

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO
PAULO - CÂMPUS CUBATÃO**

TÉCNICO EM INFORMÁTICA INTEGRADO AO ENSINO MÉDIO

MARIA JULIA CAVALCANTI CERQUEIRA LEITE

**COMO PROGRAMAR UM CONTROLE REMOTO VIA WEBSITE
UTILIZANDO ARDUINO E NODE.JS**

CUBATÃO

2023

"A tecnologia move o mundo e define o futuro"

Steve Jobs

SUMÁRIO

1. INTRODUÇÃO.....	05
1.1. VISÃO GERAL DO PROJETO.....	05
1.2. OBJETIVOS E FINALIDADES.....	05
1.2.1. DESENVOLVIMENTO DE HABILIDADES TÉCNICAS.....	05
1.2.2. INTEGRAÇÃO COM TECNOLOGIAS.....	06
1.2.3. ENTENDIMENTO DE COMUNICAÇÃO SERIAL.....	06
1.2.3.1. CONTROLE REMOTO ATRÁVES DA WEB.....	06
1.2.3.2. AUTOMAÇÃO COM HORÁRIO.....	06
1.2.3.3. EXPLORAÇÃO DA INTERNET DAS COISAS (IOT).....	06
1.2.3.4. REGISTRO E ANÁLISE DE DADOS.....	06
1.3. PRÉ-REQUISITOS E CONHECIMENTOS NECESSÁRIOS.....	07
1.3.1. CONHECIMENTO BÁSICO EM PROGRAMAÇÃO.....	07
1.3.2. NOÇÕES DE ELETRÔNICA.....	07
1.3.3. CONHECIMENTOS EM WEB DEVELOPMENT.....	07
2. CONFIGURAÇÃO DO AMBIENTE.....	07
2.1. INSTALAÇÃO DAS FERRAMENTAS NECESSÁRIAS.....	07
2.1.1. NODE.JS.....	07
2.1.2. ARDUINO IDE.....	08
2.1.3. MYSQL WORCHBENCH.....	09
3. ACESSO E UTILIZAÇÃO DOS CÓDIGO.....	10
3.1. COMO ACESSAR OS CÓDIGOS.....	10
3.2. NAVEGANDO NO REPOSITÓRIO.....	10
3.3. COMO USAR OS CÓDIGOS.....	10
3.4. CLONAR O REPOSITÓRIO.....	10
3.5. CONTEÚDO.....	11
4. DESENVOLVIMENTO DO SITE HTML, JAVASCRIPT E CSS.....	11
4.1. ESTRUTURA DO SITE HTML E ESTILIZAÇÃO DO CSS.....	11
4.1.1. CÓDIGO DO SITE CONTROLADOR.....	11

4.1.2.CÓDIGO DO SITE AUXILIAR.....	13
4.1.3. CÓDIGO DO CSS.....	15
4.2. INTEGRAÇÃO COM JAVASCRIPT.....	16
4.2.1. CÓDIGO DO SITE CONTROLADOR.....	16
4.2.2. CÓDIGO DO SITE AUXILIAR.....	18
4.3. SERVIDOR NODE.JS.....	19
4.4. ARDUINO.....	21
4.5. MYSQL WORCHBENCH.....	22
5. PROJETO ARDUINO.....	23
5.1. MATERIAIS.....	23
5.2. MONTAGEM.....	23
6. RECURSOS ADICIONAIS.....	24
6.1. LINKS ÚTEIS PARA APRENDIZADO DE NODE.JS, ARDUINO E MYSQL.....	24
6.1.1. NODE.JS.....	24
6.1.2. ARDUINO.....	24
6.1.3. MYSQL WORCHBENCH.....	24

1. INTRODUÇÃO

1.1.VISÃO GERAL DO PROJETO

Bem-vindo à apostila "Como programar um controle remoto via *website* utilizando Arduino e Node.js". Este guia explora a criação de um sistema de controle remoto de uma lâmpada, utilizando uma combinação de tecnologias *web*, programação em JavaScript, servidor Node.js e placa Arduino.

O projeto se propõe a desenvolver um *website* que ofereça controle para dispositivos inteligentes; No exemplo utilizado, há apenas alguns comandos simples, como botões de ligar e desligar, além de um horário programável para automação. Essas interações são traduzidas em comandos que acionam uma lâmpada comum conectada a uma placa Arduino.

A comunicação é intermediada por um servidor Node.js, que gerencia as requisições do *website* e as encaminha para o Arduino através de uma conexão *serial*. Os comandos são registrados em um banco de dados MySQL, permitindo análises posteriores das interações.

Ao seguir este tutorial, será possível obter uma compreensão prática de desenvolvimento *web*, manipulação de dados com bancos de dados, interação entre o mundo virtual e físico usando Arduino e a coordenação de componentes por meio do servidor Node.js.

