



Título do Projeto:

**Desenvolvimento de um Sistema Embarcado para Monitoramento do Nível e da Temperatura da Água de uma caixa d'água**

**GABRIEL DA SILVA NASCIMENTO**

**LÍLIAN HONORIO TEIXEIRA**

**LUCAS CORDEIRO VIEIRA**

**LUIZ MEDEIROS NETO**

Área do conhecimento: Engenharia de Computação

Área Temática: Tecnologia da Informação e Comunicação

Abril de 2023

## **1. INTRODUÇÃO**

A automação do controle de nível e da temperatura da água possui inúmeros benefícios, além disso é totalmente aplicável e possui um custo acessível para os seus usuários. Em questões de usabilidade, o projeto possui alta acessibilidade devido ao controle automático dos componentes eletrônicos. Logo uma vez instalado o sistema, o usuário não precisa se preocupar com a eletrônica do sistema embarcado e deve usufruir expressamente do mesmo.

Em primeiro momento, cita-se a importância do sistema de controle automático da água para a eliminação de atividades manuais constantes, como as de ligar e desligar a bomba d'água sempre que o reservatório estiver em nível baixo ou em nível alto. Assim, o usuário estará em total conforto na sua residência mediante ao ciclo de abastecimento de água e monitoramento da temperatura de maneira automatizada.

Ressalta-se, ainda, a importância ambiental desse sistema embarcado ao proporcionar uma economia enorme de água e energia reduzindo o desperdício. Além disso, ao controlar a temperatura da água é possível prolongar a vida útil da caixa d'água e da encanação da residência ou complexo industrial, citando-se ainda, que o usuário do sistema terá contato reduzido com circuitos elétricos, diminuindo portanto, os índices gerais de acidentes com equipamentos eletroeletrônicos ao manipulá-los sem os conhecimentos necessários. Caso precise, o consumidor também poderá fazer o monitoramento remoto do nível da água da sua residência, precavendo-se contra desperdícios ou falhas no sistema.

### **1.1 OBJETIVOS**

#### **1.1.1 OBJETIVO GERAL**

O objetivo principal deste projeto é desenvolver um sistema embarcado capaz de fazer o controle da temperatura e nível da água para o seu aquecimento e abastecimento, caso necessário.

#### **1.1.2 OBJETIVOS ESPECÍFICOS**

- Medir a temperatura da água;

- Medir o nível do reservatório de água;
- Controlar uma resistência (SSR/Relé);
- Controlar uma bomba d'água (SSR/Relé);
- Monitorar o nível e temperatura da água em tempo real;
- Utilizar IHM para informar e controlar a temperatura da água, o nível em tempo real da água no reservatório.

## 1.2 DESCRIÇÃO

O projeto irá integrar os componentes eletrônicos ESP WROOM 32, sensor sônico, bomba d'água, sensor de temperatura e aquecedor para controlar a temperatura e o nível da água.

Uma vez detectado pelo sensor ultrassônico, que o nível do reservatório esteja baixo, o microcontrolador deve ligar a bomba d'água e assim que a caixa estiver cheia, ela deve ser desligada.

Convém lembrar, ainda, que o esp32 irá controlar uma resistência para elevar a temperatura da água caso seja necessário, bem como fará esse controle por meio do monitoramento das informações obtidas através do sensor que detecta se a água está quente ou fria.

## 2. TABELA DE COMPONENTES

Quantidade	Descrição
1	ESP WROOM 32
~	Fios
1	Fonte de alimentação
1	Caixa d'água

1	Bomba d'água
1	Sensor sônico
1	Sensor de temperatura d'água
1	Aquecedor
1	IHM

### 3. CASOS DE USO

#### 3.1 - Medir a temperatura da água.

A. **Descrição:** A temperatura da água deve ser acompanhada pelo ESP32 a cada cinco minutos.

B. **Ator(es):** ESP32, Sensor de Temperatura, IHM.

C. **Pré-condições:** O sensor de temperatura e o IHM devem estar em perfeito estado e também devem estar introduzidos corretamente nas suas devidas posições. Além disso, a conexão entre o sensor, o IHM e o ESP32 deve ser feita corretamente e o sistema deve estar devidamente alimentado por uma fonte de energia.

