

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC JÚLIO DE MESQUITA**

**Curso Técnico em Eletrônica**

**Ailton Filgueiras Tenório**

**Daniel de Souza Silveira**

**Daniel do Nascimento Sousa**

**Gabriel Figueiredo dos Santos**

**Mikelly Dantas Reis**

**Monyse Andrade de Souza**

Relatório técnico apresentado  
junto ao Curso Técnico em  
Eletrônica, da Etec Júlio de  
mesquita, orientada pelos  
Professores Egmar Accetto e  
Adriana Mariko Yonamine  
Nakatani, como requisito  
parcial para obtenção do título  
de Técnico em Eletrônica.

**BRAIN-BS: Bloqueio e Segurança**

**Santo André - SP**

**2024**

## RESUMO

O projeto se trata de um Controle e Monitoramento de Ambiente com base em Arduino, ESP32-CAM e RFID. O seu funcionamento é dado pela detecção dos dados de uma pessoa através do cartão de acesso que a mesma apresenta ao leitor de frequência RFID, momento no qual ele irá conferir qual a informação dada à pessoa, e se ela consta no banco de dados da empresa, permitindo, ou não, seu acesso. O ESP32-CAM terá a função de capturar uma foto do titular e enviar um sinal para o microcontrolador do Arduino, que assim como já citado anteriormente, irá verificar o acesso do usuário. O objetivo da equipe com a escolha deste projeto como Trabalho de Conclusão de Curso é o considerável aumento da segurança em ambientes empresariais e industriais, cujo espaço requer qualificação e profissionalismo por parte daqueles que o frequentam. O presente trabalho tem como propósito apresentar um estudo sobre a segurança no ambiente de trabalho, como é possível controlar e monitorar os funcionários dentro de uma empresa, garantindo proteção e um espaço saudável e seguro, oferecendo bem-estar para todos os trabalhadores presentes no espaço, e um local mais produtivo. O sistema possui um preço acessível, com seu componente mais caro sendo o Arduino, e se trata de um sistema de proteção relativamente simples e confiável, com o leitor RFID exercendo sua função em um local estático. A única mudança necessária num projeto em tamanho real seria a obtenção de componentes maiores, como os próprios leitores, que teriam a necessidade de maior distância de detecção no espaço em que estiverem, e a câmera, para que pudesse fotografar a imagem de cada funcionário.

**Palavras-chave:** Empresa. Funcionário. Segurança. Trabalho. Monitoramento. Câmera.

## Sumário

<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>3</b>
2.1 SOFTWARE: TIPOS, FUNCIONAMENTO, IMPORTÂNCIA.....	3
2.2 SITE: DESENVOLVIMENTO WEB.....	3
2.3 MICROCONTROLADOR: ARDUINO UNO R3 .....	4
2.4 RFID: LEITOR DE RADIOFREQUÊNCIAS.....	5
2.5 ESP32-CAM: PARA FOTO CAPTURAS.....	6
2.6 DHT-11: SENSOR DE TEMPERATURA E UMIDADE .....	7
2.7 MYSQL: BANCO DE DADOS.....	7
2.8 XAMPP.....	8
<b>3. DESENVOLVIMENTO.....</b>	<b>9</b>
3.1 APRESENTANDO A IDEIA .....	9
3.2 METODOLOGIA .....	9
3.3 TABELA DE CUSTOS.....	12
3.4 LÓGICA.....	13
3.5 TECNOLOGIAS SEMELHANTES .....	14
3.6 SISTEMA IDEALIZADO .....	15
3.7 SISTEMA OFICIAL.....	15
<b>4 RESULTADOS .....</b>	<b>16</b>
4.5 RESULTADOS OBTIDOS.....	16
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>20</b>
<b>6 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>21</b>
<b>7 APÊNDICE .....</b>	<b>25</b>
7.5 DIÁRIO DE BORDO.....	25
<b>ANEXOS.....</b>	<b>29</b>

## 1. INTRODUÇÃO

Em diversas empresas, é perceptível a falta de controle adequado das ações dos funcionários em relação ao ambiente de trabalho onde frequentam, dificultando a forma de analisar o quão produtivo é o desempenho do operário no que diz respeito ao comportamento e segurança do mesmo e dos demais à sua volta.

A falta de segurança está fortemente ligada a vulnerabilidades em deficiências nas práticas de controle de acesso e a ausência de manutenção adequada no ambiente empresarial. Muitos gestores ainda subestimam a importância de investir em medidas de proteção, crendo que sua empresa está a salvo de possíveis ameaças. No entanto, essa visão equivocada pode desencadear vulnerabilidades, resultando em um espaço propenso a riscos.

Considerados os problemas acima, o projeto “Controle e Monitoramento de Ambiente” visa ajudar a diminuir os problemas citados acima de modo simples e intuitivo para os que dele utilizarem.

O projeto tem como objetivo melhorar a vida profissional dos funcionários e aprimorar a empresa com melhorias pontuais dos problemas, ajudando a mantê-la organizada com um sistema de proteção ambiente, auxiliando no controle de acesso e fluxo de pessoas. A ordem decorrente do uso deste sistema potencialmente diminuiria as taxas de problemas relacionados a privacidade e proteção do estabelecimento e seus respectivos empregados.

No mercado existem modelos semelhantes, mas com diferenças marcantes, como o uso de alarmes para o funcionamento e detecção ou meios diferentes do uso de câmeras para tal. Muitos trabalhos de conclusão de curso que se apropriam de um conceito parecido terão um resultado que se vê mais caro, ou desnecessariamente complicados.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Software: Tipos, funcionamento, importância.**

Software é um termo que se refere a programas de computador, que são conjuntos de instruções escritas para que um computador execute tarefas específicas. Softwares de sistemas, essenciais para o funcionamento de um aparelho digital, são responsáveis por estabelecer uma conexão simples entre o processador e os elementos externos que compõem um computador. Softwares de aplicativo, são programas desenvolvidos para realizar tarefas específicas de cada usuário.

Eles são escritos em linguagens de programação, este que é um meio de instruir comandos ao hardware do computador, como por exemplo: Python, C++, JAVA, entre outros.

Trazendo este conhecimento ao desenvolvimento do projeto em questão, não utilizaremos um software, porém um site. Por mais similar que sejam as aparências entre site e software, eles possuem estruturas e linguagens completamente diferente. Vejamos isso no tópico seguinte:

### **2.2 SITE: Desenvolvimento WEB**

O termo “site” ou “website” faz referência à uma página, ou agrupamento de páginas entre si, que permite acesso das pessoas através da internet utilizando um determinado endereço. Para a composição do site, será utilizado quatro tipos de linguagens: HTML, CSS, Java Script e PHP.

HTML é a linguagem responsável pela marcação do hipertexto, indicando a funcionalidade de cada elemento que compõe a página. O CSS, tem a função de estilizar os textos pré-indexados na página. O Java Script permite controlar os elementos do site em tempo real, sem necessariamente ter que receber dados ou respostas do servidor. E, por fim, o PHP é uma linguagem de script do lado do servidor, utilizada principalmente para a criação de páginas web dinâmicas. Ele permite a interação com bancos de dados, manipulação de formulários e a geração de conteúdo dinâmico, sendo executado no servidor antes de enviar o resultado final para o navegador do usuário.

Aplicando os conceitos de cada uma dessas linguagens, será possível desenvolver um site que seja capaz de cadastrar novos cartões, alterar suas informações ou modificar o acesso a diferentes áreas dentro de uma empresa.

Desse modo, o nosso website terá ligação direta com o hardware do BRAIN\_BS, facilitando o acesso dos funcionários ao seu respectivo histórico de acesso a ambientes, assim também possibilitando alterar esses dados, no caso do administrador da empresa.

### **2.3 MICROCONTROLADOR: Arduino UNO R3**

O Arduíno UNO R3 é um microcontrolador amplamente utilizado em projetos de eletrônica e programação, contendo muitas funcionalidades e aplicações.

Sendo uma placa de desenvolvimento baseada no microcontrolador ATmega328P, oferece uma plataforma acessível e versátil para estudantes e profissionais que desejam criar projetos eletrônicos interativos, tais como circuitos com o uso de leds, para controlar luzes, portões, sistema de irrigação, e outros dispositivos, de mesmo modo para projetos de robótica.

No contexto de desenvolvimento do nosso protótipo, o Arduino será o “cérebro” de todo o sistema de monitoramento e controle de pontos de trabalho, considerando que todo o processo de armazenamento, leitura e julgamento de acesso ou não ao respectivo ambiente, será executado pelo próprio microcontrolador Arduino.

Além disso, o Arduino terá um papel maior de unir e coordenar a comunicação entre o módulo RFID, o Esp 32 CAM e o site de configuração e monitoramento dos cartões e funcionários dentro da empresa.

**Figura 1 – foto do Arduino Uno R3**



Fonte: Montimport. Disponível em: <https://www.montimport.com/MLB-1628797484-arduino-uno-compativel-dip-atmega328p-r3-c-cabo-usb- JM>, 2024.

## 2.4 RFID: Leitor de Radiofrequências

O RFID MFRC-522 é um módulo de leitura e escrita altamente integrado para comunicação. O módulo pode se comunicar com um microcontrolador através de um SPI de 4 pinos com uma taxa de dados máxima de 10 mbps, suportando também uma comunicação através de protocolos I2C e UART. Ele é baseado no IC MFRC-522 da NXP, e é uma das opções mais acessíveis que nós encontramos para a então composição em nosso protótipo.

Um sistema RFID, do inglês Radio Frequency Identification (Identificação por radiofrequência), consiste em dois componentes principais: Uma TAG anexada ao objeto a ser identificado (Cartões, comumente) e um leitor e transcritor das informações armazenadas na própria TAG.

O módulo RFID é capaz de criar um campo eletromagnético de 13,56 MHz quando ligado a um microcontrolador. Por sua vez, a TAG ou cartão, que é considerado um dispositivo passivo (Não possui uma bateria ou alimentador de circuito), em sua composição interna possui um microchip que armazena informações, junto a antenas para transmissão e recebimento de dados, e uma bobina integrada.

Ao aproximarmos o cartão ao leitor de rádio frequências, por meio da mudança de polos indutivos gerada pelo módulo em questão, uma corrente elétrica é aplicada entre os fios internos da TAG, alimentando o microchip e a antena de modo a fazê-los funcionarem como o esperado.

Através da comunicação entre o microcontrolador e o leitor RFID, diferentes comandos são atribuídos ao chip que em seguida responde ao próprio leitor. Esses comandos podem alterar os dados guardados no microchip, lê-los ou mesmo consultá-los. Dessa forma, podemos então utilizar o módulo RFID e diferentes TAGs de identificação para obtermos o resultado esperado.

**Figura 2** – foto do módulo de proximidade para Arduino Kit Leitor RFID Rc522



Fonte: Montimport. Disponível em: <https://produto.mercadolivre.com.br/MLB-2051629589-modulo-de-proximidade-para-arduino-kit-leitor-rfid-rc522-> JM, 2024.

## 2.5 ESP32-CAM: Para foto capturas

O ESP32-CAM é um módulo de desenvolvimento baseado no chip ESP32 da Espressif. Em síntese, pode-se ser utilizado em sistema de segurança, sistema de monitoramento e sistema de controle. Dentre os detalhes do hardware encontra-se:

Chip ESP32: Microcontrolador Wi-Fi e Bluetooth de baixo consumo de energia. Ele oferece conectividade sem fio e suporte para comunicação com outros dispositivos.

