

**INSTITUTO FEDERAL DO RIO DE JANEIRO – IFRJ**

**CAMPUS NITERÓI**

**CURSO INTEGRADO TÉCNICO DE INFORMÁTICA**

**PEDRO HENRIQUE BIEGA LOPES BANDEIRA**

**MIGUEL RAMOS GIL**

**SIRENE PROGRAMÁVEL**

**NITERÓI**

**2022**

**PEDRO HENRIQUE BIEGA LOPES BANDEIRA**

**MIGUEL RAMOS GIL**

**SIRENE PROGRAMÁVEL**

**Orientador: Luiz Felipe Silva Oliveira**

Trabalho de conclusão de curso apresentado ao curso de informática do instituto federal do rio de janeiro – ifrj – Campus Niterói, para obtenção da colação de grau do curso técnico de informática integrado ao ensino médio.

**NITERÓI**

**2022**

**PEDRO HENRIQUE BIEGA LOPES BANDEIRA**

**MICHAEL MONTEIRO DE LIMA**

**MIGUEL RAMOS GIL**

**SIRENE PROGRAMÁVEL**

**Trabalho de conclusão de curso apresentado ao curso de informática do  
instituto federal do rio de janeiro – ifrj –  
campus Niterói, para obtenção da  
colação de grau do curso técnico de  
informática integrado ao ensino médio.**

**Banca examinadora**

---

---

## **Resumo**

Em uma Instituição de Ensino Médio é preciso ter elementos básicos para uma educação de qualidade. Um aviso sonoro em um ambiente escolar é uma característica marcante de uma escola, e é fundamental para marcar o começo e o fim dos horários de cada aula. Visando suprir a necessidade de uma sirene eletrônica automatizada para o IFRJ – Campus Niterói, foi desenvolvido o projeto no decorrer do curso no qual propõe-se a oferecer utilidade, segurança e informação para os alunos, servidores e docentes. Além de anunciar a mudança de aulas, o projeto desenvolvido é capaz de antecipar e informar possíveis acidentes dentro do colégio.

**Palavras chave:** ESP32, Sirene,

## **Abstract**

In a High School Institution it is necessary to have basic elements for a quality education. An audible warning in a school environment is a hallmark of a school, and is key to marking the beginning and end of each class time. In order to meet the need for an automated electronic siren for the IFRJ - Campus Niterói, the project was developed during the course in which it proposes to offer usefulness, security and information for students, servers and teachers. In addition to announcing the change of classes, the project developed is capable of anticipating and informing possible accidents within the school.

**Keywords:** ESP32, Sirene

## **Lista de Ilustrações**

Figura 1 - Conexões eletrônicas do ESP32	14
Figura 2 - Conexões eletrônicas do ESP32	14
Figura 3 - Primeiro protótipo da tela de cadastro de horários	16
Figura 4 - Tela de login da aplicação web	16
Figura 5 - Tela para o cadastro dos horários de acionamento	17
Figura 6 - Tela de aviso de dados salvos	17
Figura 7 - Arquivo JSON que será alterado no servidor	17

# Sumário

## SUMÁRIO

<b>1 Introdução</b>	<b>8</b>
<b>1.2 Objetivo</b>	<b>9</b>
<b>1.2.1 Objetivo Geral</b>	<b>9</b>
<b>1.2.2 Objetivos Específicos</b>	<b>9</b>
<b>1.3 Metodologia</b>	<b>10</b>
<b>2 Projeto</b>	<b>10</b>
<b>2.1 Levantamento dos Requisitos</b>	<b>10</b>
<b>2.1.1 Pré-projeto anterior</b>	<b>12</b>
<b>2.2 Levantamentos de Materiais que possam ser utilizados</b>	<b>12</b>
<b>2.3. Projeto proposto</b>	<b>13</b>
<b>2.3.1 Projeto Eletrônico (Circuito Eletrônico)</b>	<b>13</b>
<b>2.3.2 Projeto Eletrônico (Montagem)</b>	<b>14</b>
<b>2.4 Projeto do Sistema</b>	<b>15</b>
<b>2.4.1 Rascunho da Tela</b>	<b>16</b>
<b>3 Implementação</b>	<b>17</b>
<b>3.1 Montagem Eletrônica</b>	<b>18</b>
<b>3.2 Programa Implementado</b>	<b>18</b>
<b>4 Considerações Finais</b>	<b>19</b>

## 1 Introdução

Mesmo com o avanço do ensino a distância, o ensino presencial é dominante. Principalmente no fundamental e médio. Dentro do ensino presencial, há o uma delimitação de tempo para cada horário de aula, assim como o início e fim dos intervalos. Para que todos possam ter um entendimento claro e unanime desses horários, criou-se a necessidade de haver um item de sinalização dentro do Campus Niterói.

