraponto, por exemplo, aos *QR Codes*), bem como a facilidade para o sistema promover uma conexão à internet, são algumas das vantagens da Fechadura Automatizada Com Controle Via Web. Outro fator de destaque na sua implementação trata-se do processo de intercomunicação entre os dispositivos do sistema integrante e à flexibilidade para a autenticação dos procedimentos de acesso autorizados, que podem partir de celulares, computadores, etc (mesmo de modo simultâneo).

Ademais, é fundamental para somar ao controle de acesso a um ambiente do campus tão visitado e necessário como o MARIA, o uso de tecnologias de custos reduzidos como as do sistema RFID e, sobretudo, a do protótipo final.

Desse modo, é relevante uma breve descrição sobre as ferramentas e procedimentos explorados em cada etapa do projeto pelo grupo e seus correspondentes itens no relatório.

1.3 Etapas de realização do projeto

Inicialmente, conforme consta no capítulo 3.1**,** no contexto de estudos e aproximação com o microcontrolador Arduino e com o seu Ambiente de Desenvolvimento Integrado (IDE), o grupo se voltou, essencialmente, para o entendimento acerca da comunicação entre essa plataforma e o RFID usado.

Em seguida, conteúdo exposto na seção 3.2, tratou-se de implementar às tecnologias o meio ethernet. Assim, sua estrutura foi acoplada ao Arduino e as conexões entre o conjunto foram integralmente feitas. Por fim, em 3.3, associou-se todos os dispositivos ao hardware essencial do projeto: a fechadura.

É válido ressaltar, ainda, que o grupo se apoiou no uso do software *fritzing*; na construção de fluxogramas no ambiente de apresentação do Microsoft Powerpoint; também, na explanação acerca dos componentes, diagramas e circuitos implementados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Trancas

As trancas se enquadram, principalmente, nos seguintes tipos: eletromagnéticas, mecânicas e elétricas. As eletromagnéticas apresentam uma placa de metal, bem como um ímã com um circuito de capacitores, que levam ao processo de magnetização/desmagnetização; elas evitam o consumo elevado de energia elétrica, têm alta durabilidade, impede a intervenção dos fenômenos naturais, como vento ou tempestade. As mecânicas, em contrapartida, são relativas à movimentação manual e podem ter funcionalidade de proteção em caso de falha na estrutura elétrica (NETSEG, 2018).

O projeto fez uso de uma fechadura eletromecânica, assim como a mostrada abaixo na figura 1. Uma das suas grandes particularidades é a dupla via de aplicação: com ou sem a presença de uma fonte de eletricidade, posto que possui a propriedade mecânica salvaguardada. Além disso, tem-se uma configuração de mola ajustável (quer para portas leves ou para pesadas).

Figura 1 - Fechadura *multilaser* usada no projeto.

Fonte: Lojas Americanas (adaptado)

Por fim, como a elencada acima, as trancas elétricas são responsáveis pelo acionamento e controle de um sistema promotor de maior segurança (NETSEG, 2018). Elas contam, também, com o acesso por meio de chaveiros, cartões ou, até mesmo, pela biometria. Associado a isso, consta-se a função de autenticação, feita no projeto com base no uso de um microcontrolador, o Arduino.

2.2 Arduino

Como uma plataforma de prototipagem idealizada em 2005, na Itália, pelo professor Massimo Banzi, em uma tentativa de promover uma aproximação de seus alunos com a Eletrônica e a programação, com um custo mais acessível, e produto com fóruns de códigos abertos, o Arduino recebe cada vez mais apoio da comunidade acadêmica mundial e facilita o desenvolvimento de projetos em meio acadêmico (ARDUINO, 2015).

O criador do livro Arduino Básico, Mcroberts (2011), destaca: “a maior vantagem do Arduino sobre outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização”. Ele acrescenta, também, que pessoas leigas na área da Eletrônica não têm tantos obstáculos para encabeçar projetos com esse produto, justamente, por conta da linguagem compreensível e da disponibilidade de inúmeros programas na internet. Esse foi, pois, o principal motivador para o uso do Arduino na Fechadura Automatizada Com Controle Web.

Figura 2 - O Arduino Mega 2560.

Fonte: Portal Amazon.

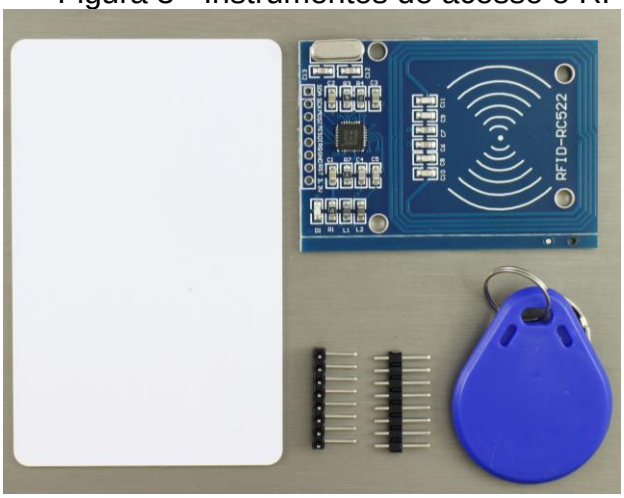
Outro importante atrativo para tal foi a possibilidade de conciliar uma rede de dispositivos com o intermédio do protocolo I2C (abordado no item 3.4) e, ainda, a oferta de uma quantidade significativa de terminais, posto que o escolhido foi o Mega 2560 (figura 2), que conta com 54 portas digitais e 16 analógicas. Esse, dentre todas as versões disponíveis no mercado da família Arduino, destina-se a trabalhos mais complexos com seus 256 KB de memória FLASH (dois quais 8 KB é para uso do bootloader), 8 KB de RAM e 4 KB de EEPROM e é, por isso, ideal para a funcionalidade do projeto (ARDUINO... 2018).

2.3 RFID

O projeto faz uso da tecnologia de Identificação por Radiofrequência (do inglês, RFID), fator promotor de um amplo armazenamento e transmissão de informações (LOUREIRO et al., 2018). Valendo-se dessa funcionalidade presente no trabalho, trata-se a RFID, nesse caso, como meio facilitador do processo de autenticação desejado.

Como produto de uma tecnologia desenvolvida no século XX, chamada RADAR (SANTINI, 2008), a RFID conta com elementos internos que compõem sua funcionalidade principal (ler e gravar informações), são esses: *transponders* (transmitem/recebem comandos); *transceivers* (usam a antena, outro componente do grupo de elementos, para captar os comandos e enviá-los ao software); e as etiquetas (encontradas nas lojas de roupas), cintos, óculos, livros, prateleiras, além de cartões e chaveiros, como mostra a figura 3 abaixo:

Figura 3 - Instrumentos de acesso e RFID.

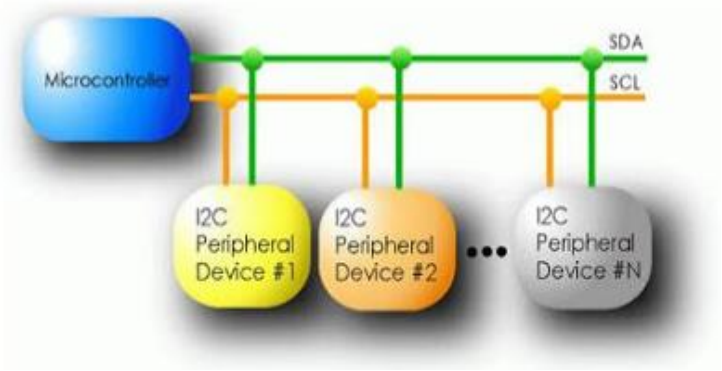


Fonte: Portal PandoraLab.

2.4 Protocolos de comunicação

Há dois tipos de comunicação das informações no sistema binário que integra o microcontrolador em uso: a serial e a paralela. No tipo serial, os sinais são enviados bit a bit, ou seja, o fluxo de dados é lento e restrito; ao passo que o segundo tipo possibilita o envio simultâneo de bits, uma transferência maior e mais rápida dos sinais, além de ficar -por consequência- mais vulnerável a interferências externas.

Figura 4 - Protocolo de comunicação I2C.

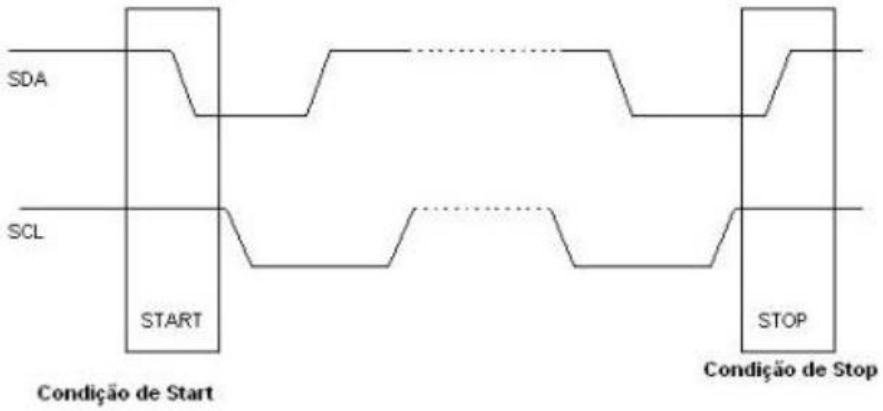


Fonte: HENRIQUE, 2012.

Como protocolo de envio paralelo, o I2C é usado para a comunicação e transporte de dados entre os dispositivos, através de seu barramento e da sua aplicabilidade mestre-escravos. Essa dinâmica permite uma forte economia de conectores no sistema, já que sua programação trabalha com diversos endereçamentos e, como um diferencial, permite o uso simultâneo de vários componentes (consonante à figura 4), de tecnologias diferentes, com base em um pareamento sem erros (HENRIQUE, 2012).

Essa comunicação se inicia com a condição *start* e finaliza com *stop*. Nessa perspectiva, para que aconteça a primeira condição, deve ocorrer a transição do nível alto para o baixo na linha SDA, enquanto o SCL permanecer com nível alto. Já na condição STOP, a linha SDA vai ter que realizar o processo inverso: transitar do nível baixo para o alto, e a linha SCL se manter constante com o nível alto (HENRIQUE, 2012). Isso pode ser visualizado na figura 5 a seguir:

Figura 5 - Condição de START e STOP da comunicação no barramento I2C.



Fonte: HENRIQUE, 2012.

Este barramento, além disso, é considerado multi-mestre, o que significa a possibilidade de mais de um dispositivo de controle presente no sistema; exceto quando em uma comunicação, na qual apenas um dispositivo mestre pode estar conectado, para se evitar um “choque” entre os dados do barramento (HENRIQUE, 2012).

2.4.1 Transmissão de dados mestre-escravo

Figura 6 - Funcionalidade mestre-escravo para uma I2C.

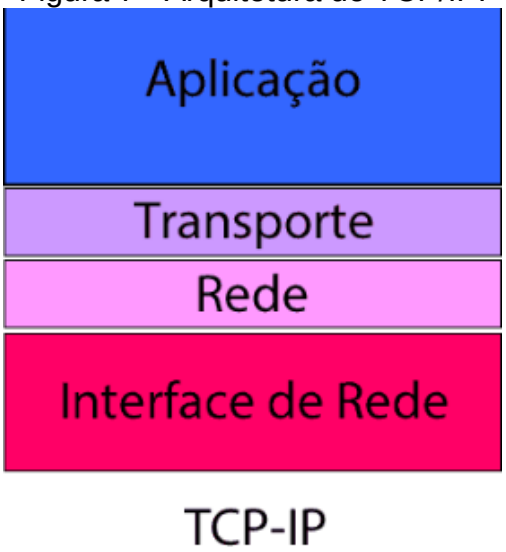
Fonte: HENRIQUE, 2012.

A transmissão de um sinal-informação para um receptor, por parte do dispositivo mestre, representa uma sistemática coordenada (conforme apresentado acima na figura 6), na qual envia-se e trata-se de um comando por vez, como no início do processo, em que o escravo aguarda a instrução *start* para saber, por meio do bit menos significativo do endereço analisado, se é para lê-lo ou escrevê-lo na memória: a fase de reconhecimento (HENRIQUE, 2012).

