

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
DE MINAS GERAIS – *CAMPUS* BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Luiz Augusto da Silva

**DESENVOLVIMENTO DE UM SISTEMA IOT PARA MONITORAMENTO DE
RESERVATÓRIOS: um estudo de caso no IFMG - *Campus* Bambuí**

Bambuí – MG

2024

LUIZ AUGUSTO DA SILVA

**DESENVOLVIMENTO DE UM SISTEMA IOT PARA MONITORAMENTO DE
RESERVATÓRIOS: um estudo de caso no IFMG - *Campus* Bambuí**

Trabalho de conclusão de curso apresentado ao Curso Bacharelado em Engenharia de Computação do Instituto Federal de Minas Gerais – *Campus* Bambuí para obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Carlos Renato Nolli

Bambuí – MG

2024

Catálogo na Fonte Biblioteca IFMG - Campus Bambuí

S586d Silva, Luiz Augusto da.
Desenvolvimento de um sistema IOT para monitoramento de reservatórios: um estudo de caso no IFMG - Campus Bambuí. / Luiz Augusto da Silva. – 2024.
45 f. : il. ; color.

Orientador: Carlos Renato Nolli.
Trabalho de Conclusão de Curso (graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG, Curso Bacharelado em Engenharia de Computação, 2024.

1. ESP 32. 2. Microcontroladores. 3. Medição de nível. I. Nolli, Carlos Renato. II. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG. III. Título.

CDD 371.3



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Bambuí
Diretoria de Ensino
Departamento de Engenharia e Computação
Faz. Varginha - Rodovia Bambuí/Medeiros - Km 05 - Caixa Postal 05 - CEP 38900-000 - Bambuí - MG
37 3431 4900 - www.ifmg.edu.br

ATA DE DEFESA DO TCC

Aos **11** dias do mês de **março** do ano de **2024**, às **18:00** horas, sob a presidência de **Carlos Renato Nolli**, o discente **Luiz Augusto da Silva** do Curso de **Bacharelado em Engenharia de Computação**, R.A nº **48504** do IFMG – Campus Bambuí, defendeu o Trabalho de Conclusão de Curso (TCC) intitulado “**SISTEMA IOT PARA MONITORAMENTO DE RESERVATÓRIOS**” e foi **APROVADO** com a nota **80,33**, que está condicionada ao cumprimento dos procedimentos pós-defesa do TCC.

Deverá apresentar o trabalho com as devidas modificações em formato pdf, em **01/04/2024** (primeiro dia útil depois de 20 dias corridos após a data da defesa) à Coordenação de TCC. O não cumprimento dos procedimentos pós-defesa de TCC até a data estipulada, implica em atribuição de nota ZERO e consequente **REPROVAÇÃO**.

Alterações sugeridas pela banca examinadora e outras observações pertinentes à defesa:

- observar as considerações entregues pela banca
- mudar o título do trabalho

Bambuí, 21 de fevereiro de 2024.



Documento assinado eletronicamente por **Carlos Renato Nolli, Professor**, em 11/03/2024, às 19:10, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Francisco Heider Willy dos Santos, Professor**, em 11/03/2024, às 19:11, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Gabriel da Silva, Professor**, em 11/03/2024, às 19:11, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Luiz Augusto da Silva, Usuário Externo**, em 11/03/2024, às 19:14, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **1840023** e o código CRC **F461173E**.

RESUMO

O problema da entrada de ar nas tubulações de água é uma questão recorrente na rede de abastecimento do IFMG - *Campus* Bambuí, acarretando uma série de complicações que afetam tanto a comunidade do *Campus* quanto o próprio sistema de fornecimento de água. Como proposta de solução para esse problema, este trabalho propôs o desenvolvimento de um sistema IoT para monitoramento de reservatórios como solução. O objetivo foi criar um sistema capaz de alertar os responsáveis pela manutenção sempre que houvesse risco de entrada de ar nas tubulações. A implementação desse projeto envolveu a utilização de diversas tecnologias, incluindo o microcontrolador ESP32, o pacote de recursos Selenium e o framework para aplicações Web Django. Após o desenvolvimento e a validação do sistema, constatou-se que essa solução, no caso, o envio do alerta de que o nível estava em um estado crítico, apresentou redução considerável no tempo de tomada de decisão, o que comprovou a eficácia da solução.

Palavras-chave: ESP 32, Microcontroladores, Medição de nível, *Whatsapp Web*, Telemetria

ABSTRACT

The problem of air entering the water pipes is a recurring issue in the IFMG - *Campus* Bambuí supply network, causing a series of complications that affect both the *Campus* community and the water supply system itself. As a proposed solution to this problem, this work proposed the development of an IoT system for monitoring reservoirs as a solution. The objective was to create a system capable of alerting those responsible for maintenance whenever there was a risk of air entering the pipes. The implementation of this project involved the use of several technologies, including the ESP32 microcontroller, the Selenium resource package and the Django web application framework. After developing and validating the system, it was found that this solution, in this case, sending an alert that the level was in a critical state, presented a considerable reduction in decision-making time, which proved the effectiveness of the solution.

Keywords: ESP 32, Microcontroller, Level measurement, Whatsapp Web, Telemetry

LISTA DE FIGURAS

Figura 1: Rede de abastecimento.....	11
Figura 2: Formas de montagem de uma chave de nível	15
Figura 3: Rede de computadores	16
Figura 4: A pilha de protocolos da Internet.....	17
Figura 5: Requisição HTTP.....	18
Figura 6: Elementos básicos de um sistema embarcado.....	19
Figura 7: ESP 32.....	20
Figura 8: Componentes ESP 32.....	21
Figura 9: Diagrama do sensor de nível.....	27
Figura 10: Sensor de Nível de Água com Boia Horizontal	28
Figura 11: Esquema do circuito elétrico.....	29
Figura 12: Placa de circuito impresso	30
Figura 13: Processo do envio de mensagens	32
Figura 14: Diagrama de máquina de estado	33
Figura 15: Diagrama da solução.....	34
Figura 16: Diagrama do reservatório principal.....	36
Figura 17: Sensor de nível.....	37
Figura 18: Placa do sistema embarcado	38
Figura 19: Sistema embarcado finalizado	38
Figura 20: Sensor de nível implantado.....	40
Figura 21: Sistema embarcado implantado	40

LISTA DE TABELAS

Tabela 1: Tabela verdade dos casos de leitura do sensor	31
Tabela 2: Sprint Backlog e Duração.....	35

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

DSR – *Design Science Research*

GND – *Ground*

GPIO – *General Purpose Input Output*

HTTP – *Hypertext Transfer Protocol*

IDE – *Integrated Development Environment*

IEEE – Instituto de Engenheiros Eletricistas e Eletrônicos

IFMG – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

IoT – *Internet of Things*

MV – Máquina Virtual

SE – Sistema embarcado

URL – *Uniform Resource Locator*

USB – *Universal Serial Bus*

SoC – *System on Chip*

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Justificativa	12
2	Objetivos	13
2.1	Objetivo Geral	13
2.2	Objetivos Específicos	13
3	REFERENCIAL TEÓRICO	14
3.1	Medição de nível	14
3.1.1	<i>Chaves de nível</i>	14
3.2	Telemetria	15
3.3	Internet	16
3.3.1	<i>Arquitetura da rede</i>	16
3.3.2	<i>Protocolo HTTP</i>	17
3.3.3	<i>Wi-fi</i>	18
3.4	Sistemas Embarcados	18
3.4.1	<i>Estrutura de um sistema embarcado</i>	19
3.4.2	<i>Microcontrolador ESP 32</i>	19
3.5	Internet of Things	21
3.6	Linguagem de programação <i>Python</i>	22
3.7	Selenium	22
3.8	Django	23
3.9	Whatsapp Messenger	23
3.10	Scrum	23
3.11	Estado-da-arte	24
4	MATERIAIS E MÉTODOS	26
4.1	Classificação do trabalho	26
4.2	Sistema embarcado medidor de nível	26
4.2.1	<i>Sensor medidor de nível</i>	26
4.2.2	<i>Microcontrolador</i>	29
4.2.3	<i>Software do sistema embarcado</i>	30
4.3	Sistema de comunicação	31
4.4	Procedimento	33
4.4.1	<i>Definição do objeto de pesquisa</i>	33

<i>4.4.2 Desenvolvimento</i>	34
4.5 Scrum	35
5 RESULTADOS E DISCUSSÕES	36
5.1 Sistema embarcado medidor de nível	36
<i>5.1.1 Sensor de nível</i>	36
<i>5.1.2 Hardware do sistema embarcado</i>	37
<i>5.1.3 Software do Sistema embarcado</i>	39
5.2 Aplicação Web	39
5.3 Implantação	39
5.4 Validação do sistema	41
6 CONCLUSÃO	42
REFERÊNCIAS	43