Vamos iniciar abordando a configuração do ambiente e, em seguida, mergulhar nos detalhes técnicos para implementar este sistema de controle remoto.

1.2.OBJETIVOS E FINALIDADES

O projeto " Como programar um controle remoto via *website* utilizando Arduino e Node.js " busca atingir uma série de objetivos que visam expandir o conhecimento e a proficiência técnica, como:

1.2.1. DESENVOLVIMENTO DE HABILIDADES TÉCNICAS: Este projeto tem como objetivo central aprimorar as habilidades dos participantes em diversas áreas tecnológicas. Desde a criação de interfaces web interativas até

a programação de dispositivos físicos, os participantes serão expostos a um conjunto variado de habilidades que têm aplicações em campos diversos.

1.2.2. **INTEGRAÇÃO COM TECNOLOGIAS:** A implementação bem-sucedida deste projeto demonstrará a capacidade de integrar tecnologias distintas em um sistema coeso e funcional. Os participantes aprenderão a interligar o desenvolvimento web, a programação de dispositivos eletrônicos e a comunicação entre sistemas através de um servidor central.

1.2.3. **ENTENDIMENTO DE COMUNICAÇÃO SERIAL:** Através da conexão serial entre o servidor Node.js e a placa Arduino Uno, os participantes adquirirão um entendimento prático da comunicação em tempo real entre dispositivos. Isso abre portas para a criação de soluções que exigem troca eficiente de informações. E tem como finalidades:

1.2.3.1. **CONTROLE REMOTO ATRÁVES DA WEB:** O projeto tem como finalidade principal criar um sistema funcional de controle remoto, no qual uma lâmpada pode ser ativada e desativada por meio de um website. Essa finalidade ilustra de maneira tangível como a tecnologia web pode ser usada para operar dispositivos no ambiente físico.

1.2.3.2. **AUTOMAÇÃO COM HORÁRIO:** Além do controle manual, a inclusão de um horário programável agrega um nível de automação ao projeto. Isso permite que a lâmpada seja acionada automaticamente de acordo com intervalos de tempo definidos, introduzindo conceitos de programação temporal.

1.2.3.3. **EXPLORAÇÃO DA INTERNET DAS COISAS (IOT):** A conexão da placa Arduino Uno à internet proporciona um vislumbre da Internet das Coisas (IoT). A possibilidade de monitorar e controlar dispositivos físicos remotamente através de uma plataforma online é uma demonstração concreta das capacidades da IoT.

1.2.3.4. **REGISTRO E ANÁLISE DE DADOS:** A integração de um banco de dados MySQL tem como finalidade registrar os comandos enviados a lâmpada. Isso oferece aos participantes a oportunidade de compreender a coleta e a análise de dados. A capacidade de interpretar padrões de uso pode ser valiosa em cenários que demandam insights de dados.

Ao concluir este tutorial, os participantes estarão equipados para criar um sistema semelhante por conta própria, compreendendo os princípios da comunicação entre diferentes tecnologias e suas aplicações práticas.

1.3.PRÉ-REQUISITOS E CONHECIMENTOS NECESSÁRIOS

Antes de iniciar este projeto, é importante possuir um conjunto básico de conhecimentos e ferramentas. Os seguintes pré-requisitos são recomendados:

- 1.3.1. **CONHECIMENTO BÁSICO EM PROGRAMAÇÃO:** Familiaridade com conceitos de programação é necessário, especialmente em linguagens como JavaScript e C++ (para programação Arduino).
- 1.3.2. **NOÇÕES DE ELETRÔNICA:** Ter uma compreensão básica de eletrônica, componentes e circuitos ajudará na implementação do Arduino.
- 1.3.3. **CONHECIMENTOS EM WEB DEVELOPMENT:** Entender conceitos básicos de desenvolvimento *web*, como HTML, CSS e JavaScript, é essencial para criar a interface do controle remoto.

2. CONFIGURAÇÃO DO AMBIENTE

2.1.INSTALAÇÃO DAS FERRAMENTAS NECESSÁRIAS

2.1.1. NODE.JS

- i. **Acesse o Site Oficial:** Vá para o site oficial do Node.js em <https://nodejs.org/>.
- ii. **Escolha a Versão:** Na página inicial, você verá duas versões do Node.js para escolher: LTS (Long-Term Support) e Current. Recomenda-se escolher a versão LTS, pois é mais estável e adequada para a maioria dos casos de uso.
- iii. **Selecione o Sistema Operacional:** Escolha o sistema operacional que você está usando. As opções mais comuns são Windows, macOS e Linux. Clique na versão relevante para o seu sistema operacional.
- iv. **Baixe o Instalador:** Após selecionar o sistema operacional, você será redirecionado para a página de download. Baixe o instalador adequado para o seu sistema.
- v. **Execute o Instalador:**
 - i. **Windows:** Após o download, execute o instalador. Siga as instruções do assistente de instalação. Certifique-se de marcar a opção "Add to PATH" durante a instalação para que o Node.js seja adicionado ao seu PATH do sistema.