D. **Pós-condições:** A medição da temperatura deve ficar registrada no sistema para ser utilizada posteriormente exibida para o usuário e integrada ao aquecedor.

#### E. Requisitos Funcionais:

a. **RF01:** O microcontrolador deve exclusivamente ler a temperatura da água por meio de um sensor de temperatura sem que haja interferências externas que provoquem erros na medição.

b. **RF02:** Disponibilizar uma variável no programa que armazene a última medição da temperatura convertida para Celsius;

- c. **RF03:** O sistema deve ser capaz de informar essa temperatura ao usuário por meio do IHM ou outro tipo de interface.
- d. **RF04:** Leituras inválidas de temperatura devem ser descartadas pelo sistema;

**F. Requisitos Não Funcionais:**

- a. **RNF01:** Confiabilidade: o sistema deve ser confiável ao ponto de sempre estar registrando as temperaturas e caso haja repetidas falhas, o driver deve ser reiniciado ou devem ser feitas manutenções no sistema.
- b. **RNF02:** Manutenibilidade: O sistema deve ser manutenível, ou seja, o IHM e o sensor de temperatura devem ser facilmente substituídos sem que haja alterações no código fonte.
- c. **RNF03:** Resistência: O IHM e o sensor de temperatura devem ser resistentes às condições que lhe forem expostas.
- d. **RNF04:** Interoperabilidade: O sensor de temperatura e o IHM devem possuir boa interoperabilidade para fazer uma comunicação estável com o ESP32.
- e. **RNF05:** Eficiente energeticamente: O sistema deve ser projetado para consumir uma menor quantidade de energia.
- f. **RNF06:** Tempo de resposta: O sensor de temperatura deve fornecer as informações o quanto antes para que o sistema funcione da melhor forma.

**3.2 - Medir o nível de água do reservatório.**

- A. **Descrição:** O microcontrolador deve ser capaz de medir o nível de água do reservatório por meio do sensor sônico a cada uma hora.
- B. **Ator(es):** ESP32, Sensor de Temperatura, IHM.
- C. **Pré-condições:** O sensor sônico deve estar instalado corretamente na caixa d'água, além disso a conexão com o ESP32 e o IHM também

deve estar feita corretamente e o sistema precisa estar alimentado por uma fonte de energia.

- D. **Pós-condições:** O sistema deve armazenar corretamente o nível da água para que possa ser acionada a bomba d'água corretamente. Além disso, os dados devem ser exibidos por meio do IHM ou outra interface.

**E. Requisitos Funcionais:**

- a. **RF01:** O nível da água do reservatório deve ser medido de forma precisa, sem que haja interferências externas.
- b. **RF02:** A última medição do nível da água deve ficar armazenada em litros numa variável no software embarcado.
- c. **RF03:** O sistema deve ser capaz de informar o nível da água em litros ao usuário por meio do IHM ou outro tipo de interface.
- d. **RF04:** Após medido o nível da água, o ESP32 deve decidir se irá ativar a bomba;

**F. Requisitos Não Funcionais:**

- a. **RNF01:** Confiabilidade: o sistema deve ser confiável ao ponto de sempre estar registrando os níveis da água corretamente e caso haja repetidas falhas, o driver deve ser reiniciado ou devem ser feitas manutenções no sistema.
- b. **RNF02:** Manutenibilidade: O sistema deve ser manutenível, ou seja, o sensor sônico deve ser facilmente substituído sem que haja alterações no código fonte.
- c. **RNF03:** Resistência: O sensor sônico e o IHM devem ser resistentes às condições que lhe forem expostas.
- d. **RNF04:** Interoperabilidade: O sensor sônico e o IHM devem prover uma boa interoperabilidade para fazer uma comunicação estável com o ESP32.