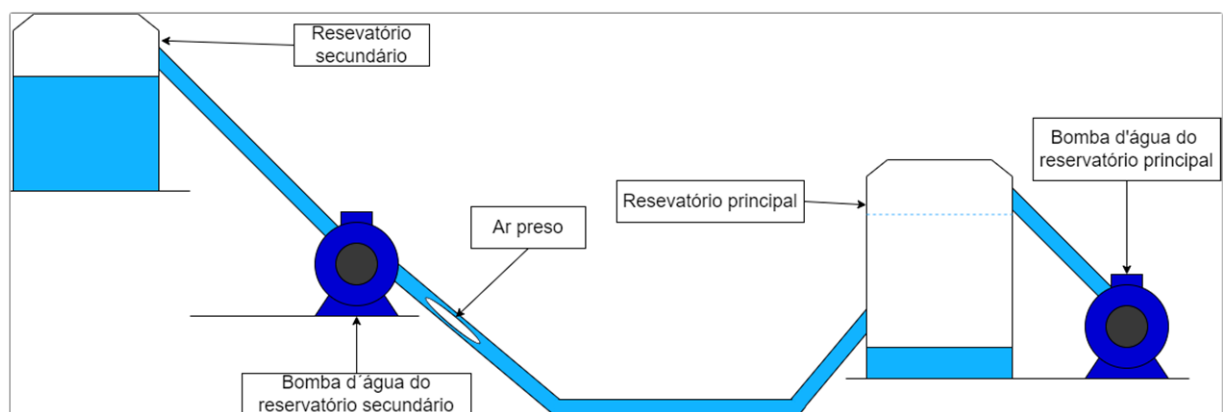
1 INTRODUÇÃO

A água desempenha um papel crucial no desenvolvimento e na sobrevivência da humanidade, mas sua disponibilidade está diminuindo na natureza. Com o aumento da população e das atividades agrícolas, a demanda por água só cresce, tornando cada vez mais desafiador garantir um fornecimento contínuo desse recurso essencial. Nos sistemas de distribuição de água, em circunstâncias especiais, como a falta de água ou energia elétrica, má gestão operacional, falta de recursos para investimento e manutenção podem interromper o serviço de abastecimento de água, o que pode resultar na introdução de ar nas tubulações e na contaminação da água (FIGUEIREDO *et al.* 2017).

A entrada de ar nas tubulações é um problema recorrente na rede de abastecimento e foi relato pelos responsáveis de manutenção do IFMG - *Campus* Bambuí, essa adversidade causa, entre outras consequências, a interrupção prolongada do abastecimento, contaminação da água e a cavitação dos equipamentos da rede de abastecimento, como relatado. Essa situação ocorre devido à falta de abastecimento do reservatório principal.

A Figura 1 mostra um diagrama da parte principal da rede de abastecimento do IFMG - *Campus* Bambuí. Pode-se observar que, quando o reservatório principal fica com o nível de água abaixo do tubo de saída, o ar entra pelas tubulações e, mesmo que o nível volte a subir, o ar ainda continuará preso nas tubulações.

Figura 1: Rede de abastecimento



Fonte: Elaborado pelo autor, 2023

Com o objetivo de propor uma solução para o problema citado, este trabalho apresenta um sistema de monitoramento do nível de água para reservatórios. A proposta consiste em desenvolver um sistema capaz de alertar sobre o nível no reservatório principal

quando esse nível atingir um estado crítico, possibilitando a tomada de providências antes mesmo que o ar consiga entrar pelas tubulações.

1.1 Justificativa

O problema de entrada de ar nas tubulações da rede de abastecimento de água potável do IFMG - *Campus* Bambuí é recorrente e foi relatado pelos responsáveis pela manutenção dos equipamentos utilizados para o abastecimento. Esses responsáveis afirmaram a necessidade de um sistema que alertasse quando o nível da água atingisse um nível crítico, ou seja, quando estivesse prestes a ficar abaixo da altura do tubo de saída. Se isso ocorrer, o ar entrará pelas tubulações, causando os problemas já citados. O desenvolvimento deste trabalho visa solucionar esse problema e trazer mais segurança no abastecimento de água para o *Campus*.

2 OBJETIVOS

2.1 Objetivo Geral

O objetivo geral deste trabalho consiste no desenvolvimento de um sistema capaz de monitorar o nível de água e emitir um alerta quando este nível atingir um estado crítico. O desenvolvimento desse sistema é uma proposta de solução para o problema de entrada de ar pelas tubulações de água do IFMG - *Campus Bambuí*.

2.2 Objetivos Específicos

- Desenvolver um dispositivo capaz de medir o nível de água.
- Implementar um sistema de comunicação para transmitir as informações sobre o nível de água.
- Integrar as partes do sistema, garantindo a interoperabilidade entre o dispositivo de medição e o sistema de comunicação.

3 REFERENCIAL TEÓRICO

Neste capítulo são abordados conceitos fundamentais para o desenvolvimento deste trabalho, sendo eles: medição de nível, sistemas embarcados, microcontroladores, *Internet*, além de temas relevantes para pesquisa e o estado-da-arte.

3.1 Medição de nível

O nível refere-se à altura do conteúdo, podendo ser sólido ou líquido, em um reservatório. Essa variável é fundamental no controle de processos contínuos, permitindo avaliar o volume armazenado em tanques, realizar balanços de materiais e manter a segurança e controle de processos. Existem três tipos básicos de medição de nível: direta, indireta e descontínua. Na medição direta, usamos referências como réguas, visores de nível ou flutuadores de forma discreta, ou seja, em uma escala graduada. Na medição indireta, o nível é determinado por grandezas físicas como pressão, empuxo, radiação ou propriedades elétricas. Nesse tipo de medição, são utilizados meios analógicos que fornecem uma medição contínua do nível. Já na medição descontínua, os medidores fornecem indicação apenas quando o nível atinge pontos específicos, como em sistemas de alarme (BEGA, 2006).

O emprego de sensores que oferecem uma leitura descontínua do nível permite que o sistema detecte a ausência do nível, em pontos específicos, indicando potenciais problemas no abastecimento do reservatório.

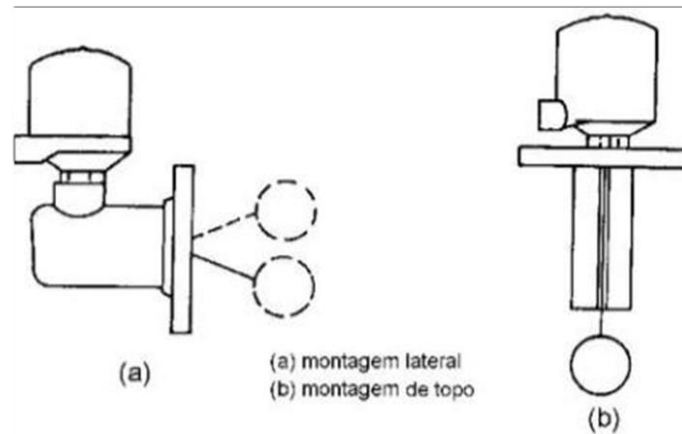
3.1.1 Chaves de nível

As chaves de nível são dispositivos do tipo medição direta, que atuam em pontos específicos de nível, desencadeando ações necessárias para garantir a segurança e operação adequada dos equipamentos. Elas possuem dois estados de saída, como energizado/desenergizado ou pressurizado/despressurizado, dependendo do tipo de chave. São compostas por um detector de nível e um circuito de saída, o detector informa ao circuito a presença ou ausência do nível, permitindo a mudança do estado de saída da chave (BEGA, 2006).

As chaves de nível do tipo flutuador consistem em um dispositivo composto por um flutuador, que monitora o nível do líquido ou a interface entre dois líquidos de densidades diferentes, acionando uma chave quando o nível atinge um valor predefinido. Os flutuadores podem apresentar diferentes formas, esférica, cilíndrica ou elíptica, e são fabricados em materiais como latão, aço inoxidável, aço-liga ou materiais leves, proporcionando equilíbrio entre a densidade do fluido e a profundidade máxima de

imersão. O termo flutuador é utilizado para descrever qualquer elemento que se mova em conformidade com a altura ou nível de um líquido, independentemente de sua forma geométrica ou material de construção (BEGA, 2006). Na Figura 2, é mostrado como uma chave de nível do tipo flutuador pode ser instalada em um reservatório.

Figura 2: Formas de montagem de uma chave de nível



Fonte: (BEGA, 2006)

Essas características das chaves de nível do tipo flutuador atendem aos requisitos deste trabalho, proporcionando uma medida direta do nível.