- ii. **macOS:** Após o download, abra o arquivo .pkg e siga as instruções do instalador.
- iii. **Linux:** No caso do Linux, as instruções podem variar de acordo com a distribuição. Geralmente, você pode usar comandos de gerenciador de pacotes. Por exemplo, no Ubuntu, você pode executar:

`sudo apt update`

`sudo apt install nodejs npm`

- vi. **Verifique a Instalação:** Após a instalação, abra um terminal e digite os seguintes comandos para verificar se o Node.js e o npm (gerenciador de pacotes do Node.js) foram instalados corretamente:

`node -v`

`npm -v`

Isso deve exibir as versões instaladas do Node.js e do npm.

2.2. ARDUINO IDE

Para instalar a IDE do Arduino, siga estas etapas:

- i. **Acesse o Site Oficial:** Vá para o site oficial do Arduino em <https://www.arduino.cc/>.
- ii. **Baixe a IDE:** No site do Arduino, clique na seção "Software" no menu principal e selecione "Arduino IDE". Isso o levará para a página de download da IDE.
- iii. **Escolha o Sistema Operacional:** Na página de download, você verá opções para diferentes sistemas operacionais, como Windows, macOS e Linux. Escolha a opção adequada para o seu sistema operacional e clique no link de download correspondente.
- iv. **Baixe e Instale:** Após o download, execute o instalador baixado.
 - i. **Windows:** Execute o instalador e siga as instruções do assistente de instalação.
 - ii. **macOS:** Abra o arquivo .dmg baixado e arraste o ícone do Arduino para a pasta Applications.
 - iii. **Linux:** As instruções de instalação podem variar de acordo com a distribuição. Geralmente, você pode extrair o arquivo baixado e executar a IDE diretamente.
- v. **Abra a IDE:** Após a instalação, abra a IDE do Arduino.

- vi. **Configuração Inicial:** Quando você abre a IDE pela primeira vez, pode ser necessário configurar algumas opções, como a pasta onde os sketches (códigos-fonte) serão salvos e a placa Arduino que você está usando. Você pode fazer isso selecionando as opções apropriadas no menu "File" (Arquivo) > "Preferences" (Preferências) e "Tools" (Ferramentas) > "Board" (Placa).
- vii. **Instale Drivers (se necessário):** Dependendo do tipo de placa Arduino que você possui, talvez seja necessário instalar drivers adicionais. A maioria das placas modernas, como a Arduino Uno, não requer drivers extras. No entanto, se estiver usando uma placa diferente, consulte a documentação do Arduino para obter informações sobre a instalação de drivers.

2.3.MYSQL WORCHBENCH

Para instalar o MySQL Workbench, uma ferramenta de administração e modelagem de banco de dados para MySQL, siga estas etapas:

- i. **Acesse o Site Oficial:** Vá para o site oficial do MySQL Workbench em <https://www.mysql.com/products/workbench/>.
- ii. **Baixe o Instalador:** Na página do MySQL Workbench, clique no botão "Download Now" (Baixar Agora) ou na opção de download correspondente ao seu sistema operacional.
- iii. **Escolha o Sistema Operacional:** Escolha a versão do MySQL Workbench que seja compatível com o seu sistema operacional. As opções mais comuns são Windows, macOS e Linux.
- iv. **Baixe e Instale:** Após o download, execute o instalador.
 - i. **Windows:** Execute o instalador baixado e siga as instruções do assistente de instalação.
 - ii. **macOS:** Abra o arquivo .dmg baixado e arraste o ícone do MySQL Workbench para a pasta Applications.
 - iii. **Linux:** As instruções de instalação podem variar de acordo com a distribuição. Geralmente, você pode extrair o arquivo baixado e seguir as instruções de instalação específicas para a sua distribuição.
 - v. **Configuração Inicial:** Quando você abrir o MySQL Workbench pela primeira vez, poderá ser solicitado que você configure uma conexão ao servidor MySQL. Você precisará fornecer as

informações de conexão, como o nome de usuário, senha, host e porta do servidor MySQL que deseja acessar.

- vi. **Uso do MySQL Workbench:** Uma vez configurado, você poderá usar o MySQL Workbench para administrar bancos de dados MySQL. A ferramenta oferece recursos como modelagem visual de bancos de dados, design de esquema, consulta SQL, administração de usuários, importação/exportação de dados e muito mais.

