

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE
COORDENAÇÃO INTERDISCIPLINAR DE TECNOLOGIAS DA INFORMAÇÃO E
COMUNICAÇÃO – CIT
CURSO TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

Diogo Weber Silveira

Desenvolvimento de uma fechadura eletrônica: um sistema de controle de acesso com registro em banco de dados e site de gerenciamento.

Araranguá

2021

Diogo Weber Silveira

Desenvolvimento de uma fechadura eletrônica: um sistema de controle de acesso com registro em banco de dados e site de gerenciamento.

Trabalho Conclusão do Curso de Graduação em Tecnologias da Informação e Comunicação do Centro de Ciências, Tecnologias e Saúde da Universidade Federal de Santa Catarina como requisito para a obtenção do Título de Bacharel em Tecnologias da Informação e Comunicação.

Orientador: Prof. Dr. Giovani Mendonça Lunardi

Araranguá

2021

Ficha de identificação da obra

Silveira, Diogo Weber

Desenvolvimento de uma fechadura eletrônica : um sistema de controle de acesso com registro em banco de dados e site de gerenciamento / Diogo Weber Silveira ; orientador, Giovani Mendonça Lunardi, 2021.

51 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Tecnologias da Informação e Comunicação,
Araranguá, 2021.

Inclui referências.

1. Tecnologias da Informação e Comunicação. 2. Fechadura eletrônica. 3. Controle de acesso. 4. Registro de acesso. 5. Desenvolvimento. I. Lunardi, Giovani Mendonça. II. Universidade Federal de Santa Catarina. Graduação em Tecnologias da Informação e Comunicação.

Diogo Weber Silveira

Desenvolvimento de uma fechadura eletrônica: um sistema de controle de acesso com registro em banco de dados e site de gerenciamento.

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Tecnologias da Informação e Comunicação” e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 17 de maio de 2021.

Prof. Vilson Gruber, Dr.

Coordenador do Curso

Banca Examinadora:

Prof. Giovani Mendonça Lunardi, Dr.

Orientador

Universidade Federal de Santa Catarina

Prof. Paulo Cesar Leite Esteves, Dr.

Universidade Federal de Santa Catarina

Doutorando Alix Ribeiro da Silva

Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus pais, familiares e amigos por seus apoios.

AGRADECIMENTOS

Aos meus pais, Valmir Silveira e Marlene Weber Silveira, por seus esforços para que eu chegasse até aqui. A minha irmã por sua ajuda e apoio durante a elaboração deste trabalho. A professora Eliane Pozzebon por realizar a conexão com os responsáveis do laboratório. O professor Giovani Mendonça Lunardi, por sua paciência e apoio.

RESUMO

A busca por um sistema que permitisse o controle de acesso a um laboratório com vários alunos, professores e servidores, que permitisse o trancamento automático de acesso ao mesmo, sem a necessidade de utilização de uma chave mecânica e que se efetua o registro de entradas e saídas automaticamente, levou ao desenvolvimento de uma fechadura eletrônica que atende as necessidades do laboratório. Este feito só foi possível realizando a implementação de uma fechadura eletrônica utilizando a plataforma Arduino, módulos RFID e relé, fechadura elétrica, scripts em Python3, banco de dados e ferramentas para desenvolvimento WEB. Concluiu-se que é possível o desenvolvimento de uma fechadura eletrônica atendendo os requisitos e foi implementado conforme demonstrado no decorrer deste trabalho.

Palavras-chave: Fechadura eletrônica. Controle de acesso. Registro de acesso. Desenvolvimento.

ABSTRACT

The search for a system that would allow access control to a laboratory with several students, teachers and servers, that would allow the automatic locking of access to it, without the need to use a mechanical key and that the registration of entries and outputs automatically, led to the development of an electronic lock that meets the needs of the laboratory. This feat was only possible by implementing an electronic lock using the Arduino platform, RFID modules and relay, electric lock, Python3 scripts, database and tools for WEB development. It was concluded that it is possible to develop an electronic lock according to the requirements and it was implemented as demonstrated in the course of this work.

Keywords: Electronic lock. Access control. Access record. Development.

LISTA DE FIGURAS

Figura 1 - Arduino Uno R3	20
Figura 2 - RFID e suas utilizações	21
Figura 3 - Relé.....	22
Figura 4 - Composição interna de uma fechadura elétrica	23
Figura 5 - Funcionamento do SPI	26
Figura 6 - Projeto montado.....	29
Figura 7 - Conexão dos componentes	30
Figura 8 - Fluxo do código do Arduino.....	31
Figura 9 - Tabelas do banco de dados	33
Figura 10 - Fluxo script do computador.....	34
Figura 11 - Fluxo script de novo cadastro.....	37
Figura 12 - Página de login	38
Figura 13 - Página do menu	39
Figura 14 - Página lista de entradas.....	40
Figura 15 - Página lista de registros	41
Figura 16 - Página lista por matrícula	42
Figura 17 - Código PC parte 1.....	47
Figura 18 - Código PC parte 2.....	47
Figura 19 - Código PC parte 3.....	47
Figura 20 - Código PC parte 4.....	47
Figura 21 - Código cadastro novos usuários	47

LISTA DE ABREVIATURAS E SIGLAS

DB – Data Base

PC – Personal Computer

RFID – Identificação por radiofrequência (sigla em inglês)

UFSC – Universidade Federal de Santa Catarina

SUMÁRIO

1	introdução	15
1.1	Justificativa e problemática	15
1.2	OBJETIVOS.....	16
1.2.1	Objetivo Geral.....	16
1.2.2	Objetivos Específicos	16
1.3	Metodologia.....	17
2	As fechaduras eletrônicas.....	18
3	COMPONENTES	19
3.1	HARDWARE	19
3.1.1	Arduíno Uno R3	19
3.1.2	RFID.....	20
3.1.3	Relé	21
3.1.4	Fechadura Elétrica	22
3.2	Software.....	23
3.2.1	Python3.....	23
3.2.2	XAMPP	24
3.2.3	IDE Arduíno.....	25
3.2.4	IDE PyCharm	25
3.3	Protocolos	25
3.3.1	SPI	26
3.3.2	UART	27
4	Desenvolvimento da fechadura eletrônica	28
4.1	Montagem da fechadura eletrônica	28
4.2	Programação do Arduíno	31
4.3	Criação do Banco de dados.....	32
4.4	Script do Computador.....	33
4.5	Script para cadastro de novos usuários	36
4.6	Paginas web	37
5	CONCLUSÃO.....	43
	REFERÊNCIAS	44
	ANEXO A – Códigos desenvolvidos.....	46

1 INTRODUÇÃO

As salas de aula e os laboratórios da Universidade Federal de Santa Catarina, Campus Araranguá, unidade do bairro Mato Alto, possuem sistema mecânico de abertura das salas, sendo necessário um professor ou funcionário da instituição para sua abertura ou fechamento.

Foi observado que os laboratórios poderiam ficar abertos por um grande período de tempo sem nenhum tipo de controle de acesso aos ambientes, ficando os laboratórios, e seus equipamentos de elevado valor econômico e técnico expostos a possíveis avarias e furtos. Alguns desses equipamentos sendo únicos, desenvolvidos nos laboratórios para suas atividades.

A partir das problemáticas observadas nesses ambientes da universidade, ficou claro aos coordenadores do laboratório LARAL, a necessidade de desenvolver um sistema que fizesse o controle de acesso a estes espaços físicos, permitindo a entrada de somente pessoas autorizadas e também a necessidade de um histórico de acessos a estes locais.

Após, foi realizada uma pesquisa de fechaduras eletrônicas existentes no mercado, sendo constatado que não havia uma fechadura que atendesse as necessidades específicas do laboratório em questão.

Ademais a UFSC possui cartões de identificação dos alunos, professores e servidores, que já contam com uma etiqueta RFID integras, que neste caso poderiam servir de chave de acesso a estes espaços da universidade.

Este trabalho divide-se em três capítulos. No primeiro, será apresentado considerações sobre fechaduras eletrônicas. No segundo capítulo, espõem-se os componentes, *hardware*, *software* e protocolos. Por fim, no terceiro capítulo, serão apresentadas as etapas do desenvolvimento da fechadura eletrônica proposta.

1.1 JUSTIFICATIVA E PROBLEMÁTICA

Durante uma conversa com o professor responsável pelo laboratório LARAL da Universidade, foi relato pelo mesmo que havia a necessidade de uma fechadura eletrônica para controlar o acesso ao laboratório devido ao sistema utilizar fechadura mecânica, este sistema não possuía um controle de acesso da sala do laboratório e o sistema de registro de acesso utilizar de um caderno para o preenchimento do histórico.

Devido a sala utilizar de uma fechadura mecânica, o seu trancamento se dá pela utilização de chave mecânica, ficando inviável a distribuição de chave aos usuários do laboratório. Com isto somente os coordenadores e funcionários da segurança possuem chave.

Ademais, quando a porta de acesso é destrancada, não existe um controle de acesso que limite o acesso de somente pessoas autorizadas, permitindo assim qualquer pessoa que passe pelo local pudesse adentrar a sala.

Por fim, o sistema de registro utilizado é um caderno onde o aluno estagiário registra o dia, a hora de entrada e hora de saída, necessitando um sistema que efetuasse o registro automaticamente.

Com estes três aspectos, fica o questionamento deste trabalho se é possível o desenvolvimento de uma fechadura eletrônica para o referido laboratório.

1.2 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste trabalho de conclusão de curso.

1.2.1 Objetivo Geral

Este trabalho tem como objetivo o desenvolvimento de uma fechadura eletrônica para o controle de acesso a um laboratório da Universidade Federal de Santa Catarina, utilizando um sistema que possa fazer das carteirinhas de identificação da universidade como chave de acesso ao laboratório.

1.2.2 Objetivos Específicos

- Utilizar a carteirinha da universidade como chave de acesso;
- Limitar o acesso apenas para pessoas autorizadas;
- Possibilidade de abertura com chave mecânica em caso de falta de energia;
- Criar um cadastro para pessoas autorizadas e seus históricos de entrada e saída;
- Possuir um site para visualização do histórico de acesso;

- Possibilidade de melhorias futuras;
- Fácil substituição dos componentes.

1.3 METODOLOGIA

O desenvolvimento deste trabalho foi realizado conforme os seguintes tópicos:

- Levantamento dos componentes necessários para o desenvolvimento;
- Execução da construção da fechadura conectando os componentes;
- Programação do Arduino, script de operação do computador e script para cadastro de novos usuários;
- Testes de funcionamento para verificação se a fechadura está operacional ou a necessidade de algum ajuste.