3.2 Telemetria

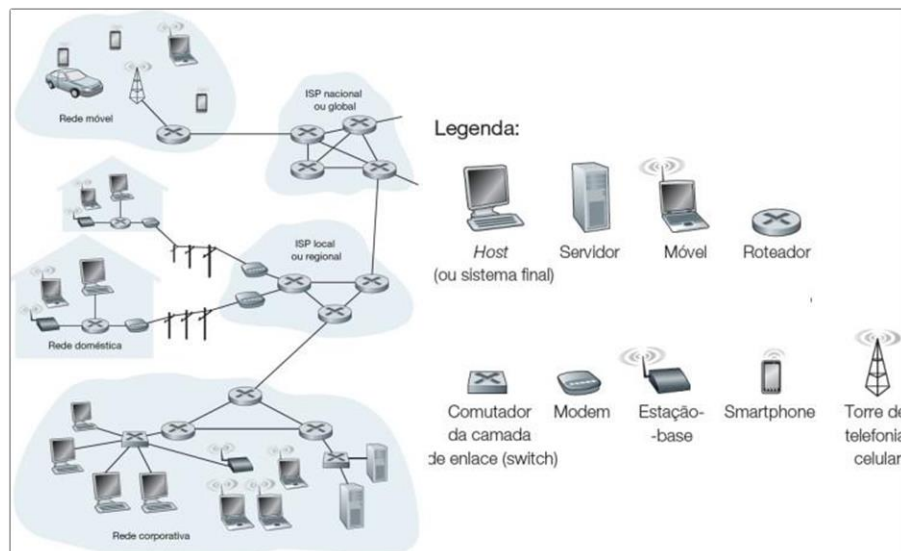
A telemetria refere-se à realização de medições à distância ou em locais remotos. Seu surgimento deu-se pela necessidade de efetuar medições em lugares inacessíveis, como a temperatura dentro de um forno. Os componentes essenciais de qualquer sistema de telemetria incluem, no mínimo, um sensor, uma antena transmissora de baixo ganho, uma antena receptora de alto ganho, um receptor e um visor. Muitos sistemas utilizam vários sensores, métodos para empacotar uma grande quantidade de dados, técnicas para desempacotar os dados no centro de controle, estratégias para marcar o tempo nas informações para análise pós-operacional, métodos de arquivamento dos dados para análise subsequente, além de técnicas sofisticadas para processar os dados antes de apresentá-los (MATTOS, 2004).

A telemetria é utilizada neste para levar as informações de nível aos interessados.

3.3 Internet

A Internet é uma rede de computadores interconectados, que funciona como uma infraestrutura global de comunicação. Ela permite a troca de informações e o acesso a recursos em todo o mundo. Em termos de componentes, a Internet é composta por uma combinação de software e hardware. O software inclui os protocolos de comunicação, que definem as regras para o envio e recebimento de dados na rede. O hardware envolve os dispositivos, como servidores, roteadores e cabos de comunicação, que são utilizados para transmitir os dados através da rede. A Figura 3 mostra os elementos básicos de uma rede e exemplos de dispositivos que podem se conectar a ela (KUROSE; ROSS, 2021).

Figura 3: Rede de computadores



Fonte: (KUROSE; ROSS, 2021)

Para realizar a integração das partes do sistema desenvolvido neste trabalho, foi utilizada a rede local de *Internet do Campus Bambuí*.

3.3.1 Arquitetura da rede

A estruturação dos projetos de redes de computadores é realizada através da organização em camadas. Essa abordagem permite que projetistas dividam as redes em diferentes camadas, cada uma com funções específicas e interligadas entre si. Cada camada oferece serviços à camada acima dela e esses serviços são realizados pela execução de ações na própria camada e pela utilização dos serviços da camada imediatamente abaixo. As camadas

podem ser implementadas em *software*, *hardware* ou em uma combinação de ambos. Protocolos de camada de aplicação e de transporte são, normalmente, realizados em *software* nos sistemas finais, enquanto as camadas física e de enlace são executadas em *hardware* associado a um enlace específico (KUROSE; ROSS, 2021). A Figura 4 mostra a pilha de protocolos que compõe a rede.

3.3.2 Protocolo HTTP

O protocolo HTTP pertence à camada de aplicação e é do tipo solicitação- resposta. Ele especifica as mensagens que os clientes podem enviar para os servidores e as respostas que recebem. Baseado em uma estrutura de métodos, como o *GET*, é utilizado para obter o conteúdo das páginas (objetos) dos servidores *Web*. Esse protocolo contribuiu para o crescimento inicial da *Web*, sendo projetado para suportar futuras aplicações orientadas a objetos (TANENBAUM, 2011). A Figura 5 mostra como pode ser uma conexão cliente/servidor, envolvendo o protocolo HTTP.

Figura 4: A pilha de protocolos da Internet



Fonte: (KUROSE; ROSS, 2021)

Figura 5: Requisição HTTP



Fonte: Elaborado pelo autor, baseado em (TANENBAUM, 2011)

Por meio de uma requisição HTTP *GET*, é possível desencadear uma série de funções em um servidor remoto. Essa requisição permite a interação entre o dispositivo e o servidor, possibilitando o processamento de dados, o acesso a recursos e à execução de ações específicas.

3.3.3 Wi-fi

Em redes locais de computadores, a tecnologia *Wi-Fi* permite a conexão de dispositivos sem a necessidade de cabos, utilizando frequências de rádio para a transmissão de dados. O padrão IEEE 802.11, utilizado para conexão *Wi-Fi*, emergiu como o mais dominante. Esse padrão define os aspectos técnicos das redes sem fio, incluindo a frequência de operação, a largura de banda disponível e os protocolos de segurança. Diversas versões do padrão foram lançadas ao longo dos anos, como 802.11b, 802.11g, 802.11n e 802.11ac, cada uma com melhorias na velocidade, alcance e capacidade de transmissão (TANENBAUM, 2011).

Para que seja possível conectar o SE à rede local de *Internet* do *Campus* é necessário utilizar a rede *Wi-Fi* para realizar esta conexão.

3.4 Sistemas Embarcados

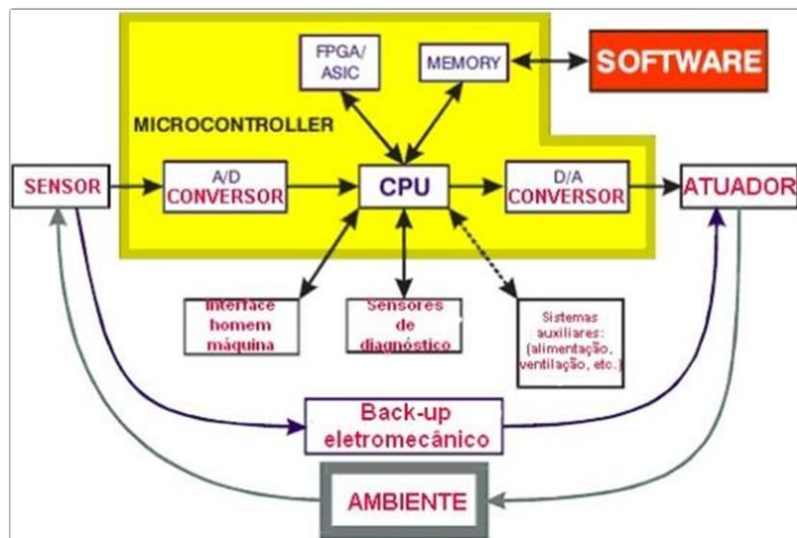
De acordo com Cunha (2007), um sistema embarcado é caracterizado por incorporar a capacidade computacional em um circuito integrado, equipamento ou sistema. Ele vai além de ser apenas um computador simples, sendo um sistema completo e autônomo, projetado para executar uma tarefa específica. Embora o usuário final não tenha acesso direto ao programa embutido no dispositivo, é possível interagir com o equipamento por meio de

interfaces como teclados e *displays*, desde que o sistema tenha sido projetado para suportar essas interações.

3.4.1 Estrutura de um sistema embarcado

Os sistemas embarcados diferem-se dos computadores convencionais, pois são projetados para executar uma única tarefa, sem a flexibilidade de executar aplicativos diversos. Geralmente, eles são programados pelos fabricantes para atualizações e melhorias e utilizam microprocessadores ou microcontroladores como "cérebros" para gerenciar suas funcionalidades, incluindo a leitura de sinais externos, execução de programas e controle de atuadores (CUNHA, 2007). Na Figura 6, os elementos básicos são mostrados em sistema embarcado e como são interligados.