Câmera OV2640: A câmera OV2640 é uma câmera de 2 megapixels com suporte para captura de imagens e vídeo. Ela é integrada ao módulo ESP32-CAM e permite a captura de fotos e streaming de vídeo.

Suporte a cartão microSD: O módulo possui um slot para cartão microSD, permitindo que você armazene imagens ou vídeos capturados diretamente no cartão.

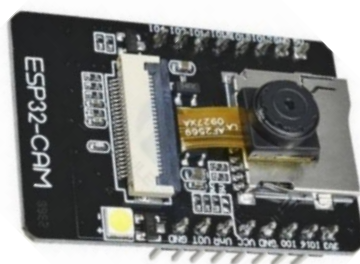
Flash onboard: O ESP32-CAM inclui um flash onboard que pode ser usado para iluminar cenas escuras durante a captura de fotos.

GPIOs: O módulo possui várias GPIOs (pinos de entrada/saída) que permitem conectar periféricos externos, como sensores, displays ou outros dispositivos que ampliam as possibilidades de uso.

Além disso, utiliza-se o programa Arduino IDE juntamente com a extensão ESP32 instalada no IDE. Concomitantemente, faz-se necessário conectar o ESP32-CAM ao computador usando um programador FTDI. Ademais, deve-se selecionar a placa correta (AI-Thinker ESP32-CAM) e a porta COM adequada na Arduino IDE.

No contexto do protótipo em questão, o ESP-32 tem como principal função capturar fotos dos funcionários quando eles solicitam o acesso a determinada área. Dessa forma será possível identificar casos de roubo de cartão ou invasores.

**Figura 3 – foto do ESP32-CAM-MB**



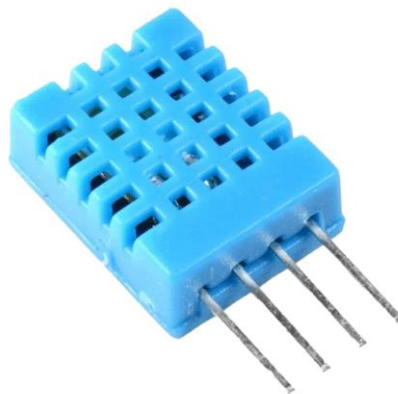
## 2.6 DHT-11: Sensor de temperatura e umidade

O DHT11 é um sensor utilizado para medir a temperatura e a umidade do ar. O seu funcionamento baseia-se em um sensor capacitivo de umidade e um termistor para medir a temperatura.

Quando alimentado, realiza leituras por meio de uma única linha de comunicação, tornando-o compatível com uma variedade de microcontroladores e placas de desenvolvimento. Ele pode medir temperaturas entre 0 a 50° Celsius com uma precisão de 2 graus, e umidade entre 20% a 90% com uma precisão de 5%.

O projeto utilizará o DHT11 para medir e registrar a temperatura ambiente quando algum usuário estiver no local e armazenar os dados no banco de dados do sistema. Dessa forma, os funcionários terão acesso às informações básicas sobre a qualidade do ambiente de trabalho apenas acessando o site.

**Figura 4** – foto do sensor de umidade e temperatura DHT11



Fonte: <https://a.co/d/9xUnRZJ>, 2024.

## 2.7 MySQL: Banco de Dados

O MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto, amplamente utilizado por desenvolvedores e empresas para armazenar, gerenciar e recuperar dados de forma eficiente. Ele foi desenvolvido pela empresa sueca MySQL AB, a qual foi adquirida pela Sun Microsystems e, posteriormente, pela Oracle Corporation. Ele é baseado na linguagem SQL (Structured Query Language), que é a linguagem padrão para a gestão de bancos de dados relacionais.

## 2.8 XAMPP

O XAMPP é uma distribuição de servidor web de código aberto que combina vários softwares em um único pacote, facilitando a instalação e execução de um ambiente de desenvolvimento local. O nome XAMPP é um acrônimo que representa seus componentes principais: X (cross-platform), A (Apache), M (MySQL/MariaDB), P (PHP) e P (Perl). Esse pacote é muito útil para criar e testar aplicações web localmente.

O XAMPP é amplamente utilizado para desenvolvimento web local, educação, teste de banco de dados, criação de protótipos e manutenção de ambientes de teste. O Apache serve como o servidor web, MySQL/MariaDB como sistema de gerenciamento de banco de dados, e as linguagens PHP e Perl são utilizadas para desenvolver páginas web dinâmicas.

A instalação do XAMPP é simples e seus serviços são gerenciados através do painel de controle incluído. Os arquivos do projeto são armazenados no diretório "htdocs", e podem ser acessados pelo navegador usando o endereço "http://localhost". Além disso, o XAMPP inclui o phpMyAdmin, uma ferramenta que facilita o gerenciamento de bancos de dados MySQL.

### **3. DESENVOLVIMENTO**

#### **3.1 Apresentando a ideia**

O projeto tem como objetivo fornecer auxílio com as seguintes funções:

Tornar a empresa num espaço mais seguro; registrar a entrada e saída de funcionários, tanto externamente quanto internamente na empresa, monitorando a circulação de pessoas no local apresentando seus dados após utilizarem o cartão de acesso; ter controle sobre a temperatura e a umidade ideal do ambiente; registrar fotograficamente o local de acesso restrito; armazenar informações sobre as pessoas que possuem o cartão de acesso.

#### **3.2 Metodologia**

Para realizar a pesquisa, utilizamos vários meios, como sites e vídeos em plataformas digitais, como o YouTube e Google. Essas fontes nos ajudaram a compreender melhor o assunto e sanar algumas dúvidas que tínhamos. Assim, o presente trabalho foi desenvolvido a partir do conhecimento que adquirimos durante o processo. Com isso, aprendemos a fazer uma conexão entre microcontroladores diferentes, como utilizar sensores, LCDs, e criar sites para o benefício do projeto.

No desenvolvimento do site e do banco de dados, utilizou-se de códigos já criados de CSS, de modo a encurtar o tempo de andamento de projeto e focar no funcionamento do projeto. As fontes utilizadas foram os canais do Youtube Gustavo Neitzke, InfoDesvenda Tecnologia, Curso em Vídeo, Rafaella Ballerini e Aprendendo do Início com Daniel. Os canais referidos, foram responsáveis pela estilização, a instalação de programas e o desenvolvimento do código para o projeto.

Na programação da tela inicial e na tela do cadastro do site foram utilizadas duas seções principais: a lógica de backend em PHP e a estrutura de frontend em HTML.

No código da página inicial, a lógica PHP inicializa quando um formulário é submetido através do método POST. Se a variável `$_POST['submit']` estiver definida, o script inclui o arquivo `conexao.php`, que contém as credenciais e a lógica de conexão ao banco de dados MySQL. Este mecanismo é crucial para estabelecer a comunicação entre a web e o banco de dados.

A estrutura HTML, construída com elementos semânticos e estilizada por um arquivo CSS externo (Rafaella Ballerini), define a interface do usuário. O cabeçalho (header) inclui uma imagem de logotipo e uma navegação com um link para a página de cadastro através de uma página de redirecionamento de tempo de 1 segundo (redic.html). O corpo principal (main) contém uma seção com descrições sobre o projeto Brain BS, destacando as principais informações, o objetivo e os componentes utilizados no projeto, como Arduino, ESP32-CAM e RFID. Este conteúdo é complementado por uma imagem do aparelho. Além disso, o uso de tags de metadados (meta) no <head> assegura a compatibilidade do documento com diferentes navegadores e dispositivos.

Na programação da tela de cadastro, o código PHP funciona de modo que os dados inseridos pelos usuários são capturados e armazenados no banco de dados MySQL. Quando o formulário é submetido, o script PHP verifica se o botão de envio (submit) foi clicado. Em caso afirmativo, ele inclui o arquivo de conexão ao banco de dados (conexao.php) e captura os dados do formulário através das variáveis \$\_POST. Em seguida, uma consulta SQL INSERT é executada para inserir esses dados na tabela usuarios do banco de dados.

A estrutura HTML da tela de cadastramento define o formulário de cadastro, incluindo campos para informações como nome, e-mail, celular, gênero, data de nascimento e endereço. Cada campo é estilizado com classes CSS (Gustavo Neitzke) e os dados são submetidos via método POST para serem processados pelo PHP. O cabeçalho da página inclui a logo da empresa e links de navegação, enquanto o corpo principal contém o formulário organizado em um <fieldset> com etiquetas e campos de entrada apropriados.

O código PHP conexão.php, o qual foi citado anteriormente, estabelece uma conexão com um banco de dados MySQL e permite operações como inserção, atualização e recuperação de dados.