2.5 TCP/IP

Um protocolo representa uma linguagem responsável pela comunicação entre dois computadores, ou um computador e um dispositivo. No projeto, faz-se uso de dois tipos: o TCP, cujo significado é Protocolo de Controle de Transmissão, e o IP, *Internet Protocol* (Protocolo de Internet). Assim, o TCP/IP trata-se da junção desses, mas também conta com outros protocolos relacionados (SILVEIRA, 2018).

Figura 7 - Arquitetura do TCP/IP.



Fonte: MARTINS, 2012.

Sua arquitetura, demonstrada pela figura 7 acima, é subdividida em 4 etapas, cada qual com sua respectiva tarefa. Inicialmente, constata-se o enviar e receber de informações através da camada de aplicação, na qual se encontra protocolos como o SMTP, FTP e HTTP. Há, pois, o processamento dos dados, seguido pelo envio à camada de transporte (MARTINS, 2012).

Logo, recebidos os dados oriundos da seção anteriormente citada, verifica-se a integridade do conjunto e é feita a sua distribuição em pacotes. A partir disso, o conteúdo separado é levado para a camada de rede, na qual ocorre o seu anexo ao endereço virtual do computador do remetente e do destinatário. Então, essas informações são entregues a uma rede de internet, mediante o uso de uma interface (MARTINS, 2012). Essa é a última das camadas, que tem sua funcionalidade intermediada por protocolos, como o ethernet, usado em várias etapas do presente projeto.

2.6 HTTP

Como um Protocolo de Transferência de Hipertexto na web, com base no recebimento de requisições e envio de um retorno entre um cliente e um servidor, o HTTP funciona da seguinte forma: um *user agent* -por meio de um dispositivo ou navegador- solicita um determinado recurso a partir do envio de pacotes de dados,

contendo cabeçalhos, a um Localizador Universal de Recurso (URL). Logo, é feita a recepção pelos servidor, e a sua consequente resposta, através de um outro cabeçalho ou recurso (VIEIRA, 2007).

Além disso, o HTTP é stateless, em seu funcionamento, a retenção de informações em requisições diferentes não é possível de acontecer. Tendo que utilizar um processo para a insistência das informações, como o uso de cookies ou sessões na URL (VIEIRA, 2007).

Para tanto, uma requisição só pode ser feita quando se define qual método será empregado no ambiente. Dentre eles, está o *get*, o qual “solicita a representação de um determinado recurso; define-se como um método seguro e não deve ser usado para disparar uma ação (remover um usuário, por exemplo)”, como sustenta Vieira (2007).

2.7 Relé

Um dos componentes básicos para o sistema eletrônico da fechadura é o relé eletromecânico, que exerce o papel de chave e cuja parte elétrica é formada por uma bobina com núcleo de ferro que, quando abastecida de uma diferença de potencial, fica polarizada magneticamente, atraindo, assim, o terminal móvel e fechando o contato. E, ao se cancelar essa alimentação, a bobina fica desmagnetizada, fazendo o terminal retornar a sua posição inicial (MARKUS, 2011).

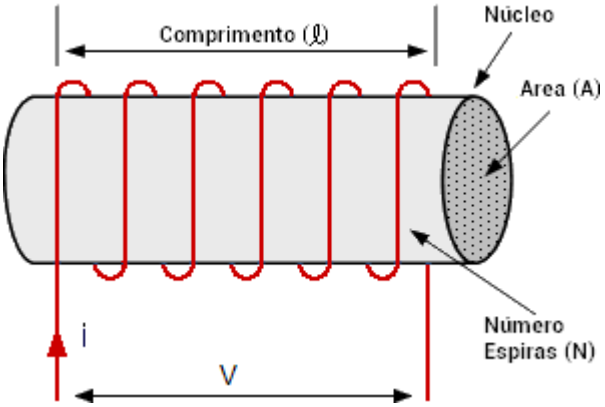
Figura 8 - Módulo de relé similar ao usado no projeto.

Fonte: Portal FILIPEFLOP.

Nessa perspectiva, é válida a compreensão acerca desse integrante básico do relé: a bobina. De acordo com Markus (2011), é um dispositivo formado por um fio esmaltado enrolado em torno de um núcleo (figura 9), na qual cria-se o campo

magnético que induz o surgimento de uma corrente elétrica contínua, e de um fluxo de elétrons que mantém sempre a mesma polaridade e sentido.

Figura 9 - Estrutura de uma bobina.



Fonte: Portal ELECTRONICA PT.

2.8 HTML

Como uma linguagem de marcação simples e baseada no emprego de tags e em uma estrutura bem delimitada, a Ethernet é de fácil visualização. Por exemplo, a parte voltada para o título do texto ou parágrafo da interface (EIS, 2011).

Figura 10 - Código introdutório da estrutura HTML.

```
<!DOCTYPE html>

<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <title>Título da página</title>
</head>
<body>
   <h1>Aqui vai o texto do título</h1>
   <p>Aqui vai o texto do parágrafo.
   Geralmente parágrafos tem muitas palavras, letras menores que as do título</p>
</body>
</html>
```

Fonte: EIS, 2011.

Consonante à estrutura acima, da figura 10, o HTML é introduzido pela *tag* "DOCTYPE", cuja função é mandar um aviso para os browsers/leitores de tela sobre o tipo de informação/documento será carregada. Em seguida, aparece a "HTML",

delimitando o *corpus* de texto que estará disposto na página; ela é seguida pela "lang", que mostra o idioma usado (EIS, 2011).

A próxima linha será a tag "Head", responsável pela indicação da tabela de caracteres que o browser usará para a renderização do texto. Outra tag bastante importante é a <tittle>, que tem o dever de indicar o título do documento. Assim, quando um usuário for pesquisar algo no seu site, basta buscar por ela. Por fim, há um encerramento, que contém o resto do código em HTML (EIS, 2011).

2.9 Ethernet

Valendo-se da linguagem de marcação explorada, é válido apontar que a comunicação dos dispositivos através de um meio físico (Ethernet), como um módulo acoplado a um Shield para a conexão ao microcontrolador (Arduino), surgiu como uma possibilidade para se conectar grandes números de computadores em uma rede local, mediante a tecnologia *Local Area Network*, a LAN (QUAIS... 2018).

Figura 11 - Cabo LAN Ethernet usado em uma CPU.

Fonte: DIAS, 2016.

Os tipos de Ethernet englobam a Fast Ethernet, Gigabit Ethernet e a 10 Gigabit Ethernet, as quais se diferenciam, sobretudo, na velocidade de transmissão da informação. A primeira apresenta uma taxa de 100 Mbits/s, ao passo que as demais têm, respectivamente, 1000 Mbits/s (1 Gbit/s) e 10 Gbits/s (QUAIS... 2018).

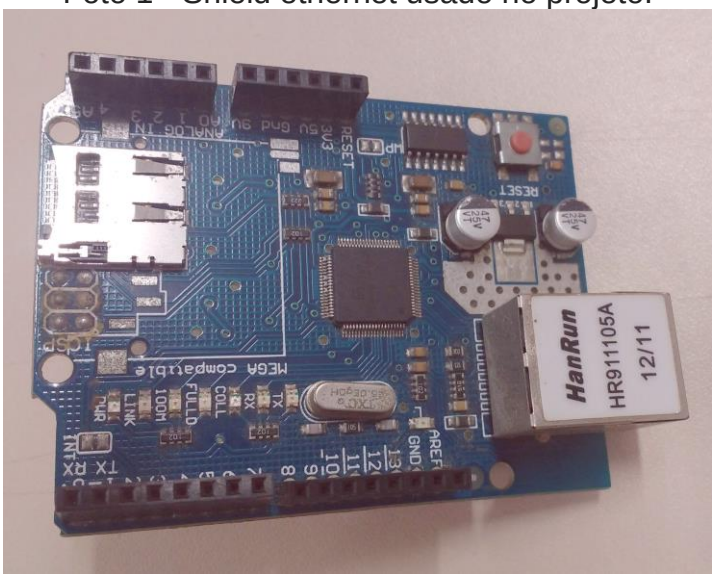
Ademais, quando se trata da meio, algumas palavras estão comumente relacionadas e abordadas no contexto de estudo, como: meio, *frame*, segmento e nó. O meio seria o caminho comum para que haja a conexão entre os sinais eletrônicos,

como os cabos de fibra-ótica ou de par trançado; o segmento designa um meio que é compartilhado; enquanto que o nó é o dispositivo que verifica-se conectado a esse último. E, por fim, o *frame*, que é a comunicação para receber e mandar mensagens, também chamada de blocos de informação (SAIBA... 2018).

Dentre os benefícios de uso da conexão Ethernet, destacam-se: a boa conexão entre vários computadores, de modo simultâneo, com a possibilidade de um alto compartilhamento de dados entre os usuários; uma alta velocidade de troca de informações em contraposição às redes Wi-fi; a acessibilidade, tendo-se em vista um menor custo no mercado; além de sua confiabilidade, devido a sua propriedade resistiva para aplicações que exigem um nível de segurança maior, pois se houver algum acidente pode ser muito prejudicial para o sistema trabalhado (REDE... 2016).

2.10 Shield ethernet

Foto 1 - Shield ethernet usado no projeto.



Fonte: Dos autores (2018).

Foi relevante para estudo e posterior implantação ao projeto, ainda, o shield ethernet. Esse meio físico fez a interface entre o sistema web construído e o microcontrolador acoplado ao sistema. Mediante sua aplicação, foi possível ao usuário da rede de autenticação o controle do pacote de dados gerados por dia, posto que se tem um cartão micro SD acoplado a sua placa.

Nessa perspectiva, a aplicação de um shield a projetos de automação é relevante, posto que, mediante a conexão de um módulo Arduino à internet, por exemplo, torna-se possível o estabelecimento de uma ligação direta entre um website

básico com um circuito eletrônico, além da realização de funções adicionais (mas também de similar importância para os processos de estudo e implementação de códigos programacionais). Essa funcionalidade pode, pois, ser contemplada na Fechadura Automatizada Com Controle Via Web, por meio da conectividade entre o website criado pelos autores com o Arduino e o decorrente controle prático do aparelho físico, a tranca.

3 METODOLOGIA

O projeto passou por três etapas principais para consolidar-se como protótipo final: de início, houve o estudo, montagem e programação de conexões entre a fechadura e o sensor RFID; seguido pela realização de procedimentos semelhantes com um shield ethernet e a implantação de uma página web dedicada ao acesso dos usuários credenciados; por fim, a compactação do sistema geral em um único produto.

3.1 Arduino e sensor RFID

Nessa fase, realizou-se o estudo da pinagem do Arduino Mega 2560, visando-se a escolha dos potenciais componentes para aplicação com o sensor de Radiofrequência, e suas respectivas funções, as quais contemplavam o fornecimento de energia ao circuito (pinos de alimentação, de 5V e 3,3V); o terminal GND e RESET; as portas relativas à passagem de dados (de natureza digital); além de pinos de entrada e saída.

O sensor RFID usado tem a referência MFRC522 e, por tal identificação, é composto pelos pinos: SDA (referente aos protocolos da informação a ser transmitida, que envolve a configuração UART, I2C e SS); SCK (*Serial Clock*), a qual diz respeito ao clock de sincronização para envio de informações de acordo com o mecanismo mestre-escravo; MOSI (Master OUT Slave IN) dados mestre-escravo; MISO (Master IN Slave OUT) dados escravo-mestre; a porta IRQ, que é utilizada em casos de interrupção de uma parte do sistema se contiver mais de um módulo RFID e se quiser usar um deles isoladamente; além das portas para aterramento, reset do sistema e alimentação de 3,3 volts. A sua configuração é representada na figura 12 abaixo:

Figura 12 - Pinagem do sensor RFID RC522.

Fonte: Portal Mercado Livre.