Em um ambiente escolar o uso de sinalizações que apontem o início/término das aulas mostra ser importante, pois auxilia o aluno e docente a ter entendimento do tempo. Ele determina melhor o uso do horário dos professores e avisa aos alunos quando eles devem voltar às salas após o intervalo.

Diante do exposto acima e a partir das necessidades observadas no Instituto Federal do Rio de Janeiro - Campus Niterói, foi idealizado um projeto para a criação de uma solução prototipada que atue como um sinalizador programável com o propósito de alertar a comunidade acadêmica da mudança de horários.

O desenvolvimento desse sinalizador, aqui denominado sirene, terá uma plataforma de gerenciamento para o funcionamento automatizado com horários previamente programados. Com os horários estabelecidos, a sirene escolar irá tocar indicando aqueles presentes na instituição à hora.

O projeto visa suprir, através do que foi ensinado na ementa do curso e de conhecimentos adquiridos por base de pesquisas, a falta de um elemento tão prático para um instituto e ajudar os alunos, os servidores e os futuros alunos.

Entretanto, como devemos desenvolvê-la? A sirene criada deve conter um Arduino<sup>1</sup>, o escolhido será o ESP32<sup>2</sup>, um Módulo Relé<sup>3</sup>, um Módulo RTC<sup>4</sup> e um Transformador. Arduino é uma plataforma de prototipagem eletrônica de hardware livre, ou seja, Open Source, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação em C++<sup>5</sup>. Já o ESP32 é uma série de microcontroladores de baixo custo e baixo consumo de energia. Também é um sistema-em-um-chip (SOC) com microcontrolador integrado, ou seja, é muito parecido com o Arduino, porém com um hardware bem melhor, já que possui um microprocessor tensilica xtensa lx6 com um poder de processamento muito maior que o Arduino, além de ter

---

<sup>1</sup> Arduino – É uma plataforma de prototipagem eletrônica que utiliza C/C++.

<sup>2</sup> ESP32 – É uma plataforma de prototipagem eletrônica que tem acesso à Internet e bluetooth.

<sup>3</sup> Relé – Um interruptor eletromagnético.

<sup>4</sup> RTC – Real Time Clock, cede acesso à data e hora.

<sup>5</sup> C++ – Linguagem de programação compilada.



funcionalidades nativas no chip como, Wi-Fi e Bluetooth, que no Arduino precisaria se utilizar uma série de Shields, o que tornaria o projeto bem mais caro, tendo isso em vista, trabalhar com o ESP32 se tornou a melhor proposta devido ao seu custo-benefício.

## **1.2 Objetivo**

Nosso objetivo é ajudar os discentes e docentes com uma necessidade que temos no campus. Hoje o campus se encontra sem uma sirene, em consequência disso os alunos tendem não saber os horários de intervalo e início das aulas. Então, construiremos uma sirene para o Campus Niterói, com o objetivo de fazer dessa um artifício para que o aluno, os professores e funcionários tomem um norte dos tempos de aula.

### **1.2.1 Objetivo Geral**

O objetivo geral do projeto é, de forma criativa, suprir a necessidade do campus de ter um alerta para os horários de aula, necessidade que se faz presente desde sua fundação, auxiliando aqueles presentes no Campus. Atualmente, os alunos não têm uma orientação clara quanto os horários de intervalo e início das aulas, podendo demorar mais para chegar as salas no horário certo, o que acarreta na demora do começo de algumas aulas. Portanto a ideia de fazer uma sirene aconteceu para que pudéssemos melhorar o ambiente escolar.

