

# **PROTÓTIPO DE SISTEMA DE CONTROLE DE ACESSO LABORATORIAL**

Discentes: Felipe da Costa Félix, Glaufe Santos de Oliveira, José Carlos de Medeiros, Rodolfo Pereira da Silva, Tatiane dos Santos Alencar Ferreira, Tiago Oliveira e Wendrio Vinicius Costa Pereira.

Orientador: Prof. Eng.º Dr. Ciro Egoavil Montero

## **RESUMO**

*Este trabalho apresenta a implementação do protótipo de um sistema embarcado que permite o controle e monitoramento online do acesso a laboratórios. O protótipo é formado por uma placa RFID Module V3 NFC PN532, responsável pelo registro de código de cartão NFC (dispositivo passivo que possui em sua estrutura um chip), uma placa de criação de código aberto Arduíno Mega, a qual é dotada de um microcontrolador ATmega 2560 responsável pela realização do comando de acionamento da tranca, um módulo ethernet W5100, sendo este o responsável pela conexão entre o microcontrolador e a internet. O módulo NFC, conectado ao microcontrolador ATmega 2560, realiza a leitura das identificações dos cartões, e as envia ao microcontrolador. No ATmega 2560, os códigos de identificação (IDs) são comparados com os definidos previamente em seu firmware, e em seguida é tomada a decisão de acionamento, onde em caso positivo, os dados de acesso são enviados através de uma porta TCP/IP à um banco de dados online contendo as informações de acesso do usuário do cartão, data e horário de entrada/saída. O projeto permite o controle eletrônico da tranca e o envio dos dados de acesso para o site de monitoramento, atuando assim como um instrumento de segurança laboratorial.*

## **1. INTRODUÇÃO**

Os laboratórios armazenam equipamentos, dispositivos e componentes de custo elevado, quantidade limitada e que requerem conhecimentos específicos para sua utilização, não podendo devido a estes fatores, permanecerem expostos e com acesso livre a qualquer indivíduo. Sendo assim, faz-se necessário o controle da entrada e a

identificação dos seus usuários. Tendo em vista essa necessidade, idealizou-se um sistema capaz de realizar o controle de acesso e enviar os dados de acesso a um banco de dados online, de forma rápida e eficiente.

O monitoramento e controle do fluxo de pessoas no ambiente laboratorial, quando realizado de forma manual e tradicional, é uma atividade que se torna cansativa e ineficaz a medida que o quantitativo de indivíduos aumenta. O trânsito de pessoas não autorizadas no laboratório e a utilização de equipamentos sem supervisão de um técnico ou professor responsável pode colocar em risco a integridade dos equipamentos, a segurança dos presentes, além da existência do risco de furto de matérias e equipamentos.

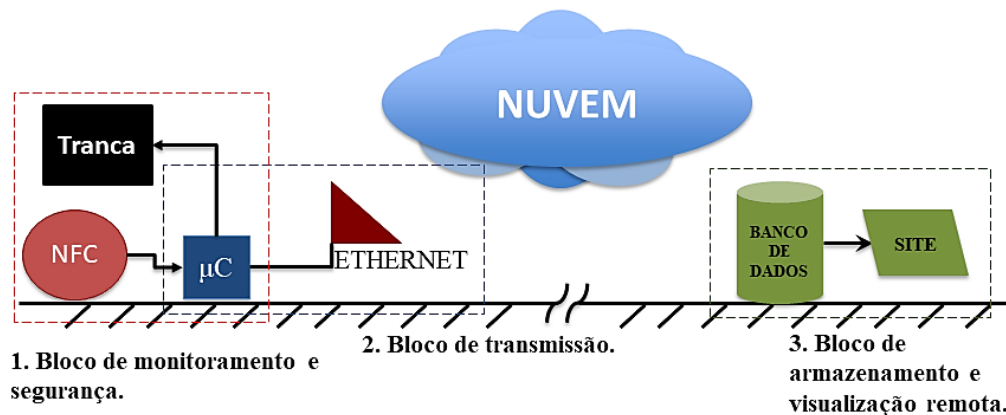
## **2. OBJETIVOS**

Desenvolver um protótipo capaz de realizar o controle eletrônico de uma tranca, a partir de uma lógica hierárquica, e o envio dos dados de acesso do usuário para um site de monitoramento, atuando assim como um instrumento de segurança laboratorial.

## **3. METODOLOGIA**

O projeto foi desenvolvido visando um sistema de controle de acesso laboratorial seguro e automático, que facilitasse o acesso dos alunos ao laboratório, e o trabalho dos técnicos e professores, os quais são responsáveis por monitorar e controlar o acesso ao laboratório. O sistema pode ser segmentado em três blocos: bloco de monitoramento e segurança, bloco de transmissão, e bloco de armazenamento e visualização remoto; como pode ser observado na figura a seguir (figura 1):

Figura 1. Esquemático do sistema completo.



Fonte: Elaborada pelo autor.

### 3.1. Bloco de monitoramento:

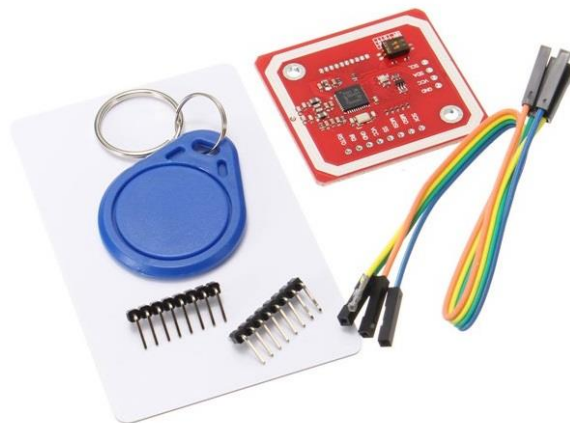
Para a leitura de dados e monitoramento dos acessos ao laboratório, se viu a necessidade de substituir as chaves comuns, por chaves identificadas com algum tipo de código único, que pudesse ser lido e identificado por um sistema eletrônico. Além disso, a chave deveria ser simples e rápida de se utilizar, e principalmente, segura.