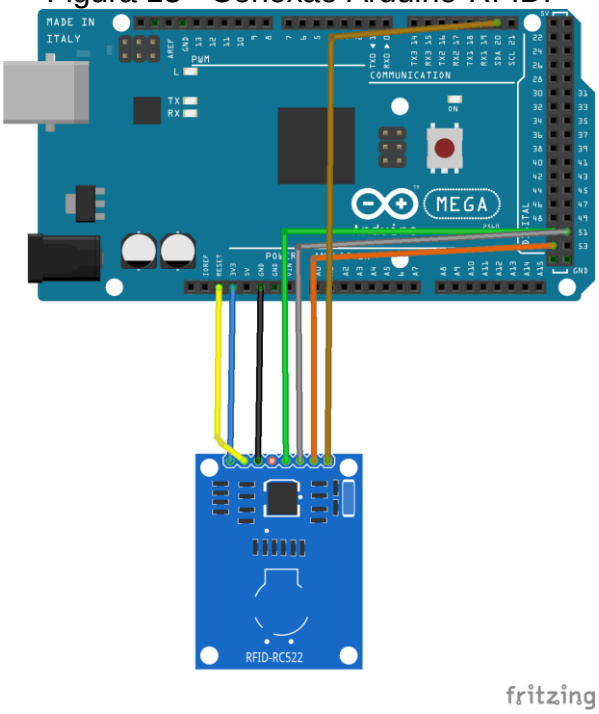
Para conexão entre o sensor e o Arduino, o grupo separou os seguintes terminais ilustrados pela tabela 1 abaixo. É válido observar, pois, que não houve a apresentação de uma fonte de alimentação diretamente exposta a esses. Fez-se, pois, um contato via cabo USB entre um computador do laboratório MARIA e a plataforma Arduino usada para tanto.

Quadro 1 - Conexões feitas entre Arduino e RFID.

ARDUINO	RFID
48	MISO
51	MOSI
50	SCK
20	SDA
3,3V	3,3V
GND	GND
RESET	RST
-	IRQ

Fonte: Elaborada pelos autores (2018).

Figura 13 - Conexão Arduino-RFID.

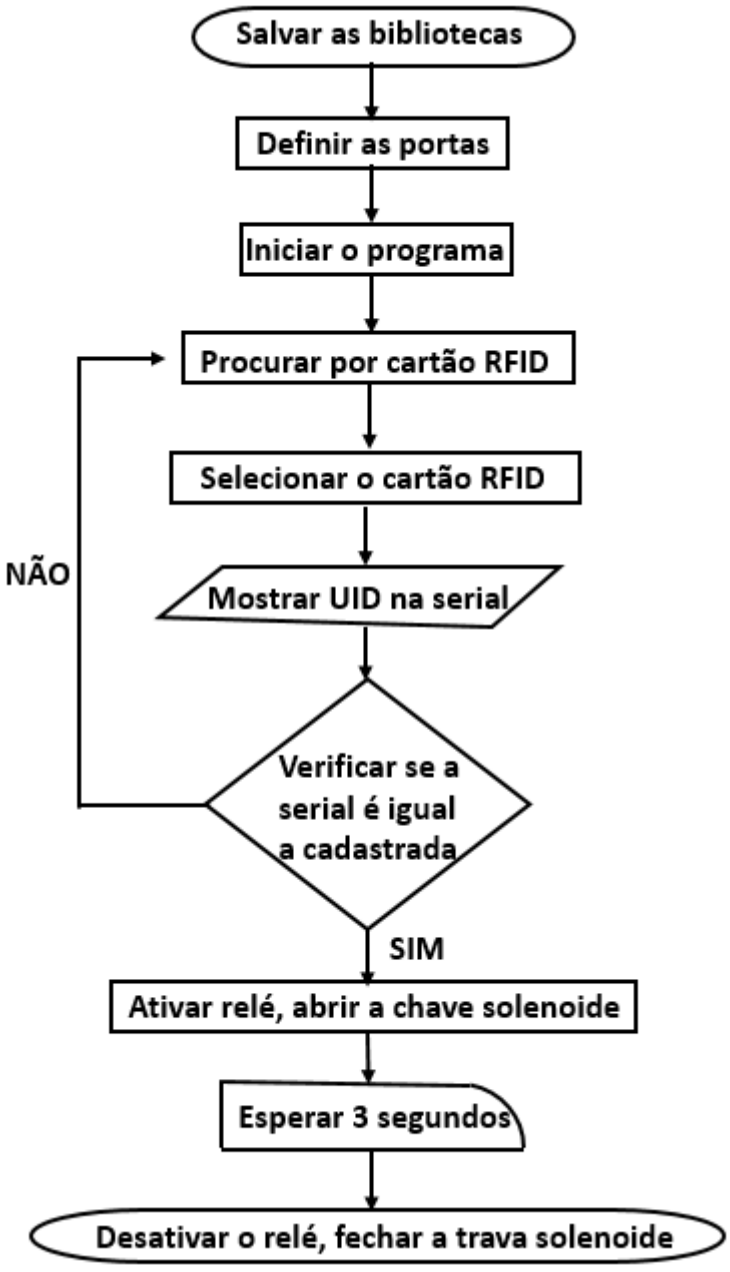


Fonte: Dos autores (2018).

Dado o enfoque à parte física do projeto, tentou-se o estabelecimento de uma comunicação entre essa e uma rede de dados virtuais, marcada por uma série de instruções destinadas a um potencial usuário da Fechadura Automatizada Via Controle Web. Isso foi possível por meio do estudo e aplicação da linguagem de programação C no IDE do Arduino.

Como procedimentos prévios à elaboração do código de autenticação das tags e consequente controle da fechadura, o grupo dedicou-se a sistemática de um fluxograma (ilustrado na figura 14) com uma série de disposições centrais:

Figura 14 - Fluxograma para tomada de ações com o RFID.



Fonte: Dos autores (2018).

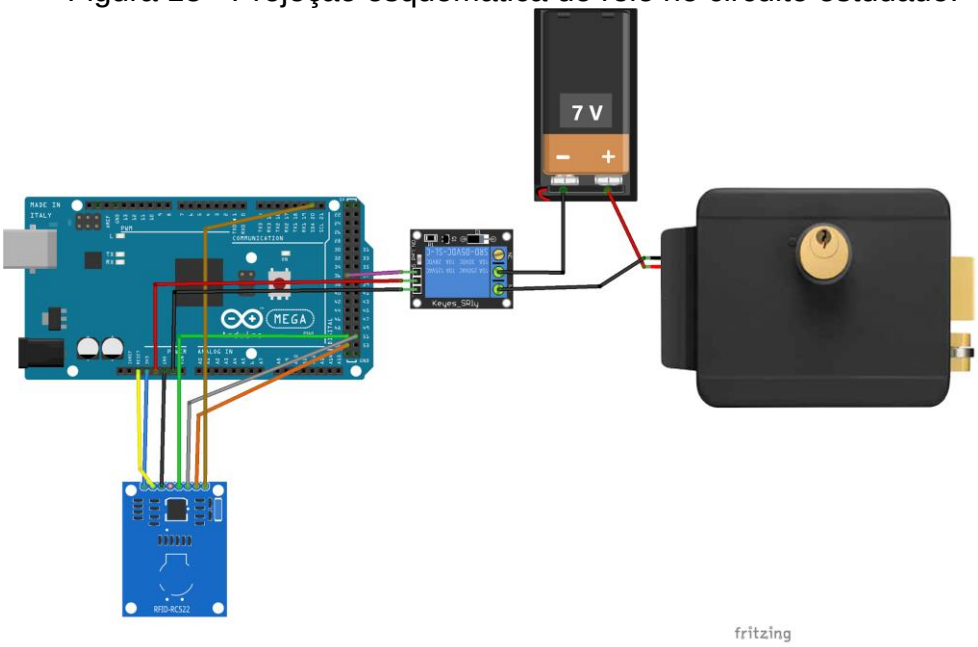
O primeiro passo destinou-se ao arquivamento das bibliotecas, como a *pwm.h*, *SPI.h* e *MFRC522.h*. Por conseguinte, definiu-se as entradas e saídas do Arduino que seriam usadas, bem como os comandos para a inicialização do sistema.

Em seguida, trabalhou-se com os procedimentos de busca, seleção e autenticação do cartão/tag previamente cadastrados no programa para que houvesse, pois, uma resultante liberação da trava pela ativação do relé (que logo cessa) e, consequentemente, sua interrupção.

Então, com o uso de uma fonte DC ajustada para fornecer 7V, acresceu-se ao presente projeto um módulo de relé com tensão correspondente a 5V, tendo-se em

vista que era necessário, para as análises desejadas com a fechadura, um elemento que pudesse gerar um controle de tensão e consolidar, portanto, a automação do sistema para que fosse possível associar o sensor RFID, o Arduino Mega e a fechadura trabalhada. Logo, o módulo de relé tem sua tensão nominal somada à da fonte, o que resulta na tensão válida para funcionamento daquela.

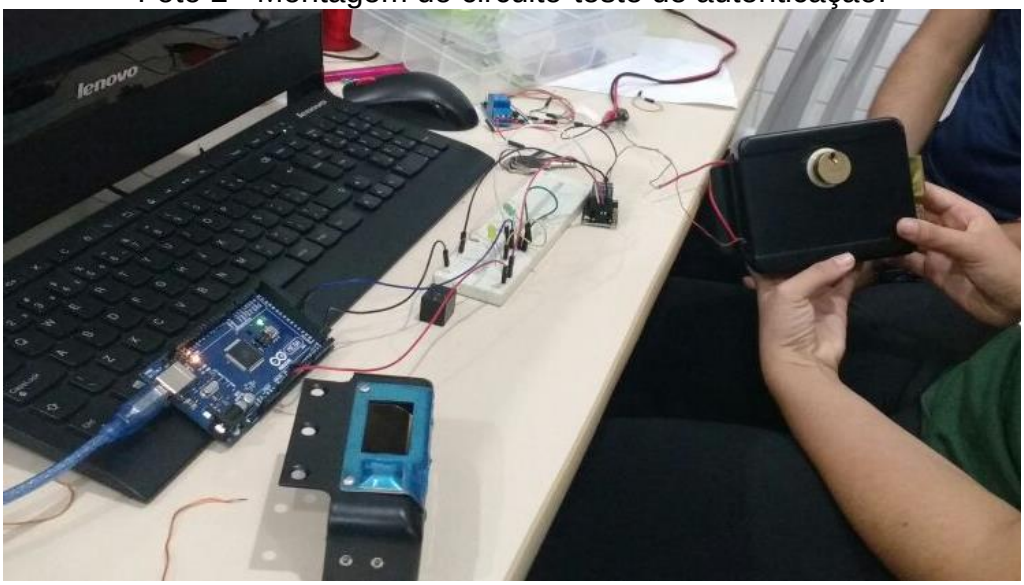
Figura 15 - Projeção esquemática do relé no circuito estudado.



Fonte: Dos autores (2018).

Somado a esse conjunto, incluiu-se, ainda, alguns leds para visualização prática da relação de funcionamento relé-fechadura, conforme ilustrado pela foto 2 abaixo:

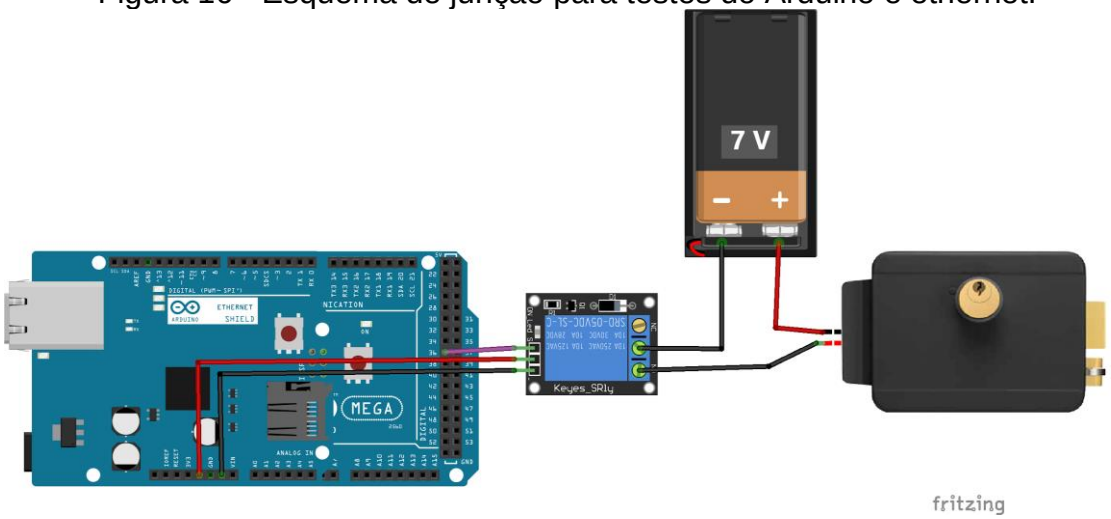
Foto 2 - Montagem do circuito-teste de autenticação.