3. ACESSO E UTILIZAÇÃO DOS CÓDIGOS

3.1.COMO ACESSAR OS CÓDIGOS

Para acessar os códigos-fonte, siga os passos simples abaixo:

- i. Abra o seu navegador web de preferência.
- ii. Digite o seguinte URL na barra de endereços:
<https://github.com/niallivmi/websocket-arduino>.
- iii. Pressione "Enter" e você será direcionado para a página principal do repositório.

3.2.NAVEGANDO NO REPOSITÓRIO

Ao chegar na página inicial, você encontrará uma variedade de informações relacionadas ao projeto. A aba "Code" permitirá que você explore os arquivos essenciais do projeto.

3.3.COMO USAR OS CÓDIGOS

Se você deseja utilizar ou estudar o código, os arquivos com extensão ".ino" são particularmente relevantes para o ambiente Arduino.

Caso pretenda realizar contribuições ou sugestões, a seção "Issues" é o local para discutir problemas existentes ou propor melhorias. A área de "Pull Requests" é reservada para submissões de alterações que você deseja que sejam consideradas pelo mantenedor do projeto.

3.4.CLONAR O REPOSITÓRIO

Se você quiser ter uma cópia local dos códigos em sua máquina, utilize o botão verde "Code" na página principal e, em seguida, copie o URL fornecido. Utilize um cliente Git ou o terminal para realizar o clone.

Esperamos que este repositório seja uma fonte valiosa de aprendizado e colaboração. Sinta-se à vontade para explorar, questionar e, se desejar, contribuir para o desenvolvimento contínuo deste projeto.

3.5.CONTEÚDO

Dentro do repositório você irá encontrar alguns arquivos, sendo eles:

- i. **Lamp.html/cronometro.html:** Este é o arquivo responsável pela comunicação WebSocket entre o cliente (navegador) e o servidor (Arduino). Nele estão definidos os parâmetros de conexão e mensagens trocadas entre os dispositivos.
- ii. **Lamp.css:** Este arquivo contém as definições de estilo utilizadas na página, como cores, fontes, posicionamento de elementos e mais.
- iii. **Lamp.js/cronometro.js:** Neste arquivo está localizado o código JavaScript que governa o comportamento interativo da página, como eventos de clique, alteração de texto e outros efeitos visuais.
- iv. **lamp.png/cronometro.png:** Este é o logo da página, que é exibido na parte superior esquerda da tela.
- v. **wslib/wslib.js:** Este é um biblioteca JavaScript que facilita a implementação de comunicação WebSocket em aplicativos web. Ela é usada no arquivo websocket.html.
- vi. **LICENSE:** Este é o arquivo de licença do projeto, que determina os termos e condições under which the software pode ser usado, modificado e distribuído.
- vii. **pomodoro.ino:** Este é o código-fonte do projeto, escrito em linguagem de programação Arduino. Ele descreve como o hardware deve se comportar e quais são suas funções.

4. DESENVOLVIMENTO DO SITE HTML, JAVASCRIPT E CSS

4.1.ESTRUTURA DO SITE HTML E ESTILIZAÇÃO DO CSS

4.1.1. CÓDIGO DO SITE CONTROLADOR

I. Definição do Documento

```
<!DOCTYPE html>  
<html lang="pt-br">
```

<!DOCTYPE html>: Declaração do tipo de documento HTML.

<html lang="pt-br">: Elemento raiz do HTML com a declaração do idioma.

II. Cabeçalho (<head>)

```
<head>
  <meta charset="UTF-8" name="lamp controller">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Controlador</title>
  <link href="https://fonts.googleapis.com/css?family=Cabin+Condensed:700" rel="stylesheet" type="text/css">
  <link rel="stylesheet" href="public/lamp.css">
  <link rel="icon" type="image/png" href="public/Lampada.png">
  <script type="text/javascript" src="public/lamp.js" defer>
</script>
</head>
```

Metadados como charset, título da página, e viewport são definidos.

Referências a folhas de estilo (lamp.css) e ao ícone da página (Lampada.png) são especificadas.

Um script JavaScript (lamp.js) é incluído e marcado como assíncrono para carregar durante o carregamento da página.

III. Corpo (<body>)

Um cabeçalho (<header>) contendo um SVG animado com a palavra "Controle".

O SVG inclui padrões (water) e uma máscara (text-mask) para a animação da água.

A animação de água é criada usando a tag <animate>.

Um artigo (<article>) contendo botões interativos.

IV. Botões Interativos

```
</header>
<article>
  <div class="Botões">
    <button class="BAnimado" id="BConexao">Conectar</button>
    <button class="BAnimado" id="BLigDes">Ligar</button>
    <button class="BAnimado" id="BTempo">Cronômetro</button>
  </div>
</article>
</body>
```

Três botões com as classes BAnimado.