Para isso, decidiu-se usar a tecnologia *NFC* (*Near Field Connection*, em português, comunicação de campo próximo), a qual é segura - muito pelo fato de exigir uma distância muito pequena para comunicação, de forma que dificulta a interceptação de dados – e de utilização simples e rápida. A mesma, é utilizada em diversas aplicações, desde a obtenção de mais informações sobre determinada obra de arte em um museu, através da aproximação de um celular com *NFC* em um determinado ponto do museu; até mesmo a pagamentos e transferências bancárias, utilizações as quais, reforçam a segurança da tecnologia, e sua versatilidade e simplicidade de uso (ALECRIM, 2017).

Em poucas palavras, o *NFC* é uma especificação que permite a comunicação sem fio (*wireless*) entre dois dispositivos mediante uma simples aproximação entre eles, sem que o usuário tenha que digitar senhas, clicar em botões ou realizar alguma ação semelhante ao estabelecer a conexão (ALECRIM, 2017).

Com o intuito de utilizar esta tecnologia, decidiu-se usar o leitor PN532 NFC/RFID (figura 2), que pode ler e grava informações utilizando a tecnologia *NFC*, em uma distância de até sete centímetros (ARDUINO E CIA, 2016), e é compatível com o microcontrolador escolhido, o qual, é o Atmega2560, que foi utilizado em uma placa Arduigo Mega 2560 (figura 3).

Figura 2. Leitor *NFC* PN532 *NFC/RFID*.



Fonte: [www.blog.filipeflop.com](http://www.blog.filipeflop.com) (2015)

Disponível em: <<https://www.filipeflop.com/produto/kit-modulo-leitor-rfid-nfc-pn532/>>. Acesso em dezembro de 2017.

Figura 3. Placa Arduino MEGA 2560.



Fonte: [www.arduino.cc](http://www.arduino.cc).

Disponível em: <<https://store.arduino.cc/usa/arduino-mega-2560-rev3>>. Acesso em dezembro de 2017.

Integralmente com o leitor *NFC*, cartões *RFID* (*Radio-Frequency IDentification*, em português, identificação por radiofrequência), foram escolhidos como chaves. A tecnologia *RFID* e *NFC*, funcionam basicamente da mesma forma, diferenciando-se basicamente, apenas pela complexidade e alcance de leitura.

Onde, o leitor *NFC*, tem a capacidade de ler mais informações, e informações mais complexas, que um leitor *RFID*, e por isso, menor alcance.

Com os cartões *RFID* como chave, o processo de abertura do laboratório seria agilizado e facilitado, enquanto a segurança permaneceria satisfatória. Além disso, fazendo-se a leitura dos códigos únicos de cada cartão, far-se-ia possível a identificação de cada cartão, conseqüentemente a pessoa que utiliza o mesmo, e por fim, a manipulação desses dados de diversas formas, inclusive a permissão diferenciada para cada usuário, e o envio desses dados para um monitoramento remoto de acesso, que é o objetivo do projeto em questão.

O microcontrolador possui uma cpu de 8 bits de arquitetura RISC avançada, 256 KB de Flash (8 KB utilizados para o bootloader), 8 KB de RAM, e 4 KB de EEPROM. Além disso, possui 16 entradas analógicas e 15 saídas PWM, comunicação SPI, I2C e 6 pinos de interrupção externa (SOUZA, 2014). É um microcontrolador com configurações mais que suficientes, que permitem um funcionamento rápido e satisfatório do sistema. Evitando atrasos na execução de suas rotinas e fornecendo confiabilidade ao sistema.

O mesmo, realizará toda a parte lógica do sistema de segurança, conferindo se os códigos dos cartões lidos pelo PN532 estão cadastrados em sua memória, e conseqüentemente ativando ou não a tranca elétrica; escrevendo mensagens pré-programadas em um *LCD* 16x2, e por fim, realizando *upload* dos dados lidos no site.

A tranca elétrica foi construída como um protótipo de teste de funcionamento. Foi construída com um servo motor, um ferrolho e uma braço de conexão entre o servo e o ferrolho. Porém, no caso de uma instalação real do sistema, deve-se comprar uma tranca elétrica confiável e ativá-la através de um relé.

Todo a parte de segurança do projeto, é independente da sua conexão com a internet, funcionando então, da seguinte forma:

Utilizando as bibliotecas *PN532\_I2C*, *PN532*, *NfcAdapter*, e *Wire*, o microcontrolador se conecta com o leitor *NFC*, e assim que algum cartão é aproximado do leitor, o mesmo faz a leitura do código hexadecimal de identificação do respectivo cartão, e envia-o para o microcontrolador, por meio da função '*nfc.read*', que o transforma em uma *string* através da função '*tag.getUidString*', e o salva em uma variável, onde, logo após, executa uma rotina de verificação desse código, dentre todos os códigos cadastrados. Caso o código não esteja cadastrado, a tranca não é acionada, e aparece a mensagem “Não Permitido” no LCD. Caso o código esteja cadastrado, é verificado se corresponde a um cartão de aluno, professor ou técnico; se for professor ou técnico, o microcontrolador irá acionar a tranca sem mais verificações, porém, se for um cartão de aluno, verificará se algum professor ou técnico abriu a porta anteriormente, e liberou o acesso de alunos. Se sim, o aluno tem livre acesso para destrancar e trancar o laboratório. Entretanto, se nenhum professor ou técnico tenha liberado o acesso ao laboratório anteriormente, ou tenham simplesmente trancado o laboratório, ao aproximar o cartão de aluno no leitor, o microcontrolador não irá acionar a tranca, e aparecerá a mensagem “Não permitido” no LCD, assim como anteriormente. Funcionando então, uma hierarquia no acesso aos laboratórios, onde professores e técnicos tem livre acesso, enquanto alunos dependem da liberação de professores ou técnicos.