2 AS FECHADURAS ELETRÔNICAS

As fechaduras eletrônicas são um tipo de sistema de controle de acesso que conta com uma lingueta, que a se fechar em direção ao batente fica trancada, diferente dos sistemas anteriores que necessita de uma chave para efetuar o trancamento. Para sua abertura conta com um sistema de abertura, que é acionado através de uma botoeira pelo lado interno ou um sistema de autenticação (podendo ser por senha, tag ou digital por exemplo), se tornando a grande vantagem do produto, ou seja, impossibilita a abertura por pessoas não autorizadas. (SOPRANO, 2020)

Além do fechamento automático, a fechadura possui outras vantagens comparada as fechaduras convencionais, tal como praticidade, conforto, segurança, e em alguns modelos mais novos a função não-perturbem, que mesmo o usuário fazendo a autenticação correta não possibilita a abertura da porta. (PORTAS DE MADEIRA)

Outros benefícios encontrados em alguns modelos de fechaduras eletrônicas, são alarme de segurança, ou seja, se o aparelho for violado ou forçado ele automaticamente dispara um alarme. Outras contam com sistema de alimentação por pilhas ou baterias, não havendo a necessidade de estar ligada a rede de energia elétrica, com isso podendo ser utilizadas mesmo na ausência de eletricidade e duram entre 4 mil e 5 mil aberturas com um conjunto de pilhas alcalinas. (SOPRANO, 2020)

Atualmente as fechaduras eletrônicas contam com, pelo menos, seis meios de abertura, sendo eles senha, onde o usuário digita uma combinação entre 3 e 20 dígitos; biometria; aplicativo; tag (ou chaveiro de proximidade), que devem ser aproximados de um leitor da fechadura. Com essa tecnologia, o usuário tem que estar com a tag o tempo todo, portando tendo a mesma utilidade de uma chave mecânica das fechaduras convencionais. As tags utilizam o sistema de rádio frequência para seu funcionamento; o sexto meio de abertura é por cartão, muito utilizado em hotéis. Utilizam-se de cartões magnéticos para seu funcionamento. (SOPRANO, 2020)

Por serem consideradas artigo de luxo e possuírem um elevado preço (com algumas passando facilmente dos dois mil reais), ainda são pouco utilizadas no âmbito residencial, sendo mais utilizadas no âmbito empresarial, onde a um grande fluxo de pessoas.

3 COMPONENTES

Este capítulo foi dividido em três seções. A primeira trata sobre *hardware*; a segunda aborda sobre software e; a terceira expõe os protocolos.

3.1 HARDWARE

O *hardware* é da parte física do sistema, responsável por seu funcionamento. Trata-se das peças, como o Arduino, RFID, relé e fechadura elétrica da fechadura eletrônica desenvolvida. (ADMINBLOG_USR, 2019).

A seguir será tratado dos componentes utilizados para a construção da fechadura eletrônica como: Arduino Uno R3, RFID (*Radio Frequency Identification*), relé e fechadura elétrica.

3.1.1 Arduino Uno R3

O Arduino Uno R3 (figura 1), é uma placa para prototipagem eletrônica que utiliza o microcontrolador Tmega328, possuindo 14 pinos digitais de entrada ou saída e 6 pinos analógicos. Utiliza de um cristal oscilador de 16MHz, uma porta USB para comunicação e alimentação, e um botão de reset. Sua voltagem de operação é de 5 volts e possui portas de alimentação para módulos externos de 5 volts e 3,3 volts, além do pino terra.(Arduino).

Seu microcontrolador é um dispositivo de 8bits, 2 kilobytes de RAM, conta com 32 KB de ROM, dos quais 512 bytes é empregado pelo *bootloader* e *clock* de 16MHz. Possui os sistemas de comunicação UARD, SPI e I2C para trocar informações com módulos a serem utilizados ou com o computador. (SOUZA, 2013)

Figura 1 - Arduino Uno R3



Fonte: ARDUINO.CC

3.1.2 RFID

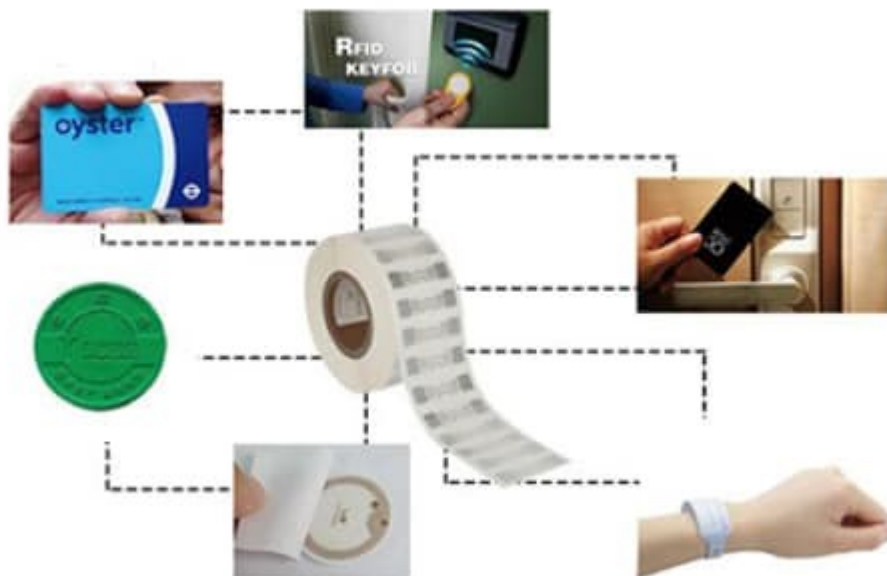
Radio Frequency Identification (RFID), em português Identificação por Rádio Frequência, é um recurso de identificação automático por rádio frequência, que é composto por uma etiqueta de rádio frequência e um transceptor (rádio transmissor e receptor). (PETENATE, 2019)

O funcionamento se dá com o transceptor (base) enviando sinais de radiofrequência que são captados por uma etiqueta RFID (tag), que ser estimulado por ondas sonoras seus circuitos transpõe as informações contidas em seu chip em ondas sonoras que retornam ao transceptor. (PETENATE, 2019)

Já as etiquetas de RFID podem ser utilizadas em diversos itens (como objetos metálicos e não metálicos, objetos bibliográficos, etc). Possuem diversas frequências de operação, com isto alterando à distância de leitura, como a 13,56MHz que tem o limite de 40 centímetros ou 125KHz que pode chegar a 70 centímetros dependendo da etiqueta. Em casos específicos pode-se chegar a nove metros. (CHASTAGNIER, 2020)

Também podem ser encontradas na forma de cartão, tag, adesivo e várias outras formas, dependendo exclusivamente para que será utilizada. Na figura 2 podemos visualizar alguns tipos de etiquetas RFID.

Figura 2 - RFID e suas utilizações



Fonte: I3C SOLUÇÕES INTELIGENTES

3.1.3 Relé

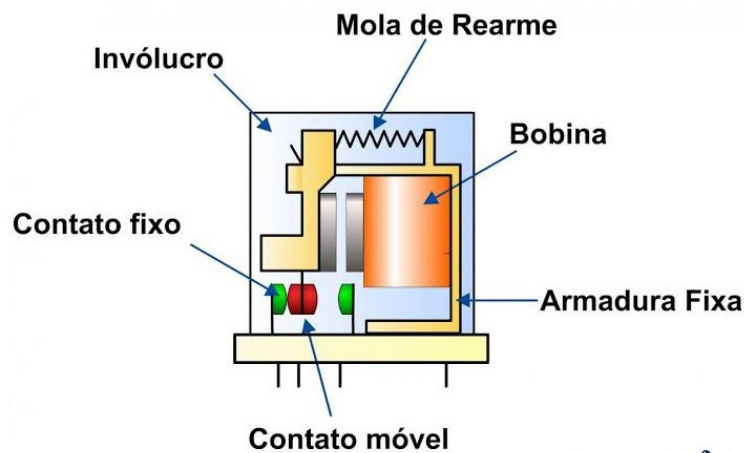
O relé é um dispositivo elétrico que tem por objetivo produzir alterações súbitas predeterminadas em, pelo menos, um circuito elétrico de saída. Possui um circuito de controle, que ao ser alimentado por uma corrente ativa, um eletroímã que faz a alteração de posição de contato do contator primário para um secundário, podendo ser resumido como um contato que abre e fecha, como pode ser visto na figura 3. (MATTEDE).

Seu funcionamento é consideravelmente simples, trabalhando quando uma corrente circula uma bobina, criando uma um campo magnético que atrai um ou mais contatos, abrindo ou fechando circuitos. Ao se cessar a corrente da bobina o campo magnético se cessa igualmente. (SANTOS)

Pode possuir contatos NF, NA ou ambos (possuindo um contato comum). O contato NF vem de normalmente fechado, ou seja, passando corrente enquanto a bobina está

desenergizada. O contato NA vem de normalmente aberto, não passa corrente enquanto a bobina está desenergizada. No caso de possuir ambos os contatos, se têm um contato central comum (C) a ambos, fechando contato entre os contatos C e NF, ao se energizar a bobina passa-se para os contatos C e NA. Ao desenergizar a bobina, volta-se ao estado normal com os

Figura 3 - Relé



contatos C e NF conectados. (SANTOS)

Fonte: MULT COMERCIAL

3.1.4 Fechadura Elétrica

Fechaduras elétricas contam com uma bobina elétrica para fazer a abertura sem a necessidade de utilizar a chave física. Seu funcionamento se dá quando a bobina ao receber uma corrente elétrica, gera um campo magnético que puxa uma trava e assim liberam a entrada, ou pode-se utilizar a chave mecânica para a abertura. Pode ser integrada a sistemas de controle de acesso ou porteiros eletrônicos. Podendo ser de sobrepor ou embutida na porta. (AGL).

Na figura 4 a seguir podemos visualizar a composição interna.

Figura 4 - Composição interna de uma fechadura elétrica



Fonte: MANUAL AGL AGLINHA

3.2 SOFTWARE

O software é da parte lógica do sistema, um conjunto de instruções que devem ser executadas por um *hardware*. (GOGONI, 2019).

Será tratado a seguir os softwares utilizados para elaboração deste trabalho.

3.2.1 Python3

Python é uma linguagem de programação de alto nível, interpretada, dinâmica, multiplataforma, modular e orientada a objetos (ROVEDA). Desenvolvida pensando no programador e tem como objetivo chave aumentar a produtividade, possibilitando programar de uma forma mais simples. (ROVEDA)

Possui uma grande quantidade de bibliotecas (conjunto de funções pré compiladas), o que facilita sua utilização em diversas áreas da programação como desenvolvimento web, machine learning, big data e inteligência artificial. (ROVEDA)

3.2.2 XAMPP

XAMPP é uma compilação dos servidores de código aberto Apache e MySQL, além de suporte as linguagens PHP e Perl. Desenvolvido para ser multiplataforma, operando facilmente no Windows ou Linux. (HIGA, 2012).