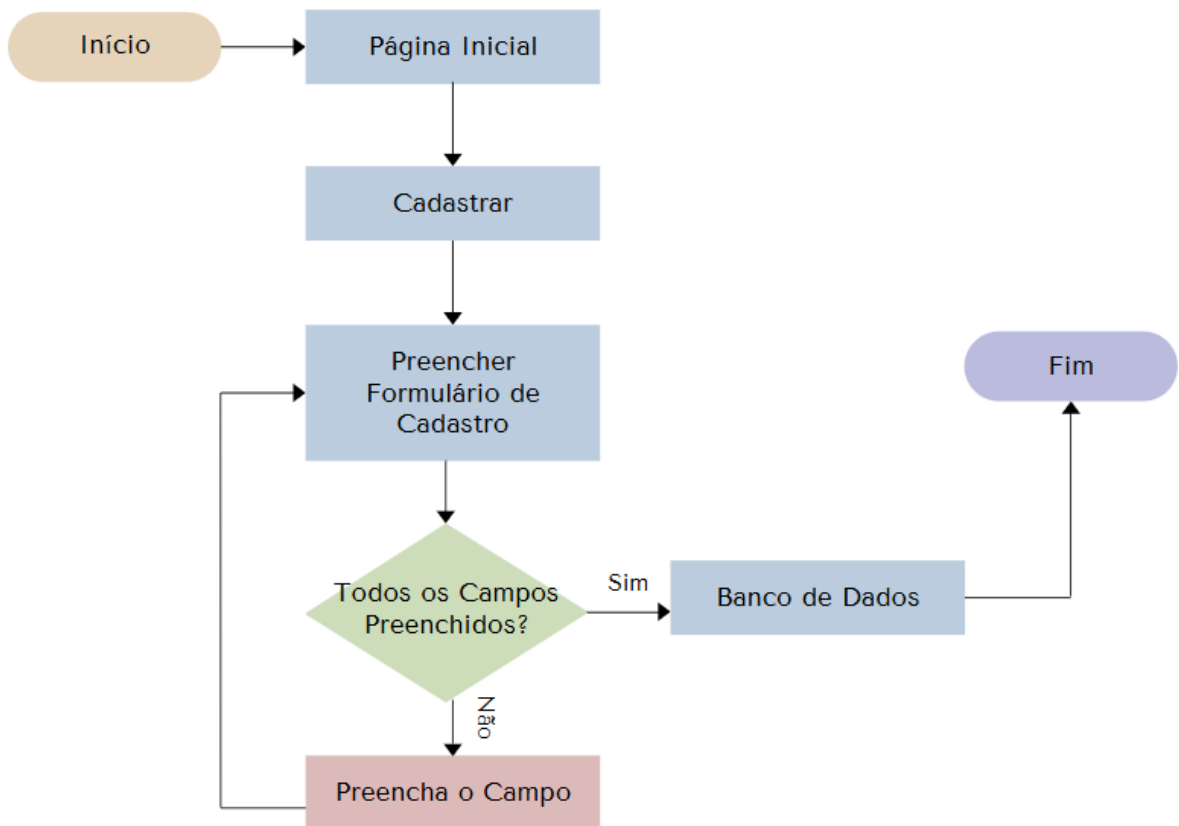
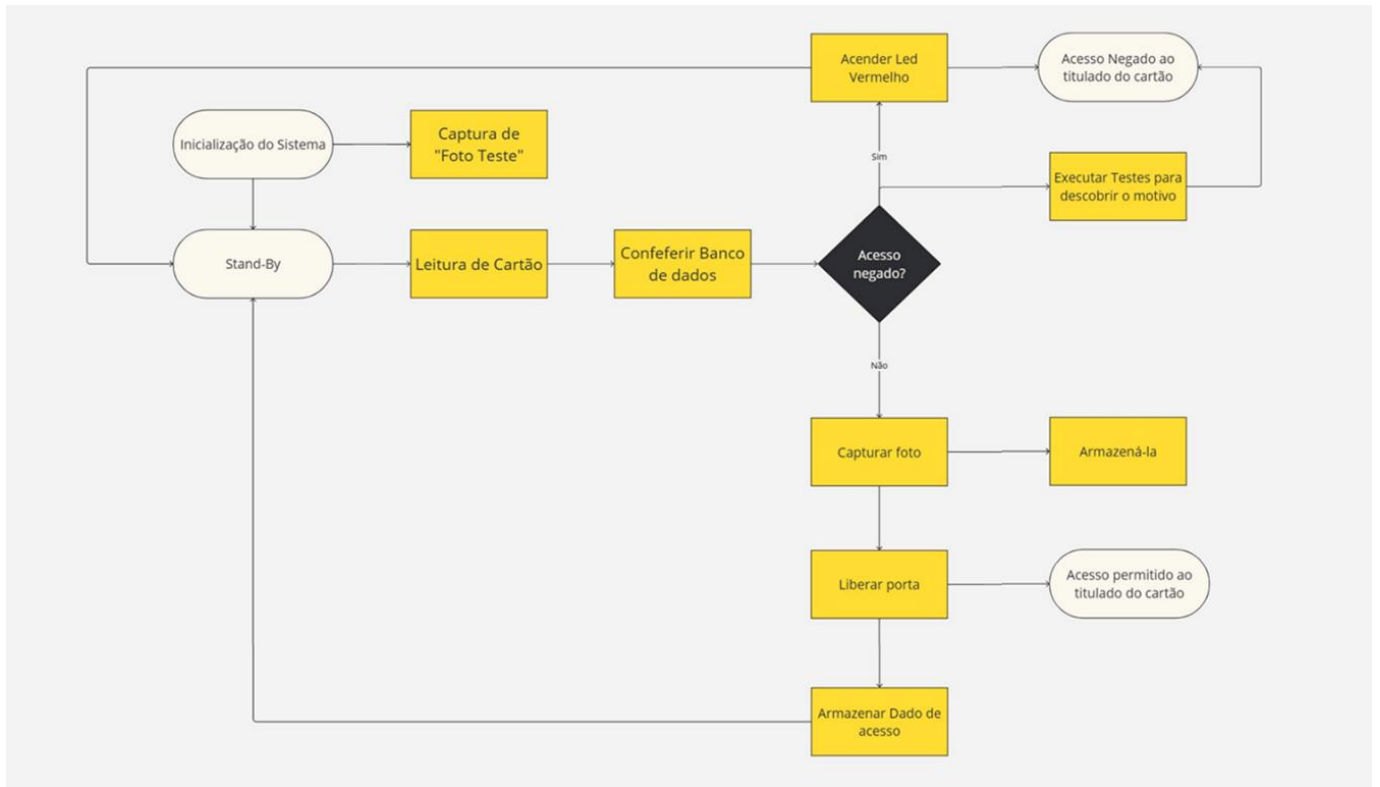
Primeiramente, são definidas variáveis com os parâmetros de conexão: `hostname` (nome do host, definido como "localhost" para desenvolvimento local), `banco\_dados` (nome do banco de dados), `usuario` (nome de usuário do banco de dados, aqui definido como "root") e `senha` (senha para o usuário, deixada vazia para desenvolvimento local).

A função `new mysqli` é utilizada para criar uma nova conexão ao banco de dados MySQL. Esta conexão é armazenada na variável `$conexao`, que é utilizada para executar consultas e interações com o banco de dados.

### 3.3 Tabela de custos

Item	Preço	Unidades	Total
Arduino Uno R3	R\$ 99,90	1	R\$ 99,90
LCD I2C	R\$ 32,00	1	R\$ 32,00
ESP 32 CAM	R\$ 89,90	1	R\$ 89,90
Jumpers M-M (40u)	R\$ 15,00	1	R\$ 15,00
RFID	R\$ 29,90	1	R\$ 29,90
Push buttons	R\$ 28,90	1	R\$ 28,90
DHT11	R\$ 16,90	1	R\$16,90
Protoboard	R\$18,00	1	R\$18,00
Led verde	R\$2,00	1	R\$2,00
Led vermelho	R\$2,00	1	R\$2,00
Resistores 330Ω	R\$0,60	3	R\$0,60
Placa de fenolite	R\$6,00	1	R\$6,00
Buzzer 5V	R\$4,00	1	R\$4,00
Cartão de memória	R\$35,00	1	R\$35,00
Fonte DC	R\$22,00	1	R\$22,00
FTDI	R\$23,00	1	R\$23,00
Cabo USB-B mini	R\$8,90	1	R\$8,90
Push Buttons	R\$6,00	3	R\$6,00
Resistores 1K	R\$0,60	3	R\$0,60
Câmera OV2640	R\$22,15	1	R\$22,15
Led branco	R\$1,00	1	R\$1,00
Cartões	R\$8,00	2	R\$8,00
Jumpers M-F	R\$15,00	40	R\$15,00
TOTAL			R\$454,25

### 3.4 Lógica



### 3.5 Tecnologias Semelhantes

O alarme de segurança Verisure, tem como base um sistema de proteção residencial que funciona contra invasões a propriedades privadas. Nesse projeto, foram implementados componentes que interagem de forma síncrona entre si. Dessa forma, a câmera que fotografa e armazena imagens no banco de dados é utilizada juntamente com o cartão de acesso, permitindo, ou não, o ingresso na empresa.

No presente trabalho, um sistema de segurança eficiente foi feito utilizando Arduino, cuja plataforma sendo a base principal do projeto, RFID, leitor de frequências para averiguar o acesso de cada usuário, e ESP32-CAM, parte fotográfica do protótipo. No sistema utilizado, o sensor quando identificar a passagem de um funcionário dentro do local, irá assimilar sua imagem com as informações contidas no cartão. Assim, analisando se é realmente seguro sua entrada na empresa, evitando acessos não autorizados e possíveis falhas na segurança.

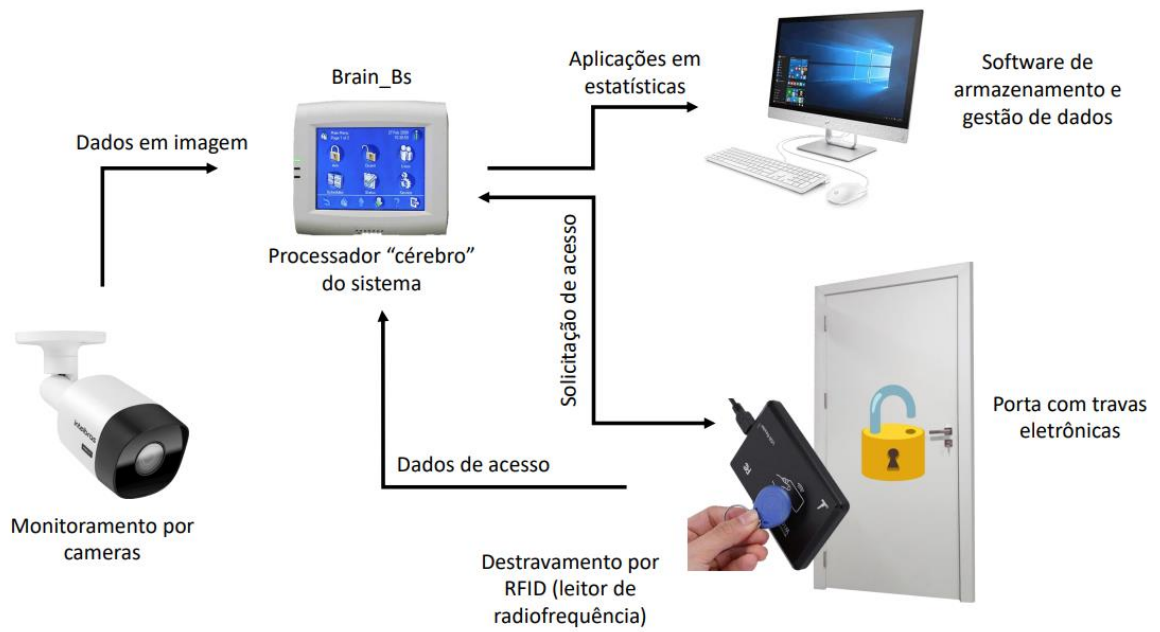
Em relação ao alarme de segurança Verisure, visamos desenvolver um projeto que obtivesse mais qualidades e funcionalidades que ele, sendo uma opção favorita entre os interessados em equipamentos relacionado a esta área de segurança.

**Figura 5** – Propaganda da Verisure



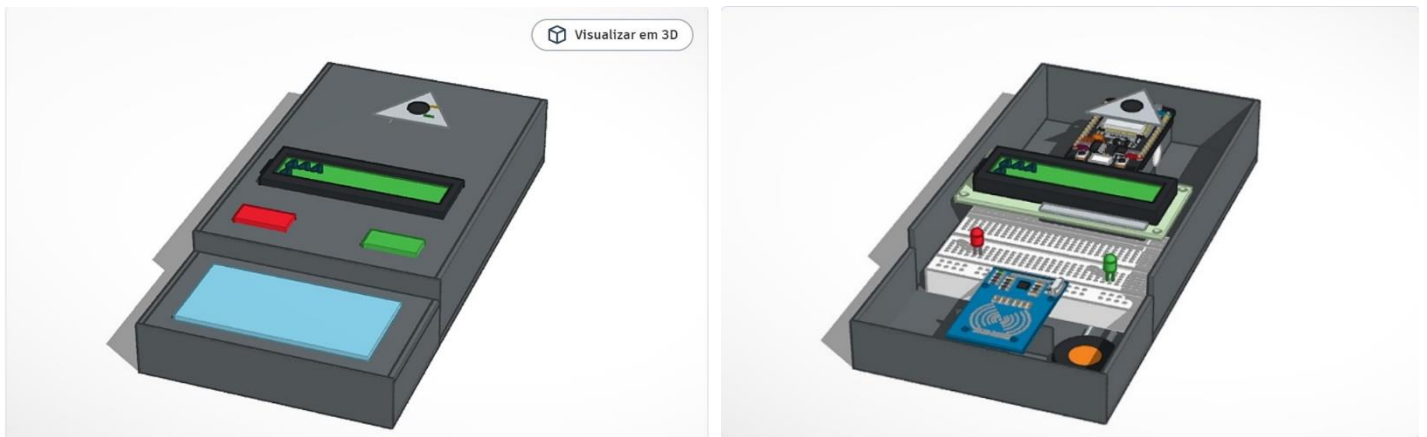
Fonte: [www.verisure.com.br](http://www.verisure.com.br), 2024.