- e. **RNF05:** Eficiente energeticamente: O sistema deve ser projetado para consumir uma menor quantidade de energia.
- f. **RNF06:** Tempo de resposta: O sensor sônico deve fornecer as informações o quanto antes para que o sistema funcione da melhor forma.

### **3.3 - Controlar uma resistência (SSR/Relé).**

- A. **Descrição:** Controlar uma resistência para aquecer a água sempre que for necessário.
- B. **Ator(es):** ESP32, Aquecedor.
- C. **Pré-Condições:** A alimentação do sistema deve estar acontecendo da maneira correta, a resistência deve estar imersa pela água do reservatório e deve ser compatível com o nível máximo da caixa d'água para que funcione corretamente, o sensor de temperatura deve estar funcionando com acerto para que não haja ativações indevidas e a conexão da resistência também deve estar conforme as regras de instalação.
- D. **Pós-Condições:** A resistência deve proporcionar que o tempo de aquecimento da água do reservatório seja breve e tenha atingido a temperatura correta. Além disso, o sistema deve identificar que a água atingiu a temperatura desejada e deve desativar a resistência.
- E. **Requisitos Funcionais:**
  - a. **RF01:** O sistema deve fornecer a temperatura ideal da água, sem que haja excessos de aquecimento.
  - b. **RF02:** Após atingir a temperatura ideal, a resistência deve ser desativada.
  - c. **RF03:** A resistência deve estar bem instalada para que sejam evitados acidentes.
- F. **Requisitos Não Funcionais:**
  - a. **RNF01:** Confiabilidade: o sistema deve ser confiável ao ponto de responder corretamente aos estímulos gerados pelo ESP32

para que falhas sejam evitadas e caso ocorram, o sistema deve ser reiniciado ou feitas manutenções.

- b. **RNF02:** Manutenibilidade: O sistema deve ser manutenível, ou seja, a resistência deve ser facilmente substituída sem que haja alterações no código fonte.
- c. **RNF03:** Resistência: A resistência deve ter um longo período de vida útil.
- d. **RNF04:** Interoperabilidade: A resistência deve possuir uma boa interoperabilidade para fazer uma comunicação estável com o ESP32.
- e. **RNF05:** Eficiente energeticamente: O sistema deve ser projetado para consumir uma menor quantidade de energia.
- f. **RNF06:** Tempo de resposta: A resistência deve ser desativada imediatamente após a água atingir a temperatura ideal.

### **3.4 - Controlar uma bomba d'água (SSR/Relé).**

- A. Descrição:** Controlar uma bomba d'água para abastecimento do reservatório.
- B. Ator(es):** ESP32, Bomba d'água.
- C. Pré-Condições:** A bomba d'água deve estar devidamente conectada ao ESP32 e deve estar recebendo a água encanada corretamente pela instalação. Além disso, a bomba deve estar sendo alimentada junto ao circuito conforme as medidas corretas, ou seja, deve estar acoplada ao SSR ou Relé para que possa ser controlada pelo ESP32. O programa embarcado também deve estar devidamente configurado para garantir o fluxo desse caso de uso.
- D. Pós-Condições:** A bomba deve ser desligada corretamente pelo ESP32 quando o reservatório estiver devidamente abastecido. O sistema deve garantir que a bomba não opere novamente até o nível da caixa d'água estar abaixo do recomendado para que não haja desperdício de energia e a vida do aparelho seja preservada.



#### **E. Requisitos Funcionais:**

- a. **RF01:** O sistema deve ativar a bomba assim que a água do reservatório estiver abaixo do recomendado.
- b. **RF02:** O sistema deve desativar a bomba assim que a água do reservatório estiver abaixo do recomendado.
- c. **RF03:** Caso ocorra erros na ativação da bomba, o sistema deve lidar com a falha, ser reiniciado ou receber manutenção.