Figura 6: Elementos básicos de um sistema embarcado



Fonte: (CUNHA, 2007)

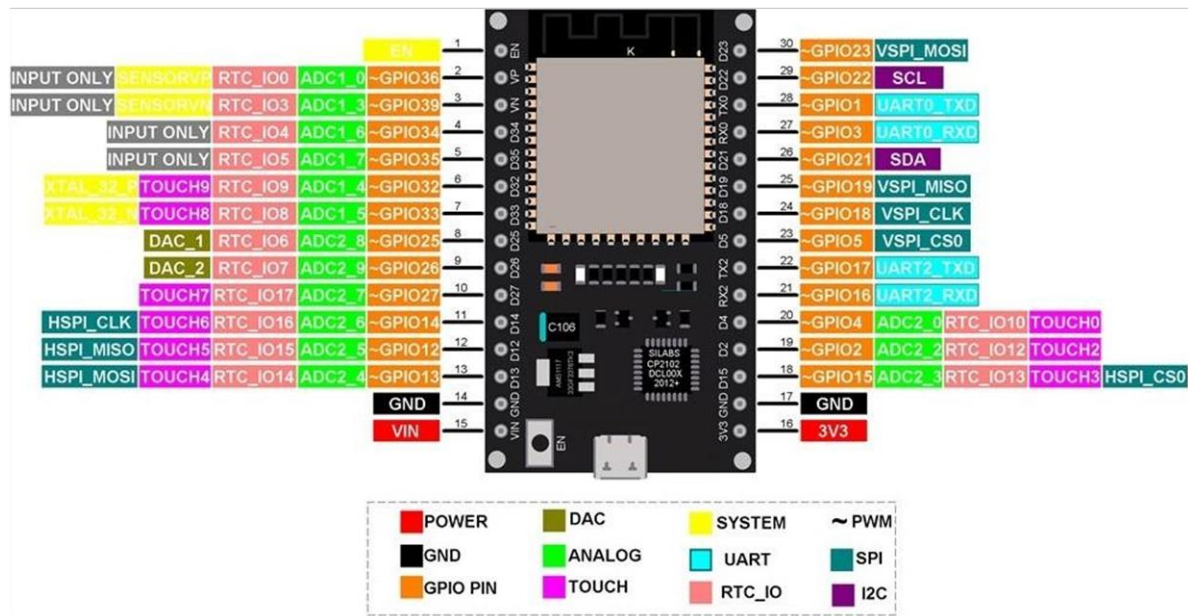
Como um dos componentes do sistema desenvolvido neste trabalho, o SE faz a leitura do nível de água e processa essa informação para decidir o ação a ser realizada, conforme a leitura do nível.

3.4.2 Microcontrolador ESP 32

O ESP 32 é um microcontrolador integrado, projetado para dispositivos móveis, eletrônicos vestíveis e aplicativos de IoT. Ele combina recursos de baixo consumo de energia,

como controle de *clock* e escalonamento dinâmico de energia. Com suporte a *Wi-Fi* e *Bluetooth*, o ESP 32 oferece uma solução com cerca de 20 componentes externos (ESPRESSIF, 2021). Na Figura 7, são mostradas as múltiplas funções de cada pino presente no microcontrolador, que está acoplado em uma placa de desenvolvimento. Observa-se que, para leitura de uma chave de nível, é possível utilizar diferentes pinos e funções.

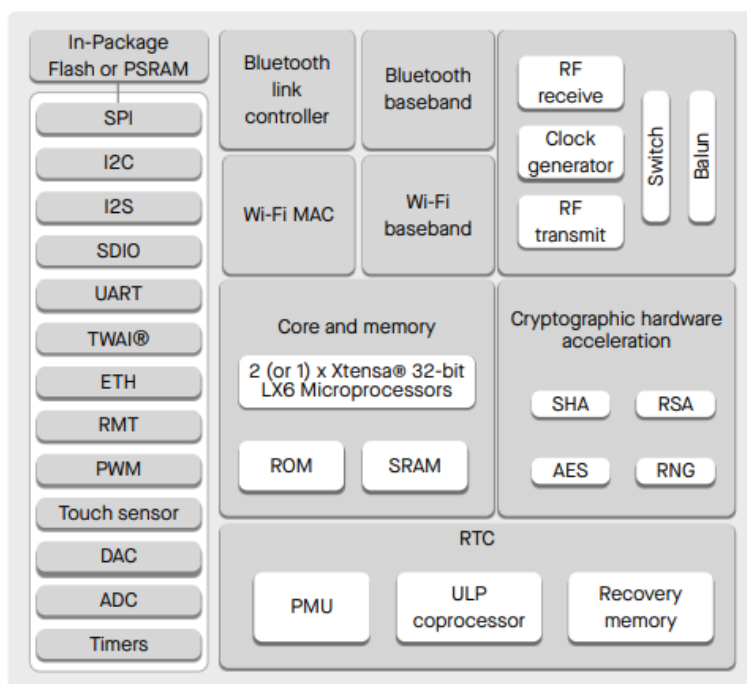
Figura 7: ESP 32



Fonte: Elaborado pelo autor, baseado em (PRAVALIKA; PRASAD et al. 2019).

O ESP32 é um chip único, que combina Wi-Fi e Bluetooth de 2,4 GHz, projetado com tecnologia de baixa potência de 40 nm da TSMC. Destinado a oferecer o melhor desempenho em potência e RF, ele demonstra robustez, versatilidade e confiabilidade em uma ampla gama de aplicações e cenários de energia (ESPRESSIF, 2021). O diagrama de blocos funcionais do SoC é apresentado abaixo.

Figura 8: Componentes ESP 32



Fonte: (ESPRESSIF, 2021)

3.5 Internet of Things

Segundo Tschofenig *et al.* (2015), O termo "*Internet of Things*" representa uma tendência em que uma quantidade de dispositivos incorporados utiliza serviços de comunicação, disponibilizados por protocolos de *Internet*. Esses dispositivos, comumente denominados "objetos inteligentes", geralmente não são operados diretamente por humanos, mas atuam como componentes de um sistema maior.

De acordo com Rose, Eldridge e Chapin (2015), o conceito de IoT foi proposto por Kevin Ashton, em 1999, que, inicialmente, se referiu a ela como objetos conectados por tecnologia de identificação por radiofrequência. A definição exata de IoT ainda está em desenvolvimento e pode variar de acordo com as perspectivas adotadas. No entanto, de modo geral, a IoT é considerada uma infraestrutura global de rede dinâmica com recursos de autoconfiguração, baseada em padrões e protocolos de comunicação interoperáveis.

Diante desse contexto, o sistema proposto neste trabalho enquadra-se como sendo um sistemas IoT. Isso se deve ao fato de utilizar um dispositivo inteligente capaz de coletar informações do ambiente e, posteriormente, enviar essas informações pela *Internet*.

3.6 Linguagem de programação *Python*

Python é uma linguagem de programação que se destaca por sua sintaxe, o que contribui para a legibilidade do código fonte e aumenta a produtividade do desenvolvedor. Além disso, a linguagem oferece diversas estruturas, como listas, dicionários, manipulação de data/hora e números complexos, assim como uma ampla variedade de módulos prontos para uso e *frameworks* de terceiros. *Python* também incorpora recursos encontrados em outras linguagens, como geradores, introspecção, persistência, metaclasses e suporte a unidades de teste. Uma das características do *Python* é ser multiparadigma, permitindo a programação modular, funcional e orientada a objetos. Mesmo os tipos básicos no *Python* são tratados como objetos. A linguagem é interpretada por meio de *bytecode* pela máquina virtual *Python*, o que confere portabilidade ao código. Isso significa que é possível compilar aplicações em uma plataforma e executá-las em outros sistemas ou executá-las diretamente a partir do código fonte (BORGES, 2014).

A linguagem *Python* é utilizada no desenvolvimento de ferramentas e recursos que foram utilizados neste trabalho para desenvolver o sistema comunicação.

3.7 Selenium

*Selenium*¹ é um projeto que engloba uma variedade de ferramentas e bibliotecas voltadas para a automação de navegadores da *Web*. Sua principal finalidade é permitir e facilitar a automação de interações com os navegadores, oferecendo recursos avançados. Utilizando o pacote de recursos disponibilizado para as linguagem de programação Java, *Python*, C#, *Ruby* e *Kotlin*, é possível construir um código capaz de acessar plataformas de bate-papo e iniciar uma conversa.

Entre as funcionalidades do *Selenium*, destacam-se as extensões que possibilitam a emulação de interações do usuário com os navegadores, como clicar em botões, preencher formulários e navegar por páginas da *Web*. Além disso, o *Selenium* disponibiliza um servidor de distribuição, o qual permite escalabilidade na alocação de navegadores para a execução de testes automatizados.

¹ <https://www.selenium.dev/>

3.8 Django

Django é um *framework* de desenvolvimento *Web* de código aberto, que utiliza a linguagem *Python* e segue o padrão de arquitetura *model-template-views*. Seu objetivo principal é simplificar a criação de sites complexos, baseados em banco de dados, promovendo a reutilização de componentes e princípios como a não repetição de código. O *framework* é mantido pela *Django Software Foundation*®, uma organização sem fins lucrativos nos Estados Unidos. Além disso, o *Django* possui um sistema de configuração que permite a integração de código de terceiros, desde que siga as convenções de aplicativos reutilizáveis. Existem diversos pacotes disponíveis que estendem as funcionalidades da estrutura original, fornecendo soluções para problemas específicos, como registro, pesquisa, fornecimento e consumo de API, sistemas de gerenciamento de conteúdo, entre outras funções (HOLOVATY; WILLISON *et al.* 2003).