Fonte: Dos autores (2018).

3.2 Arduino e Ethernet

Figura 16 - Esquema de junção para testes do Arduino e ethernet.



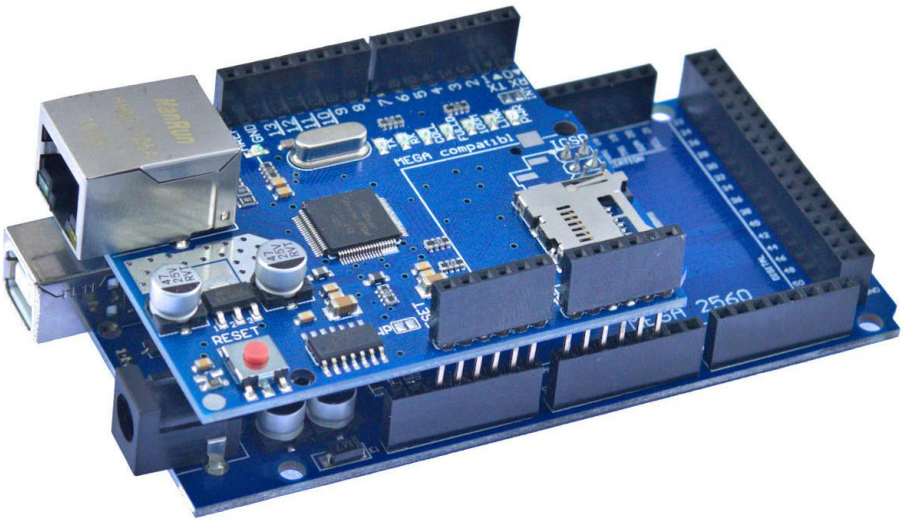
Fonte: Dos autores (2018).

Conseguinte à montagem e série de testes feitos entre o Arduino e o RFID para comprovar a funcionalidade do sistema, aplicou-se o Shield Ethernet ao projeto, posto que é responsável pela conexão do Arduino à internet. Para instalá-lo ao Arduino, o shield é conectado sobre as entradas desse microcontrolador, o que permite ao usuário ter acesso aos mesmos terminais (antes, disponíveis diretamente no Arduino; com a conexão dos dois dispositivos, pelo shield).

Ademais, se conecta ao sistema atual um cabo ethernet (de cor azulada e longa extensão) para o estabelecimento da ligação entre esse e o computador. Com isso

vinculou-se, ainda, um cabo USB, o qual promove a conversa entre os dispositivos e o computador usado.

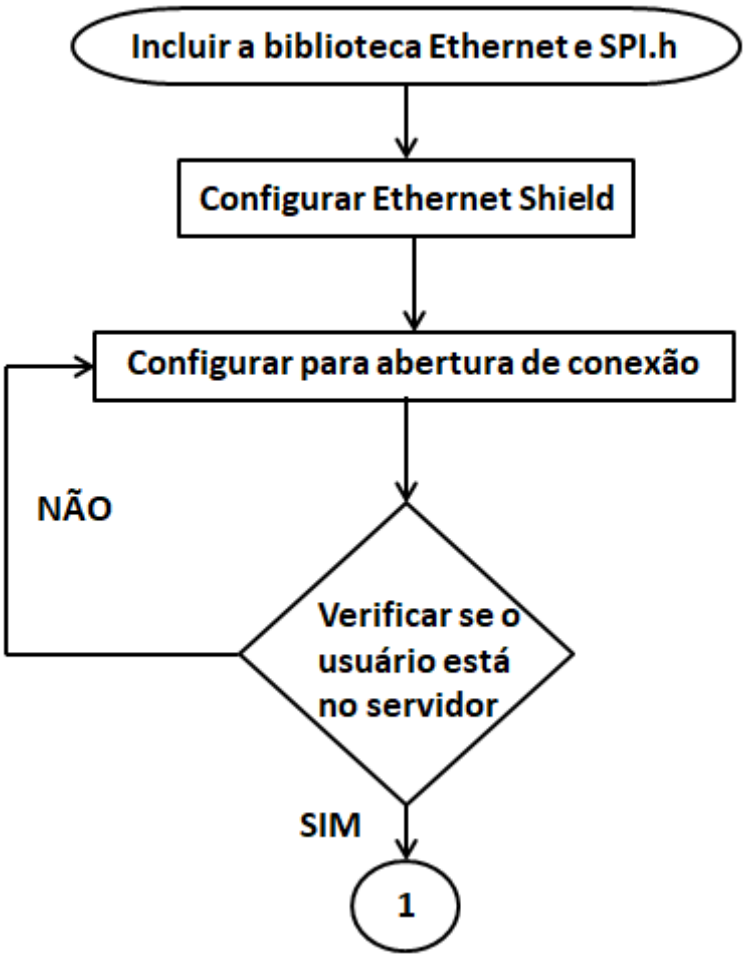
Figura 17 - Shield ethernet acoplado ao Arduino Mega.



Fonte: Portal ebay.

Antes de construir o código para programação do Arduino, idealizou-se um fluxograma com a lógica do programa, ilustrado na figura 17. Inicialmente, incluiu-se as bibliotecas necessárias para a obtenção das informações importantes das funções utilizadas no software. Em sequência, planejou-se realizar a configuração do Ethernet Shield e estruturou-se a abertura da conexão com a rede.

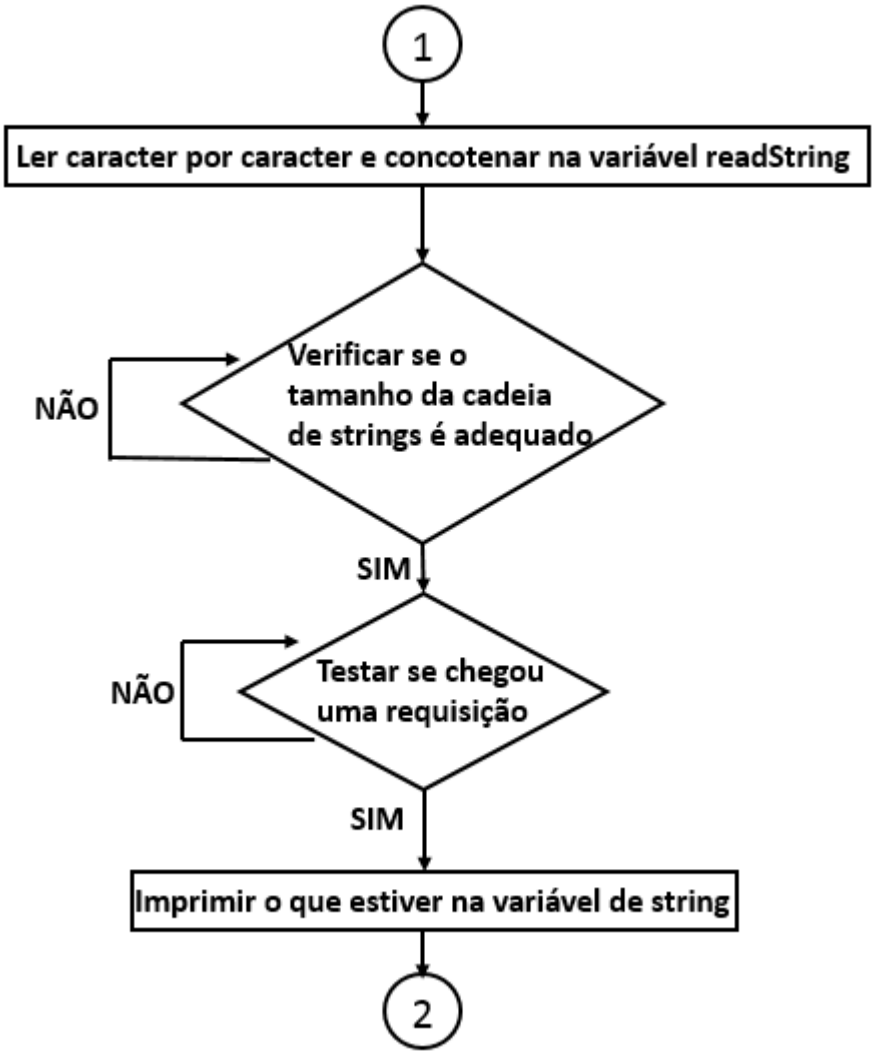
Figura 18 - Fluxograma da lógica Acesso-Ethernet.



Fonte: Dos autores (2018).

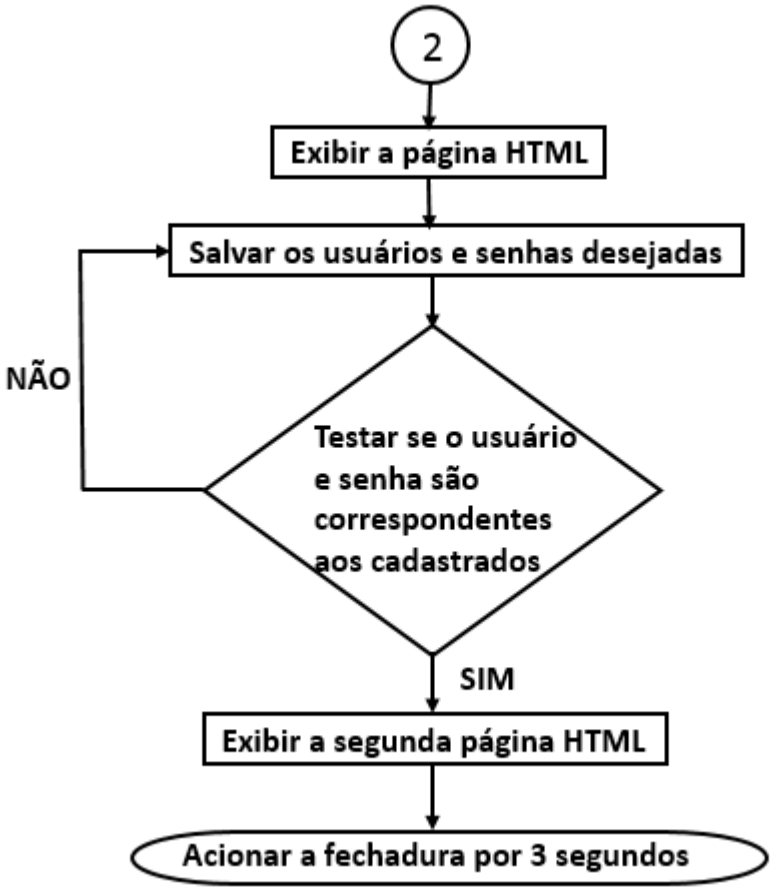
Posteriormente, incorporou-se lógicas para a verificação do usuário no servidor; constatação da chegada de uma requisição HTTP; construção do corpo da página na linguagem de marcação HTML; elaboração de um formulário para cadastro dos ícones de acesso; averiguação das informações fornecidas pelo usuário nos campos de login e senha da página, exibição da segunda página e, por fim, destrava-se a tranca, como mostrado pelas figuras 19 e 20.

Figura 19 - Continuação da série de procedimentos dessa etapa.



Fonte: Dos autores (2018).

Figura 20 - Continuação do fluxograma e etapa final do processo.

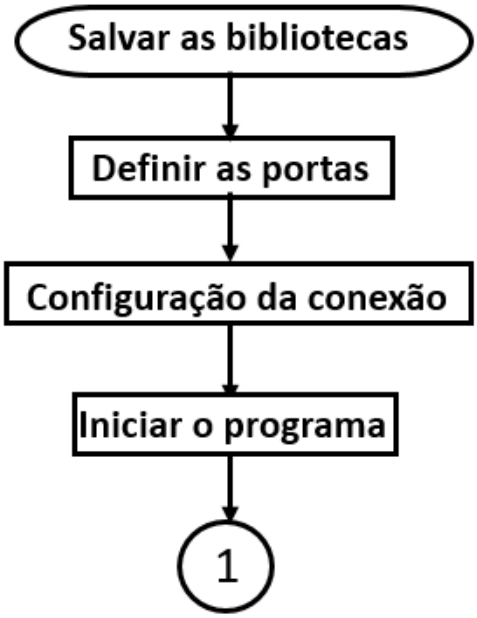


Fonte: Dos autores (2018).