Cada botão tem um id único (BConexao, BLigDes, BTempo) para identificação.

Os botões têm eventos associados a eles no arquivo JavaScript (lamp.js) para manipulação interativa.

V. Conexão com JavaScript

O script JavaScript (lamp.js) é incluído no final da página.

O código JavaScript manipula interações do usuário e se conecta à porta COM4 via WebSocket.

VI. Estilo com CSS

O estilo da página é definido no arquivo CSS (lamp.css), que não foi fornecido aqui.

Classes como BAnimado e estilos específicos para animações podem estar definidos no CSS.

4.1.2. CÓDIGO DO SITE AUXILIAR

I. Definição do Documento

```
<!DOCTYPE html>  
<html lang="pt-br">
```

<!DOCTYPE html>: Declaração do tipo de documento HTML.

<html lang="pt-br">: Elemento raiz do HTML com a declaração do idioma.

II. Cabeçalho (<head>)

```
<head>  
  <meta charset="UTF-8" name="cronometro">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Cronômetro</title>  
  <link rel="stylesheet" href="public/lamp.css">  
  <link rel="icon" type="image/png" href="public/Cronometro.png">  
  <script type="text/javascript" src="public/cronometro.js" defer></script>  
</head>
```

Metadados como charset, título da página, e viewport são definidos.

Referências a folhas de estilo (lamp.css) e ao ícone da página (Cronometro.png) são especificadas.

Um script JavaScript (cronometro.js) é incluído e marcado como assíncrono para carregar durante o carregamento da página.

III. Corpo (<body>)

```
<body>
  <header>
    <h1>Defina o tempo</h1>
  </header>
  <article>
    <form id="Formulario">
      <input type="radio" id="BRLigar" name="LigDes" value="1" title="Selecionar para ligar">
      <label for="BRLigar">Ligar</label>

      <input type="radio" id="BRDesligar" name="LigDes" value="0" title="Selecionar para Desligar">
      <label for="BRDesligar">Desligar</label>

      <br><br><label for="Hora">Defina um horário</label><br>
      <input type="time" id="Hora" name="Hora" title="Informe um horário válido"><br><br>

      <input type="submit" id="Enviar" value="Enviar">
    </form>
  </article>
</body>
```

Um cabeçalho (<header>) contendo um título "Defina o tempo".

Um artigo (<article>) contendo um formulário (<form>) com os seguintes elementos:

Dois botões de rádio (<input type="radio">) com IDs BRLigar e BRDesligar.

Rótulos (<label>) associados aos botões de rádio.

Um campo de entrada de tempo (<input type="time">) com ID Hora.

Um botão de envio (<input type="submit">) com ID Enviar.

IV. Estilo com CSS

O estilo da página é definido no arquivo CSS (lamp.css), que não foi fornecido aqui.

Classes como BAnimado e estilos específicos para o formulário podem estar definidos no CSS.

V. Conexão com JavaScript

O script JavaScript (cronometro.js) é incluído no final da página.

O código JavaScript manipula interações do usuário no formulário.

4.1.3. Código do CSS

I. Estilo do Corpo (body)

```
/*Define o estilo do corpo do HTML*/
body {
  background: linear-gradient(-45deg, #ee7752, #e73c7e, #23a6d5, #23d5ab);
  background-size: 400% 400%;
  animation: gradient 15s ease infinite;
  height: 100%;
  text-align: center;
  overflow: hidden;
}
```

Um plano de fundo é aplicado usando uma combinação linear de cores em um ângulo específico.

A animação (@keyframes gradient) cria uma transição suave entre as cores de fundo, criando um efeito de gradiente em movimento.

O texto é centralizado, a altura é definida como 100%, e o overflow é oculto para garantir um layout limpo.

II. Estilo do Texto (p)

O texto utiliza a fonte Helvetica, com espaçamento de letras, peso da fonte, tamanho e cor especificados.

O texto é centralizado no elemento <p>.

III. Estilo do Botão (button)

Os botões têm um display de bloco, margem inferior, e margem à esquerda para posicionar no centro.

O botão não tem borda, e um raio de borda (border-radius) de 20px cria cantos arredondados.

IV. Estilo do Botão pela Classe (BAnimado)

Os botões com a classe BAnimado têm uma fonte específica, espaçamento de letras, peso da fonte, tamanho e cor.

O cursor é configurado como apontador, e é adicionado um contorno (outline) ao redor do botão para criar um efeito visual.

A transição é aplicada para criar uma animação suave quando o estado do botão muda.

V. Efeito Hover da Classe (BAnimado:hover)

Quando o mouse está sobre o botão, a cor e o contorno são ajustados para criar um efeito visual de destaque.