Ademais, na parte interna do laboratório, um botão ligado a interrupção externa do pino dois do ATmega2560, aciona a tranca de forma prioritária e independente do andamento do resto do sistema. Além do mais, a situação do laboratório, “aberto”, ou “fechado”, é sempre mostrada no *LCD*, e todas as vezes em que a tranca é acionada utilizando o cartão *RFID*, o código de identificação do cartão que foi usado para acioná-la é enviado para um site, junto com a ação executada pela tranca, ou seja, se foi acionada para trancar, ou destrancar.

### **3.2. Bloco de transmissão:**

Após o desenvolvimento do sistema de segurança, para que se pudesse visualizar remotamente e em tempo real, quem acessou ou laboratório, se pensou no

conceito de internet das coisas, e, portanto, conectar o sistema à internet, e visualizar em um site os dados de quem acessou, se entrou, ou saiu do laboratório.

Nesse sentido, e no contexto de laboratório, poder-se-ia optar por dois tipos de padrão de conexão existentes no laboratório, que são: Wi-Fi (IEEE 802.11n), ou Ethernet (IEEE 802.3), onde os quais dependem diretamente da velocidade de banda larga disponível. Enfim, decidiu-se por usar o padrão Ethernet, tendo em vista que o sistema não foi desenvolvido para ser móvel, e o padrão apresenta uma melhor estabilidade, velocidade e menor atenuação e interferência, devido a sua estrutura física de funcionamento, via cabo UTP, e não conexão sem fio como o padrão Wi-Fi, o qual é muito sujeito a interferências externas.

Com o intuito de se utilizar o padrão Ethernet, optou-se pelo emprego do Ethernet Shield W5100 (figura 4), um *shield* desenvolvido para trabalhar com a placa Arduino utilizada, de forma que é necessário apenas o encaixe do *shield* na placa, e a ligação dos pinos 11, 12 e 13 do *shield*, nos pinos 51, 50 e 52 da placa Arduino, respectivamente; e consequentemente, totalmente compatível com o microcontrolador ATmega2560. O *shield* baseia-se no chip WIZnet Ethernet W5100 que fornece conexão à internet ao microcontrolador, através dos protocolos TCP ou UDP, através da biblioteca Ethernet Library, do Arduino (WIZNET, 2008).

Figura 4. Ethernet Shield W5100.



Fonte: [www.alselectro.com](http://www.alselectro.com).

Disponível em: <<http://www.alselectro.com/ethernet-shield.html>>. Acesso em dezembro de 2017.

Fazendo parte tanto da parte de monitoramento e segurança, quanto da parte de transmissão, o microcontrolador ATmega2560, se comunica com o chip WIZnet 5100, através da biblioteca Ethernet Library, e com a função '*Ethernet.begin*', define um IP, e endereço *MAC* fixos, para o chip 5100, e utiliza o mesmo para se conectar ao servidor do site **acessolabsengelet.000webhostapp.com**, através do protocolo *TCP* pela porta oficial 80, *HTTP*, reservada para servidores web (MORIMOTO, 2008), usando a função '*client.connect*'.

Após estabelecida a conexão, o *string* com as informações de *ID* do cartão *RFID*, e de entrada ou saída do usuário, é enviada através do método *POST*, usando-se a função '*client.print*' da biblioteca; com o tipo de conteúdo (*Content-Type*) padrão '*application/x-www-form-urlencoded*', onde os valores são codificados em *strings* que contém tanto o nome da variável *POST* a ser enviada, quanto o seu valor em si. A variável e seu valor são separadas com o caractere '=', enquanto variáveis são separadas entre si pelo caractere '&'. Portanto, a *string* a ser enviada com o código tem a seguinte estrutura:

*String* valorenviado = "variavelPOST=" + código;

Feito isso, a conexão com o servidor é finalizada, até que a tranca seja mais uma vez acionada utilizando-se um cartão.

### 3.3. Bloco de armazenamento e visualização remoto:

Com o intuito de permitir a análise e leitura das temperaturas lidas pelos sensores através da internet, um site foi criado e hospedado de forma gratuita pelo serviço **000webhost.com**, onde, após a criação de uma conta no site, é disponibilizado um servidor com 1,5 GB de espaço, suporte à *PHP*, *MariaDB*, *FTP*, entre outros. A criação do site se dividiu em três etapas, a criação e configuração do banco de dados *MariaDB*, o qual tem total suporte à linguagem *SQL*; o desenvolvimento do layout do site utilizando *HTML* e a ferramenta Zyro, disponibilizada pelo **000webhost.com**; e a criação dos arquivos e códigos em *PHP* e *SQL*.



O banco de dados foi criado de forma online, através do próprio 000webhost, no qual, há uma ferramenta específica para isso.

Tendo se criado o banco de dados, com a ferramenta *phpMyAdmin*, foram criadas três tabelas de dados: Acesso, Cadastro e CadastroAdmin. Na primeira tabela, foram criadas seis colunas: 'data', 'hora', 'nome', 'matricula', 'funcao' (aluno, técnico ou professor) e 'estado' (entrada ou saída), onde na qual, serão armazenados os dados de quem acessou o laboratório, e em qual data e horário. Na segunda tabela, quatro colunas: 'codigo', 'nome', 'matricula' e 'funcao'. Na mesma, serão armazenados os dados dos usuários do laboratório, e cada aluno, professor ou técnico é referenciado ao código de seu respectivo cartão. Na terceira e última tabela, criou-se três colunas: usuário, senha e e-mail, onde é armazenado os dados dos administradores do site, os quais terão acesso ao site, preenchendo o usuário e senha cadastrados, na página de login.