### 3.6 Sistema idealizado



Fonte: Os autores, 2024

### 3.7 Sistema oficial



Fonte: Os Autores, 2024

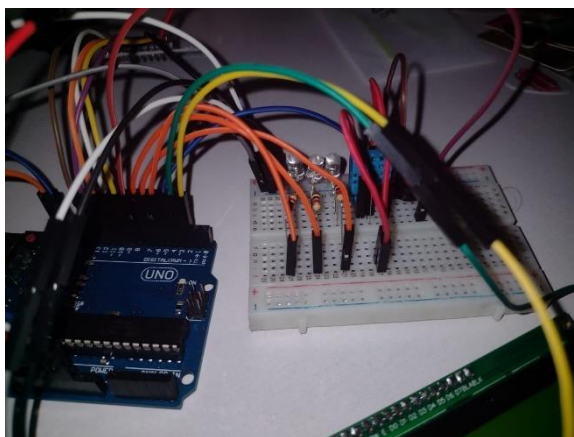
## 4 RESULTADOS

### 4.5 Resultados obtidos

Conforme foi se avançando no projeto, conseguimos atingir o objetivo de montar um sistema de monitoramento semelhante aos que são utilizados atualmente, porém, com algumas diferenças e características únicas, sendo assim, mais acessível. Sua construção foi realizada envolta do Arduino, levando como base a programação C++, sendo ela uma programação simples mesmo que extensa.

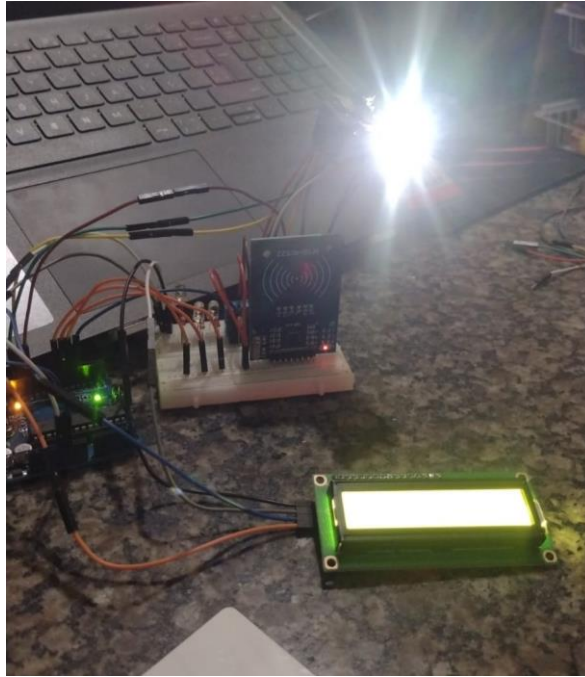


O primeiro protótipo teria como o uso a placa Raspberry Pi 3 Model B+, que permite a conexão wifi da placa Raspberry Pi com o microcontrolador Arduino. Entretanto, devido à dificuldade de realizar ligações, ao uso de linguagens de programação complexas por parte da placa Raspberry Pi, e por conta do alto custo de produção, o protótipo foi descartado logo em seguida. Consistiria em substituir o ESP32-CAM e o Arduino Uno simultaneamente.



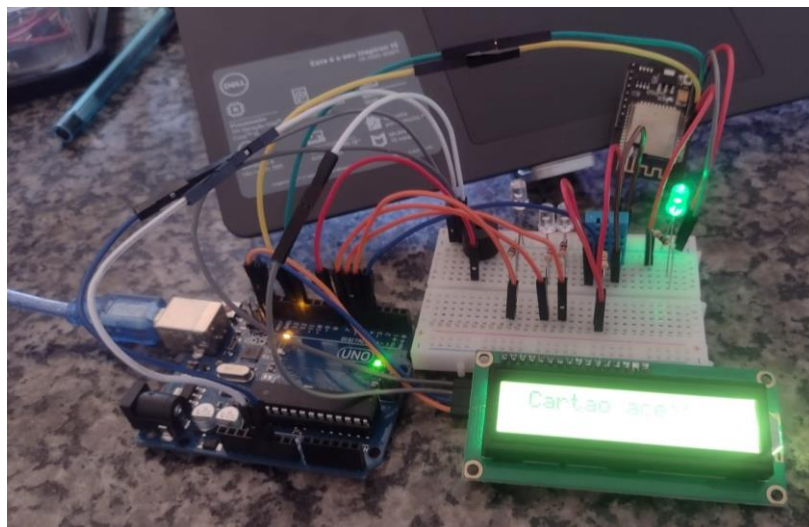
Fonte: Os Autores, 2024.

As primeiras ideias do protótipo foram realizadas em testes presenciais na placa protoboard, sendo sua construção realizada envolta do Arduino, levando como base a programação C++, sendo ela uma programação simples mesmo que extensa. No dia 2 de outubro de 2024, a primeira montagem do protótipo já havia sido feita.



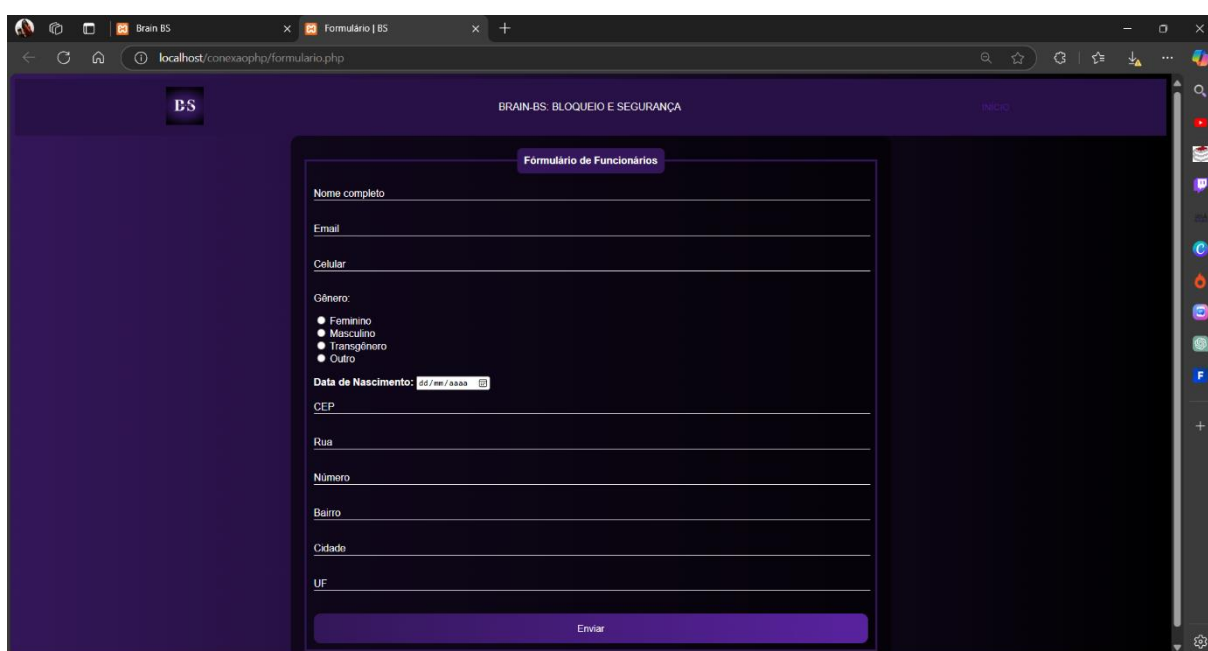
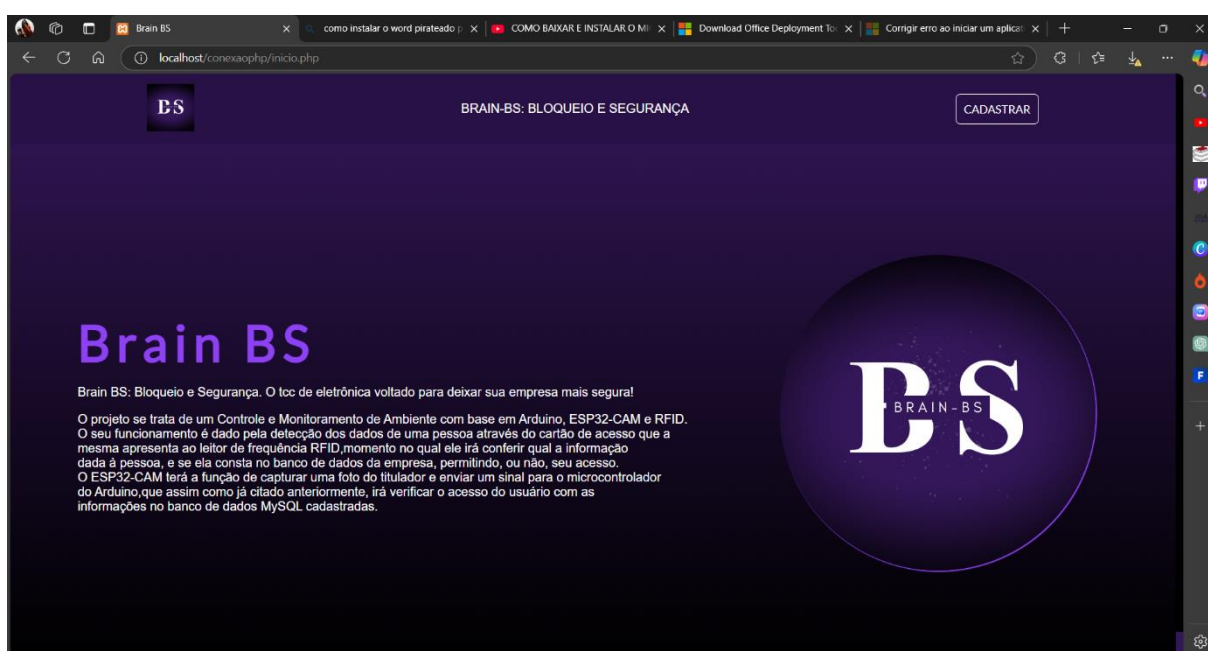
Fonte: Os Autores, 2024.

O projeto foi realizado com a ajuda de um computador e testado sem uma maquete. Nos resultados, a utilização do RFID se provou útil, com uma boa detecção da frequência emitida pela aproximação dos cartões de acesso, e logo era identificado se tinham permissão de acesso ou não de forma correta.

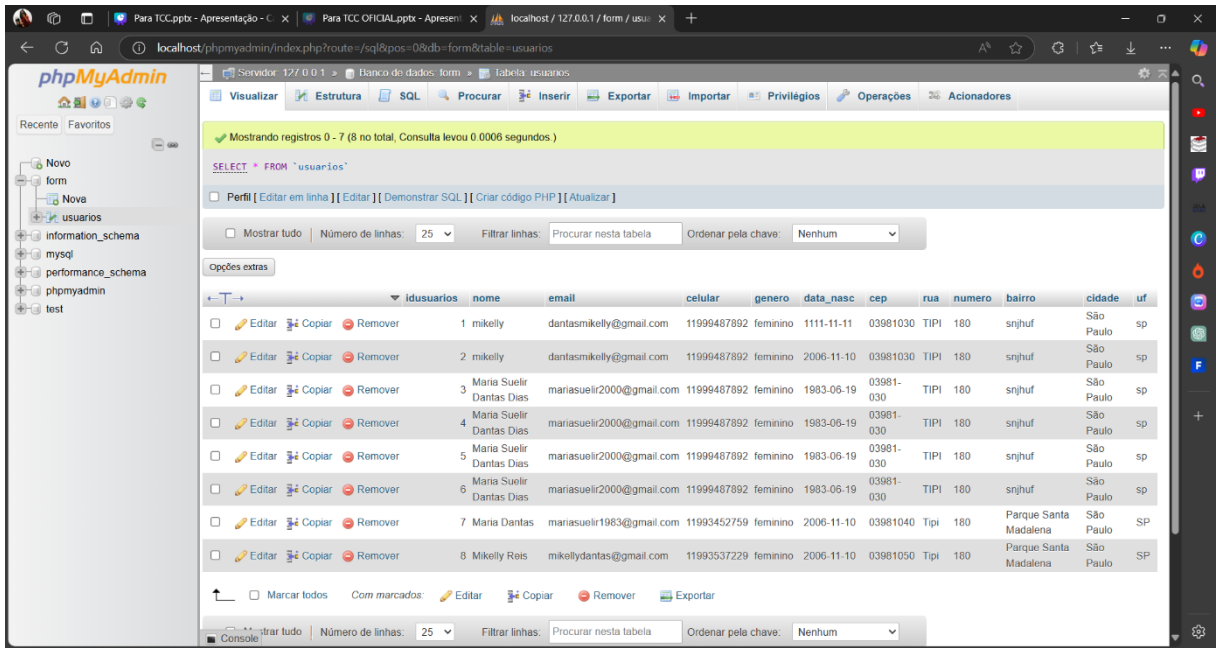


Fonte: Os Autores, 2024.