Apache é um servidor web responsável pela disponibilização de páginas e seus recursos que podem ser acessados pelo usuário, sendo compatível com os protocolos web HTTP e HTTPS. (REDAÇÃO)

Segundo Pisa (2012) o MySQL é um sistema gerenciador de banco de dados relacional, gratuito e de código aberto, utilizado em uma grande quantidade de programas para o gerenciamento de seus bancos de dados. Utiliza a linguagem SQL (Linguagem de consulta estruturada) para a inserção e remoção de dados, acesso e gerenciamento dos conteúdos armazenados em um banco de dados. (PISA, 2012)

PHP é uma linguagem de código aberto, do tipo script, de uso geral, muito utilizada no desenvolvimento web e geralmente utilizado em parceria com um código HTML. É uma linguagem de fácil aprendizado, mas que também possui recursos avançados para programadores mais avançados. Atua do lado de um servidor, gerando respostas que posteriormente é renderizado em um código HTML, este carregado em uma página do navegador. (MARQUES)

Perl é uma linguagem de código aberto, gratuita, simples, multiplataforma e de alto nível. Podendo ser utilizada em uma série de aplicações, desde sites e manipulação de arquivos de configuração de sistema, até modificar, acessar e popular tabelas de banco de dados MySQL. (TICURSOS.NET, 2009).

3.2.3 IDE Arduíno

Um Ambiente de Desenvolvimento Integrado (sigla em inglês IDE) é um programa que combina um editor de código fonte, um compilador e um debugger. As IDEs foram criadas para ajudar os programadores a desenvolver novos programas rapidamente, facilitar a organização do fluxo de trabalho e organização ao não precisar ficar trocando de aplicações. (RED HAT)

A IDE do Arduíno possui os recursos necessários para a programação de placas Arduíno e o carregamento do código as placas. Na IDE tem-se acesso o destaque de erros, inclusão de bibliotecas e correção de erros dos códigos. (TORRES, 2013)

3.2.4 IDE PyCharm

PyCharm é uma IDE para desenvolvimento Python, multiplataforma, que fornece análise gráfica, autocomplemento para código, depurador gráfico e um sistema de navegação que facilita a estrita de código (CAELUM).

A ferramenta possui suporte para: outras linguagens de programação (como SQL, JavaScript, HTML/CSS, entre outras), *frameworks* de desenvolvimento web, recursos para desenvolvimento remoto, ferramentas para desenvolvimento científico e ferramentas de banco de dados. (JETBRAINS)

3.3 PROTOCOLOS

Protocolos são conjuntos de normas que permite a comunicação de duas ou mais computadores se comuniquem, funcionando como uma linguagem universal, permitindo que seja interpretado por qualquer computador, indiferente de marca ou sistema operacional. (GOMES, 2019).

Para a realização da fechadura eletrônica foram utilizados os protocolos SPI (*Serial Peripheral Interface*) e UARD (*Universal Asynchronous Receiver-Transmitter*), no qual, serão tratados nos tópicos a seguir.

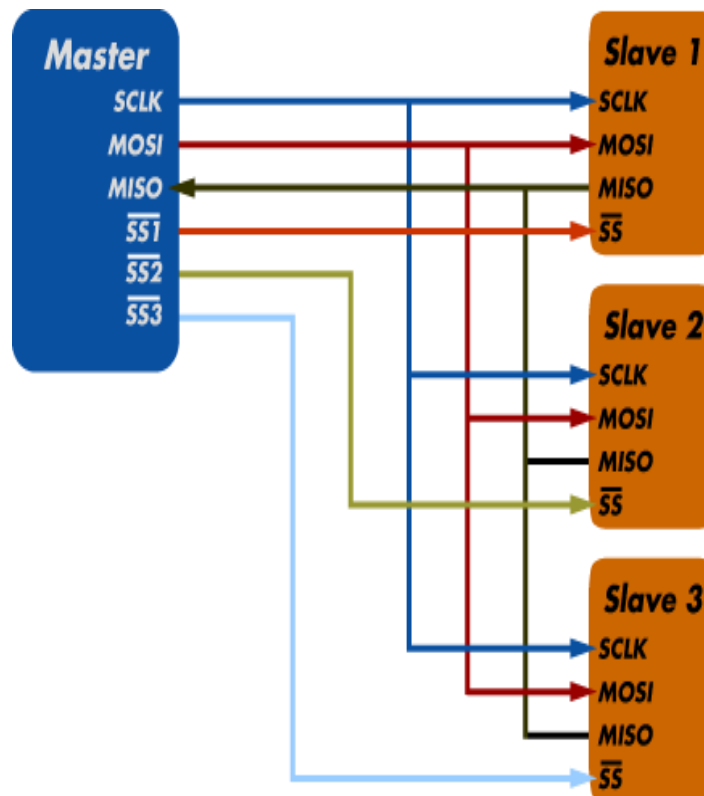
3.3.1 SPI

Serial Peripheral Interface é um protocolo de comunicação síncrona para pequenas distâncias, do tipo mestre escravo, onde só pode possuir um mestre. O SPI não possui um protocolo de interpretação de dados, ficando a cargo do desenvolvedor definir como isto ocorrerá. (GUIMARÃES, 2019)

A comunicação por SPI ocorre por 4 conexões, sendo 3 fixas e uma de seleção. A primeira conexão é denominada MISO (*Master IN Slave OUT*) onde o usuário *master* recebe dados do usuário escravo. A segunda MOSI (*Master OUT Slave In*) onde o *master* envia dados para o escravo. A terceira é chamada de SCK (*Serial Clock*) um temporizador de sincronização para a comunicação. Por último temos uma conexão para cada escravo chamada *Slave Select* (SS), onde o *master* seleciona com qual escravo receberá os dados. (LABORATÓRIO DE GARAGEM, 2012)

Na figura 5 a seguir, temos a exemplificação de como funciona o SPI.

Figura 5 - Funcionamento do SPI



Fonte: LABORATÓRIO DE GARAGEM

3.3.2 UART

Universal Asynchronous Receiver-Transmitter, é um protocolo assíncrono e *full-duplex* (onde transmissor e receptor podem enviar informações por canais distintos), responsável pela conversão da comunicação paralela em serial. No Arduino esse protocolo é convertido em USB. (ROBOCORE)

4 DESENVOLVIMENTO DA FECHADURA ELETRÔNICA

Para a criação da fechadura eletrônica e seus componentes este trabalho foi dividido em seis partes: montagem da fechadura, script do Arduino, desenvolvimento do banco de dados, script do computador, script de cadastro de novos usuários e desenvolvimento da parte WEB.

4.1 MONTAGEM DA FECHADURA ELETRÔNICA

Inicialmente para a realização deste trabalho foi necessário a seleção dos componentes essenciais no desenvolvimento da fechadura eletrônica, como o microcontrolador, leitores RFID para entrada e saída, relé, fechadura elétrica e fonte de alimentação. Foi adicionado também um LED RGB para a sinalização de qual operação estaria sendo feita pela fechadura.

O microcontrolador escolhido foi o Arduino Uno R3 pois atendia os requisitos para esse componente: barato, número de conexões e protocolos necessários. Por sua vez o leitor RFID utilizado foi Mifare RC522, que possui frequência de operação de 13.52MHz e utiliza o protocolo SPI para comunicação. Já a fechadura elétrica escolhida foi a AGLinha da fabricante AGL, que possui abertura por chave ou através da sua bobina 12V e 6W acionada eletronicamente.

Com os componentes definidos foi realizado a compra e a solda dos pinos de conexão dos módulos RFID.

Em seguida foi feito uma lista com todas as conexões que seriam necessárias a serem feitas entre os componentes ao Arduino, sendo que cada módulo RFID teria seu pino de seleção (*Slave Select*), gerando a lista a seguir:

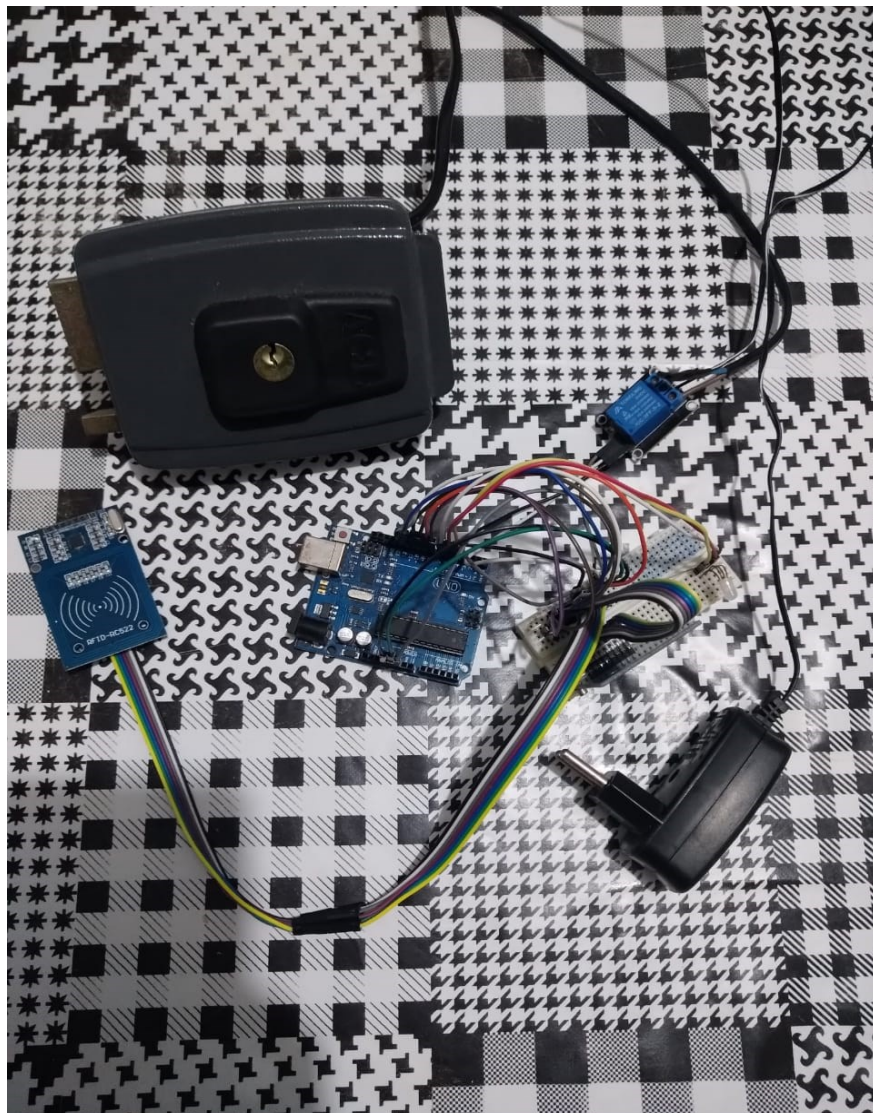
- Pino **RELE** ligado na porta 4 do Arduino
- Pino **RED** ligado na porta 5 do Arduino
- Pino **GREEN** ligado na porta 6 do Arduino
- Pino **BLUE** ligado na porta 7 do Arduino
- Pino **SDA2** ligado na porta 8 do Arduino
- Pino **RST** ligado na porta 9 do Arduino
- Pino **SDA1** ligado na porta 10 do Arduino
- Pino **MOSI** ligado na porta 11 do Arduino
- Pino **MISO** ligado na porta 12 do Arduino
- Pino **SCK** ligado na porta 13 do Arduino

Utilizando uma protoboard como apoio, foi montado as conexões RST, MOSI, MISO, SCK, GND e 3.3V, cada uma interligando os módulos RFID e o Arduino, ou seja, cada uma com 3 pinos. Foi adicionado o LED a *protoboard* e suas conexões.