3.9 Whatsapp Messenger

O *WhatsApp Messenger*® é um aplicativo de mensagens multiplataforma, que permite a troca de mensagens, por meio de dispositivos eletrônicos, através da *Internet*. Ele está disponível para uma variedade de sistemas operacionais. Com o *WhatsApp*, os usuários desses dispositivos podem trocar mensagens entre si, independentemente do sistema operacional utilizado (HONORATO; REIS, 2014).

Uma das ferramentas desse aplicativo é o *Whatsapp Web*, que permite enviar mensagem através de um computador *desktop*, sem a necessidade de autenticação ou verificação de ações automatizadas, permitindo assim o envio de mensagens de forma automáticas, como feito neste trabalho.

3.10 Scrum

O *Scrum* é uma metodologia ágil, utilizada para gerenciar e planejar projetos de *software* de forma eficaz. Nesse modelo, os projetos são divididos em ciclos, geralmente mensais, chamados de "*Sprints*". Cada *Sprint* representa um período de tempo limitado, no qual um conjunto específico de atividades deve ser concluído. As funcionalidades a serem desenvolvidas no projeto são organizadas em uma lista chamada "Product Backlog" que posteriormente são divididas para as *sprints* (SCRUM, 2023). A metodologia ágil *scrum* foi utilizada para desenvolver este projeto, como descrito no capítulo 3

3.11 Estado-da-arte

No trabalho de Galvão (2022), foi desenvolvido um sistema de monitoramento em reservatórios residenciais, para acompanhar o nível da água e informá-lo ao usuário. O sistema consiste em um SE dedicado à leitura do nível, utilizando chaves de nível, do tipo flutuador, dispostos em uma régua. Além disso, foi implementado um outro SE com um *buzzer* e um *display* para receber e fornecer as informações ao usuário.

No presente trabalho, será criado apenas um SE para monitorar o nível da água e transmitir essas informações pela *Internet* aos dispositivos móveis dos usuários, eliminando a necessidade de um segundo SE para exibir os dados ao usuário.

O trabalho de Virgens *et al.* (2020) teve como objetivo desenvolver um sistema autônomo e microprocessado, capaz de monitorar, em tempo real, o estado de um reservatório de água, bem como o fornecimento de água pela companhia de saneamento local. Esse sistema foi projetado para alertar o usuário em caso de interrupção no fornecimento, fornecer estatísticas de consumo e prever a duração da água armazenada. Para que o sistema consiga interagir com o usuário, foram construídos dois SEs, sendo um dedicado a coletar as informações sobre nível e vazão e enviar essas informações para outro SE, dedicado a exibir essas informações ao usuário.

Como já mencionado anteriormente, o presente trabalho utilizou apenas um SE, que fará a leitura no nível e enviará as informações, via *Internet*, aos dispositivos móveis dos responsáveis de manutenção.

A pesquisa conduzida por Silva *et al.* (2021) resultou no desenvolvimento de um dispositivo que realiza a medição contínua do nível de água em um reservatório, por meio de um sensor ultrassônico. Esse sensor ultrassônico é um dos componentes de um SE, que também contém um *display*, que exibe o nível de água em forma de porcentagem ao usuário. Diferentemente desse trabalho, no presente estudo, a informação do nível é primeiramente enviada, via *Internet*, aos dispositivos móveis, para que posteriormente o usuário tenha acesso a essa informação.

No trabalho realizado por Lima (2022), foi construído um SE capaz de detectar um fluxo de água e alertar sobre as condições desse fluxo, esse alerta é feito através da *Internet*, onde o SE envia uma requisição HTTP para um sistema interessado. O presente trabalho fará uma comunicação semelhante, porém levando a informação sobre o nível da água, sendo que essa informação será enviada pelo aplicativo *WhatsApp*.

Sousa (2021) desenvolveu um sistema capaz de enviar relatório sobre construção

civil através da plataforma *WhatsApp Web*. O programa capaz de realizar esse envio foi escrito na linguagem de programação *Python*, utilizando a biblioteca *Selenium*. O presente trabalho pretende criar um programa com uma função semelhante, com a finalidade de alertar sobre o estado do nível da água.

A *WhatsApp Business API*² permite que empresas se comuniquem em larga escala com os clientes. Elas podem integrar a API com sistemas de *back-end* e existem duas opções de hospedagem: local e em nuvem. Para usar a API em nuvem, as empresas fazem chamadas à *API Graph* da Meta para enviar mensagens e configuram *Webhooks* para receber eventos, como mensagens e atualizações de status. A *WhatsApp Business API* necessita de acesso a informações sensíveis, como dados de cartão de crédito. Devido a isso, ela não foi escolhida para ser utilizada para este projeto, pois o projeto está voltado para uma instituição pública.

² <https://developers.facebook.com/docs/whatsapp/cloud-api/overview>

4 MATERIAIS E MÉTODOS

Neste capítulo são descritos os materiais e métodos utilizados. O desenvolvimento do sistema de monitoramento do nível foi dividido em duas partes principais. A primeira parte consiste na construção de um SE que é responsável por medir o nível da água, a segunda parte envolve o desenvolvimento de um sistema de comunicação, que permite a transmissão das mensagens com as informações do nível para os responsáveis pela manutenção.

4.1 Classificação do trabalho

Segundo Gerhardt e Silveira (2009), o presente trabalho pode ser classificado de acordo com diferentes aspectos da pesquisa. Em relação à natureza, trata-se de uma pesquisa aplicada, pois busca aplicar conhecimentos científicos para resolver um problema específico. Quanto ao objetivo, é uma pesquisa exploratória, uma vez que busca encontrar uma solução para o problema em questão. No que diz respeito ao procedimento, é uma pesquisa enquadrada como estudo de caso, pois o escopo do trabalho está limitado ao problema de entrada de ar na tubulações de água do IFMG - *Campus* Bambuí. Por fim, em relação à abordagem, é uma pesquisa quantitativa, pois envolve a mensuração e quantificação das variáveis de pesquisa.

De acordo com Wazlawick (2009), o presente trabalho enquadra-se como sendo “Apresentação de algo Diferente”, pois propõe resolver um problema com uma solução diferentes daquelas que já existem.

4.2 Sistema embarcado medidor de nível

Para construir o SE que consegue medir o nível, são necessários dois componentes principais. O primeiro é um sensor que ficará em contato com a água e será responsável por coletar as informações do nível. O segundo componente é um microcontrolador, que receberá e processará essas informações.

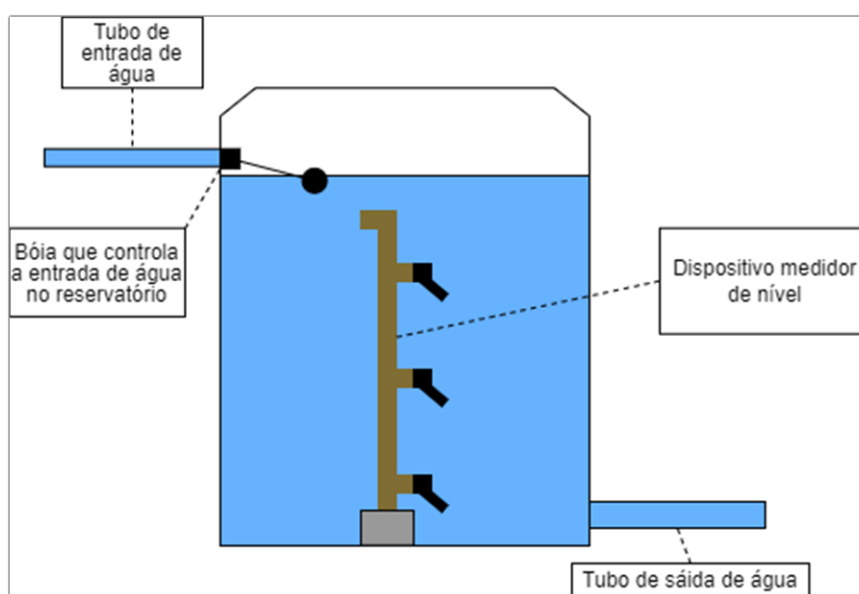
4.2.1 *Sensor medidor de nível*

Para realizar a medição do nível, foi construído um sensor composto por chaves de nível do tipo flutuador dispostos em uma régua, de forma a dar uma leitura discreta do nível. Essas chaves de nível são posicionadas de modo a indicar um nível alto, médio ou baixo, essa

indicação não faz referência à capacidade total do reservatório, mas sim à área em que o sensor está atuando. O uso de três chaves de nível trouxe redundância à informação, ou seja, a possibilidade de verificar a veracidade da informação, além de que esse tipo de sensor não exige manutenção. Embora as chaves de nível não ofereçam uma medição precisa do nível da água, essa precisão não é necessária para o problema em questão. O objetivo é apenas detectar quando o nível da água cai abaixo de um determinado ponto.

Na Figura 9, é apresentado como o sensor foi posicionado dentro do reservatório. Observa-se que o flutuador mais alto do sensor está abaixo do flutuador que controla a entrada de água no reservatório, essa configuração é feita de forma estratégica, pois quando o nível da água ficar abaixo da área de atuação do flutuador que controla a entrada de água, isso indicará algum problema com o abastecimento. Em condições normais de funcionamento, a variação no nível da água é restrita à área de atuação do flutuador que controla a entrada.