O site, o qual foi desenvolvido a partir do editor de código-fonte Visual Studio Code da Microsoft, utilizando-se das linguagens de marcação HTML e CSS, além das linguagens de programação JavaScript e PHP, provou-se útil a partir do armazenamento de dados dos funcionários que terão o acesso as áreas restritas. A utilização do banco de dados MySQL foi problemática devido à problemas na instalação de programa e da ligação com o servidor de código aberto XAMPP, entretanto, depois das descobertas sobre o que impedia o funcionamento, foi possível enviar os dados do site do formulário para o banco de dados. Desse modo, foi possível fazer o sistema web funcionar corretamente.



Fonte: Os Autores, 2024.



Fonte: Os Autores, 2024.

O protótipo final foi revestido através de uma carcaça de madeira pintada com tinta própria para MDF



Fonte: Os Autores, 2024.

## 5 CONSIDERAÇÕES FINAIS

Desenvolvemos o projeto com a finalidade de ser um produto para pequenas e grandes empresas que possibilite a existência de um ambiente bem monitorado, organizado, e seguro, criando uma experiência confortável tanto para os funcionários quanto para os proprietários da empresa, conseqüentemente elevando o nível de produtividade. Trazendo o ambiente empresarial e industrial e resolvendo problemas que o sistema atual utiliza, esse trabalho aperfeiçoou alguns protótipos semelhantes e deixou seu custo mais acessível.

O fato do baixo custo do protótipo tem a vantagem da facilidade de montar o projeto em escala maior do que outros métodos para esse sistema, como trocando os sensores por câmeras com inteligência artificial.

Durante o desenvolvimento desse projeto foi encontrado algumas dificuldades, sendo elas, problemas na comunicação serial do ESP32-CAM com o Arduino, conexão entre hardware e software, a compra de alguns materiais que demoraram para chegar e a conexão do banco de dados ao site.

Todos os objetivos do projeto foram alcançados, que foram: Fazer uma interconexão entre os componentes, possibilitando a captura de imagens pela câmera, a leitura de frequências eficiente do RFID, que permite a leitura de cada cartão apresentado, o armazenamento de dados no banco de dados MySQL, que fez ser possível o cadastro de informações, e a integração de um sistema que se comunica com o espaço em que o projeto irá atuar, permitindo uma dinâmica fluida e segura.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

**O que é HTML? [Guia para iniciantes].** Disponível em: <<https://tecnoblog.net/responde/o-que-e-html-guia-para-iniciantes/>>.

LAST MINUTE ENGINEERS. **What is RFID? How It Works? Interface RC522 RFID Module with Arduino.** Disponível em: <<https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>>.

**O que é CSS [Cascading Style Sheets]?** Disponível em: <<https://tecnoblog.net/responde/o-que-e-css-cascading-style-sheets/>>.

**O que é JavaScript? [Guia para iniciantes].** Disponível em: <<https://tecnoblog.net/responde/o-que-e-javascript-guia-para-iniciantes/>>.

TECNOLOGIA. **O que é PHP? Guia para iniciantes.** Tecnoblog, 2024. Disponível em: <<https://tecnoblog.net/responde/o-que-e-php-guia-para-iniciantes/>>.

**In-Depth: What is RFID? How It Works? Interface RC522 with Arduino.** Disponível em: <[https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/#google\\_vignette](https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/#google_vignette)>.

ALLDATASHEET.COM. **MFRC522 Datasheet(PDF).** Disponível em: <<https://www.alldatasheet.com/datasheet-pdf/pdf/227839/NXP/MFRC522.html>>.

Acesso em: 25 out. 2024.

SRC='HTTPS://SECURE.GRAVATAR.COM/AVATAR/8C3C30FE493EB74C4F8DFE F25766088C?S=42, IMG A. et al. **Interfacing MFRC522 RC522 RFID Reader Module with Arduino.** Disponível em: <<https://www.electronicshub.org/arduino-rc522-rfid-module/>>.

ADMIN. **MFRC522 RFID Sensor With Arduino Tutorial.** Disponível em: <[https://diyprojectslab.com/mfrc522-rfid-module-with-arduino/#google\\_vignette](https://diyprojectslab.com/mfrc522-rfid-module-with-arduino/#google_vignette)>.

Acesso em: 25 out. 2024.

BANCADA SKIP UP. **Arduino com Banco de Dados MySQL, RFID, Ethernet Shield e JSF.** Disponível em: <<https://www.youtube.com/watch?v=Al54cJAFY6o>>. Acesso em: 25 out. 2024.

PROFESSOR NETO PASCHOAL. **Criando um Web Server com ESP32.** Disponível em: <<https://www.youtube.com/watch?v=ZSyqNFGAF8o>>. Acesso em: 25 out. 2024.

**Arduino and ESP32 Serial Communication || UART Communication.** Disponível em: <[https://www.youtube.com/watch?v=\\_A2KwuiE7Ww](https://www.youtube.com/watch?v=_A2KwuiE7Ww)>. Acesso em: 1 jun. 2023.

ELETROGATE. **Introdução ao ESP32-CAM.** Disponível em: <<https://blog.eletrogate.com/introducao-ao-esp32-cam/>>.

**Medindo temperatura e umidade usando o sensor DHT11 – Blog da Robótica.** Disponível em: <<https://www.blogdarobotica.com/2020/10/29/medindo-temperatura-e-umidade-usando-o-sensor-dht11/>>.

YouTube. **LANDING PAGE com HTML e CSS.** Disponível em: <<https://youtu.be/IIF6vD-RlJE?si=G8VcGWkeV3ZP2PFv>>.

YouTube. **Instalando MySQL e Workbench.** Disponível em: <<https://youtu.be/s0YoPLbox40?si=ijCyHXNkaT7iAA9t>>.

YouTube. **Aprenda como ligar um FORMULÁRIO com BANCO DE DADOS #01.** Disponível em: <<https://youtu.be/QOeDE7nPDq0?si=hXbRBmZWY0Jn2UEf>>.

YouTube. **APRENDA como criar um FORMULÁRIO com HTML e CSS.** Disponível em: <<https://youtu.be/VCsNIRXNsmY?si=oKi6ALeM7fWgc0BK>>.

SOUZA, F. **Como programar o ESP32 na Arduino IDE?** Disponível em: <<https://embarcados.com.br/como-programar-o-esp32-na-arduino-ide/>>.

ROBOCORE. **ESP32-CAM - ESP32 com Câmera.** Disponível em: <<https://www.robocore.net/wifi/esp32-cam-esp32-com-camera>>.

YELLOW PURPLE. **[SOLVED] ESP32CAM Error - Failed to connect to ESP32 Cam || Problems Uploading Code to ESP32 Camera.** Disponível em: <<https://www.youtube.com/watch?v=lqNRj4kQCcA>>. Acesso em: 25 out. 2024.

LOBO DA ROBÓTICA. **Como INSTALAR a ESP32 na IDE do Arduino? (Método DEFINITIVO).** Disponível em: <<https://www.youtube.com/watch?v=Ozr1SXwir9g>>.

Acesso em: 25 out. 2024. **ESP32 Explained In 1 Minute! #IoT #robonyx #electronics.** Disponível em: <<https://www.youtube.com/shorts/XTm-YL-5G34>>. Acesso em: 25 out. 2024.

ENGENHEIRO DE PLANTÃO. **Tutorial TinkerCAD Português Como modelar 3D com o TinkerCAD Introdução.** Disponível em: <<https://www.youtube.com/watch?v=L4exjlrXsNw>>. Acesso em: 25 out. 2024.

BRINCANDO COM IDEIAS. **O que é Tinkercad? Transforme as suas ideias em realidade!** Disponível em: <<https://www.youtube.com/watch?v=WTQ87oo9LsA>>. Acesso em: 25 out. 2024.

**DHT11 - SENSOR DE UMIDADE E TEMPERATURA | Curso de Arduino #109.** Disponível em: <<https://www.youtube.com/watch?v=onVxqTngED8>>. Acesso em: 9 abr. 2024.

GUSE, R. **Câmera IP: Cuide do seu bebê com ESP32-CAM - MakerHero.** Disponível em: <<https://www.makerhero.com/blog/esp-32-camera-ip/>>. Acesso em: 25 out. 2024.

**Como abrir sua porta com o cartão do ônibus #ManualMaker Aula 8, Vídeo 2.** Disponível em: <[https://www.youtube.com/watch?v=l\\_4cJWRUbnU](https://www.youtube.com/watch?v=l_4cJWRUbnU)>. Acesso em: 29 nov. 2022.

SATYAM SINGH. **RFID Simulation in proteus with Arduino.** Disponível em: <<https://www.youtube.com/watch?v=kWaYomvnpUU>>. Acesso em: 25 out. 2024.

ALAN MELLO. **Simulador de Leitor RFID (Servidor Socket) em Java.** Disponível em: <<https://www.youtube.com/watch?v=NZP1ECX7roY>>. Acesso em: 25 out. 2024.

**Como usar uma identificação por rádio #ManualMaker Aula 8, Vídeo 1.** Disponível em: <[https://www.youtube.com/watch?v=gcBN4NLqz\\_U](https://www.youtube.com/watch?v=gcBN4NLqz_U)>. Acesso em: 21 set. 2022.

SCHEMATIC WORLD. **ESP32 Library in Proteus: Everything You Need to Know.** Disponível em: <<https://www.youtube.com/watch?v=Rm-4W0hP-dY>>. Acesso em: 25 out. 2024.

THAT PROJECT. **USB Camera to ESP32-S2 (UVC Camera).** Disponível em: <<https://www.youtube.com/watch?v=4l8ohXMIhjs>>. Acesso em: 25 out. 2024.

**RFID door lock access control system | How to make an RFID door lock system using Arduino.** Disponível em: <<https://www.youtube.com/watch?v=GOO84CGBPz8>>. Acesso em: 6 jun. 2023.

**How to Make Fingerprint Door Lock | Arduino Project.** Disponível em: <<https://www.youtube.com/watch?v=gpXuEghz1zc>>. Acesso em: 20 mar. 2023.

TECH STUDYCELL. **ESP32-CAM Motion Sensor Security Camera with Notification using Blynk - DIY Home surveillance system.** Disponível em: <<https://www.youtube.com/watch?v=LqX9EMFSODA>>. Acesso em: 25 out. 2024.