Ainda para a criação do site, como já informado, todo o layout foi desenvolvido com *HTML* e a ferramenta Zyro, desde as imagens, botões, e inclusive as seis páginas principais do site. São elas:

#### 3.3.1. Login:

Página na qual o administrador faz o login para ter acesso ao site. Enquanto não houver nenhum administrador autenticado no site, todas as outras páginas redirecionam para a página Login. Além disso, caso o administrador esqueça sua senha, na página de Login, ele pode selecionar a opção 'Esqueceu sua senha? ', e colocar o e-mail cadastrado à sua conta. Feito isso, uma senha aleatória de seis dígitos, com letras maiúsculas, minúsculas e números, é gerada, substitui a senha atual do usuário, e é enviada ao e-mail cadastrado.

#### 3.3.2. Página Inicial:

Logo que se é efetuado o login, o site é redirecionado para a página inicial, a qual, contém a descrição do site, e a descrição e link, de todas as outras páginas do mesmo.

### 3.3.3. Dados de Acesso:

Página na qual os dados da tabela ‘Acesso’ são mostrados em uma tabela *HTML*. Além disso, na página, contém dois botões com opções de manipulação dos dados: ‘imprimir’, onde o botão redireciona o site para uma página contendo apenas a tabela em *HTML*, permitindo uma impressão limpa dos dados. O outro botão, é o botão ‘download’, que redireciona o site a um arquivo *PHP*, no qual seleciona os dados da tabela ‘Acesso’, exporta-os para um arquivo .csv, e faz o download dos mesmos. Após isso, o site redirecionado à página Dados de Acesso outra vez.

### 3.3.4. Cadastro Usuários:

Nesta página, é possível cadastrar novos usuários do laboratório, referenciando os dados do usuário, ao cartão já cadastrado no microcontrolador, usado pelo mesmo. Ademais, nessa página é possível substituir os dados referenciados a um cartão, por outros dados, caso haja a necessidade do cartão ser passado para outro usuário. E por fim, é possível deletar usuários cadastrados. Todos os dados são adicionados, substituídos e deletados da tabela ‘Cadastro’.

### 3.3.5. Cadastro Administradores:

Assim como na página ‘Cadastro Usuários’, nesta página é possível adicionar e deletar novos administradores do site e laboratório, e os dados são adicionados à tabela ‘CadastroAdmin’. Para adicioná-los, é necessário preencher um nome de usuário não cadastrado ainda, uma senha, e um e-mail também não cadastrado. Para deletá-los, é necessário especificar o usuário, e a senha do mesmo. É possível também, trocar a senha e trocar o e-mail do usuário, caso tenha-se a senha do usuário.

### 3.3.6. Usuários Cadastrados:

Os dados da tabela ‘Cadastro’, são mostrados em uma tabela *HTML* nesta página. Ou seja, sempre que um novo usuário do laboratório é adicionado, substituído, ou deletado, isso é mostrado nesta página.

Finalizando a criação do site, foram criados os arquivos em *PHP*, e os códigos em *PHP* e *SQL* incluso nas páginas já citadas. Dos arquivos *PHP*, temos:

#### 3.3.7. connect.php:

Para facilitar a conexão com o banco de dados, a função '*Connection*' foi criada neste arquivo. A mesma, utiliza a função '*mysqli\_connect*', e retorna a conexão com o banco de dados utilizado, quando usada.

#### 3.3.8. add.php:

Esse arquivo foi criado para receber o código enviado pelo microcontrolador, selecionar os dados do usuário referente ao código recebido, e adicionar esses dados na tabela 'Acesso'. O código do mesmo, inicia-se incluindo o arquivo 'connect.php', logo após, o código recebido é salvo em uma variável, e os dados referentes aos códigos são selecionados utilizando as funções '*mysqli\_query*' e '*Connection*', junto com a instrução *SQL* '*SELECT*'. Então, confere-se se houve algum resultado, com a função '*mysqli\_num\_rows*'; caso sim, os dados são inseridos na tabela 'Acesso', junto com a data e a hora do acesso ao laboratório, com as funções '*mysqli\_query*', '*Connection*' e as instruções *SQL* '*INSERT INTO*', '*CURRENT\_DATE*', e '*CURRENT\_TIME*'. Caso não, é adicionado 'Usuário Desconhecido' à tabela, com a data e a hora do acesso, assim como no caso anterior.

O 'Usuário Desconhecido' é inserido para indicar que algum cartão cadastrado no microcontrolador, não está cadastrado no site, e, portanto, não deveria estar sendo utilizado.

#### 3.3.9. adicionar.php:

Inclui-se o arquivo '*connect.php*', e confere-se com as funções '*Connection*', '*mysqli\_query*' e '*mysqli\_num\_rows*', e a instrução *SQL* '*SELECT*', se o código submetido no formulário de adição na página 'Cadastro Usuários', já está referenciado a algum usuário, e caso não, adiciona os dados na tabela 'Cadastro', utilizando outra vez, as funções '*Connection*', '*mysqli\_query*' e a instrução *SQL* '*INSERT*

*INTO*'. Logo após, a conexão é fechada, e a página é redirecionada para a página 'Cadastro Usuários'.

#### 3.3.10. substituir.php:

Funciona da mesma forma do arquivo anterior, porém, ao contrário do código anterior, caso o código já esteja cadastrado, então, os dados submetidos no formulário de substituição da página 'Cadastro Usuários' substituem os dados anteriores referenciados ao código submetido. A substituição é feita com as mesmas funções do arquivo anterior, porém, a instrução *SQL* usada é a instrução '*UPDATE*'.