4.2.INTEGRAÇÃO COM JAVASCRIPT

4.2.1. CÓDIGO DO SITE CONTROLADOR

I. Variáveis Globais

```
var BLigDes // Variável global para o botão de Ligar e Desligar herdado do HTML
var BConexao // Variável global para o botão de Conectar e Desconectar herdado do HTML
var socket = null // Variável global para a conexão WebSocket
let janela // Variável para abrir janela do Cronômetro
var BTempo // Variável global para o botão Cronômetro herdado do HTML
var connectDataQueue = [] // Fila para armazenar os dados de conectar/desconectar
var horaClique
```

BLigDes: Variável global que representa o botão de Ligar/Desligar no HTML.

BConexao: Variável global para o botão de Conectar/Desconectar no HTML.

BTempo: Variável global para o botão do Horário no HTML.

socket: Variável global para a conexão WebSocket.

janela: Variável para armazenar a referência à janela do horário.

connectDataQueue: Fila para armazenar dados de conectar/desconectar.

horaClique: Variável para armazenar a hora do clique em botões.

II. Inicialização no Carregamento da Página

A função associada ao evento load da janela é acionada quando a página é carregada.

Elementos HTML relevantes são associados às variáveis globais.

Event listeners são configurados para os botões Ligar/Desligar, Conectar/Desconectar e do Horário.

III. Função de Ligar e Desligar (LigarDesligar)

```
// Função de Ligar e Desligar
function LigarDesligar() {
  // Altera a palavra do botão
  var BValor = B LigDes.textContent === 'Ligar' ? 'Desligar' : 'Ligar'
  B LigDes.textContent = BValor

  // Captura a hora do clique
  horaClique = new Date().toISOString()

  // Envia os dados para a porta COM5
  if (socket !== null && socket.readyState === WebSocket.OPEN) {
    if (BValor === 'Ligar') {
      socket.send('0')
    } else if (BValor === 'Desligar') {
      socket.send('1')
    }
  }
}
```

Ao clicar no botão de Ligar/Desligar, a função alterna o texto do botão entre 'Ligar' e 'Desligar'.

A hora do clique é capturada e os dados são enviados para a porta COM5 via WebSocket, dependendo do estado do botão.

IV. Função de Conectar e Desconectar (ConectarDesconectar)

Semelhante à função de Ligar/Desligar, a função alterna o texto do botão entre 'Conectar' e 'Desconectar'.

A hora do clique é capturada.

Se o botão for pressionado para 'Desconectar', uma nova conexão WebSocket é estabelecida. Dados na fila são enviados e, em seguida, dados são enviados para a porta COM4.

Se o botão for pressionado para 'Conectar', a conexão existente é encerrada.

V. Função de Abrir o Horário (Abrir)

Ao clicar no botão do Horário, a função envia um comando via WebSocket para a porta COM4.

Uma nova janela (cronometro.html) é aberta.

VI. Eventos WebSocket

O código lida com eventos como `onopen`, `onclose`, e `onerror` para gerenciar a conexão WebSocket.

Quando a conexão é estabelecida, dados na fila são enviados e a porta COM4 recebe dados dependendo do estado do botão.

4.2.2. CÓDIGO DO SITE AUXILIAR

I. Variáveis Globais

`socket`: Variável global para a conexão WebSocket.

`BRLigDes`: Variável global para o botão de rádio "Ligar" no HTML.

`IHora`: Variável global para o campo de horário no HTML.

`Formulario`: Variável global para o formulário no HTML.

`Enviar`: Variável global para o botão de envio.

II. Inicialização no Carregamento da Página

A função associada ao evento `load` é acionada quando a página é carregada.

Elementos HTML relevantes são associados às variáveis globais.

Event listeners são configurados para o formulário, prevenindo o envio padrão, e para o botão de envio.

III. Função de Enviar Dados (EnviarDados)

A função é chamada ao clicar no botão de envio.

O valor do botão de rádio e o valor do campo de horário são capturados.

Se a hora for válida, os dados são enviados para a porta COM4 via WebSocket, formatados como um objeto JSON.

IV. Manipulação do Formulário

O formulário tem um event listener para o evento submit, que impede o envio padrão do formulário.

V. Event Listener do Botão de Envio (Enviar)

Ao clicar no botão de envio, a função captura o valor do botão de rádio e o valor do campo de horário.

Se a hora for válida, a função EnviarDados é chamada para enviar os dados formatados corretamente via WebSocket.

VI. Verificação de Hora Válida

Antes de enviar os dados, o código verifica se a hora inserida é válida.

Se válida, os dados são enviados; caso contrário, uma mensagem de alerta é exibida.

VII. Envio de Dados via WebSocket

Os dados são enviados para a porta COM4 utilizando WebSocket, formatados como um objeto JSON contendo informações sobre a ação agendada, a hora e a ação a ser executada.