A fechadura teve um dos seus terminais, ligado diretamente a fonte, enquanto o outro foi ligado ao terminal NO do relé, no qual no relé foi ligado ao terminal comum o outro fio da fonte.

Depois de montada ficou conforme a figura 6 a seguir:

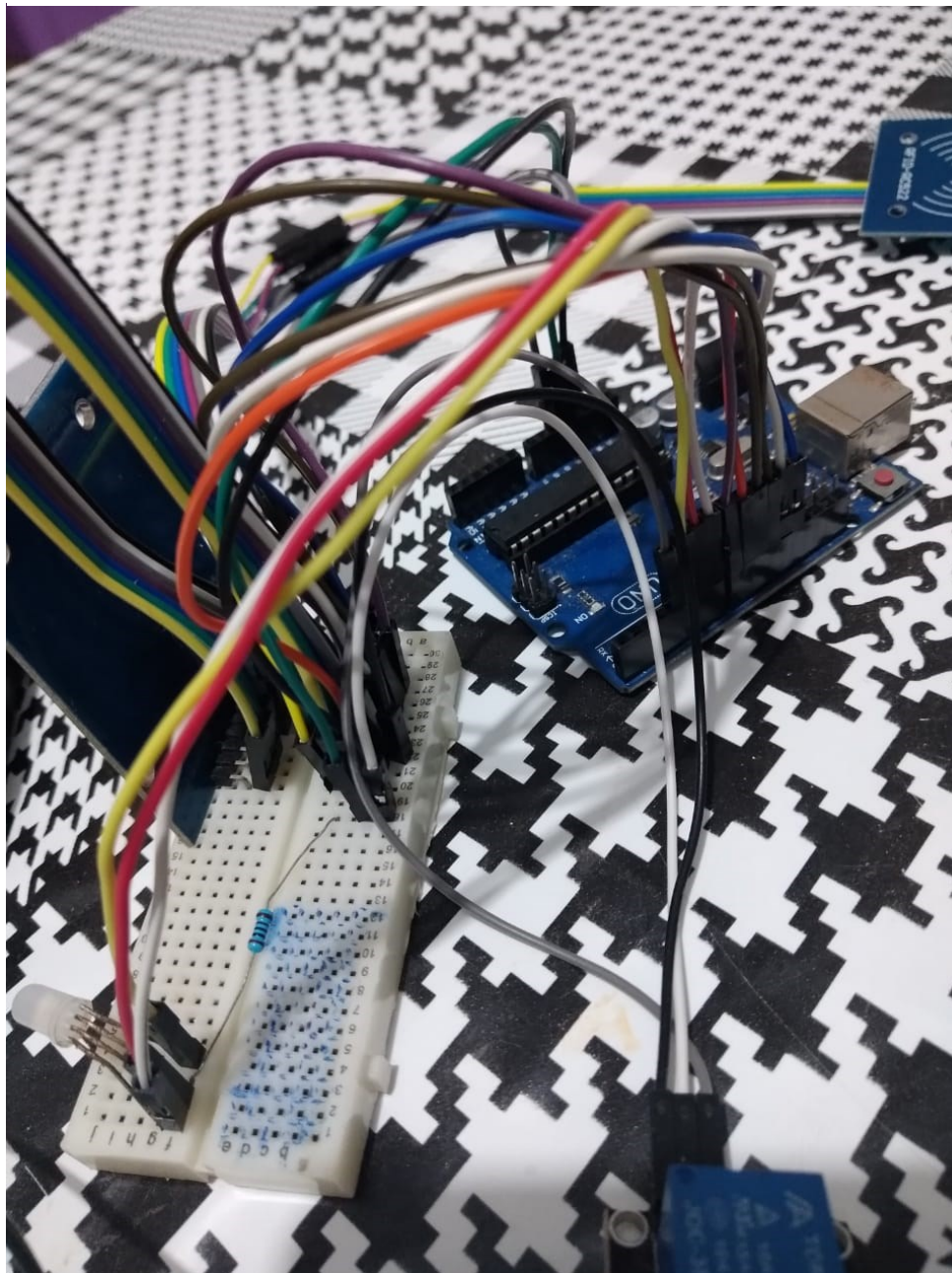
Figura 6 - Projeto montado



Fonte: AUTORIA PRÓPRIA

Já as conexões podem ser vistas conforme mostrado pela figura 7 a seguir:

Figura 7 - Conexão dos componentes



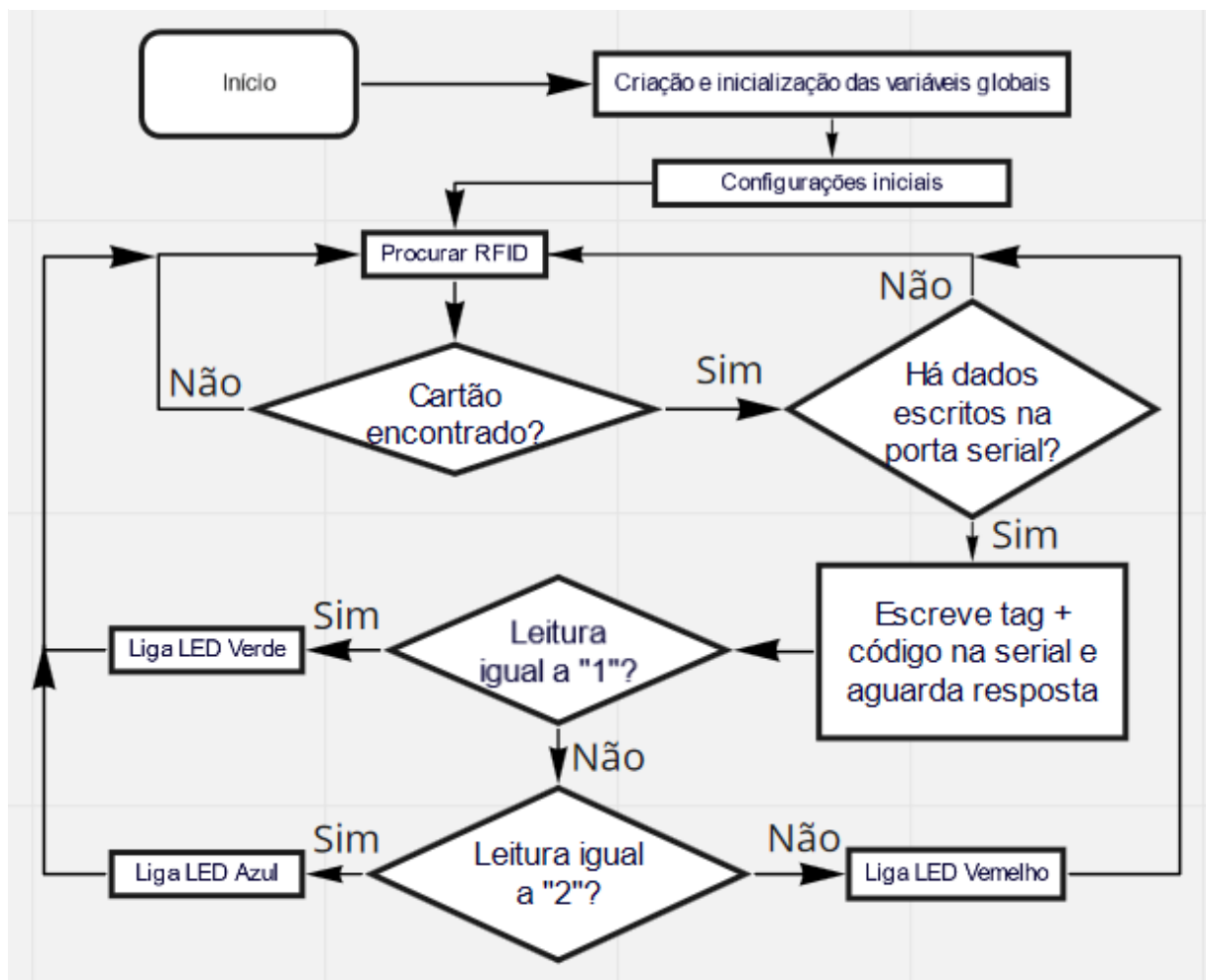
Fonte: AUTORIA PRÓPRIA

4.2 PROGRAMAÇÃO DO ARDUÍNO

Com a conclusão da montagem dos componentes físicos da fechadura teve-se o início da parte de controle, primeiramente sendo feito o código de controle do Arduino.

A programação do Arduino (figura 8) é constituída de em 3 partes principais: primeira a inclusão das bibliotecas necessárias e inicialização de variáveis globais, em segundo vem a função *setup* para configuração inicial de operação e por último vem a função *loop*, onde ocorre a sua repetição indeterminadamente até o desligamento do sistema (como demonstrado na figura a seguir). Pode-se também definir outras funções que se repetirão durante o correr do

Figura 8 - Fluxo do código do Arduino



loop.

Fonte: AUTORIA PRÓPRIA

Foram incluídas as bibliotecas do protocolo SPI e do módulo MFRC522. Foram definidas as variáveis das portas utilizadas onde ocorreria controle, sendo seleção do módulo RFID, acionamento do relé ou LED.

Na sequência, a função *setup*, que define as configurações de operação inicial. Nela foram definidas a inicialização da porta serial (para comunicação com o computador), da SPI, dos módulos RFID. Em sequência foram definidos os pinos do LED e relé.