3.3 Arduino, ethernet e RFID

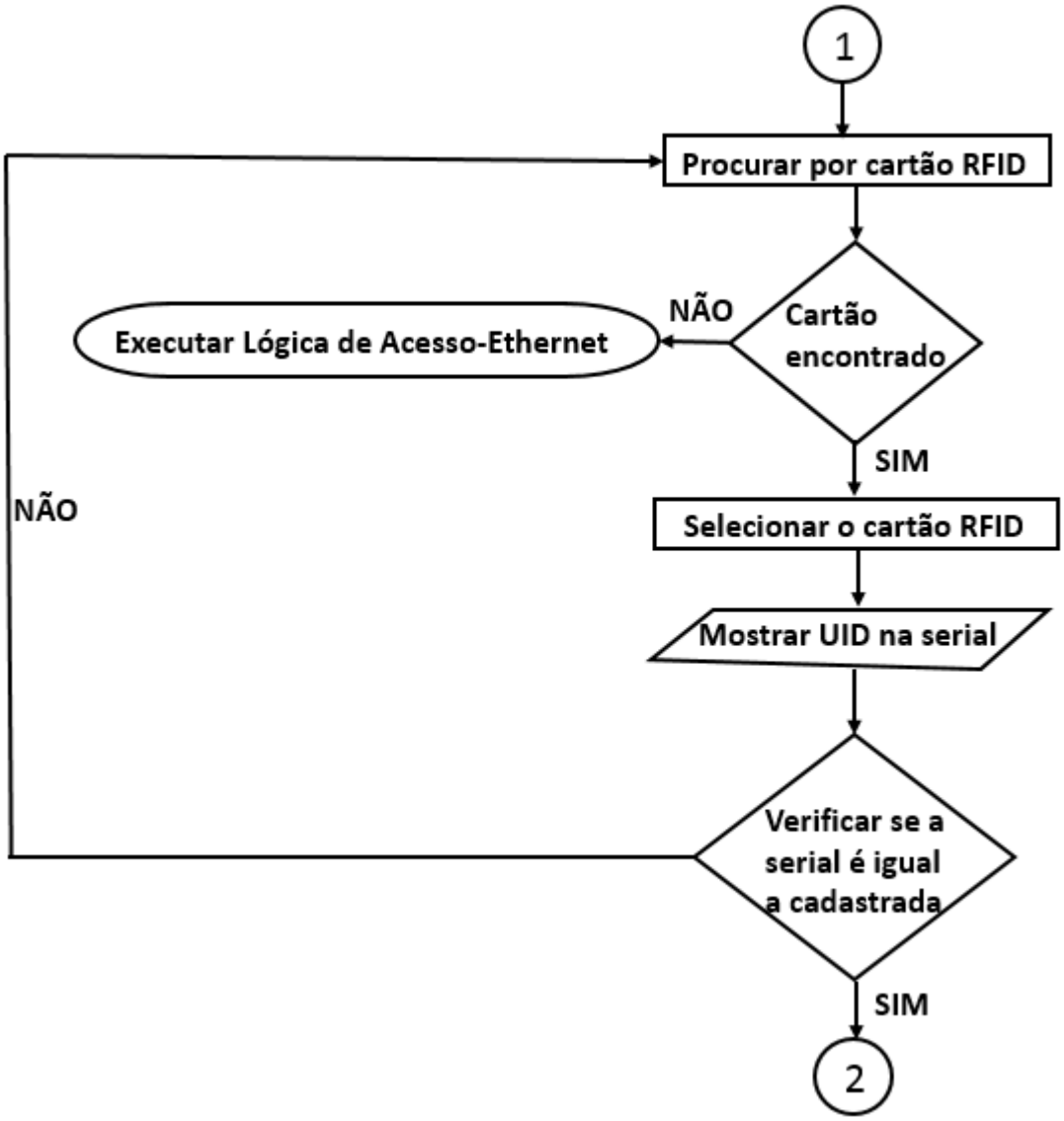
Como procedimento prévio à finalização do projeto, foi feita a idealização da lógica do código final, o qual no instante que é realizada a busca por uma tag RFID, não obtendo êxito na procura e inicializada a execução da lógica Acesso-Ethernet, como alternativa secundária para liberação da tranca, representada por um fluxograma nas figuras 21, 22 e 23.

Figura 21 - Lógica Acesso-Ethernet (parte I).



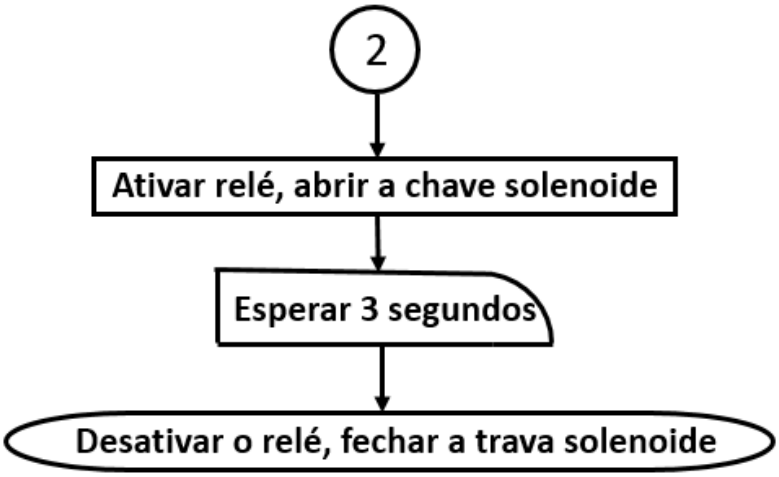
Fonte: Dos autores (2018).

Figura 22 - Lógica Acesso-Ethernet (parte II).



Fonte: Dos autores (2018).

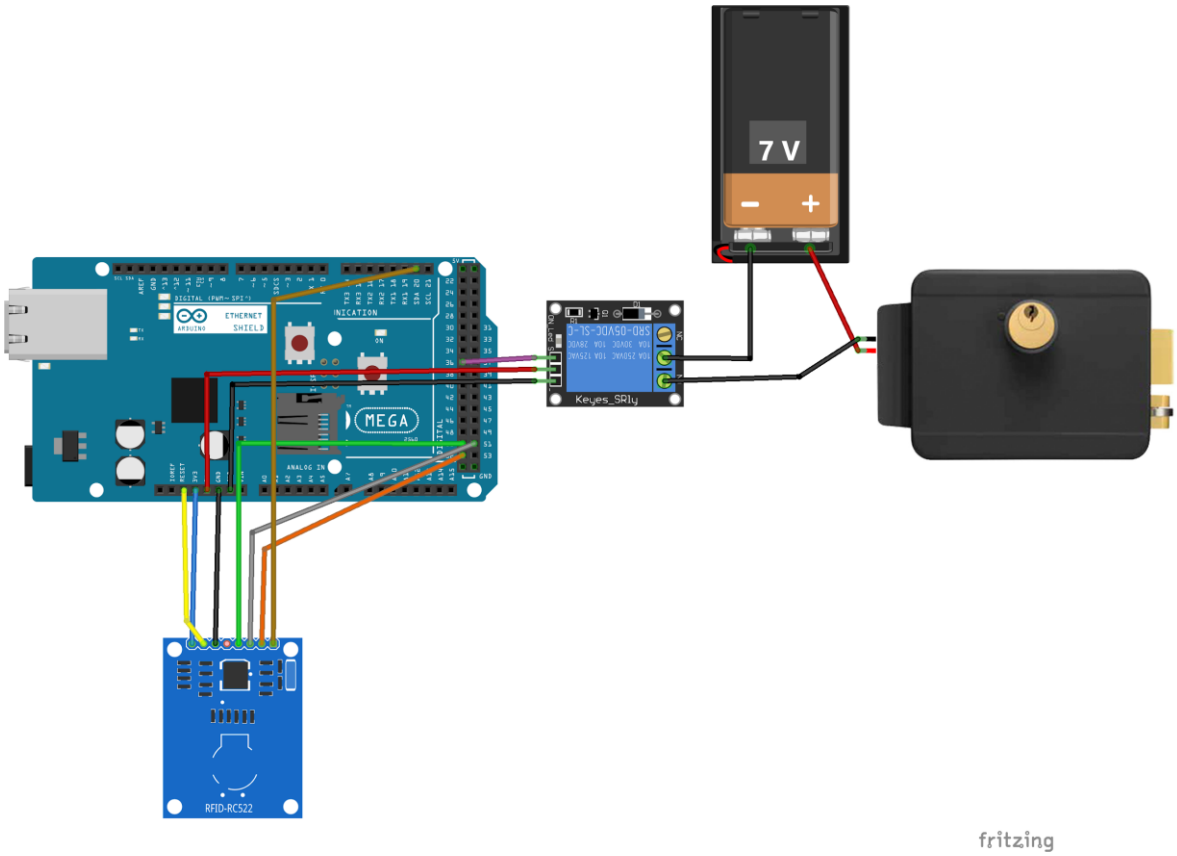
Figura 23 - Lógica Acesso-Ethernet (parte III).



Fonte: Dos autores (2018).

Posteriormente, o grupo desenvolveu o esquemático final do projeto, unindo as propostas citadas nos itens 3.1 e 3.2, representado na figura 24:

Figura 24 - Esquema do circuito final.



Fonte: Dos autores (2018).

4 RESULTADOS E DISCUSSÕES

4.1 Testes básicos

Para os testes iniciais com o sistema Arduino construído, programas como o demonstrado abaixo, na figura 25 (de modo paralelo ao circuito da foto 3) foram desenvolvidos:

Figura 25 - Construção de sistema Arduino-RFID e leds.

Teste_led_fechadura §

```
void setup() //etapa de inicialização do programa
{
  pinMode(38, OUTPUT); //configuração do pino 38 como saída
}

void loop()
{
  digitalWrite(38, HIGH); //atribuição do nível alto para a porta 38 do hardware
  delay(3000); //tempo de espera entre uma ação e outra no loop
  digitalWrite(38, LOW); //atribuição do valor 0V à porta usada
  delay(3000);
}
```

Fonte: Dos autores (2018).

Foto 3 - Circuito eletrônico usado para testes prévios.

Fonte: Dos autores (2018).

Como retorno a essa etapa, obteve-se a visualização prática da transição entre os estados solicitados do programa para o microcontrolador, de acordo com a variação na luminosidade do led conectado ao circuito principal.

Em sequência, foi trabalhado um prospecto de ligação já entre o Arduino e o RFID apresentado na figura 13, no final da seção 3.1. Nesse sistema construído, é excluída a presença de elementos visuais de resposta, como no caso citado há pouco. Assim, posto o objetivo máximo de leitura e apresentação das tags, logrou-se os respectivos códigos de identificação (UIDs):

Tag1: 56 A8 1F 49 (cuja característica diferencial foi ausência de adesivos externos);
Tag2: D9 40 5D 9E.

4.2 Etapas para conexão Arduino-RFID-ethernet

A conjugação do circuito elementar (microcontrolador, sensor de identificação por radiofrequência e shield Ethernet) foi feita, pois, com a fechadura elétrica e o relé. Logo, buscou-se uma programação adequada para a autenticação das tags, conexão Web, exibição de uma página web na linguagem HTML e, consonante a isso, do comportamento desse dispositivo eletromecânico.

Para tanto, o grupo se baseou em dois códigos originais: um da página FilipeFlop e outro de um colaborador, localizado na internet (pertencente a Arthur Oliveira). Então, investiu-se nas adequações necessárias ao código para a construção de um código final.

Adaptou-se, assim, o campo das tags e as configurações do computador utilizado; fez-se a caracterização da página com base nas necessidades da equipe: a inclusão do campo de login e o cadastro de mais de um usuário e das respectivas respostas.

Na introdução do programa, figura 26, além de incluir-se as bibliotecas necessárias para leitura (do sensor de radiofrequência) e conexão com o shield Ethernet, foi feita a configuração dos pinos destinados à saída do circuito e dos protocolos de comunicação trabalhados.