#### 3.3.11. deletar.php:

Com a matrícula submetida no formulário de exclusão da página 'Cadastro Usuários', confere-se se há algum usuário do laboratório matriculado, e caso sim, assim como nos outros arquivos, a conexão é estabelecida, e os dados referentes à matrícula submetida, são deletados com a instrução *SQL* '*DELETE*'. Então, a conexão é fechada e a página é redirecionada para a página 'Cadastro Usuários'.

#### 3.3.12. exportcsv.php:

Arquivo para onde o site é direcionado quando o botão de download na página 'Dados de Acesso' é clicado. Nele, o arquivo '*connect.php*' é incluído, e com as funções '*Connection*', '*fopen*', '*fputcsv*', '*mysqli\_query*' e '*mysqli\_fetch\_assoc*', um arquivo .csv é criado, e os arquivos da tabela 'Acesso', são exportados, e adicionados nele. Após isso, a conexão é fechada, o arquivo é baixado no computador do administrador que está acessando o site, e o site é redirecionado para a página 'Dados de Acesso'.

#### 3.3.13. passgen.php:

Usando as funções '*str\_shuffle*' e '*substr*', cria-se uma função que gera e retorna, uma senha aleatória de seis dígitos, com letras maiúsculas, minúsculas e números.

#### 3.3.14. sair.php:

Por último, esse arquivo foi criado para que o administrador possa se desconectar da sua conta no site. Nela, verifica-se se a variável '*SESSION*' 'logado', existe, e se seu valor é '*TRUE*', caso sim, a função '*session\_destroy*' é usada, e a variável 'logado' é apagada. Logo, então, o site é redirecionado para a página de login.

Especificando os códigos de cada página, temos:

#### 3.3.15. Login:

Inicialmente, nesta página, verifica-se se há uma variável *SESSION* de nome 'logado', criada, e se o valor dela é *TRUE*, caso sim, o site é redirecionado para a Página Inicial, através da função *header*. Caso não, a página Login é acessada. Nela, o arquivo 'connect.php' é incluído, e logo após, há um formulário *PHP*, com os campos 'Usuário' e 'Senha'. Caso haja a submissão do formulário, é verificado se há algum campo em branco, caso não, utilizando a função '*Connection*' contida no arquivo incluído, a função *mysqli\_query*, e a instrução *SQL*, '*SELECT*', a senha referente ao usuário submetido é selecionada. Então, com a função *mysqli\_num\_rows*, confere-se se há alguma senha foi encontrada, caso não, mostra-se uma mensagem de que o usuário não está cadastrado. Caso sim, a senha submetida é no formulário é criptografada com a função '*md5*' e comparada com a senha, já criptografada, selecionada no banco de dados, se elas forem diferentes, uma mensagem indicando isso é mostrada, se forem iguais, uma variável *SESSION* de nome 'logado', é criada, a ela é dado o valor '*TRUE*', a conexão é fechada pela função *mysqli\_close*, e então, a página é redirecionada para a página inicial do site.

#### 3.3.16. Página Inicial:

Verifica-se se a variável *SESSION* foi criada, e se seu valor é *TRUE*, caso não, o site é redirecionado para a página de login.

#### 3.3.17. Dados de Acesso:

É verificado se a variável *SESSION* foi criada, e se seu valor é *TRUE*, caso não, o site é redirecionado para a página de login. Caso sim, o arquivo ‘connect.php’ é incluído, e usando a função ‘Connection’, a função *mysqli\_query*, e a instrução *SQL* ‘*SELECT*’, todos os dados da tabela ‘Acesso’ são selecionados, e ordenados de forma decrescente pela data e hora em que foram acessados. Após serem selecionados, os dados são salvos como um vetor, com a função *mysqli\_fetch\_array*, e então cada parte deste vetor, é igual ao valor de uma das colunas da tabela. Esses valores são então, adicionados à tabela *HTML* da página através da função *printf*. A conexão com o banco de dados é então, fechada.

#### 3.3.18. Cadastro Usuários:

Inicia-se verificando se a variável *SESSION* foi criada, e se seu valor é *TRUE*, caso não, o site é redirecionado para a página de login. Nesta página, há três formulários *PHP*: o de adição, o de substituição, e o de exclusão. O primeiro é o formulário de adição de usuários à tabela ‘Cadastro’, com os campos ‘Código’, ‘Nome’, ‘Matrícula’, e ‘Função’, e que quando é submetido, redireciona a página ao arquivo ‘adicionar.php’. O segundo formulário, substitui os dados referenciados à um cartão, e tem os campos iguais ao primeiro, porém ao ser submetido, redireciona o site para o arquivo ‘substituir.php’. O terceiro e último formulário, deleta usuários da tabela ‘Cadastro’, e seu único campo é o campo ‘Matrícula’.

#### 3.3.19. Cadastro Administradores:

Nesta página, há quatro subpáginas. Nas quatro, inicialmente se verifica se a variável *SESSION* foi criada, e se seu valor é *TRUE*, caso não, o site é redirecionado para a página de login. Nas quatro também, o arquivo ‘connect.php’ é incluído. As quatro são:

##### 3.3.19.1. Adicionar Administrador:

Nesta subpágina, há um formulário com os campos ‘Usuário’, ‘Senha’, ‘Repita a Senha’, ‘E-mail’ e ‘Repita o E-mail’. Quando o formulário

é submetido, primeiramente, confere-se algum campo está em branco, e mostra uma mensagem indicando isso, caso aconteça; se não, com as funções '*Connection*' e *mysqli\_query*, e a instrução '*SELECT*', procura-se o usuário que foi preenchido no formulário, na tabela 'CadastroAdmin', e com a função '*mysqli\_num\_rows*' verifica-se se houve alguma resposta. Se sim, mostra uma mensagem indicando que o usuário já está cadastrado, se não, compara-se as duas senhas e os dois e-mails que foram escritos; se as duas senhas e os dois e-mails forem iguais, os dados são inseridos na tabela 'CadastroAdmin', com as funções '*Connection*' e *mysqli\_query*, e a instrução '*INSERT INTO*'. Feito isso, a conexão é fechada.