#### **F. Requisitos Não Funcionais:**

- a. **RNF01:** Confiabilidade: o sistema deve ser confiável ao ponto da bomba d'água responder imediatamente aos estímulos gerados pelo ESP32 para que falhas sejam evitadas e caso ocorram, o sistema deve ser reiniciado ou feitas manutenções.
- b. **RNF02:** Manutenibilidade: O sistema deve ser manutenível, ou seja, a bomba d'água deve ser facilmente substituída sem que haja alterações no código fonte.
- c. **RNF03:** Resistência: A bomba d'água deve ter um longo período de vida útil.
- d. **RNF04:** Interoperabilidade: A bomba d'água deve possuir uma boa interoperabilidade para fazer uma comunicação estável com o ESP32 juntamente ao SSR ou relé.
- e. **RNF05:** Eficiente energeticamente: O sistema deve ser projetado para consumir uma menor quantidade de energia.
- f. **RNF06:** Tempo de resposta: A bomba d'água deve ser desativada imediatamente após a água atingir o nível ideal programado no microcontrolador.

### **3.5 - Controlar o IHM.**

**A. Descrição:** O IHM deve exibir corretamente os dados enviados pelo ESP32.

**B. Ator(es):** ESP32, IHM.

**C. Pré-Condições:** É necessário que o circuito esteja alimentado corretamente por uma fonte de energia elétrica, o IHM deve ser compatível com o ESP32 e deve estar conectado corretamente, além disso as informações a serem exibidas pelo IHM precisam estar armazenadas corretamente no ESP32 por meio do uso correto de variáveis.

**D. Pós-Condições:** O valor exibido precisa ser o correto e o IHM deve responder caso seja requisitado sempre que houver atualizações no ESP32 sendo administradas pelos sensores de temperatura e o sônico.

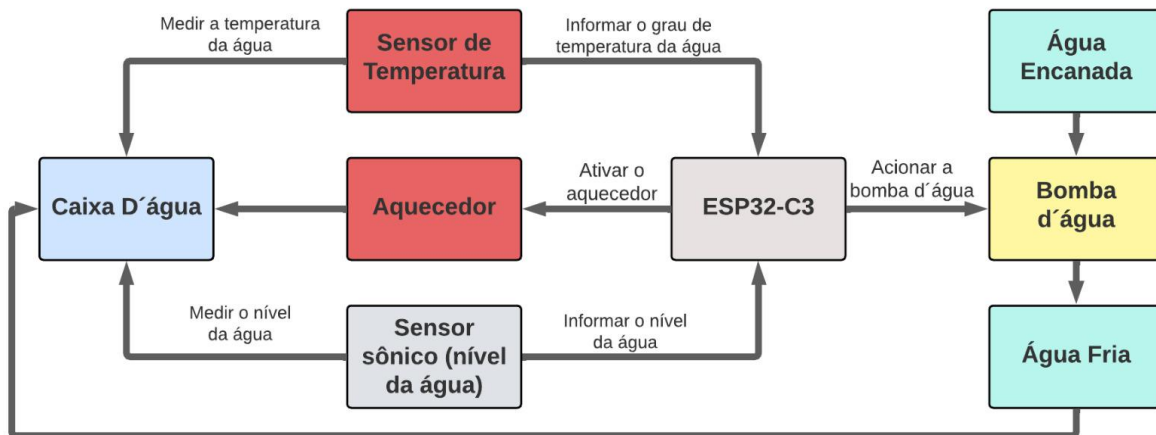
**E. Requisitos Funcionais:**

- a. **RF01:** O IHM deve exibir corretamente as informações relacionadas ao reservatório e medidas pelos sensores.
- b. **RF02:** O sistema deve exibir uma mensagem de alerta no IHM caso ele não esteja operando corretamente.