Figura 26 - Bibliotecas usadas no código modificado em C.

Incluindo_a_fechadura_e_o_rfid §

```
#include <deprecated.h>
#include <MFRC522.h>
#include <MFRC522Extended.h>
#include <require_cpp11.h>
#include <PWM.h>
#include <SPI.h>
#include <Ethernet.h> //Inclui as bibliotecas.

#define SS_PIN 10
#define RST_PIN 9  //define as portas
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Fonte: Dos autores (2018).

Por conseguinte, fez-se a configuração da conexão (figura 27), em conjunto com a inclusão de dados do computador no qual o programa irá funcionar, como o valor do MAC, o IPv4 (com o último número modificado), o IP *gateway*, a máscara de subrede e a porta de acesso do Ethernet. Esses foram encontrados no *Prompt* de Comando (cmd.exe) do computador.

Figura 27 - Configuração da conexão (Ethernet-shield).

```
//*******************************CONFIGURAÇÃO DA CONEXÃO *********************************
//Configuraçao Ethernet Shield
byte mac[] = { 0xF8, 0x0F, 0x41, 0x8F, 0xC7, 0x33 }; // Entre com o valor do MAC
IPAddress ip(169,254,48,140); // Configure um IP válido
byte gateway[] = {10,83,0,1}; //Entre com o IP do Computador onde a Câmera esta instalada
byte subnet[] = { 255, 255, 0, 0 }; //Entre com a Máskara de Subrede
EthernetServer server(8080); /*Inicializa a biblioteca EthernetServer com os valores de IP
acima citados e configura a porta de acesso(80)
//=============================================================================================
```

Fonte: Dos autores (2018).

Posteriormente, como registra a figura 28, declarou-se as variáveis do tipo *boolean* e *string*, usadas para a verificação do estado da conexão e armazenamento da requisição, respectivamente. Afora isso, vê-se um *void setup*, responsável pela abertura da conexão; por definir o pino de saída do circuito (nesse caso, o 36); pela velocidade da taxa de transmissão; e por inicializar o protocolo de comunicação serial e a biblioteca *MFRC522*.

Figura 28 - Declaração de variáveis e abertura da conexão.

```
boolean logado = false; // variavel que vai guardar se esta logado ou nao.
String StringOne;
String readString;// Irá guardar a string das requisições HTTP

void setup(){ //configuração para abertura de conexão
    pinMode(36, OUTPUT); //configuração do pino 36 como saída
    Ethernet.begin(mac, ip);
    Serial.begin(9600); //velocidade da taxa de transmissão
    SPI.begin();       // Inicializa o protocolo de comunicação serial
    mfrc522.PCD_Init();   // Inicia MFRC522 e ativa a biblioteca
    Serial.println("Aproxime o seu cartao do leitor..."); //texto a ser impresso
    Serial.println();
    server.begin();
}//FIM VOID SETUP
```

Fonte: Dos autores (2018).

Ao iniciar a função *void loop ()*, verifica-se se há a presença de uma tag RFID, como apresentado na figura 29. Senão, o programa irá averiguar se existe alguma requisição no sistema web, pegá-la e fazer o armazenamento de caractere por

caractere na variável *readString.* Após isso, quando na detecção de quebras de linha, imprimirá no monitor serial que aquela requisição condiz com a configurada. Assim, permite a impressão da página HTML e define, inicialmente, as propriedades de tamanho e cor da interface.

Figura 29 - Verificação da presença da tag RFID.

```
void loop(){

  // Procura por cartão RFID
  if ( ! mfrc522.PICC_IsNewCardPresent())
  { //Se o cartão não for encontrado, verifique se há alguma requisição no sistema web.
// ***************************************Pagina HTML*********************************
 EthernetClient client = server.available();
 if (client) {
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      //Realiza a leitura de caracter por caracter  e vai concatenando no readString.
      if (readString.length() < 100) {
        readString += c;
      }
      //Inicio da pagina html, imprimindo o cabeçalho.
      if (c == '\n') { //Verifica se houve uma quebra de linha na requisição
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println();
        //Inicio dos códigos html.
        client.println("<HTML>");
        client.println("<head>");
        client.println("<meta name=\"viewport\" content=\"width=320\">");
        client.println("<meta name=\"viewport\" content=\"width=device-width\">");
        client.println("<meta charset=\"utf-8\">");
        client.println("<meta name=\"viewport\" content=\"initial-scale=1.0, user-scalable=no\">");
        client.println("</head>");
        client.println("<BODY bgcolor='#457FCL'>"); //Define a cor do corpo da página
        //Termino do cabecalho, entra quando estiver ou nao logado
```

Fonte: Dos autores (2018).

Em seguida, verifica-se o estado da variável booleana *logado*. Na sequência, em caso de ser falsa essa condição, expõe-se o formulário. Caracteriza-se, também, a fonte e cores usadas, os ícones de login e senha, além do botão de confirmação, como mostra a figura 30:

Figura 30 - Condição para o formulário.

```
        //Verificar o estado da variavel logado, caso false realiza a impressão de tudo que compor esse if
        //Página de login
if(logado == false){
        client.println("<H1><font face='arial' color='#FFFFFF'><center> Faça login para abrir a fechadura!</center></font></H1>");
        //Define a fonte do texto, a cor e o alinhamento ao centro.
        client.println("<center><img src='LINK DA IMAGEM DESTAQUE'</img></center><br>");
        //Define a imagem a ser utilizada.
    //aqui é um formulario basico
        client.println("<center><form>");
        client.println("<p>Login: <input type='login' name='feinl' /></p>"); //Inserção do login.
        client.println("<p>Senha: <input type='password' name='feinp' /></p>"); //Inserção da senha.
        client.println("<input type='submit' value='Ok' />"); //Inserção do botão.
        client.println("</form></center>");
```

Fonte: Dos autores (2018).

Por conseguinte, a análise se debruça sobre as informações digitadas nos campos do formulário por parte dos usuários. Caso correspondam ao esperado, imprimir-se-á na tela a mensagem de “login ok” e será invertida a variável *logado* para verdadeira, como exibe a figura 31 abaixo:

Figura 31 - Comparação entre dados fornecidos e os cadastrados.

```
  //Verificar o que vem pela string.
  //Ao digitar a senha ela vai aparecer na url
  //O arduino realiza a leitura e verifica se é igual ao que está nesse if.
if(readString.indexOf( "?feinl=eva&feinp=123") > 0){
    //Verifica se é igual ao primeiro login e senha cadastrados
    //Emite a menssagem de "login ok" e altera o valor de logado de false para TRUE.
    //Quando estiver em true, exibirá o else.
       client.println("<center><H1>Login ok</H1></center>"); //Exibe a mensagem de "login ok".
       logado = !logado;  //Nega a variavel logado.
       client.println("<meta http-equiv=refresh content=0;URL=/>");
           }
      else if (readString.indexOf( "?feinl=laryssa&feinp=456") > 0){
        //Verifica se é igual ao segundo login e senha cadastrados
       client.println("<center><H1>Login ok</H1></center>");
       logado = !logado;
       client.println("<meta http-equiv=refresh content=0;URL=/>");
           }
       else if (readString.indexOf( "?feinl=rodolfo&feinp=789") > 0){
        //Verifica se é igual ao terceiro login e senha cadastrados
       client.println("<center><H1>Login ok</H1></center>");
       logado = !logado;
       client.println("<meta http-equiv=refresh content=0;URL=/>");
           }
  }//Fim do logado false.
```

Fonte: Dos autores (2018).

Com a inversão da variável, sucede que o programa exibirá a segunda página, correspondente a abertura da tranca por um período de 3 segundos. Em tal caso, tem-se a interrupção do acesso e a finalização da página HTML, como demonstra a figura 32:

Figura 32 - Exibição da segunda página e acesso à tranca.

```
  //************************ parte depois de logado = true ***********************************************
  else{
// Tratando-se de um loop, verificação se a variavel logado agora está TRUE, substituirá a tela pela seguinte.
      client.println("<center>");
      client.println("<h1><font face=\"Comic Sans\" color='#FFFFFF'>Olá, bem-vindo ao MARIA!</font></h2>");
      client.println("<center><img src='LINK DA IMAGEM CENTRAL'</img></center><br>");
      client.println("<br><br><br>");
      client.println("</form> <br />");
      client.println("</center>");
  }// fim do else quando vc estiver logado
      client.println("</BODY>");
      client.println("</HTML>");
      client.stop();
      readString="";

      Serial.println ("Pode entrar");
          digitalWrite(36, HIGH);
          delay(3000);
          digitalWrite(36, LOW);
          delay(3000);
          client.stop();

      }//fim if c == '\n'
    }// fim if client.available
  } //fim while cliente logado
 }// fim if client

//************************************************** FIM HTML ********************************************
```

Fonte: Dos autores (2018).

A continuidade do código trata da leitura dos endereços vinculados a cartões ou *tags* RFID identificadas pelo sistema. Nesse processo, é feito o recolhimento de caractere por caractere, além do armazenamento subsequente na variável *conteúdo*.

Figura 33 - Reconhecimento e leitura de cartões/*tags* RFID.

```
// Seleciona o cartao RFID
if ( ! mfrc522.PICC_ReadCardSerial())
{
  return;
}
//Mostra UID na serial
Serial.print("UID da tag :");
String conteudo= "";
byte letra;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
   Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
   Serial.print(mfrc522.uid.uidByte[i], HEX);
   conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
   conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Mensagem : ");
conteudo.toUpperCase();
```

Fonte: Dos autores (2018).

Por fim, é feita uma comparação entre a UID armazenada na string e a referente ao cartão cadastrado, como ilustra a figura 34. Em caso de correspondência, o relé será ativado e há a liberação da trava solenoide, durante 3 segundos, e desativa-se o relé.

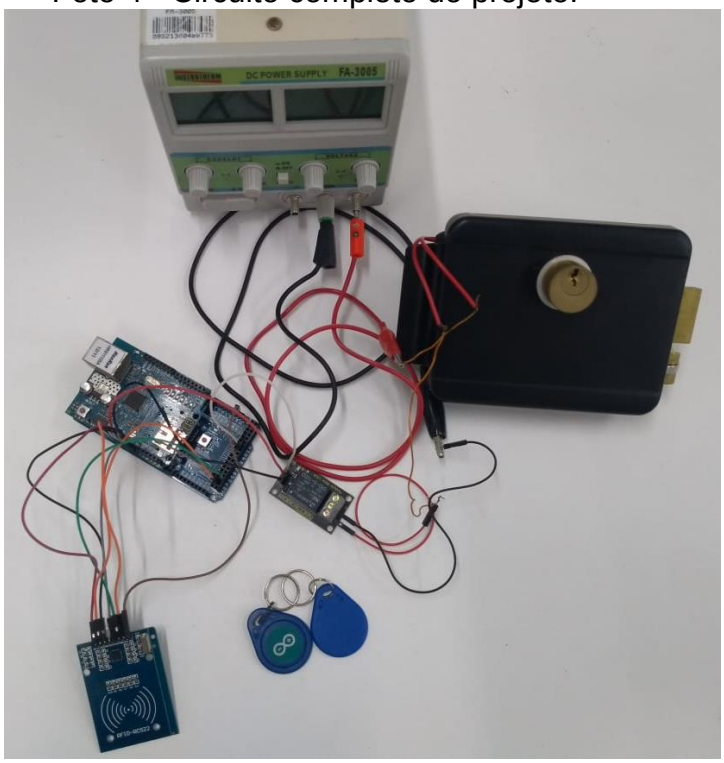
Figura 34 - Comparação de UIDs.