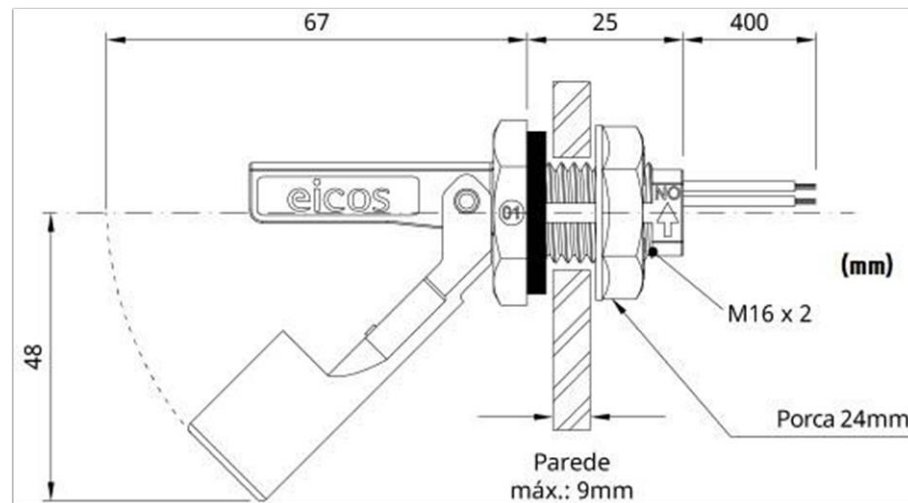
Figura 9: Diagrama do sensor de nível



Fonte: Elaborado pelo autor, 2023.

As três chaves de nível, utilizadas para a construção do dispositivo, são do modelo "Sensor de Nível LA16M-40", como mostrado na Figura 10. O sensor fornece um sinal *ON/OFF* para indicar quando o nível de líquido foi alcançado e é instalado na lateral do tanque pelo interior, por meio de um furo de 16mm de diâmetro. Para detectar cada nível de líquido desejado, é necessário um sensor de nível instalado no ponto correspondente. (EICOS, 2023).

Figura 10: Sensor de Nível de Água com Boia Horizontal



Fonte: (EICOS, 2023)

O sensor é feito de POM - Poliacetal, suportando até dois bar de pressão e temperaturas entre -10°C e 100°C . Ele possui saída de contato *ON/OFF*, com conexão elétrica por cabo de $2 \times 0,5\text{mm}^2$. Sua montagem é lateral interna em furo de 16mm, vedada por arruela NBR, e sua espessura máxima de parede do reservatório é de 9mm. O sensor suporta diferentes tensões de trabalho e tem uma potência máxima e corrente máxima específicas para cada uma delas (EICOS, 2023).

Além das chaves de nível, os seguintes materiais também foram utilizados.

- 150 cm de cano PVC de 32 mm.
- 3 emendas tipo T de cano PVC de 32 mm.
- tampões de PVC de 32 mm.
- cola para tubo PVC.
- 3 Kg de cimento.
- 24 metros de fio encapado de 2 mm.

Para que o sensor de medição de nível fique com as medidas corretas, foram coletadas as medidas do reservatório principal do IFMG - *Campus* Bambuí e, posteriormente, foram feitos os cálculos das medidas de tal forma que o dispositivo fique proporcional ao reservatório. Após a construção do dispositivo de medição de nível, foram obtidos quatro fios de saída, sendo um fio positivo para cada flutuador e um fio para o negativo. Esses fios serão conectados ao ESP 32.

4.2.2 Microcontrolador

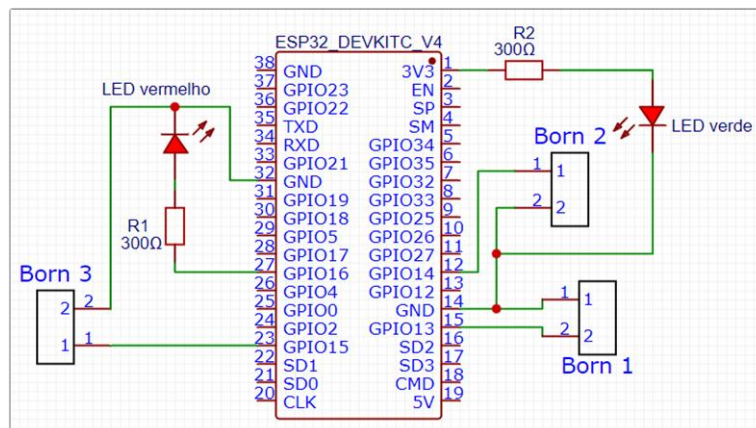
O SE utiliza o microcontrolador ESP 32 para processar informações do sensor de nível, este microcontrolador está integrado a uma placa de desenvolvimento, que contém pinos para prototipação, regulador de tensão, interface USB, entre outros componentes que facilitam a utilização do ESP 32. O sensor de nível possui quatro fios: três fios positivos conectados aos pinos 13, 14 e 15 (GPIO 13, 14 e 15) e um fio negativo ligado ao pino GND do ESP32. Essa configuração permite verificar o contato elétrico entre os pinos e o GND.

Além disso, dois LEDs foram conectados ao ESP 32, um LED verde está ligado ao pino de alimentação de 3,3 Volts e ao GND, indicando que o ESP 32 está ligado, e um LED vermelho, conectado ao pino 16 (GPIO 16) e ao GND, com duas funções: piscar a cada dois segundos, quando o nível está baixo, e piscar a cada 300 milissegundos, em caso de problemas de conexão com o *Wi-Fi*.

Para conectar o ESP 32 à rede de internet do *Campus*, utilizou-se a conexão *Wi-Fi*, utilizando módulo *Wi-Fi* integrado ao ESP 32. A alimentação do microcontrolador é feita pela entrada micro-USB da placa de desenvolvimento e utilizando uma fonte de corrente contínua de cinco Volts.

O projeto do circuito eletrônico do SE foi feito, utilizando a plataforma *Easy eda*, onde foi desenvolvido o esquema elétrico e a placa de circuito impresso, como mostrado nas Figuras 11 e 12 respectivamente.

Figura 11: Esquema do circuito elétrico

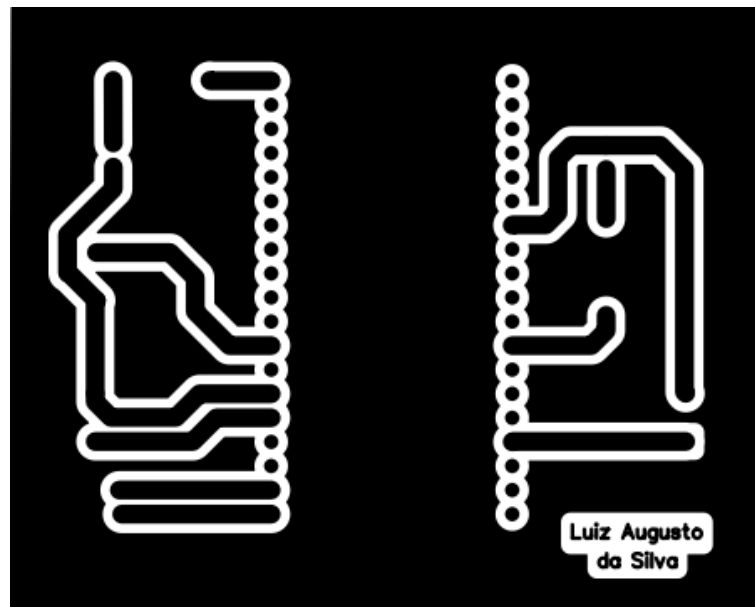


Fonte: Elaborado pelo autor, 2023.

A construção do SE precisou da seguinte lista de componentes.

- 1 ESP32-WROOM-32u.
- 1 Antena *wifi*.
- 1 sensor de nível (desenvolvido neste projeto)
- 1 LED verde
- 1 LED vermelho
- 2 resistores de 300 Ohms
- 1 placa de fenolite de 10x7 cm
- Bornes 2 polos - KF-301 2T

Figura 12: Placa de circuito impresso



Fonte: Elaborado pelo autor, 2023.

4.2.3 Software do sistema embarcado

Para estabelecer as ações a serem executadas quando o ESP 32 receber as informações do nível d'água, foi desenvolvido um código-fonte na linguagem de programação C. Esse código foi implementado na plataforma de desenvolvimento Arduino IDE.

Devido à natureza do problema e ao *design* do sensor de nível, a Tabela 1 apresenta uma tabela verdade dos possíveis casos de leitura do sensor. Observa-se que o valor 0 indica que o flutuador está com o contato aberto, enquanto o valor 1 indica o contato está fechado

A partir desta tabela, as principais condições de leitura foram consideradas e tratadas no código-fonte. O código-fonte em C inclui as condições para verificar os valores dos

flutuadores "Alto", "Médio" e "Baixo" e determinar a saída correspondente, de acordo com a tabela fornecida.