**F. Requisitos Não Funcionais:**

- a. **RNF01:** Confiabilidade: o sistema deve ser confiável ao ponto da IHM responder imediatamente aos estímulos gerados pelo ESP32, caso contrário uma mensagem de alerta deve ser exibida no IHM.
- b. **RNF02:** Manutenibilidade: O sistema deve ser manutenível, ou seja, a IHM deve ser facilmente substituída sem que haja alterações no código fonte.
- c. **RNF03:** Resistência: A IHM deve ter um longo período de vida útil.
- d. **RNF04:** Interoperabilidade: A IHM deve possuir uma boa interoperabilidade para fazer uma comunicação estável com o ESP32.
- e. **RNF05:** Eficiente energeticamente: O sistema deve ser projetado para consumir uma menor quantidade de energia.

#### 4. DIAGRAMA ILUSTRATIVO



#### 5. MATERIAIS E MÉTODOS

##### 5.1 DESCRIÇÃO DAS ATIVIDADES NO DESENVOLVIMENTO DOS COMPONENTES

Para o desenvolvimento dos componentes, primeiramente foi necessário a construção das funcionalidades mediante aos requisitos estabelecidos, tanto funcionais, quanto não funcionais. Assim, o esp32 foi conectado junto aos componentes de hardware como o sensor de temperatura ds18b20, o HCSR04 e o motor, para estabelecer o controle de diferentes tensões, foi necessária a utilização de relés conectados a fontes de alimentação externas.

No componente ds18b12 foram construídas funções para realizar escrita e leitura, além de seleção, resolução e configuração do sensor, como a `ds18b20_requestTemperatures()` e a `ds18b20_getTempC()` para retornar a temperatura em graus celsius.

Na realização da leitura de distância, primeiramente foi necessário inicializar as configurações dos pinos de *trigger* e *echo* do esp32 em modo de output e input mediante a função de realização da leitura da distância com a calibração escolhida.

Dessa forma, para a ativação e desativação do motor e da resistência são necessários relés chaveados a fontes de alimentação externa. Assim, cita-se como exemplo, a construção das funções de `MotorControl_active()` e `MotorControl_desactive()` foram elaboradas com essa finalidade.

No ponto de start da aplicação, são estados os modos de operação dos pinos do esp32, para controlar o motor, resistência, temperatura, sensor ultrassônico e fornecer a interface de usuário por meio da lib de wifi e servidores do ESP-IDF e é nessa função onde acontece toda a lógica de controle e atuação dos componentes do projeto.