Em sequência foi criado funções para ligar o LED, que o liga-o por um determinado tempo junto do relé (no caso das luzes verde e azul) e então desliga-os. Outra função foi para os módulos RFID (*lerrfid*), já que o projeto possui 2 módulos RFID e suas funções seriam iguais, assim facilitando a legibilidade do código e economia de memória ao não se repetir o mesmo código.

A função *lerrfid* quando chamada inicia procurando por uma tag RFID, então testa se tem dados escrito na serial. Possuindo dados é feito o tratamento do dado (pois o protocolo SPI não possui interpretação de dados) e adiciona o código de operação (entrada ou saída), por fim então escrito na serial e então a função fica aguardando resposta do computador. Ao receber a resposta efetua um teste para verificar se libera a fechadura para entra, chamando a função de ligar o LED verde, em sequência, ou saída, chamando a função de LED azul. Caso não seja uma das duas respostas esperadas ficou definido que seria negada o acesso e chamaria a função do LED vermelho.

Por fim a função *loop* chama a função *lerrfid* para o módulo RFID de entrada (que possui código de operação 1) e posteriormente a chama de novo, só que desta vez para o RFID de saída (este que possui código de operação 2).

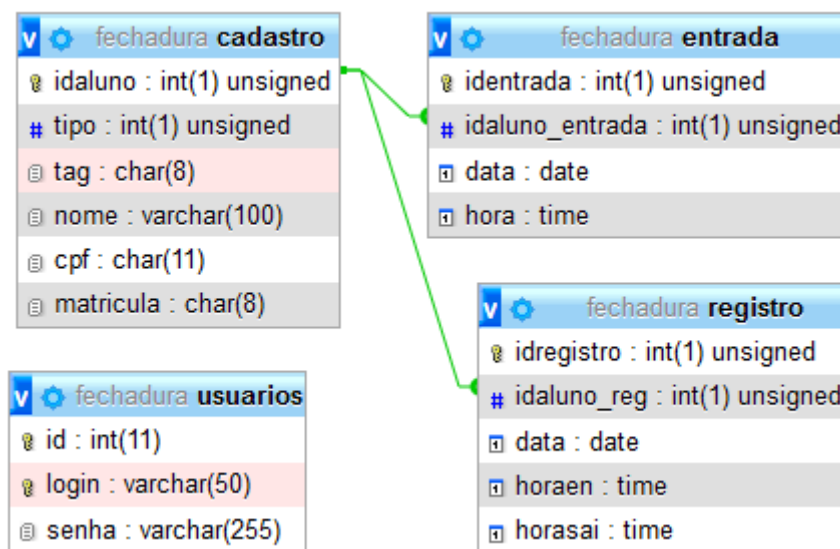
4.3 CRIAÇÃO DO BANCO DE DADOS

Para o armazenamento organizado de cada usuário cadastrado ou o registro de cada entra ou saída foi necessário a criação de um banco de dados, no qual o banco foi dividido em 4 tabelas: cadastro, entrada, registro e usuários.

Na tabela cadastro é onde ficam armazenados os dados dos usuários e conta com os campos: id do aluno, nome, CPF, matricula, tag e tipo de usuário. Já na tabela entrada é onde estão armazenados os dados de entrada e conta com os campos: id de entrada, id do aluno, data e hora da entrada. Por sua vez a tabela registro é onde ficam armazenados os registros de saída, contando com os campos: id do registro, id do aluno, data, hora de entrada e hora de saída.

Por fim a tabela usuários, onde ficam armazenados os logins e senhas para os usuários que possuem acesso ao site, conforme demonstrado pela figura 9 a seguir.

Figura 9 - Tabelas do banco de dados



Fonte: AUTORIA PRÓPRIA

4.4 SCRIPT DO COMPUTADOR

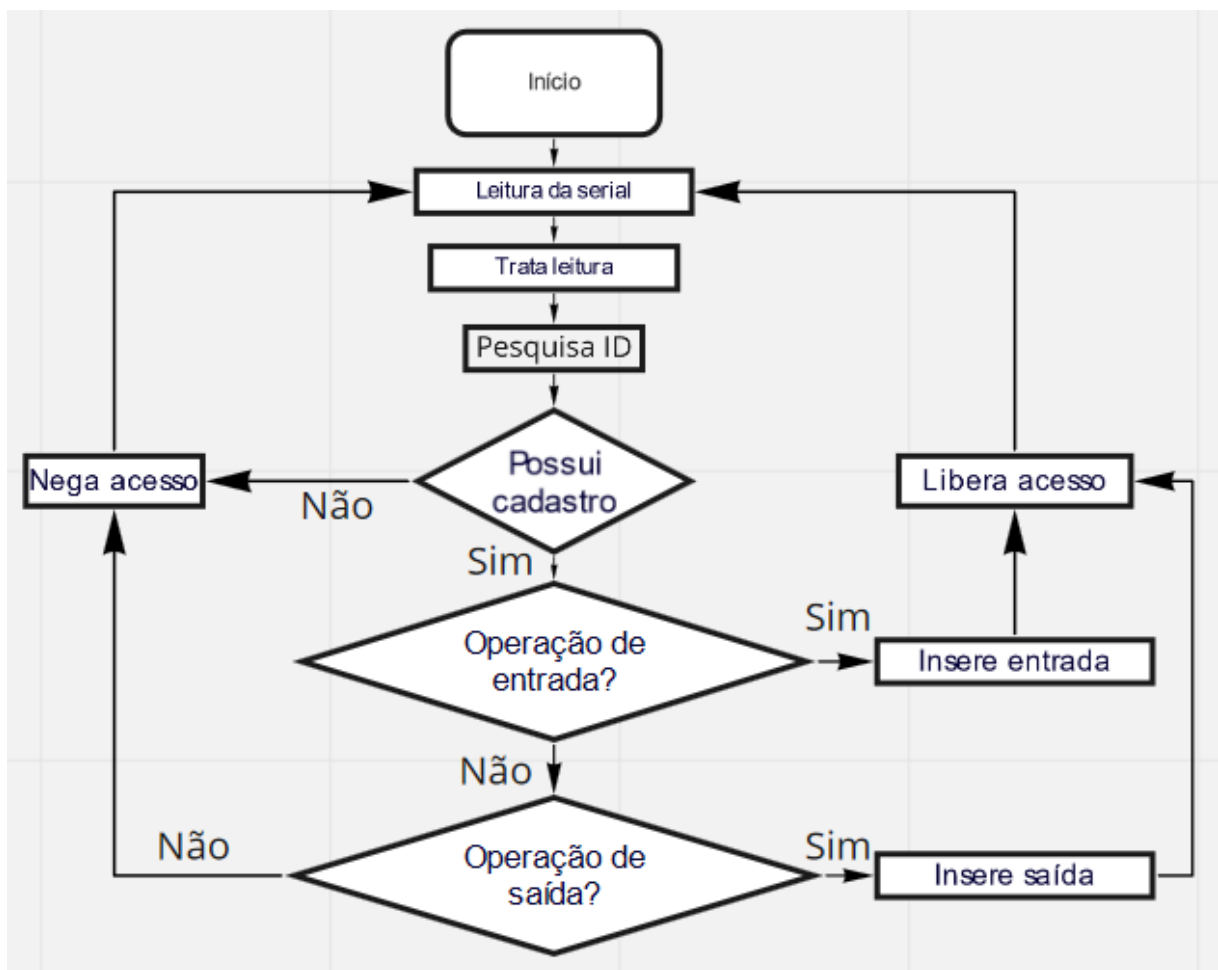
Posteriormente a programação do Arduino foi necessário a criação de um programa no computador (figura 10), onde seria efetuado a pasquina no banco de dados para efetuar ou não a liberação da fechadura e o cadastro das entradas e saídas dos usuários.

Foi escolhido o Python3 como a linguagem que seria utilizada para a programação do PC, pois como já citado no tópico 3.2.1, o Python3 é multiplataforma (por ser interpretada) e possui um grande acervo de bibliotecas, no qual foram utilizadas três neste trabalho. A primeira foi a “pyserial 3.4”, no qual conta com funções de manipulação de porta serial, como leitura e

escrita. A segunda foi a biblioteca “PyMySQL 0.9.3”, que permite a manipulação de bancos de dados, como inserção, seleção, atualização e remoção de dados. A terceira foi a biblioteca “date”, está que possibilita a manipulação de data e hora.

Com as bibliotecas já determinadas teve-se o início da programação do script, no qual foi definido as seguintes funções necessárias para o funcionamento: conexão ao banco de dados, leitura da serial, tratamento dos dados da serial, consulta de tag no banco de dados, cadastro de entrada, remoção de entrada, cadastro de saída e função principal.

Figura 10 - Fluxo script do computador



Fonte: AUTORIA PRÓPRIA

A primeira função, conexão ao banco de dados, permite a conexão do script com o banco de dados e seleciona a base de dados da fechadura; em seguida, a segunda função, leitura da serial, realiza a leitura de dados da porta serial advindos do Arduíno.

Posteriormente, a função de tratamento de dados da porta serial, que faz uma filtragem nos dados lidos da serial, selecionando somente o código da tag e código de operação (entrada ou saída), e descartando os demais dados.

Na sequência, a função de consulta da tag no banco de dados, que verifica no banco de dados se possui alguma tag registrada com aquele valor lido, retornando o id de usuário que a possui aquela tag, ou retorna 0 caso não possua registro e assim não travar o script.

Pouco depois, a função cadastro de entrada, função esta que inicialmente verifica se o ID de usuário é válido, então se já possui algum registro de entrada anterior e o exclui, caso já possua. Posteriormente faz a inserção no banco de dados um novo registro de entrada contendo o dia e a hora do momento de entrada; na sequência, tem-se a função remoção de entrada, função tem como objetivo excluir do banco de dados algum registro de entrada que possa se ter.

Adiante, a função cadastro de saída, que verifica se possui um usuário cadastrado, caso não possua, é negado a abertura da fechadura. Caso tenha um usuário válido verifica a existência do cadastro de entrada, verifica se o dia é o mesmo e faz a inserção na tabela de saída (chamada de registro) um novo registro contendo o dia, hora de entrada e hora de saída, então faz a remoção do registro de entrada. Ficou definido por padrão que se não possuir entrada ou os dias forem diferentes será negada a abertura da fechadura.

Por fim a função principal, que é um laço de repetição infinito. Inicialmente fica aguardando o Arduíno enviar via serial a tag e código de operação, faz o tratamento destes dados. Então verifica se é uma operação de entrada, chamando a função de registro de entrada e envia ao Arduíno o código de liberação de entrada, ou operação de saída, chamando a função de cadastro de saída e encaminhando ao Arduíno o código de saída. Por medida de segurança qualquer operação que não seja entrada ou saída é negada a liberação da fechadura e enviado ao Arduíno o código de negação de liberação.