INTERNET E COISAS. **ESP32-CAM - Introdução - leC112.** Disponível em: <<https://www.youtube.com/watch?v=JPgmergAnd4I>>. Acesso em: 25 out. 2024.

**Como usar com Arduino - KIT RFID MFRC522 - BLOG MASTERWALKER SHOP.** Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-kit-rfid-mfrc522>>.

BANCADA SKIP UP. **Arduino com Banco de Dados MySQL, RFID, Ethernet Shield e JSF.** Disponível em: <<https://www.youtube.com/watch?v=Al54cJAFY6o>>. Acesso em: 25 out. 2024.

**ESP32-CAM HTTP Post Images/Photos to Server | Random Nerd Tutorials.**

Disponível em: <<https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/>>.

**ESP32-CAM Take Photo and Save to MicroSD Card | Random Nerd Tutorials.**

Disponível em: <<https://randomnerdtutorials.com/esp32-cam-take-photo-save-microsd-card/>>.

**How to Program / Upload Code to ESP32-CAM AI-Thinker (Arduino IDE) |**

**Random Nerd Tutorials.** Disponível em: <<https://randomnerdtutorials.com/program-upload-code-esp32-cam/>>.

**DE. Principais problemas de segurança enfrentados pelas empresas.** Disponível

em: <<https://g1.globo.com/sp/presidente-prudente-regiao/especial-publicitario/security-seguranca-e-servicos/noticia/2023/01/31/principais-problemas-de-seguranca-enfrentados-pelas-empresas.ghtml>>. Acesso em: 25 out. 2024.

**Sistema de alarmes: Como funciona?** Disponível em:

<<https://www.verisure.com.br/blog/sistema-de-alarmes-como-funciona>>. Acesso em: 25 out. 2024.

## **7 APÊNDICE**

### **7.5 Diário de bordo**

08/02/2024

Decidimos qual projeto iríamos executar (Sistema de Segurança)

15/02/2024

Definimos os integrantes permanentes do grupo.

22/02/2024

Criamos o cronograma e definimos a logotipo do projeto.

29/02/2024

Dedicamos o dia a fazer pesquisas e evoluir com o projeto.

21/03/2024

Definimos o orçamento do projeto e dividimos a quantia de pagamento para cada componente.

28/03/2024

Definimos a estrutura base do projeto e estamos ensaiando para a apresentação que seria feita em breve.

04/04/2024

Apresentamos o protótipo do projeto para a sala e os professores.

09/04/2024

Realizamos a solda do RFID.

11/04/2024

Compramos e testamos os componentes e materiais necessários.

18/04/2024

Nos reunimos e conversamos sobre como faríamos a comunicação entre o hardware e o software.

02/05/2024

Concluimos a fundamentação e o referencial teórico.

16/05/2024 – 30/05/2024

Realizamos testes referentes a comunicação entre os componentes.

13/06/2024

Apresentamos nossos avanços aos professores e recebemos orientações de como progredir ainda mais.

01/08/2024 – 05/09/2024

Trabalhamos duro para conseguir fazer o ESP32-CAM funcionar, e conseguimos ligar um Led através de um site.

30/09/2024

Obtivemos êxito na comunicação serial.

03/10/2024

Realizamos testes com o protótipo, cujo o mesmo estava apresentando falhas no funcionamento.

10/10/2024

Fizemos a tabela do banco de dados, mas não foi possível fazer a conexão com o site.

18/10/2024

Não foi possível resolver a conexão do banco de dados com o site.

25/10/2024

Descobrimos que o problema de conexão estava sendo causado por falta do XAMPP. Fizemos a instalação do programa, entretanto, o problema não foi resolvido devido a falhas de conexão entre o XAMPP e o MySQL.

06/11/2024

Exploramos a possibilidade de fazer o design do projeto em 3D.

08/11/2024

Alteramos a estética da tela inicial do site. Tentamos resolver o problema com o banco de dados, mas não foi possível.

13/11/2024

Tentamos fazer a conexão a partir de outro notebook, entretanto, apresentou as mesmas falhas.

15/11/2024

Resolvemos o problema do erro da conexão do banco de dados com o site. Utilizamos o MySQL a partir do <http://localhost> com a utilização da ferramenta que facilita o gerenciamento de bancos phpMyAdmin.

17/11/2024

Descobrimos que, para a programação funcionar, corretamente, junto com o banco de dados, os arquivos do projeto deveriam ser armazenados no diretório "htdocs". Logo, fizemos a mudança das pastas dos arquivos do projeto.

22/11/2024

Terminamos a parte teórica do projeto. Entretanto, a conexão entre a página inicial e a página de cadastro começaram a dar erro.

26/11/2024

Descobrimos o que estava impedindo a conexão das pastas, resolvemos e finalizamos a parte web e do banco de dados do projeto.

## ANEXOS

### CÓDIGO: style.css

```
@import url('https://fonts.googleapis.com/css2?family=Lato');

:root{
  --white: #fff;
  --dark1: #34175a;
  --dark2: #34175a;
  --dark3: #000000;
  --dark4: #000000;
  --dark5: #8f41f5;
  --cyan1: #8f41f5;
}
*{
  margin: 0;
  padding: 0;
}
::-webkit-scrollbar {
  width: 10px;
}
::-webkit-scrollbar-track {
  background: var(--dark2);
}
::-webkit-scrollbar-thumb {
  background: var(--dark3);
}
::-webkit-scrollbar-thumb:hover {
  background: var(--dark4);
}
body{
  background-color: var(--dark1);
  background-image: linear-gradient(#34175a, rgb(0, 0, 0)); /* cor preta desfocada */
  background-size: 400%;
  height: 100vh;
  font-family: 'Lato';
}
header{
  display: flex;
  justify-content: space-around;
  align-items: center;
  gap: 1rem;
  flex-wrap: wrap;
  padding: .8rem;
  background-color: #281147;
  min-height: 50px;
}
.box-img-header{
  height: 100%;
}
.img-logo{
  height: 100%;
  max-height: 60px;
}
nav{
  display: flex;
  gap: 2rem;
  justify-content: center;
```

```

        align-items: center;
        flex-wrap: wrap;
    }
    .nav-links{
        text-decoration: none;
        color: var(--white);
    }
    .nav-links:hover{
        color: var(--dark4);
        text-decoration: underline;
        transition: .4s;
    }
    .box-btn-actions{
        display: flex;
        gap: 1rem;
    }
    :is(button,.link-action){
        padding: .6rem;
        font-size: .9rem;
        border-radius: 6px;
        cursor: pointer;
        transition: .4s;
    }
    .btn-action{
        background-color:
transparent;
        border: 1px solid var(--
white);
        color: var(--white);
        text-decoration: none;
        padding: .6rem;
        font-size: .9rem;
        border-radius: 6px;
        cursor: pointer;
        transition: .4s;
    }
    .btn-action:hover{
        background-color: var(--
dark3);
        border: 1px solid var(--
dark3);
    }
    .link-action{
        text-decoration: none;
        width: 200px;
        text-align: center;
    }
    main{
        display: flex;
        flex-direction: column;
        gap: 2rem;
    }
    section{
        padding: 1rem;
    }
    .first-section{
        display: flex;
        justify-content: space-
evenly;
        align-items: center;
        gap: 2rem;
        min-height: 650px;
        flex-wrap: wrap;
    }
    .first-section > div{
        display: flex;

```

```

    flex-direction: column;
    gap: 1rem;
    min-width: 300px;
}
h1{
    font-size: 4rem;
    letter-spacing: 8px;
    color: var(--white);
}
strong{
    color: var(--cyan1);
}
.first-section p{
    color: var(--white);
}
.first-section button{
    width: 200px;
}
.box-img-main{
    max-width: 400px;
    padding: 1rem;
}
.img-profile{
    width: 100%;
    border-radius: 50%;
}
:is(.second-section, .fourth-
section){
    background-color: var(--
dark2);
    display: flex;
    flex-direction: column;
    gap: 1rem;
    align-items: center;
}
.title{
    color: var(--dark5);
    font-size: 200%;
}
.second-section div{
    display: flex;
    justify-content: center;
    align-items: center;
    gap: 2rem;
    flex-wrap: wrap;
}
.second-section p{
    color: var(--dark4);
    font-weight: bold;
    letter-spacing: 2px;
}
.second-section p:hover{
    color: var(--dark5);
    transition: .4s;
    cursor: default;
}
.third-section{
    display: flex;
    flex-direction: column;
    gap: 2rem;
}
.third-section > div{
    display: flex;
    align-items: center;
    justify-content: center;
    gap: 4rem;

```

```

        flex-wrap: wrap;
    }
    .third-section > div img{
        width: 250px;
        border-radius: 10px;
    }
    .third-section > div p{
        max-width: 600px;
        color: var(--white);
    }
    #my-projects-list{
        display: flex;
        gap: 2rem;
        padding: 2rem;
        align-items: center;
        justify-content: center;
        flex-wrap: wrap;
    }
    #my-projects-list a{
        text-decoration: none;
        color: var(--dark3);
    }
    #my-projects-list a:hover{
        transition: .4s;
        color: var(--dark4);
    }
    :is(.img-profile,.img-who-am-
i){
        box-shadow: 1px 1px 1px 1px
var(--cyan1);
    }
    footer{
        padding: 4rem;
        display: flex;
        flex-direction: column;
        gap: 2rem;
    }
    footer > div{
        display: flex;
        justify-content: center;
        gap: 2rem;
        flex-wrap: wrap;
    }
    .text-formatation-center {
        font-family: Arial;
        text-align: center;
        font-size: 125%;
        word-spacing: 200%;
        letter-spacing: 100%;
        margin-left: 10%;
        margin-right: 10%;
    }
    .text-formatation-left {
        font-family: Arial;
        text-align: left;
        font-size: 100%;
        word-spacing: 200%;
        letter-spacing: 100%;
        margin-left: 0%;
        margin-right: 0%;
        height:auto;
        width: auto;
    }

```

**CÓDIGO: styleformulario.css**

```

@import
url('https://fonts.googleapis.c
om/css2?family=Lato');

body {
  font-family: Arial,
Helvetica, sans-serif;
  background-image: linear-
gradient(to right, #34175a,
rgb(0, 0, 0));
}
header {
  display: flex;
  justify-content: space-
around;
  align-items: center;
  gap: 1rem;
  flex-wrap: wrap;
  padding: .8rem;
  background-color: #281147;
  min-height: 50px;
}
.box-img-header {
  height: 100%;
}
.img-logo {
  height: 100%;
  max-height: 60px;
}
nav {
  display: flex;
  gap: 2rem;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
}
.nav-links {
  text-decoration: none;
  color: var(#ffffff);
}
.nav-links:hover {
  color: var(#000000);
  text-decoration: underline;
  transition: .4s;
}
.box-btn-actions {
  display: flex;
  gap: 1rem;
}
:is(button, .link-action) {
  padding: .6rem;
  font-size: .9rem;
  border-radius: 6px;
  cursor: pointer;
  transition: .4s;
}
.btn-action {
  background-color:
transparent;
  border: 1px solid
var(#ffffff);
  color: var(#ffffff);
  text-decoration: none;
  padding: .6rem;