## **5.2 LISTA DE MATERIAIS**

### **5.2.1 ESP-WROOM-32**



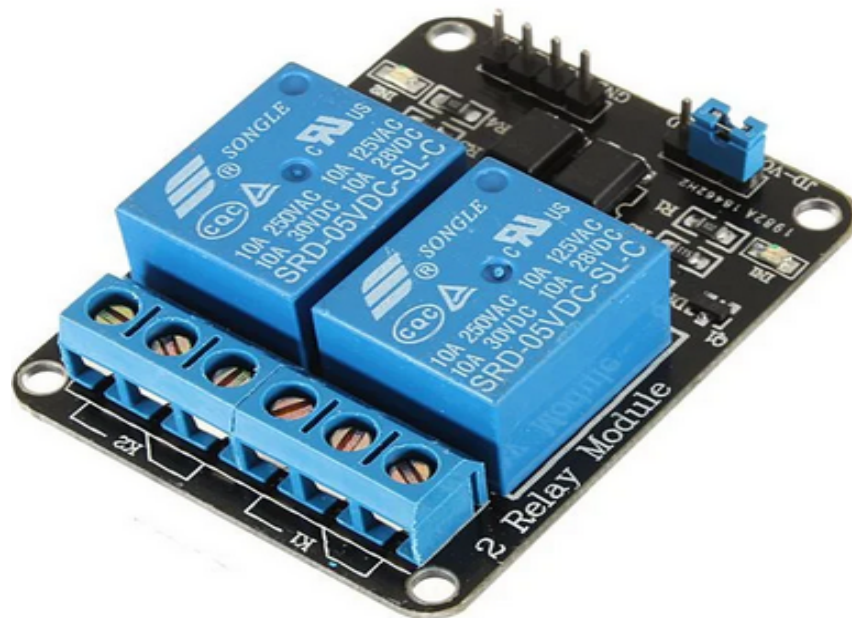
### **5.2.2 HC-SR04**



### 5.2.3 DS18B20



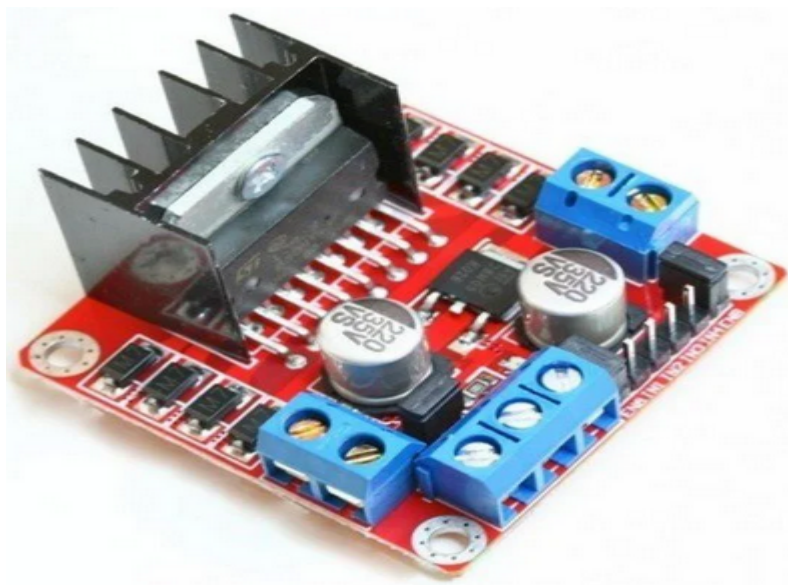
### 5.2.4 MÓDULO RELÉ 2 CANAIS OPERAÇÃO EM 3.3V



#### 5.2.5 AQUECEDOR FLEXÍVEL DE SILICONE 100W 220V



#### 5.2.6 PONTE H DUPLA L298N



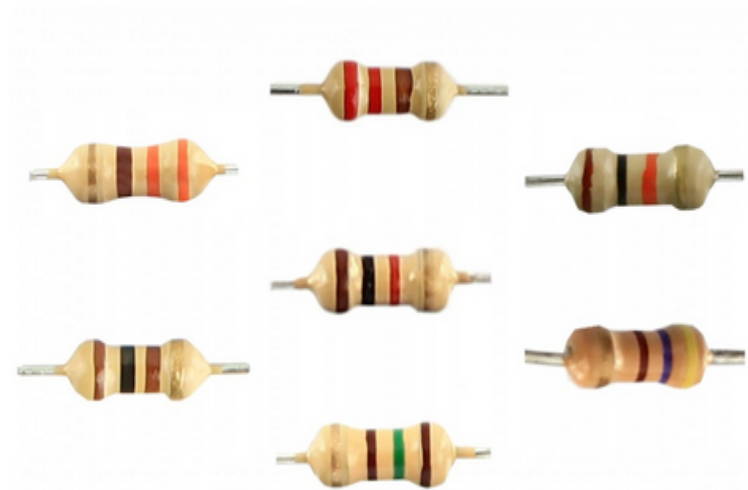
#### 5.2.7 BOMBA D'ÀGUA MINI 12V



### 5.2.8 JUMPERS



### 5.2.9 RESISTORES



## 5.3 LISTA DAS BIBLIOTECAS UTILIZADAS

Biblioteca	Importância para o Projeto
stdio.h	Leitura e escrita em variáveis a serem tratadas pelas requisições http, lógica de strings.
stdint.h	Permite estabelecer tamanhos específicos para variáveis do tipo inteiro como foi feito na variável <i>uint8_t ds18b20_rom_code</i> na elaboração do componente ds18b20.
stdbool.h	Trabalho com variáveis booleanas em true/false.