Posteriormente com o Arduíno programado e o *script* do computador pronto, foi necessário alguns pequenos ajustes de sincronização nos dois códigos, devido aos componentes

operarem com velocidades de processamento consideravelmente grande. Enquanto o Arduino opera com um *clock* na casa dos 16MHz, o computador opera na casa dos 2GHz a 4GHZ.

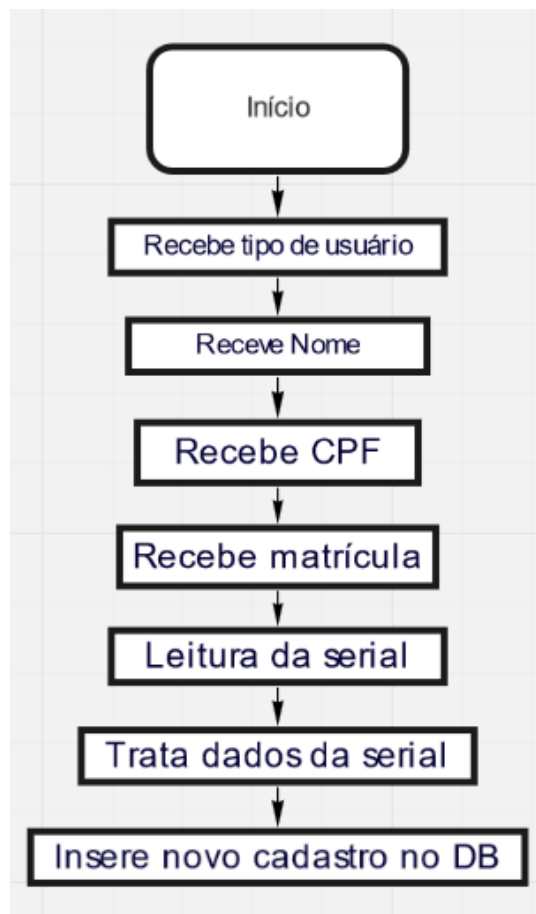
4.5 SCRIPT PARA CADASTRO DE NOVOS USUÁRIOS

Ademais, para o cadastro de novos usuários (figura 11), foi necessário a implementação de um script em Python3 para o cadastro de novos usuários. Inicialmente foi incluso as seguintes bibliotecas: “PyMySQL 0.9.3” para acesso e manipulação de banco de dados e “pyserial 3.4” para utilização da porta serial.

Em seguida foi criada uma função chamada *trata*, que faz o tratamento da leitura da serial, parecida com a função *trata* do tópico 4.3, só que esta função pega somente o valor da tag, ignorando o código de operação enviado pelo Arduino. Também foi criada a função *cadcadastro* que faz a inserção dos dados de um novo cadastro no banco de dados.

Por fim tem-se a parte principal do programa, que se inicia pedindo nome, cpf, matrícula e tipo de usuário (1 para administrador e 2 para aluno). Com estes dados coletados o programa fica aguardando o Arduino encaminhar uma nova tag (do qual, quem está realizando a operação de cadastro tem que ir até o leitor rfid, podendo ser o de entrada ou saída e aproximar a tag que se deseja cadastrar. Após a leitura da tag o Arduino envia normalmente a tag mais o código de operação), então chama a função de tratamento de leitura e posteriormente chama a função *cadcadastro*, encerrando o *script*.

Figura 11 - Fluxo script de novo cadastro



Fonte: AUTORIA PRÓPRIA

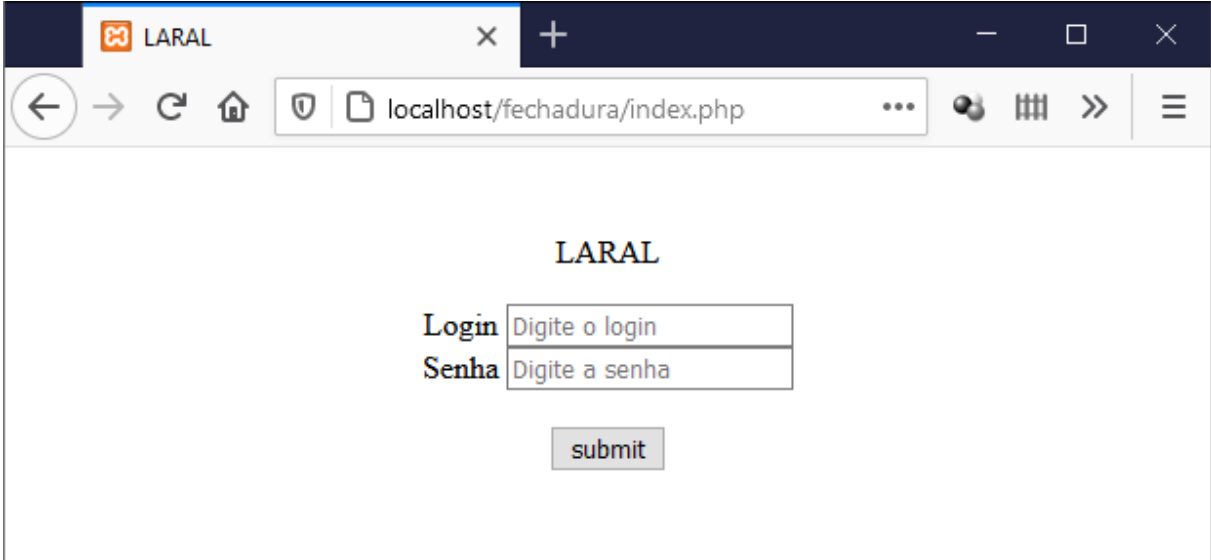
4.6 PAGINAS WEB

Foi produzido um site para que se possa visualizar os registros de entradas e saídas. Este site conta com página de login, menu, lista de entrada, lista de saída e pesquisa de registros por matrícula.

A página inicial ou de login (figura 12), possui um campo para digitação do login e outro para digitação da senha e botão enviar. Ao se clicar no botão enviar é feita uma pesquisa no banco de dados procurando pelo usuário e compara se login e senha são iguais

aos armazenados. Possui um sistema de proteção contra *sql injection* (tentativa de invasão ao se alterar as características da consulta SQL, assim liberando acesso sem precisar de login e

Figura 12 - Página de login



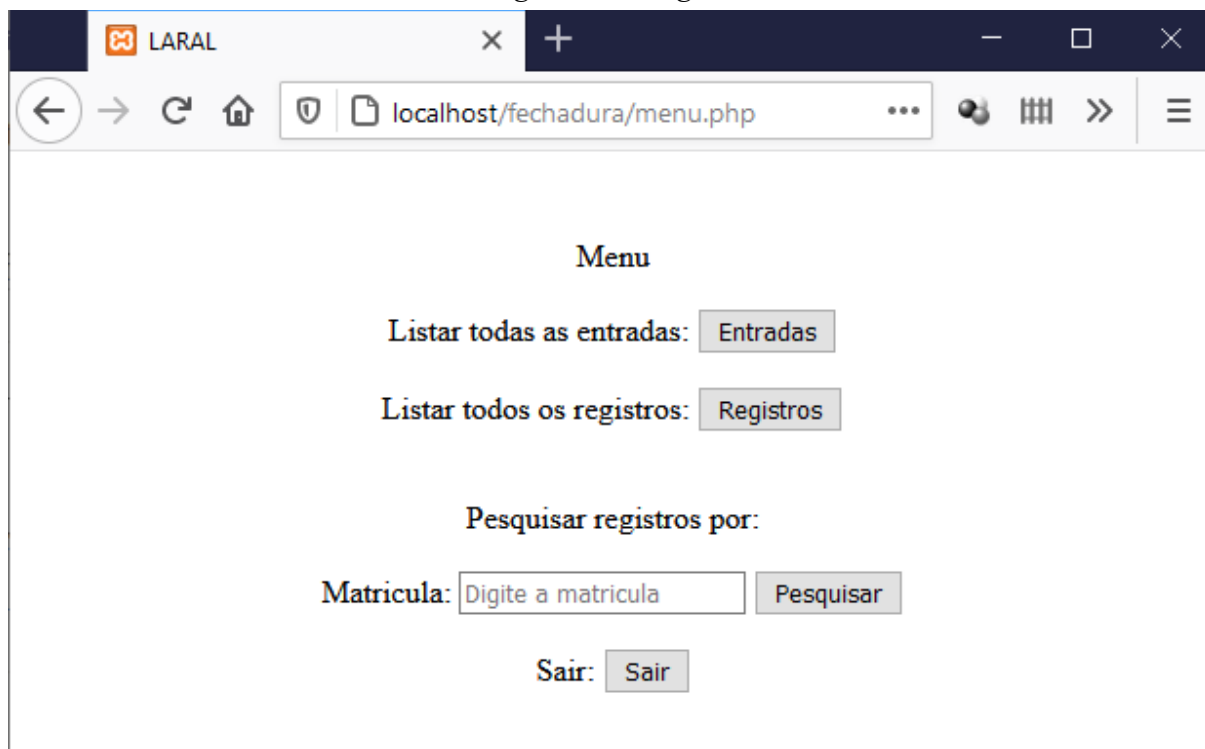
The image shows a web browser window with a single tab titled 'LARAL'. The address bar displays 'localhost/fechadura/index.php'. The page content is centered and features the title 'LARAL' at the top. Below the title, there are two input fields: 'Login' with the placeholder text 'Digite o login' and 'Senha' with the placeholder text 'Digite a senha'. A 'submit' button is positioned below these fields.

senha válidos) no campo da senha.

Fonte: AUTORIA PRÓPRIA

A segunda página é a página do Menu (figura 13), nela há a opção de ser redirecionado para as páginas de listagem de entradas e listagem de registros (saída). Também pode-se realizar a pesquisa de registros por matrícula.