4.3.SERVIDOR NODE.JS

I. Conexão Serial (Arduino)

```
// Requisição SerialPort para conectar COM5 - Arduino
const { SerialPort } = require("serialport")

// Construtor Port COM4
const port = new SerialPort({
  // Porta Serial
  path: "COM5",
  // Baudrate
  baudRate: 9600,
  dataBits: 8,
  stopBits: 1,
  parity: "none",
})
console.log('Conexão estabelecida: COM5')
```

O trecho inicial do código incorpora a biblioteca SerialPort para estabelecer uma comunicação serial com um dispositivo Arduino conectado à porta COM5.

O objeto port é instanciado com a configuração da porta serial, incluindo o caminho da porta, taxa de baud, número de bits de dados, bits de parada e paridade.

A mensagem "Conexão estabelecida: COM5" é exibida no console para indicar uma conexão bem-sucedida.

II. WebSocket

```
// Requisição WebSocket para conectar 8080 - Site HTML
const WebSocket = require('ws')

// Importe a biblioteca node-schedule
const schedule = require('node-schedule');

// Construtor WS 8080 - localhost:8080
const server = new WebSocket.Server({ port: 8080 })

// "Conectado"
server.on('connection', (socket) => {
  console.log('Conexão estabelecida: localhost:8080')
```

O módulo WebSocket é requisitado para criar um servidor WebSocket na porta 8080, possibilitando a comunicação bidirecional entre o Node.js e um site HTML hospedado localmente em localhost:8080.

O servidor WebSocket é instanciado através do objeto server.

III. MySQL Connection

```
// Requisição Mysql para conectar com o MySQL
const mysql = require('mysql2')

// Configurações da conexão
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'Batata600237084!',
  database: 'conexao_node'
})
```

A biblioteca mysql2 é solicitada para possibilitar a comunicação com um banco de dados MySQL.

As configurações de conexão, como host, usuário, senha e nome do banco de dados, são definidas no objeto connection.

A mensagem "Conexão estabelecida: MySQL" é exibida no console ao verificar uma conexão bem-sucedida.

IV. Funcionalidades Específicas

i. Recebimento de Mensagens

Dentro do evento 'connection', o código monitora o evento 'message' para tratar mensagens recebidas do WebSocket.

Cada mensagem é exibida no console e encaminhada para a porta serial (Arduino) para execução do comando correspondente.

ii. Armazenamento de Eventos

A função storeButtonClickEvent é invocada para registrar informações sobre eventos no banco de dados MySQL.

O horário do evento é obtido utilizando a biblioteca moment com ajuste de fuso horário para 'America/Sao_Paulo'.

iii. Agendamento de Ações

A função agendarAcao permite a programação de ações (como ligar ou desligar) em horários predefinidos.

A biblioteca node-schedule é empregada para criar regras de agendamento, e as ações são executadas conforme a programação.

4.4.ARDUINO

```

int statusRelay; // Variável para armazenar o estado do relé

void setup() {
  Serial.begin(9600);
  // Configurar o pino do relé como saída
  pinMode(13, OUTPUT); // Suponha que você esteja usando o pino 13 para o controle do relé
  digitalWrite(13, HIGH); // Inicialmente, ligue o relé (HIGH)
}

void loop() {
  if (Serial.available() > 0) {
    statusRelay = Serial.read();
    if (statusRelay == '0') {
      digitalWrite(13, HIGH); // Desliga o relé (LOW)
    } else if (statusRelay == '1') {
      digitalWrite(13, LOW); // Liga o relé (HIGH)
    }
  }
}

```

I. Declaração de Variável

statusRelay é uma variável inteira utilizada para armazenar o estado do relé, que será recebido pela porta serial.

II. Configuração Inicial (setup)

Serial.begin(9600);: Inicia a comunicação serial com uma taxa de 9600 bps.

pinMode(13, OUTPUT);: Configura o pino 13 como saída para controlar o relé.

digitalWrite(13, HIGH);: Inicialmente, o relé é ligado (nível lógico HIGH).

III. Loop Principal (loop)

if (Serial.available() > 0): Verifica se há dados disponíveis na porta serial.

statusRelay = Serial.read();: Lê o caractere recebido da porta serial e armazena em statusRelay.

Se statusRelay for '0', o relé é desligado (digitalWrite(13, HIGH);).

Se statusRelay for '1', o relé é ligado (digitalWrite(13, LOW);).

4.5.BANCO DE DADOS MYSQL

I. Criação da tabela

Cria uma tabela chamada "registros" se ela não existir.

A tabela tem as seguintes colunas

ID: Um inteiro não nulo e autoincremental, que serve como chave primária.

ID_BOTÃO: Uma cadeia de caracteres de até 255 caracteres, com um valor padrão nulo.