stdlib.h	Alocação de memória, conversão entre tipos, string to float/int retornadas por requisições.
string.h	Funções para comparação de strings, que podem ser constantes ou respostas de uma requisição http, por exemplo.
freertos/FreeRTOS.h	Implementar as tarefas para o controle dos componentes utilizados no projeto, disparar eventos. Função <i>app_main()</i> .
freertos/task.h	Elaborar as tasks do programa a serem gerenciadas pela função main para estabelecer o controle correto dos componentes.
esp_timer.h	Funções do temporizador para medição das distâncias por meio da função “ <i>esp_timer_get_time()</i> ” utilizada por exemplo.
freertos/timers.h	Funções para controle dos timers utilizados nos componentes.
esp_log.h	Registro de logs por meio do SDK do ESP32.
esp_system.h	Funções macro para a manipulação do sistema.
esp_wifi.h	Estabelecer conexão wifi por meio da API, disparar e registrar eventos relacionados ao controle dos componentes.
esp_event.h	Gerenciar callbacks de eventos relacionados aos componentes, por exemplo.
esp_http_server.h	Implementar um http server por meio do esp32, a fim de definir as rotas do servidor http e gerenciar requisições feitas por clientes.
DS18B20.h	Funções para a comunicação com o sensor por meio do OneWire.
DallasOneWire\onewire\onewire.h	Estabelecer o protocolo de comunicação com o sensor de



	temperatura.
DallasOneWire\drivers\ow_driver.h	Estabelecer o protocolo de comunicação com funções de conversão e calibre.
RelayModule.h	Funções para inicializar, ativar e desativar os relés.
MotorControl.h	Estabelecer a interface de controle da ativação da bomba
HC_SR04.h	Módulo para controle do sensor ultrassônico por meio da elaboração dos componentes

## 6. RESULTADOS

Durante a elaboração do projeto, os resultados alcançados foram o controle da bomba d'água por meio do sensor ultrassônico e relés utilizados para medir o nível do reservatório e acionar a bomba, faltando apenas a calibração para o limite mínimo. Além disso, também foram alcançados resultados quanto à medição da temperatura da água, dessa vez de forma completa para acionar e desacionar a resistência para aquecimento, desta vez com todas as especificações do projeto cumpridas.

Destaca-se, ainda, que a princípio foi elaborada uma interface de usuário baseada no módulo de wifi do ESP-IDF, que apesar de funcionar corretamente nos primeiros testes, ocorreram problemas posteriormente depois de algumas atualizações na última versão da biblioteca.

Desse modo, alguns objetivos e especificações não foram alcançadas. Entretanto, o projeto resultou em uma versão funcional, ativando e desativando a bomba e a resistência conforme níveis e temperatura da água, mesmo que sem tanto controle por parte do usuário final.

## 7. CONCLUSÃO

Infere-se, que, apesar dos desafios encontrados pelo caminho, o projeto foi bastante produtivo, ao pôr os membros em prova quanto a utilização dos drivers de

controle de periféricos de um projeto com o framework ESP-IDF. O projeto pôs em prática, tanto conhecimentos de hardware quanto conhecimentos de software, estimulando também práticas de pesquisa e leitura de materiais de especificações. Por mais que todos os objetivos estabelecidos não foram todos alcançados, para uma primeira versão final, grande parte dos objetivos foram alcançados e boas práticas foram trabalhadas.