```
 if (conteudo.substring(1) == "F3 DE 90 55") //UID 1 - Cartao
 {
   Serial.println("Bem Vindo ao Maria!");
   Serial.println();
   digitalWrite(36, HIGH); // ativa rele, abre a trava solenoide
   delay(3000);            // espera 3 segundos
   digitalWrite(36, LOW);  // desativa rele, fecha a trava solenoide
 }
}//FIM VOID LOOP
```

Fonte: Dos autores (2018).

Finalizado o código, implementou-se um circuito (foto 4), construído com base no esquemático estudado. Nessa configuração, o terminal positivo é conectado à fechadura, que recebe também o terminal normalmente aberto (NO, do inglês *Normally open*) do relé, cujo terminal comum (ao centro) se dirige ao outro receptor da trava.

Foto 4 - Circuito completo do projeto.

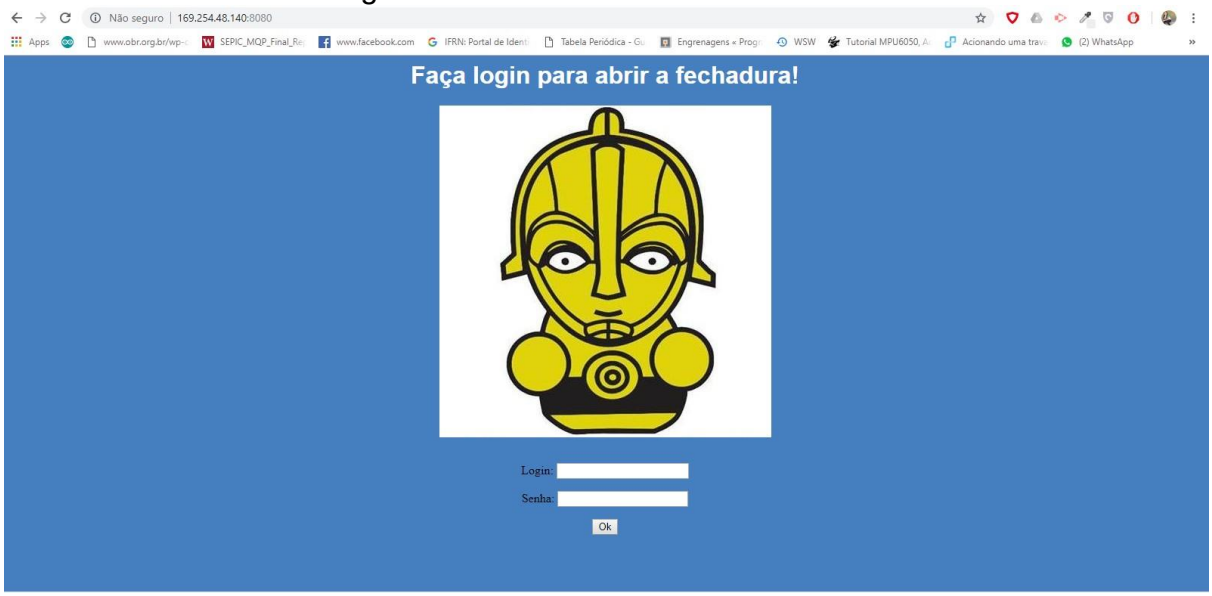


Fonte: Dos autores (2018).

4.2.1 Interfaces código-usuários

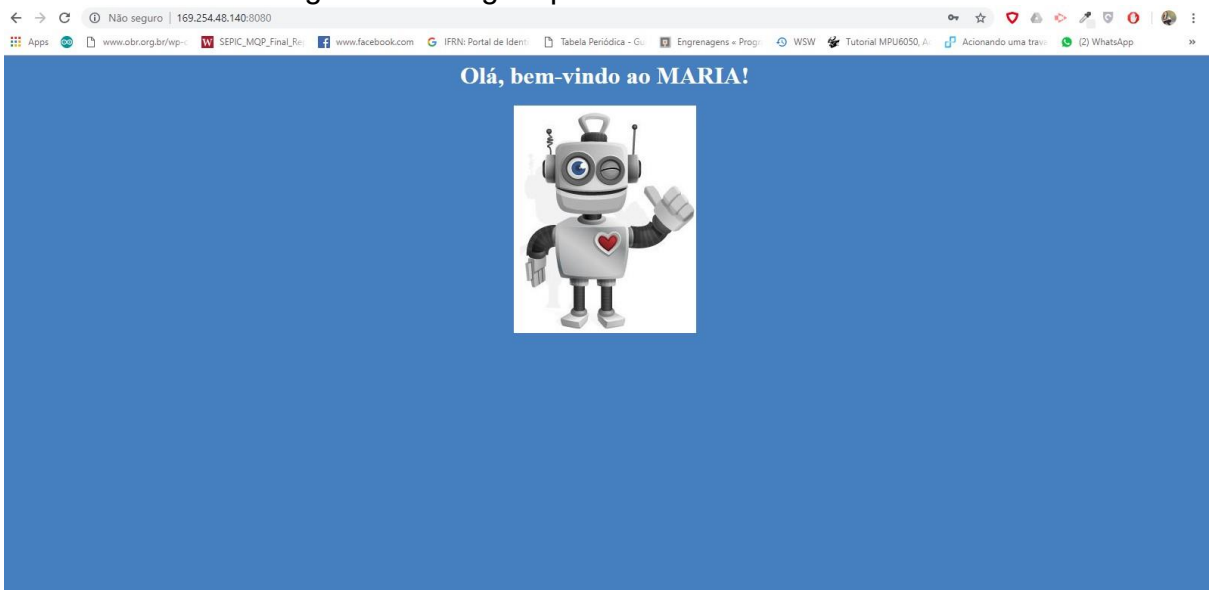
O grupo obteve, como produtos do código final, as seguintes páginas (figura 35 e 36). A primeira delas contempla os espaços com os quais os usuários têm contato no processo de autenticação de acesso, no qual é exposto o campo para login e outro para senha, e o ícone de confirmação dessas informações.

Figura 35 - Site de acesso ao sistema.



Fonte: Dos autores (2018).

Figura 36 - Página para usuários cadastrados.



Fonte: Dos autores (2018).

A página resposta, ilustrada acima, é instantaneamente revelada quando o acesso é permitido, isto é, no momento em que as informações cedidas pelos usuários são equivalentes às registradas no sistema de armazenamento do projeto, apresenta-se com uma saudação de entrada e uma ilustração simbólica da receptividade do ambiente.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento desse projeto possibilitou ao grupo um contato frequente e profundo com o sistema Arduino e tecnologias amplamente relacionadas à Informática, sobretudo. Durante os estudos, testes em laboratório foram muito presentes e isso promoveu a interação entre grande parte das propostas teóricas constantemente almejadas nas reuniões e o desafio máximo de conciliar uma página web e o sistema físico em potencial.

Com base na metodologia utilizada, foi possível compreender a utilidade de um projeto simples mas também de muita praticidade para o ambiente em que será disposta, tendo-se em vista a pluralidade de opções para acesso que os visitantes autorizados terão.

A implementação de sua estrutura no laboratório de robótica e automação do campus designa, assim, a primeira das expectativas futuras que o projeto alcança. Visa-se, também, a correção de um erro na etapa de transição entre uma página web e outra: os dados vinculados aos campos preenchidos pelos usuários autorizados são exibidos na URL, ou seja, é oportuna a substituição do método *get*.

Associado a isso, posto que foi feita uma construção básica que contempla os ícones e elementos figurativos, pretende-se incrementá-la em momentos póstumos. Por fim, como outro fator a ser melhorado, tem-se a correspondência entre o hardware e software, tendo-se em vista que a atualização da página de acesso só é feita mediante o reenvio do código no IDE do Arduino.

REFERÊNCIAS

ANDROID + Arduino + Ethernet shield (socket). Disponível em: <https://www.portugal-a-programar.pt/forums/topic/63113-snippset-android-arduino-ethernet-shield-socket/>. Acesso em: 25 out. 2018.

ARDUINO Mega 2560. Disponível em: <http://www.mantech.co.za/datasheets/products/A000047.pdf>. Acesso em: 25 out. 2018.

ARNDT, Fabiano. **Arduino - Autenticação HTTP de Usuário e Senha - Ethernet Shield.** 2015. Disponível em: <http://fabianoallex.blogspot.com/2015/02/arduino-autenticacao-http-de-usuario-e.html>. Acesso em: 30 out. 2018.

BAUERMEISTER, Giovanni. **Acionando uma trava elétrica com RFID.** 2017. Disponível em: <https://www.filipeflop.com/blog/acionando-trava-eletrica-com-rfid/>. Acesso em: 30 out. 2018.

BIANCHIN, J. A. Pereira; Carlos G.; LANGNER, Cristiane G.; CHUEIRI, Ivan J. **Fechadura Eletrônica.** 2017. 3 f. TCC (Graduação) - Curso de Eletricidade, Instituto de Tecnologia Para O Desenvolvimento, Rio de Janeiro, 2007.

BOBINA - Indutor: Bobinas e Indutores. Bobinas e Indutores. Disponível em: <https://www.electronica-pt.com/bobina>. Acesso em: 25 out. 2018.

BONSOR, Kevin; FENLON, Wesley. **How RFID Works.** Disponível em: <https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid.htm>. Acesso em: 25 out. 2018.

CASSIOLATO, César. **Relés x Sensores.** 2018. Disponível em: <http://www.smar.com/brasil/artigo-tecnico/reles-x-sensores>. Acesso em: 20 out. 2018.

COMMUNICATIONS, Axis. **Network technologies.** 2018. Disponível em: <https://www.axis.com/pt-br/learning/web-articles/technical-guide-to-network-video/lan-and-ethernet>. Acesso em: 18 set. 2018.

CUNHA, Alessandro. **RFID – Etiquetas com eletrônica de ponta.** 2016. Disponível em: <https://www.embarcados.com.br/rfid-etiquetas-com-eletronica-de-ponta/>. Acesso em: 20 out. 2018.

DIAS, Rafa. **O que é uma rede LAN Ethernet?** Disponível em: <http://www.drivermax.com.br/driver/drivers-lan-ethernet/>. Acesso em: 21 set. 2018.

EIS, Diego. **O básico: O que é HTML?** Disponível em: <https://tableless.com.br/o-que-html-basico/>. Acesso em: 06 set. 2018.

ETACARINAE. **Arduino 10: funções, listas, bibliotecas.** 2017. Disponível em: <http://eletronicaparaartistas.com.br/arduino-10-funcoes-listas-bibliotecas/>. Acesso em: 30 out. 2018.
FECHADURAS Digitais: Bem vindo ao mundo sem chaves. Disponível em: <http://www.webprom.net.br/fechaduras-e-controle-de-acesso.html>. Acesso em: 26 out. 2018.

GRUPO OITO ARDUINO. **História do Arduino e seus modelos.** 2015. Disponível em: <https://arduinoaprendizes.wordpress.com/2015/04/22/historiaarduino/>. Acesso em: 06 set. 2018.

GUGELMIN, Felipe. **Internet: qual a diferença entre os protocolos UDP e TCP?** 2014. Disponível em: <https://www.tecmundo.com.br/internet/57947-internet-diferenca-entre-protocolos-udp-tcp.htm>. Acesso em: 20 out. 2018.

HACHOUCHE, Anwar S. **Apostila Arduino Básico V1.0.** 2018. Disponível em: <http://apostilas.eletrogate.com/Apostila_Arduino_Basico-V1.0-Eletrogate.pdf>. Acesso em: 30 out. 2018.

HENRIQUE, Tiago. **Comunicação I2C.** 2012. Disponível em: <http://microcontrolandos.blogspot.com/2012/12/comunicacao-i2c.html>. Acesso em: 20 set. 2018.

ISPBLOG. **Rede ethernet industrial: 4 benefícios dessa tecnologia.** 2016. Disponível em: <https://www.ispblog.com.br/2016/06/17/rede-ethernet-industrial-4-beneficios-dessa-tecnologia/>. Acesso em: 18 set. 2018.

LOUREIRO, Gabriel da Silva Martins et al. **RFID:** Identificação por Rádio Frequência. Disponível em: <https://www.gta.ufrj.br/grad/15_1/rfid/segurepriv.html>. Acesso em: 27 set. 2018.

MADEIRA, Daniel. **SHIELD ETHERNET W5100 - CRIANDO UM SERVIDOR WEB COM ARDUINO:** Primeiros passos na criação de um Servidor Web com Arduino. Disponível em: <https://portal.vidadesilicio.com.br/shield-ethernet-w5100-servidor-web/>. Acesso em: 25 out. 2018.

MCROBERTS, Michael. **Arduino:** Básico. São Paulo: Novatec Editora, 2011. 456 p.
MEIRELLES, Adriano. **Redes, Guia Prático 2ª Ed.:** Portas TCP e UDP. 2008. Disponível em: <https://www.hardware.com.br/livros/redes/portas-tcp-udp.html>. Acesso em: 20 out. 2018.

MORIMOTO, Carlos E. REDES: Guia prático. Disponível em: <https://www.hardware.com.br/livros/redes/portas-tcp-udp.html>. Acesso em: 09 out. 2018.

NUNES, Felipe Vilela. **DESENVOLVIMENTO DE SISTEMA DE SEGURANÇA UTILIZANDO MICROCONTROLADOR PIC18F4550.** 2017. 46 f. Monografia (Especialização) - Curso de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2013.

O QUE as fechaduras podem fazer pela sua segurança. Disponível em: <http://www.netseg.com.br/not.php?id=6547>. Acesso em: 25 set. 2018.

O QUE é o Arduino? Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 06 set. 2018.

OLIVEIRA, Euler. **Como usar com Arduino – Ethernet Shield W5100 (Web server).** 2018. Disponível em: <http://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-ethernet-shield-w5100-via-web-server/>. Acesso em: 20 out. 2018.

POR QUE OFERECER UM SISTEMA DE SEGURANÇA RESIDENCIAL COM AUTOMAÇÃO? Disponível em: <https://www.neocontrol.com.br/news/sistema-de-seguranca-residencial/>. Acesso em: 26 out. 2018.

PREDIGER, Daniel; FREITAS, Edison Pignaton de; SILVEIRA, Sidnei Renato. **Modelo de Aplicabilidade de Sistema RFID para Rastreabilidade na Indústria Alimentícia.** 2018. Disponível em: <http://w3.ufsm.br/frederico/images/ModelodeAplicabilidadedeSistemaRFIDparaRastreabilidadena_Ind%C3%BAstriaAliment%C3%ADcia.pdf>. Acesso em: 20 out. 2018.

PTCOMPUTADOR. **Quais são os benefícios do protocolo Ethernet**. 2017. Disponível em: <http://ptcomputador.com/Networking/ethernet/66156.html>. Acesso em: 18 set. 2018.

RFID TAGS AND HARDWARE. 2018. Disponível em: <https://www.rfidinc.com/>. Acesso em: 20 out. 2018.

SACCO, Francesco. **Comunicação SPI – Parte 1.** 2014. Disponível em: <https://www.embarcados.com.br/spi-parte-1/>. Acesso em: 28 jun. 2017.

SAIBA O QUE É E COMO FUNCIONA A REDE ETHERNET. Disponível em: <https://www.impressorajato.com.br/o-que-e-e-como-funciona-a-rede-ethernet>. Acesso em: 18 set. 2018.

SANTINI, Arthur Gambin. **RFID: Conceitos, Aplicabilidades e Impactos.** Rio de Janeiro: Ciência Moderna Ltda., 2008.

SEGURANÇA residencial: 4 equipamentos para uma casa mais segura. 2017. Disponível em: <https://blog.diprel.com.br/seguranca-residencial-4-equipamentos-para-uma-casa-mais-segura/>. Acesso em: 30 out. 2018.

SILVEIRA, Cristiano Bertulucci. **O que é TCP/IP?** Disponível em: <https://www.citisystems.com.br/protocolo-tcp-ip/>. Acesso em: 25 out. 2018.

SIMPÓSIO debate expansão do mercado de sistemas eletrônicos de segurança no Rio Grande do Sul. 2017. Disponível em: <http://revistasegurancaeletronica.com.br/simposio-debate-expansao-do-mercado-de-sistemas-eletronicos-de-seguranca-no-rio-grande-do-sul/>. Acesso em: 30 out. 2018.

THOMSEN, Adilson. **Como comunicar com o Arduino Ethernet Shield W5100**. 2014. Disponível em: <https://www.filipeflop.com/blog/tutorial-ethernet-shield-w5100/>. Acesso em: 20 out. 2018.

THOMSEN, Adilson. **Controle de Acesso usando Leitor RFID com Arduino**. 2014. Disponível em: <https://www.filipeflop.com/blog/controle-acesso-leitor-rfid-arduino/>. Acesso em: 30 out. 2018.

THOMSEN, Adilson. **Qual Arduino Comprar? Conheça os Tipos de Arduino:** Tipos de Arduino. Disponível em: <https://www.filipeflop.com/blog/tipos-de-arduino-qual-comprar/>. Acesso em: 06 set. 2018.

TUDO Sobre Relés. 2018. Disponível em: <http://www.newtoncbraga.com.br/index.php/como-funciona/597-como-funcionam-os-reles?showall=1&limitstart>. Acesso em: 20 out. 2018.

VIDAL, Vitor. **Ethernet Shield W5100 com Arduino – Parte 1**. 2017. Disponível em: < http://blog.eletrogate.com/ethernet-shield-w5100-com-arduino/>. Acesso em: 20 out. 2018.

VIEIRA, Nando. **Entendendo um pouco mais sobre o protocolo HTTP**. 2007. Disponível em: <https://nandovieira.com.br/entendendo-um-pouco-mais-sobre-o-protocolo-http>. Acesso em: 30 out. 2018.

W5100 Datasheet Version 1.1.6. 2008. Disponível em: <https://img.filipeflop.com/files/download/Datasheet_W5100_v1_1_6.pdf>. Acesso em: 20 out. 2018.

APÊNDICE A - Código final do projeto