### **1.2.2 Objetivos Específicos**

No projeto da sirene, usamos o hardware ESP32 como plataforma de desenvolvimento do projeto, sua programação é feita em C/C++ pelo Arduino IDE<sup>6</sup>. Um dos nossos objetivos é desenvolver uma plataforma (site) que se conecte com o microcontrolador do projeto e mude os parâmetros de comportamento, os horários que ela será acionada. Pretende-se desenvolver esse site em html<sup>7</sup>, css<sup>8</sup>, php<sup>9</sup>, javascript<sup>10</sup> e um arquivo JSON<sup>11</sup> para retenção dos dados. Para se comunicar, o

---

<sup>6</sup> Arduino IDE – É uma plataforma de programação.

<sup>7</sup> html – Linguagem de marcação de hipertexto.

<sup>8</sup> css – Cascading Style Sheet, linguagem de marcação usada para estilização de páginas.

<sup>9</sup> php – Hypertext Preprocessor, usado para processar informação providas do html.

<sup>10</sup> javascript – Linguagem de programação interpretada.

<sup>11</sup> JSON – JavaScript Object Notation, usada para troca de dados simples.

ESP32 fará uma requisição http<sup>12</sup> com a página web e logo após fará a desserialização do conteúdo armazenado dentro do arquivo JSON, logo após a captura de dados os o microcontrolador irá salvá-los dentro de sua memória na volátil, a EEPROM<sup>13</sup>, assim permitindo que a sirene permaneça funcionando baseada no último horário salvo por ela. O usuário irá conectar os ESP32 a internet para informar os horários que a sirene deve ser acionada e o módulo RTC irá conceder a data e hora em tempo real para o microcontrolador.

Os códigos do projeto serão open source<sup>14</sup> e está disponível na direção, com o coordenador do curso de informática e com os criadores do projeto para aqueles que quiserem fazer manutenção ou melhorar o projeto fazendo upgrades.

### **1.3 Metodologia**

Para organizar o desenvolvimento estabelecemos uma metodologia para esse projeto tomando como base os passos utilizados no desenvolvimento de sistemas (PRESSMAN, 2010), que são: análise de requisitos, análise de sistemas, projeto, implementação, testes, implantação, operação. Com isso em mente, podemos entender os passos tomados para que o projeto seja realizado.

## **2 Projeto**

O projeto idealizado deverá cumprir alguns processos até que esteja estabelecida a versão final do projeto. Para estabelecer isso é preciso analisar o que devemos implementar na sua estrutura de hardware e software, pesquisar sobre projetos base para fundamentar o desenvolvido e saber quais materiais devemos usar. Tendo isso em mente, o projeto pode ser concluído.

### **2.1 Levantamento dos Requisitos**

Para desenvolvermos o projeto, o seu desenvolvimento foi separado em duas partes: hardware (montagem) e software (site). Para obter êxito na conclusão, trabalhamos em conjunto para sabermos como seria a estrutura da montagem e do sistema deveríamos desenvolver.

---

<sup>12</sup> http – Protocolo de transferência de hipertexto.

<sup>13</sup> EEPROM – Memória não volátil do microcontrolador.

<sup>14</sup> Open source – Software de código aberto.

O projeto, em sua idealização, prevê que deverá ser um dispositivo que controle a passagem dos horários de aula, alarmando a instituição do IFRJ quando um tempo de aula começar ou terminar. Com isso estabelecido, a ideia foi aprimorada para que os horários de alarme sejam automatizados, e funcionem em toda a escola, foi dado o nome do projeto de “Sirene programável”.

Notamos que o hardware necessitará de alguns módulos que pudessem armazenar os horários que o alarme iria tocar e um módulo pra ligar e desligar a sirene, além de um dispositivo programável que possa definir agrupar e definir ação que cada componente deve executar. Todos esses dispositivos foram pensados para que o projeto seja executado no horário certo em que a Sirene deverá alarmar de forma que seja propensa à mudança do que foi pré-programado.

Após fazer as análises de como iríamos instalar o projeto na instituição, criou-se a necessidade de termos 2 sirenes para cobrimos todo o espaço da escola com o sinal. Com as pesquisas, foi estabelecida a necessidade de um Módulo ESP32 como microcontrolador, um Módulo RTC para manter as informações do horário e um Módulo Relé para abrir ou fechar a corrente que liga a sirene.

Para o armazenamento dos componentes optamos por manter os módulos usados dentro de uma caixa, tenha uma bandeja removível e que seja de fácil troca dos componentes. Essa caixa terá uma dobradiça para que ela feche e mantenha os dispositivos protegidos, diminuindo os riscos de danos aos componentes, dentro dela há uma bandeja que comportará todos os módulos do projeto, esta será removível, dando vantagem a troca e reparo dos dispositivos, visando o futuro reparo das peças caso futuramente haja um mau funcionamento de uma.