#### 3.3.19.2. Mudar Senha:

Nesta, há outro formulário com os campos 'Usuário', 'Senha Atual', 'Nova Senha' e 'Repita a Senha'. Ao formulário ser submetido, novamente confere-se se há algum campo em branco e mostra uma mensagem indicando isso. Caso não aconteça, seleciona-se a senha referente ao usuário escrito no formulário, da mesma forma que na subpágina anterior, e verifica-se também se há algum resultado da procura. Porém diferentemente da subpágina anterior, caso não haja usuário cadastrado, mostra-se uma mensagem indicando, e caso haja usuário, criptografa-se a senha escrita no formulário, e compara a mesma, com a senha selecionada anteriormente, já criptografada. Se as senhas forem iguais, compara-se as duas senhas escritas no formulário, se forem iguais, a nova senha é criptografada e substitui a senha antiga na tabela 'CadastroAdmin', usando as funções '*Connection*' e *mysqli\_query*, e a instrução *SQL* '*UPDATE*'. Logo após, a conexão é fechada.

#### 3.3.19.3. Mudar E-mail:

Excluindo a parte de criptografia, o código *PHP* desta subpágina funciona exatamente como o código da página anterior, porém, não se tratando mais das senhas, e sim dos e-mails cadastrados.

#### 3.3.19.4. Deletar Administrador:

Nesta última subpágina, o formulário contém apenas dois campos: ‘Usuário’, e ‘Senha’. Assim que submetido o formulário, confere-se se há algum campo em branco e mostra-se uma mensagem indicando isso, caso verdadeiro. Se não, a senha relacionada ao usuário é selecionada, assim como em todos os outros códigos, e compara-se se é igual à senha escrita no formulário, já criptografada. Caso as senhas sejam iguais, os dados do usuário selecionado são deletados usando as funções ‘*Connection*’, *mysqli\_query*’, e a instrução SQL ‘*DELETE*’. Por fim, a conexão é fechada.

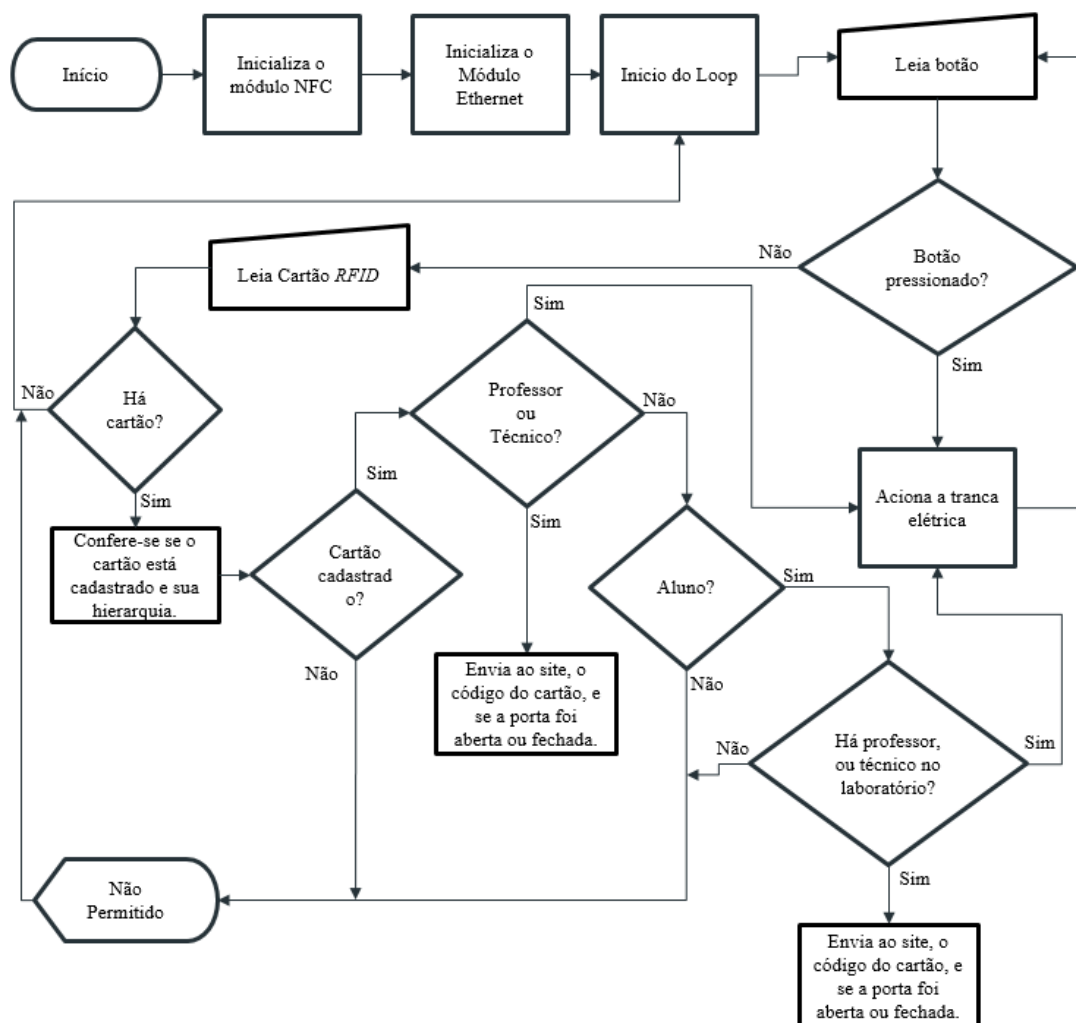
Concluindo o desenvolvimento e montagem do projeto, todo o circuito é alimentado por uma fonte de 9 V, e 1 A de corrente, com uma regulação de tensão para 5 V na placa Arduino, para o ATmega2560, e o leitor *NFC*.