No caso de envio de alerta, o código-fonte contém instruções para que seja realizado esse envio. Isso é feito através de uma requisição HTTP, utilizando a biblioteca *http Client ESP32*. Essa biblioteca disponibiliza um conjunto de APIs para realizar solicitações HTTP/S em aplicativos ESP-IDF (SYSTEMS, 2023).

Tabela 1: Tabela verdade dos casos de leitura do sensor

Chave "Alto"	Chave "Médio"	Chave "Baixo"	Saída
1	1	1	"Full"
1	1	0	"Don't care"
1	0	1	"Don't care"
1	0	0	"Don't care"
0	1	1	"Middle"
0	1	0	"Don't care"
0	0	1	"Low"
0	0	0	"Empty"

Fonte: Elaborado pelo autor.

4.3 Sistema de comunicação

Quando o ESP 32 realiza uma requisição HTTP, ela é encaminhada a uma aplicação *Web*, desenvolvida, utilizando o *framework Django* e hospedada em uma máquina virtual no datacenter do *Campus*. Essa requisição atua como um gatilho, desencadeando uma sequência de instruções a serem executadas.

Nessa requisição HTTP, a URL³ representa o endereço da aplicação *Web* e inclui um caminho, que direciona para a função correta dentro da aplicação. Além do caminho da função, existe também um parâmetro denominado "*token*", que age como uma senha, concedendo acesso à função de envio de mensagens.

Utilizando o pacote de recursos *Selenium*, disponível na linguagem *Python*, foi

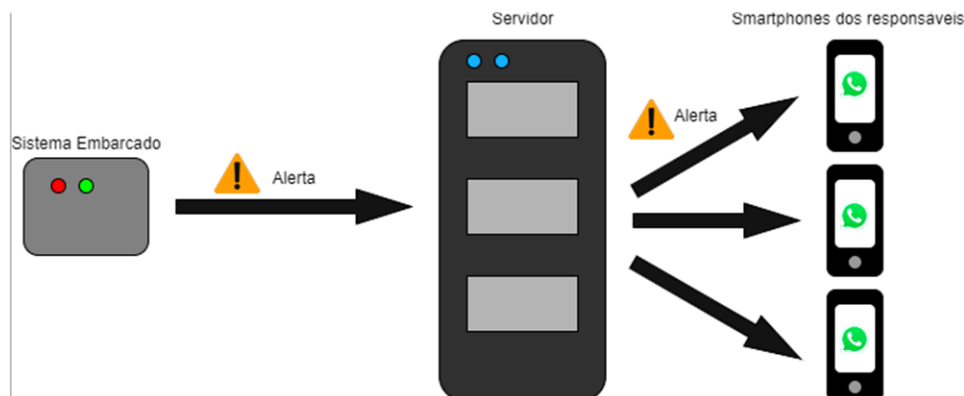
³ http://VMAdress/send_msg/msg_1/?token=123

realizada a automação de um navegador *Web*, que envia mensagens previamente escritas aos responsáveis pela manutenção. Esse envio ocorre por meio da plataforma *Whatsapp Web*. Existem soluções que fazem esse envio, como a API do *WhatsApp Business*, que requer acesso a informações sensíveis, como dados de cartão de crédito. Por essa razão, não foi selecionada para ser utilizada neste projeto, uma vez que este se destina a uma instituição pública.

Para que seja possível o envio das mensagens através do *Whatsapp Web*, é necessário ter um número de telefone para registrar uma conta de usuário. Portanto, foi criada uma conta de usuário, usando um número fixo designado para esse fim, do próprio *Campus*, com a devida autorização para sua utilização nesse contexto. Essa conta é mantida ativa no navegador que é empregado para o envio das mensagens.

Todos os códigos da aplicação *Web* foram desenvolvidos do editor de código-fonte *Visual Studio Code*, na linguagem de programação *Python*. A Figura 13 mostra, de forma geral, como o envio do alerta ocorre, envolvendo todas as partes do sistema.

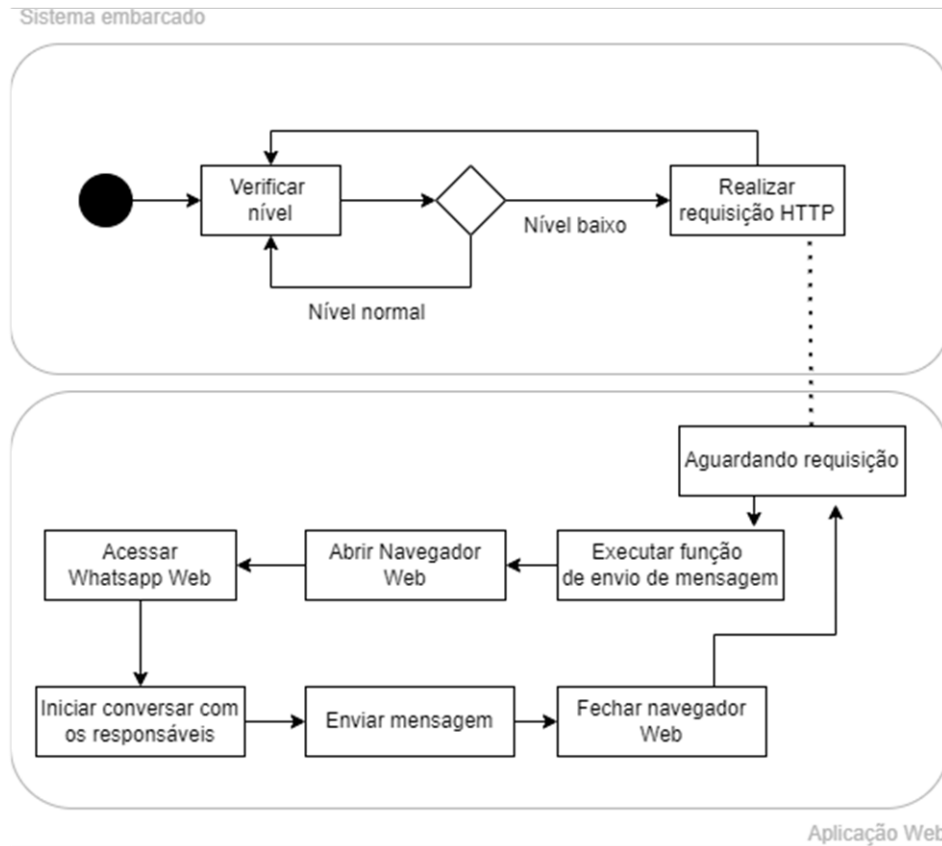
Figura 13: Processo do envio de mensagens



Fonte: Elaborado pelo autor, 2023.

Para ilustrar o funcionamento do sistema, o diagrama de máquina de estado, apresentado na Figura 14, descreve os estados do sistema.

Figura 14: Diagrama de máquina de estado



Fonte: Elaborado pelo autor, 2023.

Observe que, tanto o SE quanto a aplicação *Web*, operam em um *loop* contínuo. Essa abordagem é essencial, pois o sistema deve funcionar de forma ininterrupta, enviando alertas sempre que necessário, isso significa que, após o envio de um alerta, o sistema não encerra, mas permanece ativo, aguardando a próxima situação, que demande o envio de um alerta.

4.4 Procedimento

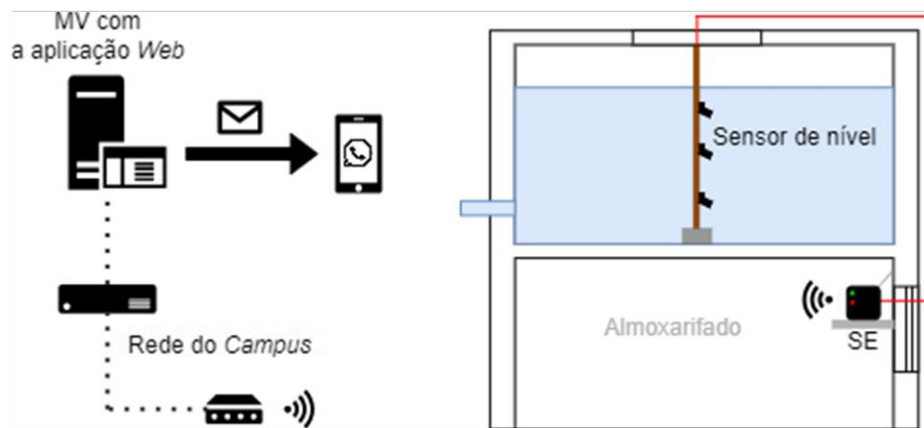
Como um estudo de caso, o presente trabalho busca solucionar o problema da entrada de ar pelas tubulações de água do IFMG - *Campus* Bambuí.