Quanto ao software, foi idealizada uma aplicação web que iria se conectar com o ESP32 por meio do protocolo MQTT<sup>15</sup>, por esse protocolo os horários de ativação seriam enviados para o ESP32, assim o microcontrolador já com o módulo RTC acoplado, acionaria a sirene nos horários pré-definidos.

---

<sup>15</sup> MQTT – Message Queuing Telemetry Transport, um protocolo de troca de mensagens.

### **2.1.1 Pré-projeto anterior**

Previamente, antes da idealização do projeto, o campus possuía uma solução de automação em ESP32 LoRa915mhz<sup>16</sup> para uma sirene escolar, porém, com o tempo, entrou em desuso. Entretanto os seus componentes e seu circuito eletrônico servirão como base para a montagem do novo projeto.

O projeto desenvolvido em 2019, possuía um ESP32 LoRa915mhz, um módulo relé, um módulo RTC e uma protoboard para unir os componentes. A alimentação da sirene era feita diretamente através da tomada e seu circuito era fechado pelo módulo Relé. Sua ativação era feita através do LoRa915mhz, onde era ativado manualmente através de uma página e um comando de ON ou OFF.

Com o modelo antigo em mãos, pudemos ter um ponto de partida para o começo da montagem do projeto, sabendo quais peças e como montá-las.

### **2.2 Levantamentos de Materiais que possam ser utilizados**

Para o desenvolvimento de um projeto de Hardware Programável, será necessário ter materiais que desempenhem funções para que a sirene funcione, como ter uma conexão com o site que controlará os horários de toque do alarme.

Devemos ter alguns dispositivos que permitem o funcionamento do projeto. Um dos materiais usados será um módulo ESP32, uma placa de desenvolvimento e prototipação eletrônica, que funciona como um microcontrolador tal como um Arduino, porém com muito mais conectividade e recursos contando com Wi-Fi e outras opções que ajudam no desenvolvimento do projeto. Outro dispositivo será um módulo Relé, que desempenhará função de interruptor eletromecânico com o objetivo de prover modificações súbitas e predeterminadas, permitindo a abertura e fechamento de um circuito elétrico, bloqueando ou não a passagem elétrica. Por último, notamos a necessidade de um dispositivo para armazenar a hora atual e real de onde a sirene está, para solucionar essa demanda utilizaremos o módulo RTC, um módulo Real Time Clock é módulo de relógio de tempo real voltando para a prototipação eletrônica.

---

<sup>16</sup> LoRa915mhz – É uma tecnologia de rede de área de baixa distância.

## 2.3. Projeto proposto

Ao desenvolvermos o projeto e aplicarmos o que estava em sua idealização o tivemos que mudar a idealização e solucionar os erros e problemas encontrados. Propondo melhorias e lidando com o que encontrávamos, assim o projeto proposto é o resultado do que pensamos na idealização.

### 2.3.1 Projeto Eletrônico (Circuito Eletrônico)

Executando a montagem do circuito e das peças que a compõem, foi feita a ligação do ESP32 com os demais dispositivos citados anteriormente. Os dispositivos como o RTC e Relé se ligam ao ESP32 por meio de uma porta de energia, de ground e uma porta de dados, conforme mostra as imagens abaixo. Todos eles se mantem dentro da bandeja removível, possibilitando o futuro reparo das peças.

Abaixo está exposto o diagrama do circuito que o ESP32 está conectado, e sua devidas portas, assim como no projeto montado foi feito.