### **3.4. Fluxogramas de funcionamento:**

3.4.1. Fluxograma do monitoramento e segurança, ao envio de dados (figura5):



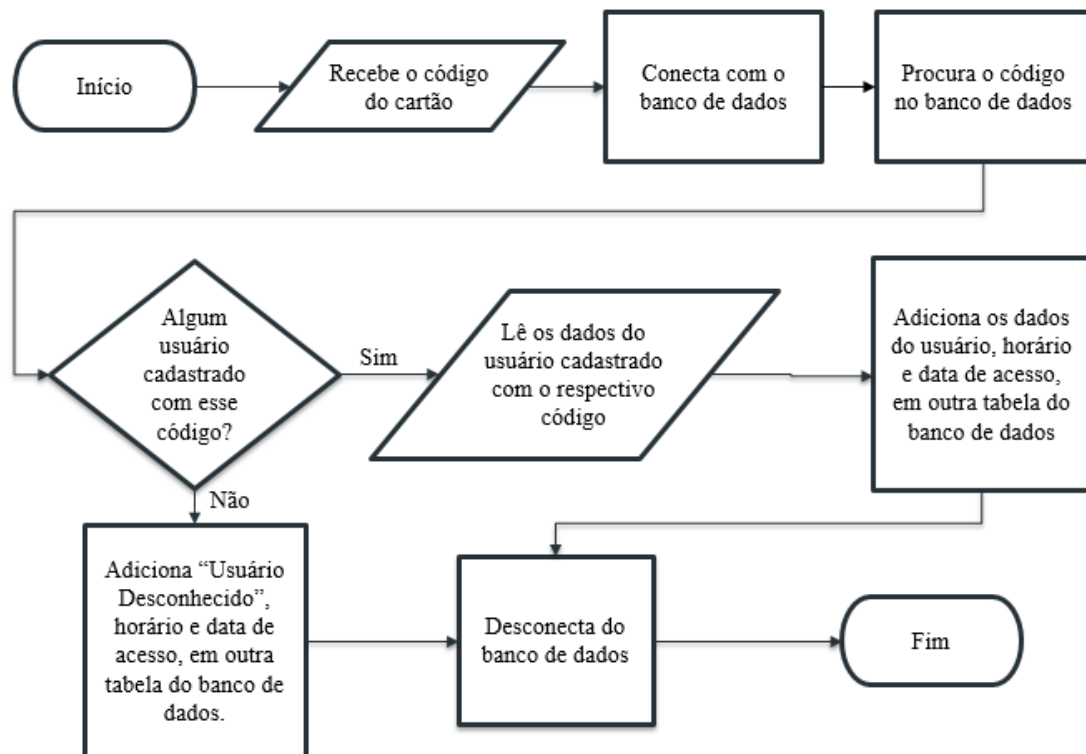
Figura 5. Fluxograma de monitoramento e segurança, ao envio de dados.



Fonte: Elaborado pelo autor.

### 3.4.2. Fluxograma do recebimento de dados e adição no banco de dados:

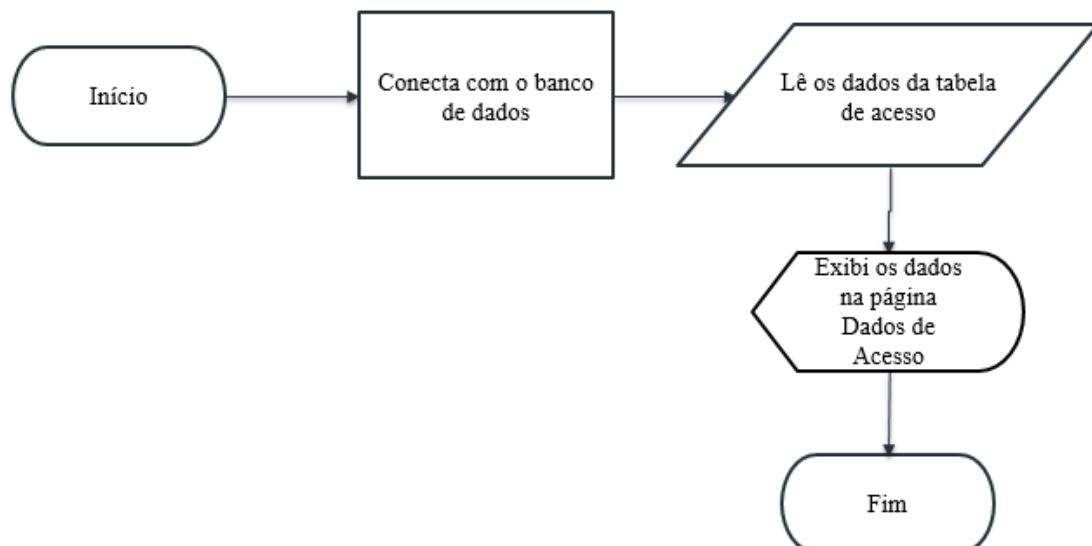
Figura 6. Fluxograma do recebimento de dados e adição no banco de dados.



Fonte: Elaborada pelo autor.

### 3.4.3. Fluxograma da exibição dos dados de acesso, na página 'Dados de Acesso':

Figura 7. Fluxograma da exibição dos dados de acesso:



Fonte: Elaborada pelo autor.