```

```

    font-size: .9rem;
    border-radius: 6px;
    cursor: pointer;
    transition: .4s;
}
.btn-action:hover {
    background-color:
var(#000000);
    border: 1px solid
var(#000000);
}
.link-action {
    text-decoration: none;
    width: 200px;
    text-align: center;
}
main {
    display: flex;
    flex-direction: column;
    gap: 2rem;
}
.box-img-main {
    max-width: 400px;
    padding: 1rem;
}
.img-profile {
    width: 100%;
    border-radius: 50%;
}
.box {
    color: white;
    position: relative;
    top: 50%;
    left: 50%;
    transform: translate(-50%,
0%);
    background-color: rgba(0,
0, 0, 0.6);
    padding: 20px;
    border-radius: 15px;
    width: 50%;
}
fieldset {
    border: 3px solid #34175a;
}
legend {
    border: 1px solid #34175a;
    padding: 10px;
    text-align: center;
    background-color: #34175a;
    border-radius: 8px;
}
.inputBox {
    position: relative;
}
.inputUser {
    background: none;
    border: none;
    border-bottom: 1px solid
white;
    outline: none;
    color: white;
    font-size: 15px;
    width: 100%;
    letter-spacing: 2px;
}

```

```

.labelInput {
  position: absolute;
  top: 0px;
  left: 0px;
  pointer-events: none;
  transition: .5s;
}
.inputUser:focus~.labelInput,
.inputUser:valid~.labelInput {
  top: -20px;
  font-size: 12px;
  color: #34175a;
}
#data_nascimento {
  border: none;
  padding: 8px;
  border-radius: 10px;
  outline: none;
  font-size: 15px;
}
#submit {
  background-image: linear-
gradient(to right, #34175a,
rgb(89, 35, 159));
  width: 100%;
  border: none;
  padding: 15px;
  color: white;
  font-size: 15px;
  cursor: pointer;
  border-radius: 10px;
}
#submit:hover {

```

```

background-image: linear-
gradient(to right, #34175a,
rgb(89, 35, 159));
}
footer {
  padding: 4rem;
  display: flex;
  flex-direction: column;
  gap: 2rem;
}
footer>div {
  display: flex;
  justify-content: center;
  gap: 2rem;
  flex-wrap: wrap;
}

```

### CÓDIGO: redic.html

```

<!-- redic.html -->
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">

  <title>Redirecionando</title>
  <script>
    setTimeout(function() {
window.location.href =
"formulario.php";
    }, 1000); // 1 segundos
  </script>

```

```

</head>
<body>
  <p>Você será redirecionado
em 1 segundo...</p>
</body>
</html>

```

### CÓDIGO: main.js

```

function getProjects(){
  const urlGitHub =
'https://api.github.com/users/G
utoneitzke/repos'
  var loadingElement =
document.getElementById('loadin
g')
  fetch(urlGitHub,{
    method: 'GET',
  })
  .then((response) =>
response.json())
  .then((response) => {
console.log(response)
showProjects(response)
loadingElement.style.display =
'none'
  })
  .catch((e) => {
    console.log(`Error
-> ${e}`)
  })
}
function showProjects(data){

```

```

  var listElement =
document.getElementById('my-
projects-list')
  for(let i = 0; i <
data.length; i++)
  {
    let div =
document.createElement("div")
    let a =
document.createElement("a")
    a.href =
data[i]['clone_url']
    a.target = '_blank'
    a.title =
data[i]['description']
    let linkText =
document.createTextNode(data[i]
['name']);
    a.appendChild(linkText);
    div.appendChild(a)
    listElement.appendChild(div)
  }
}
getProjects()

```

### CÓDIGO: conexao.php

```

<?php
$hostname = "LocalHost";
$bancodedados = "form";
$usuario = "root";

```

```

$senha = "";
$conexao = new
mysqli($hostname, $usuario,
$senha, $bancodedados);
// if ($conexao->connect_errno)
{
//     echo "falha ao
conectar:(" . $conexao-
>connect_errno . ")" .
$conexao->connect_errno;
//}
//else {
//     echo "Conectado ao Banco
de Dados";
//}
?>

```

### CÓDIGO: inicio.php

```

<?php
if (isset($_POST['submit']))
{
    include_once('conexao.php');
}
?>
<!DOCTYPE html>
<html lang="br">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-
Compatible" content="IE=edge">

```

```

    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet"
href="style.css">
    <title>Brain BS</title>
</head>
<body>
    <header>
        <div class="box-img-
header">
            
        </div>
        <nav>
            <div class="text-
formatation-left">
                <span
style="color: #fff;">BRAIN-BS:
BLOQUEIO E SEGURANÇA</span>
            </div>
        </nav>
        <div class="box-btn-
actions">
            <a target="_blank"
class="btn-action"
href="redic.html">CADASTRAR</a>
        </div>
    </header>
    <main>
        <section class="first-
section">

```

```

<div>
<h1><strong>Brain
BS</strong></h1>
<div class =
'text-formatation-
left'><p>Brain BS: Bloqueio e
Segurança. O tcc de eletrônica
voltado para deixar sua empresa
mais segura!</p>
</div>
<div class =
'text-formatation-left'>
<p>O
projeto se trata de um Controle
e Monitoramento de Ambiente com
base em Arduino, ESP32-CAM e
RFID.</p>
<p>O seu
funcionamento é dado pela
detecção dos dados de uma
pessoa através do cartão de
acesso que a</p>
<p>mesma
apresenta ao leitor de
frequência RFID, momento no qual
ele irá conferir qual a
informação</p>
<p>dada à
pessoa, e se ela consta no
banco de dados da empresa,
permitindo, ou não, seu
acesso.</p>

```

```

<p>O ESP32-
CAM terá a função de capturar
uma foto do titular e enviar
um sinal para o
microcontrolador</p>
<p>do
Arduino, que assim como já
citado anteriormente, irá
verificar o acesso do usuário
com as</p>
<p>informações no banco de
dados mysql cadastradas.</p>
</div>
</div>
<div class="box-
img-main">

</div>
</section>
</main>
</body>
<script src="main.js"></script>
</html>

```

### **CÓDIGO: formulario.php**

```

<?php
if (isset($_POST['submit']))
{

```

```

        // print_r('Nome: ' .
$_POST['nome']);
        // print_r('<br>');
        // print_r('E-mail: ' .
$_POST['email']);
        // print_r('<br>');
        // print_r('Celular: ' .
$_POST['celular']);
        // print_r('<br>');
        // print_r('Genero: ' .
$_POST['genero']);
        // print_r('<br>');
        // print_r('Data de
nascimento: ' .
$_POST['data_nascimento']);
        // print_r('<br>');
        // print_r('Endereco: ' .
$_POST['endereco']);
        // print_r('<br>');
        // print_r('Rua: ' .
$_POST['rua']);
        // print_r('<br>');
        // print_r('Numero: ' .
$_POST['numero']);
        // print_r('<br>');
        // print_r('Bairro: ' .
$_POST['bairro']);
        // print_r('<br>');
        // print_r('Cidade: ' .
$_POST['cidade']);
        // print_r('<br>');
        // print_r('UF: ' .
$_POST['uf']);

        include_once('conexao.php');
        $nome = $_POST['nome'];
        $email = $_POST['email'];
        $celular =
$_POST['celular'];
        $genero = $_POST['genero'];
        $data_nasc =
$_POST['data_nasc'];
        $cep = $_POST['cep'];
        $rua = $_POST['rua'];
        $numero = $_POST['numero'];
        $bairro = $_POST['bairro'];
        $cidade = $_POST['cidade'];
        $uf = $_POST['uf'];
        $result =
mysqli_query($conexao , "INSERT
INTO
usuarios(nome,email,celular,gen
ero,data_nasc,cep,rua,numero,ba
irro,cidade,uf)
        VALUES
('$nome','$email','$celular','$
genero','$data_nasc','$cep','$r
ua','$numero','$bairro','$cidad
e','$uf')");
    }
?>
<!DOCTYPE html>
<html lang="br">
<head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-
Compatible" content="IE=edge">

```

```

    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet"
href="styleformulario.css">
    <title>Formulário |
BS</title>
</head>
<body>
    <header>
        <div class="box-img-
header">
            
        </div>
        <nav>
            <div class="text-
formatation-left">
                <span
style="color: #fff;">BRAIN-BS:
BLOQUEIO E SEGURANÇA</span>
            </div>
        </nav>
        <div class="box-btn-
actions">
            <a target="_blank"
class="btn-action"
href="inicio.php">INÍCIO</a>
        </div>
    </header>
    <div class="box">

```

```

    <form
action="formulario.php"
method="POST">
        <fieldset>
            <legend><b>Fórmulaário de
Funcionários</b></legend>
                <br>
                <div
class="inputBox">
                    <input
type="text" name="nome"
id="nome" class="inputUser"
required>
                        <label
for="nome"
class="labelInput">Nome
completo</label>
                </div>
                <br><br>
                <div
class="inputBox">
                    <input
type="email" name="email"
id="email" class="inputUser"
required>
                        <label
for="email"
class="labelInput">Email</label
                    >
                </div>
            </div>
        </form>
    </div>

```

```

        <div
class="inputBox">
            <input
type="tel" name="celular"
id="celular" class="inputUser"
required>
                <label
for="celular"
class="labelInput">Celular</lab
el>
        </div>
<br>
<p>Gênero:</p>
<input
type="radio" id="feminino"
name="genero" value="feminino"
required>
        <label
for="feminino">Feminino</label>
<br>
<input
type="radio" id="masculino"
name="genero" value="masculino"
required>
        <label
for="masculino">Masculino</labe
l>
<br>
<input
type="radio" id="transgenero"
name="genero"
value="transgenero" required>

```

```

        <label
for="transgenero">Transgênero</
label>
<br>
<input
type="radio" id="outro"
name="genero" value="outro"
required>
        <label
for="outro">Outro</label>
<br><br>
<label
for="data_nasc"><b>Data de
Nascimento:</b></label>
<input
type="date" name="data_nasc"
id="data_nasc" required>
<br><br>
<div
class="inputBox">
        <input
type="text" name="cep" id="cep"
class="inputUser" required>
        <label
for="cep"
class="labelInput">CEP</label>
</div>
<br><br>
<div
class="inputBox">
        <input
type="text" name="rua" id="rua"
class="inputUser" required>

```

```

        <label
for="rua"
class="labelInput">Rua</label>
    </div>
    <br><br>
    <div
class="inputBox">
        <input
type="text" name="numero"
id="numero" class="inputUser"
required>
        <label
for="numero"
class="labelInput">Número</label>
    </div>
    <br><br>
    <div
class="inputBox">
        <input
type="text" name="bairro"
id="bairro" class="inputUser"
required>
        <label
for="bairro"
class="labelInput">Bairro</label>
    </div>
    <br><br>
    <div
class="inputBox">
        <input

```

```

id="cidade" class="inputUser"
required>
        <label
for="cidade"
class="labelInput">Cidade</label>
    </div>
    <br><br>
    <div
class="inputBox">
        <input
type="text" name="uf" id="uf"
class="inputUser" required>
        <label
for="uf"
class="labelInput">UF</label>
    </div>
    <br><br>
    <input
type="submit" name="submit"
id="submit">
    </fieldset>
</form>
</div>
</body>
<script src="main.js"></script>
</html>

```

**CÓDIGO: Arduino**

```

#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

#define SS_PIN 10      // Pino
SS (SDA) do MFRC522

#define RST_PIN 9     // Pino
RST do MFRC522

#define bsinal 8      //
Buzzer

#define off 6         // Led
vermelho

#define on 7          // Led
verde

LiquidCrystal_I2C lcd
(0x27,16,2); // Instância LCD

MFRC522 mfrc522(SS_PIN,
RST_PIN); // Instância RFID

SoftwareSerial espSerial(2, 3);
// Pinos para comunicação
Serial (RX = 2, TX = 3)

String uid = "";
String data = "";

boolean status = false;
//estado do Led (HIGH/LOW)

float h;
//valor relativo à Umidade

float t;
//valor relativo à Temperatura

int i = 0;
//contador para piscadas de led
- a cada 2i temos uma piscada

int s = 0;
//contador para notas
executadas

int note1 = 665;
int note2 = 498;
int note3 = 996;
int note4 = 1330;
int note5 = 587;

void setup(){

    SPI.begin();
    //Comunicação com RFID

    Serial.begin(9600);      //
Comunicação com o Serial
Monitor

    espSerial.begin(9600);  //
Comunicação com o ESP32

    Serial.println("Arduino
pronto para comunicação");

    pinMode(on, OUTPUT);    //
Led verde