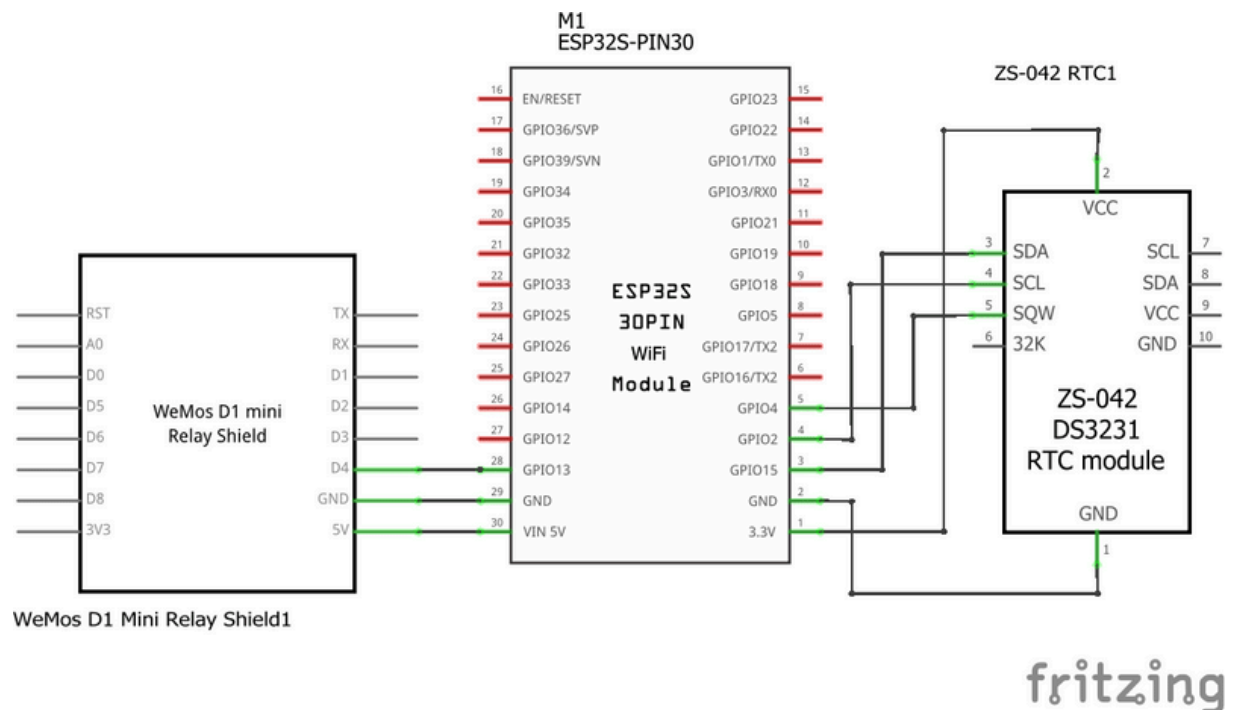


Figura 1 - Conexões eletrônicas do ESP32

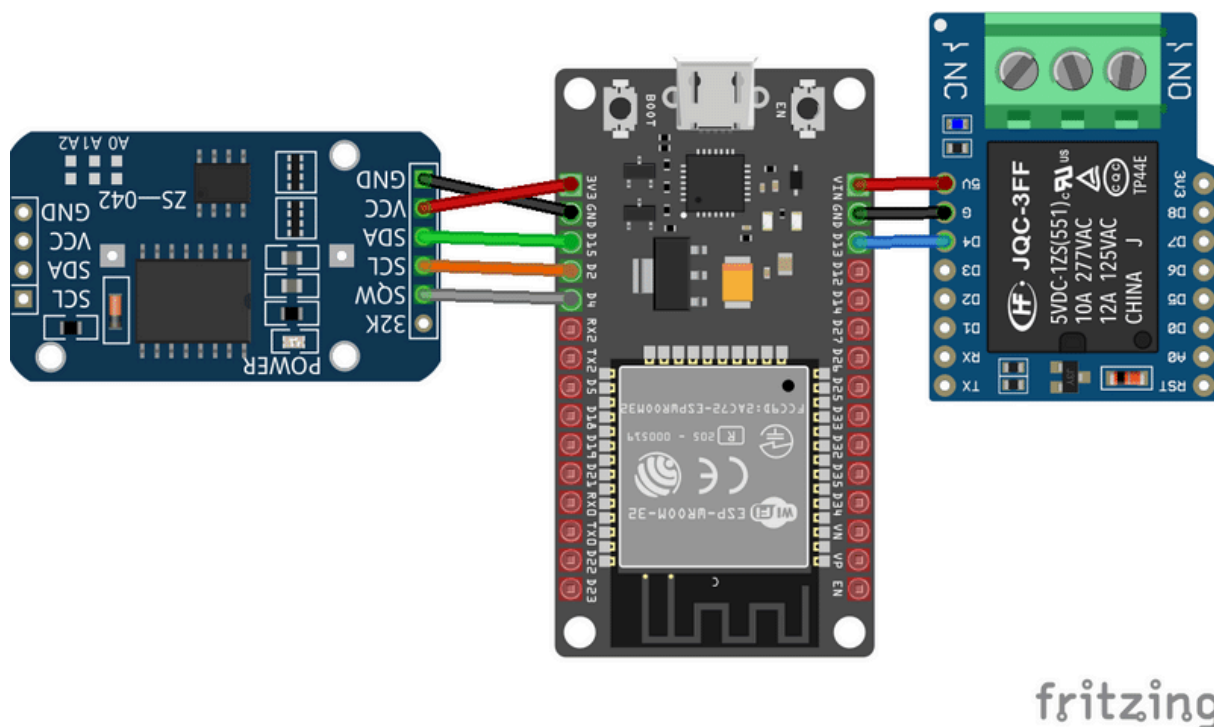


Figura 2 - Conexões eletrônicas do ESP32

### 2.3.2 Projeto Eletrônico (Montagem)

Para começar o projeto, destacou-se a necessidade de um caixa que comporte todos dos componentes do projeto. Em resolução, uma caixa com dobradiças foi adicionada ao projeto e uma bandeja removível dentro dela para o comporte dos dispositivos eletrônicos, onde todos os módulos foram instalados.

Quanto a alimentação de força para a sirene, foi instalado na caixa um plug AC tripolar macho, assim podendo alimentar a sirene por meio de um cabo de força conectável ao plug. Dentro da caixa, foi soldado um cabo diretamente da sirene ao plug de força. A sirene tem seu circuito partindo da força e passando pelo módulo relé que abre ou fecha o circuito para sua ativação, passa por um conector que permite a abertura e mudança dos fios, e assim termina no plug mais uma vez.

Conforme o projeto foi feito, notou-se a necessidade de um dispositivo que transformasse e regulasse a entrada de 220V 20A que era distribuída à sirene e alimentasse também o ESP32 com a saída de 5V 700mA. Para resolver isso, foi adotado ao projeto um Módulo transformador de 200V 2A para 5V 700mA.

Percebendo que teríamos que atingir toda a escola com o alerta, optamos por fazer outra sirene idêntica ao primeiro dispositivo, ela foi feita com os componentes da antiga sirene que havia na escola. Com ela podemos alcançar o som em todo o Campus.

## **2.4 Projeto do Sistema**

Quando o foi dado início ao desenvolvimento da parte de software notamos que usar o protocolo MQTT tornaria o projeto muito inseguro, já que usaríamos um broker público, dessa maneira qualquer pessoa mal intencionada e com conhecimentos da área de informática poderia acessar o broker<sup>17</sup> e a alterar os horários de acionamento.