#### 4. RESULTADOS E DISCUSSÕES

Nos testes realizados dentro do laboratório, o sistema (figura 5) funcionou corretamente dentro do esperado, executando de forma correta e eficiente o processo de leitura dos cartões *NFC* e na sequência o acionamento preciso da tranca. A rotina implementada como firmware executou os procedimentos de maneira adequada, incluindo o requisito de hierarquia professor/técnico e aluno no processo de abertura da tranca. A transmissão de dados via ethernet foi executada de forma eficiente e rápida, levando em média três segundos, em uma rede doméstica com velocidade de 10 Mbps. O microcontrolador ATmega2560 apresentou um bom desempenho geral, atingindo a marca de 23% de utilização de sua memória dinâmica e apenas 10% de seu espaço de armazenamento para programas com 36 cartões *NFC* cadastrados, ou seja, o microcontrolador ainda possuía um poder de processamento considerável frente a esta aplicação.

O site (figura 6) demonstrou robustez no tocante a quantidade de informações colhidas do microcontrolador e possibilidades de edições por parte do administrador. A página foi desenvolvida utilizando-se critérios consideráveis de proteção, sendo acessível apenas aos seus gestores.

Dentre as principais funções disponíveis no sítio eletrônico, pode-se citar:

- Cadastro de administradores e usuários;
- Gerenciamento de administradores e usuários;
- Modo de recuperação de senha via e-mail;
- Demonstração de data, horário e dados de acesso dos administradores e usuários;
- Exportação dos dados do banco de dados de acesso em formato .csv e página de impressão em *HTML*.

Figura 8. Maquete construída para testes.



Fonte: Elaborada pelo autor.

Figura 9. Página inicial do site.



Fonte: Elaborada pelo autor.

## 5. CONCLUSÃO

O projeto desenvolvido proporcionou a criação de uma rotina lógica consistente, podendo ser adaptada para posterior aplicação com a utilização de uma tranca eletrônica real na prática. Também foi elaborado um site capaz de receber os dados de acesso via

ethernet, criando assim um sistema de monitoramento e registro, sendo possível sua exportação em formato .csv ou impressão em uma página *HTML*. Estes podem inclusive servir como indicadores de uso laboratorial em visitas do Ministério da Educação - MEC, contribuindo assim com a avaliação do curso de Engenharia Elétrica da Universidade Federal de Rondônia.

## REFERÊNCIAS BIBLIOGRÁFICAS

CUNHA, Alessandro (2016). “*NFC (Near Field Communication) – Aplicações e uso*”. Disponível em: <<https://www.embarcados.com.br/nfc-near-field-communication/>>. Acesso em: dez. 2017.

ALECRIM, Emerson (2017). “*O que é NFC (Near Field Communication)?* ”. Disponível em: <<https://www.infowester.com/nfc.php>>. Acesso em: dez. 2017.

ARDUINO (2017). “*Arduino Mega*”. Disponível em: <<https://www.arduino.cc/en/Main/arduinoBoardMega>>. Acesso em: dez. 2017.

MOTA, Allan (2017). “*Usando a Interrupção Externa no Seu Arduino*”. Disponível em: <<https://portal.vidadesilicio.com.br/usando-a-interruptcao-externa-no-seu-arduino/>>. Acesso em: dez. 2017.

CARLOS, Robert (2014). “*RFID X NFC – Tecnologias Emergentes*”. Disponível em: <<https://robertcg.wordpress.com/2014/03/23/rfid-x-nfc-tecnologias-emergentes/>>. Acesso em: dez. 2017.

PINHEIRO, José Maurício S. “*Padrão Ethernet*”. Disponível em: <[https://www.projetoderedes.com.br/aulas/ugb\\_redes\\_I/ugb\\_redes\\_I\\_material\\_de\\_apoio\\_02.pdf](https://www.projetoderedes.com.br/aulas/ugb_redes_I/ugb_redes_I_material_de_apoio_02.pdf)>. Acesso em: dez. 2017.

PANTAS, Willians (2017). “*Wi-Fi vs Ethernet: A conexão por cabo é melhor?*”. Disponível em: <<http://understech.com.br/wi-fi-vs-ethernet-conexao-por-cabo-melhor/>>. Acesso em: dez. 2017.

WIZNET (2008). “*W5100 Datasheet – Version 1.1.6*”. Disponível em:  
<[https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100\\_Datasheet\\_v1\\_1\\_6.pdf](https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf)>. Acesso em: dez. 2017.

ARDUINO (2017). “*Ethernet/Ethernet 2 library*”. Disponível em:  
<<https://www.arduino.cc/en/Reference/Ethernet>>. Acesso em: dez. 2017.

MORIMOTO, C. E. Redes, Guia Prático 2ª Ed. 2008. Disponível em:  
<<http://www.hardware.com.br/livros/redes/>>. Acesso em: dez. 2017.

MDN WEB DOCS (2017). “*POST Method*”. Disponível em:  
<<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/POST>>. Acesso em: dez. 2017.

ARDUINO E CIA (2016). “*Comunicação sem fio com módulo PN532 NFC RFID*”. Disponível em: <<https://www.arduinoecia.com.br/2016/10/modulo-pn532-nfc-rfid-arduino.html>>. Acesso em: dez. 2017.

INSTRUCTABLES. “*Send Arduino Data to the Web (PHP/MySQL/D3.js)* ”. Disponível em: <<http://www.instructables.com/id/PART-1-Send-Arduino-data-to-the-Web-PHP-MySQL-D3js/>>. Acesso em: dez. 2017.

ARDUINO (2017). “*Web Client*”. Disponível em:  
<<https://www.arduino.cc/en/Tutorial/WebClient>>. Acesso em: dez. 2017.