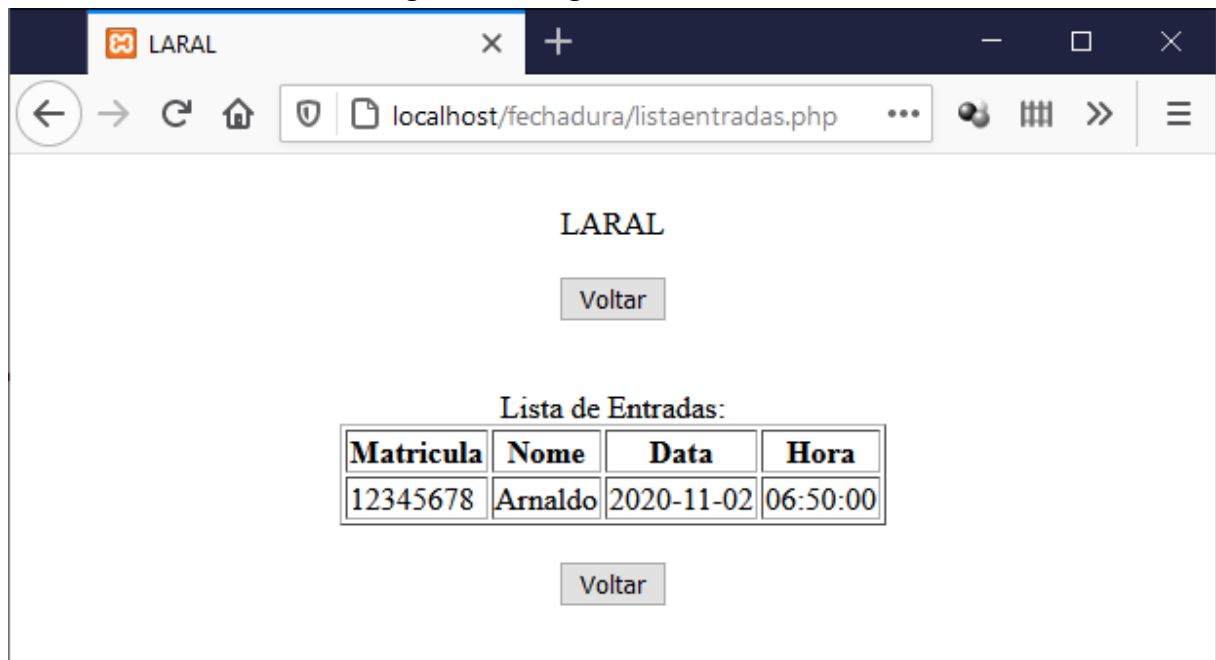
Figura 13 - Página do menu



Fonte: AUTORIA PRÓPRIA

Em seguida vem a página Lista de Entradas (figura 14), que possui uma tabela contendo o registro de todas as entradas registradas até o momento, que ainda não efetuaram a saída. Nesta tabela há os dados de matrícula, nome, data e hora da entrada. Possui dois botões um no início e outro no final da página que levam o usuário de volta a página Menu, chamados de voltar. Possui dados utilizados nos testes da fechadura.

Figura 14 - Página lista de entradas



Fonte: AUTORIA PRÓPRIA

Posteriormente a página Lista de Registros (figura 15), que possui uma tabela contendo todos os registros de saída, apresentando matrícula, nome, data, hora de entrada e hora de saída. Possui também dois botões, um no início e outro no final da página que levam o usuário de volta a página Menu, chamados de voltar. Apresenta os dados utilizados nos testes da fechadura.

Figura 15 - Página lista de registros

LARAL

[Voltar](#)

Registros:

Matricula	Nome	Data	Entrada	Saida
18201412	Diogo W Silveira	2020-10-29	02:19:00	02:19:10
18201412	Diogo W Silveira	2020-10-29	04:24:02	04:24:08
18201412	Diogo W Silveira	2020-10-29	04:42:46	04:42:57
18201412	Diogo W Silveira	2020-10-29	06:49:13	06:49:25
18201412	Diogo W Silveira	2020-10-29	06:49:38	06:49:45
18201412	Diogo W Silveira	2020-10-29	07:14:44	07:15:36
18201412	Diogo W Silveira	2020-10-29	14:06:56	14:47:29
18201412	Diogo W Silveira	2020-10-30	05:54:06	05:56:15
18201412	Diogo W Silveira	2020-10-30	05:56:19	05:56:22
18201412	Diogo W Silveira	2020-10-30	06:01:34	06:01:38
18201412	Diogo W Silveira	2020-11-03	16:30:25	16:41:51
18201412	Diogo W Silveira	2020-11-03	16:44:47	16:44:50
18201412	Diogo W Silveira	2020-11-04	01:20:10	01:20:17
18201412	Diogo W Silveira	2020-11-04	01:33:33	01:33:37
12345678	Arnaldo	2020-10-29	04:17:14	04:18:27
12345678	Arnaldo	2020-10-30	12:38:31	12:38:35
12345678	Arnaldo	2020-11-02	06:49:35	06:49:55

[Voltar](#)

Fonte: AUTORIA PRÓPRIA

Por fim temos a página de pesquisa por matrícula (figura 16), no qual possui uma tabela contendo todos os registros da matrícula procurada, apresentando matrícula, nome, data, hora de entrada e hora de saída. Possui também dois botões, um no início e outro no final da página que levam o usuário de volta a página Menu. Possui dados utilizados nos testes da fechadura.

Figura 16 - Página lista por matrícula

LARAL

Voltar

Registros de 18201412:

Matricula	Nome	Data	Entrada	Saida
18201412	Diogo W Silveira	2020-10-29	02:19:00	02:19:10
18201412	Diogo W Silveira	2020-10-29	04:24:02	04:24:08
18201412	Diogo W Silveira	2020-10-29	04:42:46	04:42:57
18201412	Diogo W Silveira	2020-10-29	06:49:13	06:49:25
18201412	Diogo W Silveira	2020-10-29	06:49:38	06:49:45
18201412	Diogo W Silveira	2020-10-29	07:14:44	07:15:36
18201412	Diogo W Silveira	2020-10-29	14:06:56	14:47:29
18201412	Diogo W Silveira	2020-10-30	05:54:06	05:56:15
18201412	Diogo W Silveira	2020-10-30	05:56:19	05:56:22
18201412	Diogo W Silveira	2020-10-30	06:01:34	06:01:38
18201412	Diogo W Silveira	2020-11-03	16:30:25	16:41:51
18201412	Diogo W Silveira	2020-11-03	16:44:47	16:44:50
18201412	Diogo W Silveira	2020-11-04	01:20:10	01:20:17
18201412	Diogo W Silveira	2020-11-04	01:33:33	01:33:37

Voltar

Fonte: AUTORIA PRÓPRIA

5 CONCLUSÃO

Durante o desenvolvimento do projeto foi necessário um estudo sobre os componentes e tecnologias utilizadas. Ficou evidente as diversas aplicações dos *hardwares* e *softwares*, tanto para este projeto, quanto para uma variedade de outros projetos.

O objetivo deste trabalho foi concluído com sucesso, objetivo esse que era a criação de uma fechadura eletrônica para manter a porta da sala trancada, realizar o controle de acesso ao laboratório e realizar automaticamente o registro do histórico de entradas e saídas.

A fechadura é de fácil utilização, bastando que o usuário porte sua carteirinha de identificação da Universidade como chave de acesso, de baixo custo (custou aproximadamente R\$300,00), e que os dados de acesso possam ser visualizados através da página WEB desenvolvida.

Com um *script* em Python3 foi possível fazer o cadastro de novos usuários. Com a utilização da fechadura elétrica AGLinha, que possui um sistema de abertura mecânica (por chave física), ficando possível a abertura, por ocasião, em caso de falta de energia.

Ficou evidente também que este projeto possui um grande potencial de refinamento, ao se fazer as conexões utilizando uma placa de solda em vez de uma *protoboard*, mas mantenha terminais para uma fácil substituição dos componentes. Também foi observado a possibilidade de inclusão de um segundo LED para que, enquanto um LED indique para quem está do lado de dentro a operação que está ocorrendo no Arduíno (entrada, saída ou negação de acesso), quem está do lado de fora também possa o visualizar.

Uma evolução em potencial se dá no site desenvolvido, podendo ser adicionados: pesquisa por nome, relatórios de acesso, acesso remoto (podendo ser adicionado ao site do laboratório, para que os professores supervisores possam verificar remotamente o acesso), exclusão de usuários cadastrados, entre outras.

Em anexo se encontram os códigos do Arduíno e *scripts* Python3.

REFERÊNCIAS

ADMINBLOG_USR. O que é Hardware?. bringIT/blog. 2019. Disponível em: <https://www.bringit.com.br/blog/duvidas-frequentes/que-hardware/>. Acesso 21 abr. 2021

AGL. Fechadura elétrica. Disponível em: <https://www.aglfechaduras.com.br/fechadura-eletrica>. Acesso em: 26 abr. 2021.

ARDUINO. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em: 21 abr. 2021.

CAELUM. Introdução ao Pycharm. Disponível em: <https://www.caelum.com.br/apostila-python-orientacao-a-objetos/introducao-ao-pycharm#ide>. Acesso em: 26 abr. 2021.

CHASTAGNIER, Paola. Etiquetas RFID, seus modelos e aplicações. Disponível em: <https://rfidbrasil.com/blog/etiquetas-rfid-seus-modelos-e-aplicacoes/>. Acesso em: 02 maio 2021.

GOGONI, Ronaldo. O que é software?. Disponível em: <https://tecnoblog.net/311647/o-que-e-software/>. Acesso em: 24 abr. 2021.

GOMES, Pedro César Tebaldi. Conheça os principais protocolos de rede e seus usos!. Disponível em: <https://www.opservices.com.br/protocolos-de-rede/>. Acesso em: 26 abr. 2021.

GUIMARÃES, Fábio. SPI. Disponível em: <http://mundoprojetado.com.br/spi/>. Acesso em: 27 abr. 2021.

HIGA, Paulo. O que é XAMPP e para que ser. Disponível em: <https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/02/o-que-e-xampp-e-para-que-serve.html>. Acesso em: 24 abr. 2021.

JETBRAINS. PyCharm. Disponível em: <https://www.jetbrains.com/pt-br/pycharm/>. Acesso em: 26 abr. 2021.

LABORATÓRIO DE GARAGEM. Tutorial: Comunicação SPI (Serial Peripheral Interface) com Arduino. Disponível em: <https://labdegaragem.com/profiles/blogs/tutorial-comunica-o-spi-serial-peripheral-interface-com-arduino>. Acesso em 27 abr. 2021.

MARQUES, Rafael. O que é PHP: entenda de forma descomplicada. Disponível em: <https://www.homehost.com.br/blog/tutoriais/php/o-que-e-php/>. Acesso em: 25 abr. 2021.

MATTEDE, Henrique. O que é relé? Como funciona um relé?. Disponível em: <https://www.mundodaeletrica.com.br/o-que-e-rele-como-funciona-um-rele/>. Acesso em 23 abr. 2021.

PETENATE, Marcelo. Identificação por Radiofrequência, ou RFID, Como Funciona Esse Sistema Seguro. 2019. Disponível em: <https://www.escolaedti.com.br/rfid-o-que-e>. Acesso em: 23 abr. 2021

PISA, Pedro. O que é e como usar o MySQL?. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>. Acesso em: 25 abr. 2021.