Tendo isso em vista decidimos junto de nosso orientador mudar o plano de ação e usar um arquivo JSON para a obtenção dos horários de acionamento, assim tornado o projeto mais seguro, já que para ter acesso à página de cadastro de horários será necessário ter acesso às credenciais de acesso.

A escolha da plataforma web para desenvolvimento da aplicação se deu em razão da maior acessibilidade, tendo em vista que todo o aparelho atual do mercado já vem com um navegador.

---

<sup>17</sup> Broker – É o elemento responsável por gerir as publicações e as subscrições do MQTT

### 2.4.1 Rascunho da Tela

A wireframe of a time registration screen. It features a dark blue header and footer. The main content area has a gray background. There are ten input fields arranged in two columns. The left column contains fields labeled 'HORARIO 1:', 'HORARIO 2:', 'HORARIO 3:', 'HORARIO 4:', 'HORARIO 5:', and 'HORARIO 6:'. The right column contains fields labeled 'HORARIO 7:', 'HORARIO 8:', 'HORARIO 9:', and 'HORARIO 10:'. Each label is in a black box, and each input field is a white rectangle.

Figura 3 - Primeiro protótipo da tela de cadastro de horários

Após mudanças significantes no design da página web e a criação de novas telas a página web está completamente mudada visualmente

A mockup of a login screen. It has a yellow header with a black 'N' logo on the left and the word 'Login' in the center. Below the header is a white rounded rectangle containing two input fields: 'Usuário' and 'Senha'. Below these fields are two buttons: 'Acessar' and 'Apagar'.