```

```

    pinMode(off, OUTPUT);    //
Led Vermelho
    lcd.init();
    lcd.setBacklight(1);
    lcd.setCursor(0,0);
    lcd.clear();

    mfr522.PCD_Init();    //
Inicializa o leitor RFID
    mfr522.PICC_HaltA(); //
Apaga dados lidos pelo RFID
    Serial.println("Leitor RFID
inicializado");

    delay(400);

    lcd.print("Iniciando o
seu:");

    delay(1500);

    tone(bsinal, note2);

    delay(250);

    start();

    stand();
}

void loop(){

    //Verifica alguma informação
vinda do ESP SERIAL

    if (espSerial.available()) {

        String msgIno =
espSerial.readStringUntil('\n')
;

        msgIno.trim();

        Serial.println(msgIno);

    }

    //Envia dados Seriais para o
ESP

    if (Serial.available()) {

        String msgEsp =
Serial.readStringUntil('\n');

        msgEsp.trim();

        espSerial.println(msgEsp);

    }

    // Verifica se existe um novo
cartão presente

    if
(!mfr522.PICC_IsNewCardPresent
()) {

        return;

    }

    // Verifica se o cartão pode
ser lido

    if
(!mfr522.PICC_ReadCardSerial()
) {

        return;

    }
}

```

```

    uid = "";
    Serial.print("UID do cartão:
");
    for (byte i = 0; i <
mfr522.uid.size; i++) {
        // Formata o UID como uma
string hexadecimal
        uid +=
String(mfr522.uid.uidByte[i] <
0x10 ? "0" : "");
        uid +=
String(mfr522.uid.uidByte[i],
HEX);
    }
    Serial.println(uid);
    // Verifica se o UID
corresponde a um cartão
autorizado
    if
(uid.equalsIgnoreCase("2316F9A3
") ||
uid.equalsIgnoreCase("43bb4695"
)) {
        Serial.println("Cartão
autorizado.");
        accept();
    }
    else {
        Serial.println("Cartão não
autorizado.");
        recuse();
    }
    mfr522.PICC_HaltA();
}
void start(){
    s = 0;
    lcd.clear();
    delay(250);
    while(i <= 2){ // Executa
a inicialização do BRAIN_BS //
Futuramente deve fazer teste de
inicialização
        lcd.setCursor(4,0);
        lcd.print("BRAIN_BS");
        s++; abertura();
        delay(250);
        s++; abertura();
        delay(250);
        s++; abertura();
        delay(250);
        s++; abertura();
        delay(250);
        s++; abertura();
        lcd.clear();

```

```

        delay(1000);
        i = i + 1;
    }
    s = 0;
    delay(1000);
}

void stand(){ // Aparência
do LCD pré-leitura do cartão

    lcd.clear();
    delay(250);
    lcd.setCursor(4,0);
    lcd.print("Apresente");
    lcd.setCursor(2,1);
    lcd.print("o seu cartao");
}

void recuse(){ // Comando
para cartão não autorizado

    i = 0; // zera
contadores de piscadas e
efeitos sonoros

    s = 0;

    lcd.clear();

    while(i <4){ // Pisca led
vermelho

        delay(250);

        status = !status;

        digitalWrite(off,
status);

        lcd.setCursor(1,0);

        lcd.print("Cartao
recusado");

        s = s + 1;

        sinalrecusado();

        delay(250);

        status = !status;

        digitalWrite(off,
status);

        i = i + 1;

    }

    lcd.clear();

    delay(250);

    mfrc522.PICC_HaltA();

    stand(); // Retorna ao
estado pré-leitura

}

void acept(){ // Comando
para cartão autorizado

    i = 0; // zera
contadores de piscadas e
efeitos sonoros

    s = 0;

    lcd.clear();

```

```

    while(i <4){ // Pisca Led
verde
        sinalaceito();

        delay(250);

        status = !status;

        digitalWrite(on, status);

        lcd.setCursor(1,0);

        lcd.print("Cartao aceito");

        i == 0 ? tone(bsinal,
note3) : void();

        delay(250);

        status = !status;

        digitalWrite(on, status);

        i = i + 1;

        s = s + 1;

    }

    espSerial.println("takePic");
// Autoriza a fotocaptura para
o ESP 32 CAM
while(!espSerial.available()){

    espSerial.println(uid);
// Transfere o uid do cartão

    lcd.clear();

    stand();

}

```

```

void sinalrecusado(){
// Efeitos sonoros para cartão
inválido

    switch(s){

        case 1:

            tone(bsinal, note1);

            break;

        case 2:

            tone(bsinal, note2);

            lcd.clear();

            break;

        case 3:

            noTone(bsinal);

            break;

    }

}

void sinalaceito(){
// Efeitos Sonoros para cartão
válido

    switch(s){

        case 1:

            tone(bsinal, note4);

            break;

        case 2:

            noTone(bsinal);

            lcd.clear();

```

```

        break;
    }
}
void abertura(){          //
Efeitos Sonoros
    switch(s){
        case 1:
            tone(bsinal, note1);
            break;
        case 2:
            tone(bsinal, note2);
            break;
        case 3:
            tone(bsinal, note5);
            break;
        case 4:
            tone(bsinal, note1);
            break;
        case 5:
            noTone(bsinal);
            break;
    }
}

```

## CÓDIGO: ESP

```

#include "esp_camera.h"
#include "Arduino.h"
#include "SD_MMC.h"
#include "time.h"
#include "FS.h"
#include <WiFi.h>
#include <EEPROM.h>
// Definições
#define EEPROM_SIZE 1
int pictureNumber = 0;
String data = "";
const char *ssid =
"Brastelecom.Janaina";
const char *password =
"Genetacce";
// Configurações do NTP para
obter a data e hora
const char* ntpServer =
"pool.ntp.org";
const long  gmtOffset_sec = -
10800;          // Ajustável
ao fuso horário
const int  daylightOffset_sec
= 0;           // Ajustável ao
horário de verão

```

```

#define RX_PIN 3
// Pino RX para comunicação com
o Arduino

#define TX_PIN 1
// Pino TX para comunicação com
o Arduino

HardwareSerial mySerial(2);
// Usando UART2

// Define os pinos da câmera
para ESP32-CAM

#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM      5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

    mySerial.begin(9600,
SERIAL_8N1, RX_PIN, TX_PIN);

    mySerial.println("Iniciando..."
);

    WiFi.begin(ssid, password);

    while (WiFi.status() !=
WL_CONNECTED) {

        delay(1000);

        mySerial.println("Conectando ao
Wi-Fi...");

    }

    mySerial.println("Conectado
ao Wi-Fi");

    // Configuração NTP para
obter a data/hora

    configTime(gmtOffset_sec,
daylightOffset_sec, ntpServer);

    // Configuração da câmera

    camera_config_t config;

    config.ledc_channel =
LEDC_CHANNEL_0;

    config.ledc_timer =
LEDC_TIMER_0;

    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;

void setup() {

```

```

config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk =
XCLK_GPIO_NUM;

config.pin_pclk =
PCLK_GPIO_NUM;

config.pin_vsync =
VSYNC_GPIO_NUM;

config.pin_href =
HREF_GPIO_NUM;

config.pin_sscb_sda =
SIOD_GPIO_NUM;

config.pin_sscb_scl =
SIOC_GPIO_NUM;

config.pin_pwn =
PWDN_GPIO_NUM;

config.pin_reset =
RESET_GPIO_NUM;

config.xclk_freq_hz =
20000000;

config.pixel_format =
PIXFORMAT_JPEG;

config.frame_size =
FRAMESIZE_SVGA; // Tamanho da
imagem (800x600)

config.jpeg_quality = 10;
// Qualidade da imagem

config.fb_count = 1;
// Número de framebuffers

// Verificar PSRAM
if (!psramFound()) {
    mySerial.println("PSRAM não
detectado");
} else {
    mySerial.println("PSRAM
detectado");
}

// Inicializando a câmera
if (esp_camera_init(&config)
!= ESP_OK) {
    mySerial.println("Erro na
inicialização da câmera");
    return;
}

// Inicializando o cartão SD
if (!SD_MMC.begin()) {
    mySerial.println("Erro ao
montar o cartão SD");
    return;
}

uint8_t cardType =
SD_MMC.cardType();

```

```

    if(cardType == CARD_NONE){
        mySerial.println("Cartão SD
não detectado");
        return;
    }

    uint64_t cardSize =
SD_MMC.cardSize() / (1024 *
1024);

    mySerial.printf("Tamanho do
cartão SD: %lluMB\n",
cardSize);

    EEPROM.begin(EEPROM_SIZE);

    pictureNumber =
EEPROM.read(0);

    mySerial.printf("Número
inicial de imagens: %d\n",
pictureNumber);

    takepic("imagem_descarte");

    mySerial.println("ESP32
pronto");
}

void takepic(String folderName)
{
    camera_fb_t *fb =
esp_camera_fb_get(); //
Captura a imagem

        if (!fb) {
            mySerial.println("Falha ao
capturar imagem");
            return;
        }

        pictureNumber++; //
Incrementa o número da imagem

        // Obter a data e hora
        struct tm timeinfo;
        if (!getLocalTime(&timeinfo))
        {
            mySerial.println("Falha ao
obter a data/hora");
            return;
        }

        // Formatar a data e hora
para o nome do arquivo

        char timeStamp[16];

        strftime(timeStamp,
sizeof(timeStamp),
"%Y%m%d_%H%M%S", &timeinfo);

        // Gerar o nome da pasta e do
arquivo

        String folderPath = "/" +
folderName;

        String path = folderPath +
"/" + String(pictureNumber) +
"_" + timeStamp + ".jpg";

```

```

    // Criar a pasta se ela não
existir

    if
(!SD_MMC.exists(folderPath)) {
        if
(!SD_MMC.mkdir(folderPath)) {
            mySerial.println("Erro ao
criar a pasta");
            return;
        }
    }

    // Abrir o arquivo para
escrita dentro da pasta

    File file =
SD_MMC.open(path.c_str(),
FILE_WRITE);

    if (!file) {
        mySerial.println("Erro ao
abrir o arquivo para escrita");
        return;
    }

    // Escrever os dados da
imagem no arquivo

    file.write(fb->buf, fb->len);

    // Atualizar a EEPROM com o
novo número da imagem

    EEPROM.write(0,
pictureNumber);

    EEPROM.commit();

    // Fechar o arquivo e liberar
o framebuffer

    file.close();

    esp_camera_fb_return(fb);

    mySerial.printf("Imagem
capturada e salva como: %s\n",
path.c_str());

    delay(1000);
}

void loop() {
    if (mySerial.available()) {
        data =
mySerial.readStringUntil('\n');
        data.trim();

        if (data.isEmpty()) return;

        if (data.equals("takePic"))
        {
            mySerial.println("data");

            while(!mySerial.available()){
                data =
mySerial.readStringUntil('\n');
                data.trim();

                takepic(data);
            } else {

```

```
        mySerial.println("Você      }  
disse: " + data);  
    }  
}
```