PORTAS DE MADEIRA. Fechadura eletrônica – Guia completo de como funcionam, quais os tipos e como podem melhorar a segurança. Disponível em: <https://allmadloja.com.br/fechadura-eletronica-guia-completo-de-como-funcionam-quais-os-tipos-e-como-podem-melhorar-a-seguranca/>. Acesso em: 05 maio 2021.

RED HAT. Ambiente de Desenvolvimento Integrado O que é IDE?. Disponível em: <https://www.redhat.com/pt-br/topics/middleware/what-is-ide>. Acesso em: 25 abr. 2021.

REDAÇÃO. O que é servidor Apache?. Disponível em: <https://canaltech.com.br/internet/o-que-e-servidor-apache/>. Acesso em: 24 abr. 2021.

ROBOCORE. Comparação Entre Protocolos de Comunicação Serial. Disponível em: <https://www.robocore.net/tutoriais/comparacao-entre-protocolos-de-comunicacao-serial.html>. Acesso em: 27 abr. 2021

ROVEDA, Ugo. O que é Python, para que serve e por que aprender?. Disponível em: <https://kenzie.com.br/blog/o-que-e-python/>. Acesso em: 24 abr. 2021.

SANTOS, Diego Marcelo dos. Relê. Disponível em: <https://www.infoescola.com/eletronica/rele/>. Acesso em: 24 abr. 2021.

SOPRANO. Fechadura eletrônica, fechadura elétrica e fechadura digital: existe diferença?. Disponível em: <https://www.soprano.com.br/blog/fechadura-eletronica-fechadura-eletrica-e-fechadura-digital-existe-diferenca>. Acesso em: 05 maio 2021.

SOUZA, Fábio. Arduino Uno. 2013. Disponível em: <https://www.embarcados.com.br/arduino-uno/>. Acesso em 23 abr. 2021.

TELES, Elaine. Arduino: O que é? Pra que serve? Quais as possibilidades?. Coletividade. 2016. Disponível em: <https://medium.com/nossa-coletividade/arduino-o-que-%C3%A9-pra-que-serve-quais-as-possibilidades-efbd59d33491>. Acesso em: 21 abr.2021.

TORRES, Vitor. Socorro: o que é a IDE do Arduino?!. Disponível em: <http://www.natalnet.br/ura/?p=438>. Acesso em: 25 abr. 2021.

ANEXO A – Códigos desenvolvidos

Neste anexo estão contidos os códigos desenvolvidos na construção da fechadura eletrônica.

Código do Arduíno:

```
#include <SPI.h> // Biblioteca do SPI (Serial Peripheral Interface)
#include <MFRC522.h> // Biblioteca do módulo RFID-RC522

#define SS_PIN1 10 // Define o pino 10 como select do rfid1 (entrada)
#define SS_PIN2 8 // Define o pino 8 como select do rfid2 (saida)
#define RST_PIN 9 // Define o pino 9 como Reset
#define blue 7 // Define o pino 7 para a cor azul do LED
#define green 6 // Define o pino 6 para a cor verde do LED
#define red 5 // Define o pino 5 para a cor vermelho do LED
#define rele 4 // Define pino 4 como do do ralay da fechadura
#define tr 700 // Define o tempo para acionamento do rele 0 - 1000

MFRC522 rfid1(SS_PIN1, RST_PIN); // Cria o objeto rfid de entrada
MFRC522 rfid2(SS_PIN2, RST_PIN); // Cria o objeto rfid de saida

int recebe = 0;

void setup() {
  Serial.begin(9600); // Inicia a serial
  SPI.begin(); // Inicia SPI bus
  rfid1.PCD_Init(); // Inicia MFRC522 1
  rfid2.PCD_Init(); // Inicia MFRC522 2
  pinMode(blue, OUTPUT); //Define pino do led azul como saida
  pinMode(red, OUTPUT); //Define pino do led azul como saida
  pinMode(green, OUTPUT); //Define pino do led azul como saida
```

```
pinMode(rele, OUTPUT); //Define pino do relay como saida
digitalWrite(blue, HIGH);
digitalWrite(green, HIGH);
digitalWrite(red, HIGH);
digitalWrite(rele, LOW);
}
```

```
void ligablue(){
  digitalWrite(blue, LOW);
  digitalWrite(rele, HIGH);
  delay(tr);
  digitalWrite(rele, LOW);
  delay(1000-tr);
  digitalWrite(blue, HIGH);
}
```

```
void ligagreen(){
  digitalWrite(green, LOW);
  digitalWrite(rele, HIGH);
  delay(tr);
  digitalWrite(rele, LOW);
  delay(1000-tr);
  digitalWrite(green, HIGH);
}
```

```
void ligared(){
  digitalWrite(red, LOW);
  delay(1000);
  digitalWrite(red, HIGH);
}
```

```

void lerrfid(MFRC522 mfrc522, String i){
  if (! mfrc522.PICC_IsNewCardPresent()){ // Aguarda a aproximacao do cartao
    return;
  }
  if (! mfrc522.PICC_ReadCardSerial()){ // Seleciona um dos cartoes
    return;
  }
  String conteudo= "";
  for (byte i = 0; i < mfrc522.uid.size; i++){
    conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  String sinais = i + "|" + conteudo;
  Serial.println(sinais);
  recebe = Serial.parseInt();
  Serial.print(recebe);
  if (recebe == 1 || recebe == '1'){
    ligablue();
    return;
  }
  else if (recebe == 2 || recebe == '2'){
    ligagreen();
    return;
  }
  else{
    ligared();
    return;
  }
  recebe = 0;
}

```

```

void loop() {
  lerrfid(rfid1, "1"); // rfid1 entrada
  delay(500);
  lerrfid(rfid2, "2"); // rfid2 saida
  delay(500);
}

```

Código do PC (script em Python3):

Figura 17 - Código PC parte 1

```

def insertentrada(id):
    if id <= 0:
        print("Id entrada invalido")
        ser.write(b'3')
        return
    id = str(id)
    query = "SELECT identrada FROM entrada WHERE idaluno_entrada = " + id
    c.execute(query)
    a = c.fetchall()
    if a: # testa se nao é nulo
        deleteentrada(id)
        print("entrada deletada")
    query = "INSERT INTO entrada VALUES (DEFAULT, " + id + ", DEFAULT, DEFAULT)"
    c.execute(query)
    con.commit()
    print("Inserido entrada com sucesso")
    ser.write(b'1')

```

Fonte: AUTORIA PRÓPRIA

Figura 18 - Código PC parte 2

```
import pymysql as sql
import serial
from datetime import date

usb = "COM3"
host = "localhost"
user = "diogo"
password = "121925"
db = "fechadura"
port = 3306
con = sql.connect(host, user, password, db, port)
c = con.cursor()

def selectidcadastro(tag):
    query = "SELECT idaluno FROM cadastro WHERE tag = '" + tag + "'"
    c.execute(query)
    a = c.fetchall()
    if a:
        x = a[0][0]
        return x
    else:
        return 0

def deleteentrada(id):
    i = int(id)
    if (i < 1):
        return
    id = str(id)
    query = "DELETE FROM entrada WHERE idaluno_entrada = " + id
    c.execute(query)
    con.commit()
```

Fonte: AUTORIA PRÓPRIA

Figura 19 - Código PC parte 3

```
def trata(x):
    x1 = str(x)
    a = x1.split("|")
    t1 = a[0].split("b'")
    s1 = a[1].split(" ")
    s2 = s1[4].split("\\r")
    s3 = s1[1] + s1[2] + s1[3] + s2[0]
    lista = (int(t1[1]), s3)
    return lista

ser = serial.Serial(usb, 9600)
print("Serial Iniciada...")

while 1:
    leitura = ser.read(16)
    y = trata(leitura)
    cod = selectidcadastro(y[1])
    if cod > 0: # verifica se ta cadastro (possui tag cadastrada)
        if y[0] == 1:
            insertentrada(cod)
        elif y[0] == 2:
            insertregistro(cod)
        else: # garantia que nao vai liberar
            print("Valor invalido")
            ser.write(b'3')
    else:
        print("Não possui cadastro")
        ser.write(b'3')
    ser.read(1)

    print("Aguardando nova leitura")
```

Fonte: AUTORIA PRÓPRIA

Figura 20 - Código PC parte 4

```
def insertregistro(id):
    print("Entrou no insere")
    if id < 1:
        print("Id saida invalido")
        return
    id = str(id)
    query = "SELECT data, hora FROM entrada WHERE idaluno_entrada = " + id #seleciona entrada com a id
    c.execute(query)
    a = c.fetchall()
    if a: # testa se existe alguma entra existente
        dia = date.today()
        select = (str(a[0][0]), str(a[0][1]))
        d1 = select[0].split("-")
        d2 = d1[0] + d1[1] + d1[2]
        t1 = select[1].split(":")
        t2 = t1[0] + t1[1] + t1[2]

        if a[0][0] == dia:
            query = "INSERT INTO registro VALUES (DEFAULT, " + id + ", " + d2 + ", " + t2 + ", DEFAULT)"
            c.execute(query)
            con.commit()
            print("Registro inserido com sucesso")
            ser.write(b'2')
        else:
            print("Dias diferentes")
            ser.write(b'3')
        deleteentrada(id)
    else:
        print("Não possui entrada")
        ser.write(b'3')
```

Fonte: AUTORIA PRÓPRIA

Código cadastro (script em Python3)

Figura 21 - Código cadastro novos usuários

```

import pymysql as sql
import serial

host = "localhost"
user = "diogo"
password = "121925"
db = "fechadura"
port = 3306
con = sql.connect(host, user, password, db, port)
c = con.cursor()
ser = serial.Serial("COM3", 9600)

def trata(x):
    teste = str(x)
    a = teste.split("|")
    s1 = a[1].split(" ")
    s2 = s1[4].split("\n")
    s3 = s1[1] + s1[2] + s1[3] + s2[0]
    l = (str(s3))
    return l

def cadcadastro(tipo, tag, nome, cpf, matricula):
    query = "INSERT INTO cadastro VALUES (DEFAULT, " + tipo + ", " + tag + ", " + nome + ", " + cpf + ", " + matricula + ")"
    print(query)
    c.execute(query)
    con.commit()

# Principal
tipo = str(input("Digite o tipo de usuario (1 para admin, 2 para aluno): "))
nome = input("Digite o nome: ")
cpf = str(input("Digite o cpf: "))
matr = str(input("Digite a matricula: "))

print("Agora passe a tag no leitor de entrada")

rfid = ser.read(16)
print(rfid)
tag = trata(rfid)
print(tag)

cadcadastro(tipo, tag, nome, cpf, matr)

```

Fonte: AUTORIA PRÓPRIA