COMANDO: Uma cadeia de caracteres de até 255 caracteres, com um valor padrão nulo.

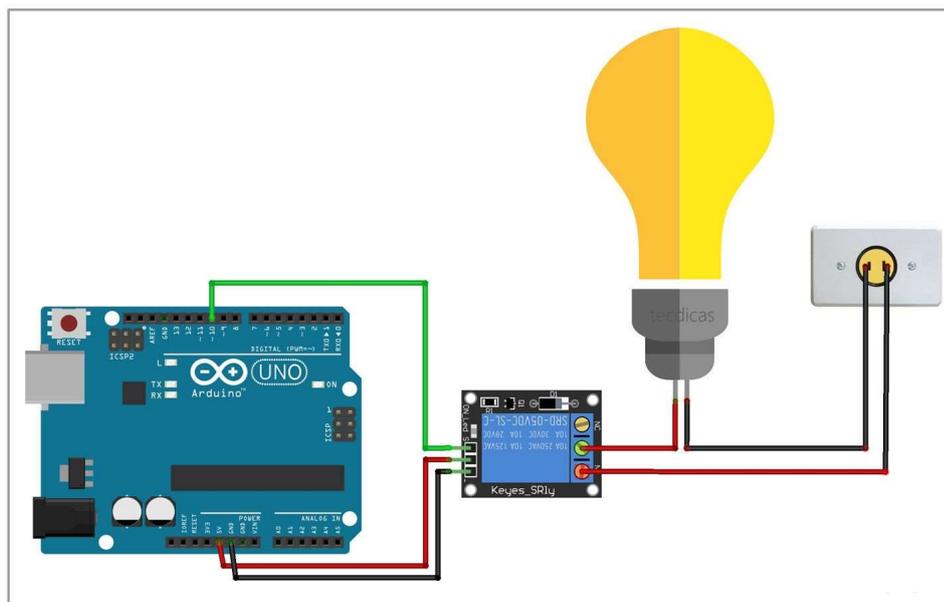
HORÁRIO: Um campo de data e hora, com um valor padrão nulo.

5. PROJETO ARDUINO

5.1. MATERIAIS

- i. 1 Arduino Uno Rev 3 com cabo USB
- ii. 1 Módulo relé de 1 canal ou 2 canais 5V - 10A
- iii. 1 Lâmpada de 60W - Base E27
- iv. 1 Soquete E27
- v. Fios jumpers Macho - Fêmea
- vi. 2 Pedacos de fios para ligar no soquete e relé
- vii. 1 Cabo com plug de tomada para ligar a lâmpada
(Recomendável para segurança)
- viii. 1 Alicates de corte de sua preferência
- ix. 1 Chave do tipo Philips ou Fenda, depende do soquete.
- x. 1 Fita isolante

5.2. MONTAGEM



FONTE: Como ligar uma lâmpada no Arduino Uno. Disponível em: <<https://tecdicas.com/como-ligar-uma-lampada-no-arduino-uno/#:~:text=Como%20ligar%20uma%20%C3%A2mpada%20no%20Arduino%20Uno%201%3E>>. Acesso em: 28 nov. 2023.

6. RECURSOS ADICIONAIS

6.1. LINKS ÚTEIS PARA APRENDIZADO DE NODE.JS, ARDUINO E MYSQL

6.1.1. NODE.JS

- Documentação Oficial do Node.js: A documentação oficial é uma fonte completa para aprender sobre o Node.js, desde conceitos básicos até tópicos avançados. (Acesso: <https://nodejs.org/en/docs>).
- Node.js Crash Course: Um vídeo tutorial introdutório que cobre os fundamentos do Node.js em um formato de crash course. (Acesso: https://www.youtube.com/watch?v=zb3Qk8SG5Ms&ab_channel=NetNinja).

6.1.2. ARDUINO

- Arduino Official Website: O site oficial da Arduino oferece uma variedade de recursos, desde tutoriais até documentação técnica. (Acesso: <https://www.arduino.cc/>).
- Arduino Programming Course: Um curso online na plataforma Udemy que abrange a programação de placas Arduino para iniciantes. (Acesso: <https://www.udemy.com/course/arduino-programming-for-beginners/>).

6.1.3. MYSQL WORCHBENCH

- MySQL Documentation: A documentação oficial do MySQL fornece informações detalhadas sobre o uso e administração do banco de dados. (Acesso: <https://dev.mysql.com/doc/>).
- MySQL Tutorial: Um tutorial interativo no W3Schools que oferece uma introdução prática ao MySQL e suas consultas. (Acesso: <https://www.w3schools.com/mysql/>).

Lembrando que a busca por tutoriais, fóruns e plataformas de aprendizado online também pode ser uma abordagem eficaz para aprofundar seus conhecimentos nessas tecnologias.