Figura 4 - Tela de login da aplicação web





Horários

Primeiro Horário	00:00	⌚	Nono horário	00:00	⌚
Segundo Horário	00:00	⌚	Décimo horário	00:00	⌚
Terceiro horário	00:00	⌚	Décimo primeiro horário	00:00	⌚
Quarto horário	00:00	⌚	Décimo segundo horário	00:00	⌚
Quinto horário	00:00	⌚	Décimo terceiro horário	00:00	⌚
Sexto horário	00:00	⌚	Décimo quarto horário	00:00	⌚
Sétimo horário	00:00	⌚	Décimo quinto horário	00:00	⌚
Oitavo horário	00:00	⌚			

Versão de configuração

Configuração

Duração em segundos

Duração

Enviar

Apagar

\* Caso a versão de configuração seja igual a versão anterior os horários não serão carregados

**Figura 5 - Tela para o cadastro dos horários de acionamento**

Os horários foram enviados

Em minutos eles serão carregados na sirene

**Figura 6 - Tela de aviso de dados salvos**

```
[{"horario0": "07:30", "horario1": "09:00", "horario2": "09:20", "horario3": "10:50", "horario4": "12:20", "horario5": "13:10", "horario6": "14:40", "horario7": "16:10", "horario8": "16:30", "horario9": "18:00", "horario10": "00:00", "horario11": "00:00", "horario12": "00:00", "horario13": "00:00", "horario14": "00:00", "config_version": "1", "duration": "10"}]
```

**Figura 7 - Arquivo JSON que será alterado no servidor**

### 3 Implementação

O resultado implementado no projeto está separado em dois resultados conclusivos do dispositivo criado: a montagem eletrônica e a implementação do programa. A montagem eletrônica corresponde a parte física, mostrando os resultados a do que foi construído. Já a implementação do programa corresponde a parte da codificação do projeto.

### **3.1 Montagem Eletrônica**

A montagem final é composta pela caixa de madeira que comporta todos os dispositivos, a ligação de força do plug para a sirene, assim como para o transformador converter para 5V 700mA e alimentar o ESP32.

Os dispositivos eletrônicos do projeto são anexados na badeja removível e todos os seus fios são facilmente trocáveis, pois possuem conectores removíveis. Os dispositivos do projeto são: o ESP32, que contém as informações de quando a sirene irá tocar e controla os demais aparelhos, o módulo Relé, que tem função de interruptor eletrônico, abrindo ou fechando o circuito, o módulo RTC, que comporta as horas reais e o transformador, que alimenta o ESP32.

Devido a dimensão da escola e o objetivo de ter um alarme para toda a escola, construímos duas sirenes idênticas com o mesmo circuito eletrônico e montagem, assim podendo alcançar toda a escola com o sinal.

### **3.2 Programa Implementado**

Após as diversas mudanças feitas durante o processo implementação, o fim da sirene se tornou completamente diferente quando comparado com a visão que tínhamos dela no início do projeto, ao final de todo o processo de implementação a sirene está funcionando da seguinte maneira, o usuário acessa a página de login, insere as credencias e é direcionado para o formulário de cadastro de horários de acionamento ao inserir os horários ele é direcionado para uma página, que utiliza de uma função feita em php para extrair os dados da página de cadastro, após isso, essa função cria um arquivo JSON com os dados da página anterior.

Tendo isso funcionando, o ESP32, que já esta com o modulo para a obtenção de data e hora acoplado, ele se conecta com a internet, faz uma requisição http e se conecta com o arquivo JSON, após isso, caso o versão de configuração seja diferente da versão de configuração anterior o ESP32 salva os dados contidos no JSON dentro da memória EEPROM, com isso, o microcontrolador usa as informações salvas na EEPROM e as compara com a data e a hora fornecida pelo modulo RTC, caso ele verifique que chegou no horário de ativação o ESP32 aciona a sirene durante o tempo determinado pelo usuário na página de cadastro.

#### **4 Considerações Finais**

O objetivo geral desse projeto foi desenvolver uma solução criativa, prototipada e pratica para uma dor que existiu no instituto federal do rio de janeiro – Campus Niterói durante os anos de 2021 e 2022.

Com base nas soluções e métodos de implementação propostos durante esse trabalho, verificamos que os métodos usados para conclusão deste trabalho foram os melhores encontrados tendo em vista as nossas limitações, apesar disso, seguindo tudo que foi exposto durante esse projeto é fato que a solução para a dor será sanada.