```
#include <deprecated.h>
#include <MFRC522.h>
#include <MFRC522Extended.h>
#include <require_cpp11.h>
#include <PWM.h>
#include <SPI.h>
#include <Ethernet.h> //Inclusão das bibliotecas.

//(http://169.254.48.140:8080/) // Login para entrada na página.
//***************CONFIGURAÇÃO DA CONEXÃO*****************************
//Configuração Ethernet Shield.
byte mac[] = { 0xF8, 0x0F, 0x41, 0x8F, 0xC7, 0x33 }; // Entre com o valor do MAC.
IPAddress ip(169,254,48,140); // Configure um IP válido.
byte gateway[] = {10,83,0,1}; //Entre com o IP do Computador onde a Câmera está instalada.
byte subnet[] = { 255, 255, 0, 0 }; //Entre com a Máscara de Subrede.
EthernetServer server(8080); //Inicializa a biblioteca EthernetServer com os valores de IP acima citados e configura a porta de acesso(80).
//============================================================

#define SS_PIN 10 //Define as portas.
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);

boolean logado = false; // Variável que vai guardar se está logado ou não.
String StringOne;
String readString;// Irá guardar a string das requisições HTTP.

void setup() { //Configuração para abertura de conexão.
    pinMode(36, OUTPUT); //Configuração do pino 36 como saída.
    Ethernet.begin(mac, ip);
    Serial.begin(9600); //Velocidade da taxa de transmissão.
```

```
    SPI.begin();// Inicializa o protocolo de comunicação serial.
    mfrc522.PCD_Init();  // Inicia MFRC522 e ativa a biblioteca.
    Serial.println("Aproxime o seu cartao do leitor..."); //Texto a ser impresso.
    Serial.println();
    server.begin();
}//FIM VOID SETUP

void loop(){
  // Procura por cartão RFID.
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {//Se o cartão não for encontrado, verifique se há alguma requisição no sistema web.
// ***************************************Página HTML*********************************
 EthernetClient client = server.available();
 if (client) {
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();//Realiza a leitura de caracter por caracter  e vai concatenando no readString.
      if (readString.length() < 100) {
        readString += c;
      }
      //Inicio da página html, imprimindo o cabeçalho.
      if (c == '\n') { //Verifica se houve uma quebra de linha na requisição.
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println();
        //Início dos códigos html.
        client.println("<HTML>");
        client.println("<head>");
        client.println("<meta name=\"viewport\" content=\"width=320\">");
        client.println("<meta name=\"viewport\" content=\"width=device-width\">");
        client.println("<meta charset=\"utf-8\">");
        client.println("<meta name=\"viewport\" content=\"initial-scale=1.0, user-scalable=no\">");
```

```
        client.println("</head>");
        client.println("<BODY bgcolor='#457FCL'>"); //Define a cor do corpo da página
        //Termino do cabecalho, entra quando estiver ou nao logado

        //Verificar o estado da variavel logado, caso false realiza a impressão de tudo que compor esse if
        //Página de login
if(logado == false){
        client.println("<H1><font face='arial' color='#FFFFFF'><center>   Faça login para abrir a fechadura! </center></font></H1>");
        //Define a fonte do texto, a cor e o alinhamento ao centro.
client.println("<center><imgsrc='https://scontent.fcpv2-2.fna.fbcdn.net/v/t1.15752-9/45008076_328905974330485_1964461609615097856_n.png?_nc_cat=101&_nc_ht=scontent.fcpv2-2.fna&oh=0bf4775a00c186357fd325baf8175ec2&oe=5C899203'</img></center><br>");
        //Define a imagem a ser utilizada.
    //Aqui é um formulário básico.
        client.println("<center><form>");
        client.println("<p>Login: <input type='login' name='feinl' /></p>"); //Inserção do login.
        client.println("<p>Senha: <input type='password' name='feinp' /></p>"); //Inserção da senha.
        client.println("<input type='submit' value='Ok' />"); //Inserção do botão.
        client.println("</form></center>");

  //Verificar o que vem pela string.
  //Ao digitar a senha, ela vai aparecer na url, o arduino realiza a leitura e verifica se é igual ao que está nesse if.

if(readString.indexOf( "?feinl=rodolfo&feinp=123") > 0){
    //Verifica se é igual ao primeiro login e senha cadastrados, emite a mensagem de "login ok" e alterna o valor de logado de false para TRUE.
    //Quando estiver em true, exibirá o else.
```

```
      client.println("<center><H1>Login ok</H1></center>"); //Exibe a mensagem de "login ok".
      logado = !logado;  //Nega a variável logado.
      client.println("<meta http-equiv=refresh content=0;URL=/>");
         }

      else if (readString.indexOf( "?feinl=eva&feinp=456") > 0){
       //Verifica se é igual ao segundo login e senha cadastrados.
      client.println("<center><H1>Login ok</H1></center>");
      logado = !logado;
      client.println("<meta http-equiv=refresh content=0;URL=/>");
         }

      else if (readString.indexOf( "?feinl=laryssa&feinp=789") > 0){
       //Verifica se é igual ao terceiro login e senha cadastrados.
      client.println("<center><H1>Login ok</H1></center>");
      logado = !logado;
      client.println("<meta http-equiv=refresh content=0;URL=/>");
         }

  }//Fim do logado false.
  //*************************** parte depois de logado = true ***************************
  else{
// Tratando-se de um loop, verificação se a variável logado agora está TRUE, substituirá a tela pela seguinte.
      client.println("<center>");
      client.println("<h1><font face=\"Comic Sans\" color='#FFFFFF'>Olá, bem-vindo ao MARIA! </font></h2>");
client.println("<center><imgsrc='https://cdn-images-1.medium.com/max/1200/1*5Aw5-fmKGyN3CMM7f2jC1Q.jpeg'</img></center><br>");
      client.println("<br><br><br>");
      client.println("</form> <br />");
      client.println("</center>");
```

```
  }// Fim do else quando estiver logado.
      client.println("</BODY>");
      client.println("</HTML>");
      client.stop();
      readString="";

      Serial.println ("Pode entrar");
          digitalWrite(36, HIGH);
          delay(3000);
          digitalWrite(36, LOW);
          delay(3000);
          client.stop();

      }//Fim if c == '\n'.
    }// Fim if client.available.
  } //Fim while cliente logado.
 }// Fim if client.
//******************************** FIM HTML **********************************
  }
  // Seleciona o cartão RFID.
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Mostra UID na serial.
  Serial.print("UID da tag:");
  String conteudo= "";
  byte letra;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     Serial.print(mfrc522.uid.uidByte[i] < 0x10? " 0" : " ");
     Serial.print(mfrc522.uid.uidByte[i], HEX);
     conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
```

```
}
Serial.println();
Serial.print("Mensagem: ");
conteudo.toUpperCase();

if (conteudo.substring(1) == "F3 DE 90 55") //UID 1 - Cartão.
{
  Serial.println("Bem Vindo ao Maria!");
  Serial.println();
  digitalWrite(36, HIGH); // Ativa relé, abre a trava solenóide.
  delay(3000);           // Espera 3 segundos.
  digitalWrite (36, LOW);  // Desativa relé, fecha a trava solenóide.
}  }//FIM VOID LOOP.
```