4.4.1 Definição do objeto de pesquisa

Como descrito no capítulo 1, o problema da entrada de ar nas tubulações foi descrito pelos responsáveis de manutenção da rede de abastecimento do IFMG - *Campus* Bambuí. Para

resolver este problema, o presente trabalho propõe o desenvolvimento de um sistema de monitoramento do nível da água como forma de evitar que o ar entre pelas tubulações. A solução compreende a intercomunicação dos diversos elementos do sistema, juntamente à implementação de cada um desses componentes. O Diagrama, apresentado na Figura 15, ilustra os componentes do sistema e a maneira pela qual eles se interligam.

Figura 15: Diagrama da solução



Fonte: Elaborado pelo autor, 2023.

4.4.2 Desenvolvimento

O sistema de monitoramento do nível de água foi dividido em duas principais etapas, o desenvolvimento do SE, capaz de realizar a medição do nível, e o desenvolvimento do sistema de comunicação, capaz de enviar uma mensagem, pelo *Whatsapp*, para os responsáveis de manutenção. Todo o desenvolvimento do sistema de monitoramento seguiu a metodologia ágil *Scrum*, pois essa é uma metodologia de desenvolvimento que permite alterações ao longo do processo. No *Product Backlog*, foram definidos todos os requisitos do sistema de monitoramento, posteriormente, no *Sprint Planning* foram definidos o cronograma e as tarefas a serem realizadas em cada *Sprint*.

Para demonstrar e validar o sistema de monitoramento, ele foi implementado no reservatório principal do IFMG - *Campus Bambuí*, também o sistema de comunicação foi implementado em uma máquina virtual, hospedada no *datacenter* do *Campus*. Para verificar a eficácia do sistema, foi simulada uma falta de abastecimento do reservatório, desligando a bomba d'água, que alimenta o reservatório principal, e também forçando a abertura dos contatos das chaves de nível.

4.5 Scrum

O desenvolvimento deste trabalho teve início com uma análise de requisitos funcionais e não funcionais do sistema, que foram os seguintes:

Requisitos funcionais:

- Monitoramento constante do nível de água
- Transmissão de informações sobre no nível de água
- Envio de mensagens de alerta

Requisitos não-funcionais:

- O sistema depende de conexão estável
- O sistema depende de alimentação elétrica constante
- O sistema depende do *datacenter* do *Campus*

Com base na análise de requisitos, foi feito o seguinte *Product Backlog*:

- O sistema deve ser capaz de detectar o nível de água
- O sistema deve ser capaz de transmitir o nível de água
- O sistema deve ser capaz de enviar mensagens de alerta

A tabela 2 mostra o *Sprint Planning*, que detalha as *sprints* planejadas, onde o título da *sprint* representa a tarefa a ser realizada.

Tabela 2: Sprint Backlog e Duração

Sprint	Duração
Sprint 1: Desenvolvimento do código da aplicação Web	1 semana
Sprint 2: Desenvolvimento do código do sistema embarcado	1 semana
Sprint 3: Construção do sensor de nível	2 semanas
Sprint 4: Construção do sistema embarcado	2 semanas
Sprint 5: Implantação do sistema	2 semanas

Fonte: Elaborado pelo autor.

A implementação do sistema de monitoramento exigiu um total de oito semanas para ser concluída. Durante esse intervalo, todas as etapas necessárias foram realizadas para a criação da solução. Inicialmente, cada *sprint* estava planejada para durar uma semana, no entanto, devido a diversos contratemplos, as últimas *sprints* tiveram uma duração de duas semanas. Todo o gerenciamento de projeto foi realizado, utilizando a plataforma *Jira software*.

5 RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados os resultados obtidos com o desenvolvimento deste projeto. Seguindo a metodologia descrita no Capítulo 3, foi possível desenvolver, implementar e validar o projeto em questão, verificando sua eficácia e se os objetivos propostos foram atendidos.

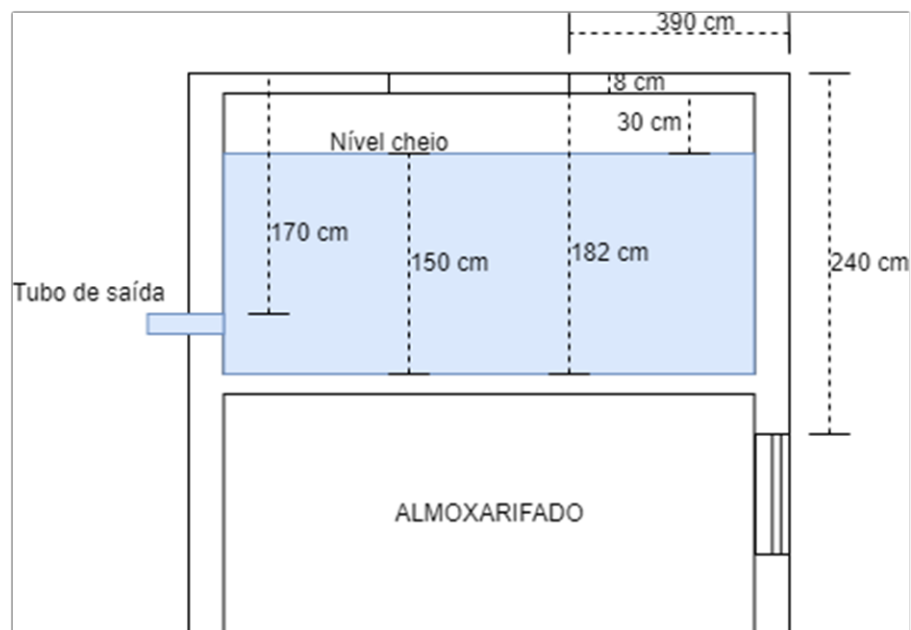
5.1 Sistema embarcado medidor de nível

O SE foi desenvolvido e implantado no reservatório principal do *Campus*, juntamente à montagem e instalação do sensor de nível correspondente. A partir do desenvolvimento do código na linguagem de programação C até a instalação final do projeto, foram necessárias cinco semanas para concluir todas as etapas deste processo.

5.1.1 Sensor de nível

Após a realização das coletas das medidas, foi determinado quais as medidas de interesse para o projeto, como mostrado na Figura 16.

Figura 16: Diagrama do reservatório principal



Fonte: Elaborado pelo autor, 2023

Com base nas medições realizadas e levando em consideração que a área de atuação do flutuador, que regula o fluxo de entrada de água no reservatório, é de 30 cm, com o nível máximo, foi determinado que a primeira chave de nível do sensor deveria ser posicionada a uma altura de um metro em relação ao fundo do reservatório. A chave de nível mais profunda foi instalada 15 cm acima do fundo do reservatório, enquanto a chave intermediária foi colocada 45 cm acima do fundo do reservatório e a altura total do sensor de nível foi de 180 cm. Por fim, a base do sensor foi feita com cimento e água, utilizando uma garrafa plástica como molde. A Figura 17 mostra a sensor de nível ao final da construção. Além do sensor de nível, também foi utilizado cerca de 24 m de fio encapado, sendo que, a cada seis metros, foram utilizadas cores diferentes.

Figura 17: Sensor de nível



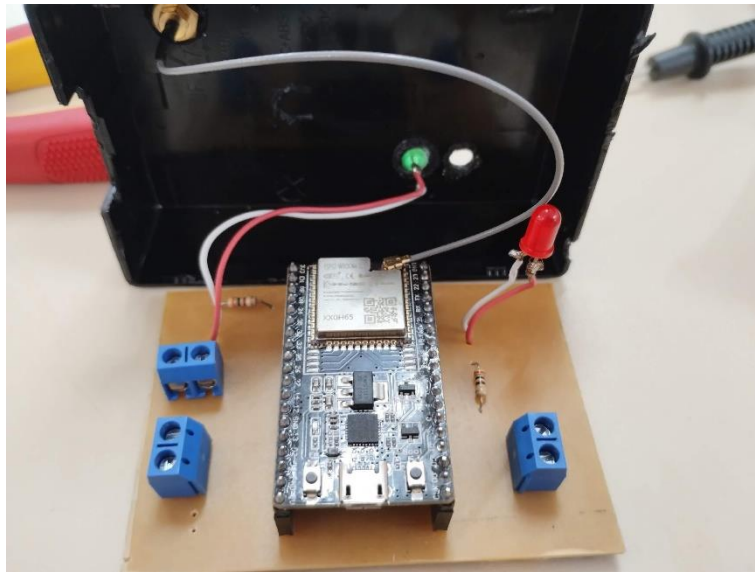
Fonte: Elaborado pelo autor, 2023

5.1.2 Hardware do sistema embarcado

Além dos materiais já mencionados no capítulo 3, também foi necessário utilizar fio de estanho, um ferro de solda, ácido ferroso e outros materiais essenciais para a fabricação de uma placa de circuito impresso, que ilustrada na figura 18. Essa placa foi acondicionada dentro de uma caixa de plástico, com o propósito de proteger os componentes, conforme

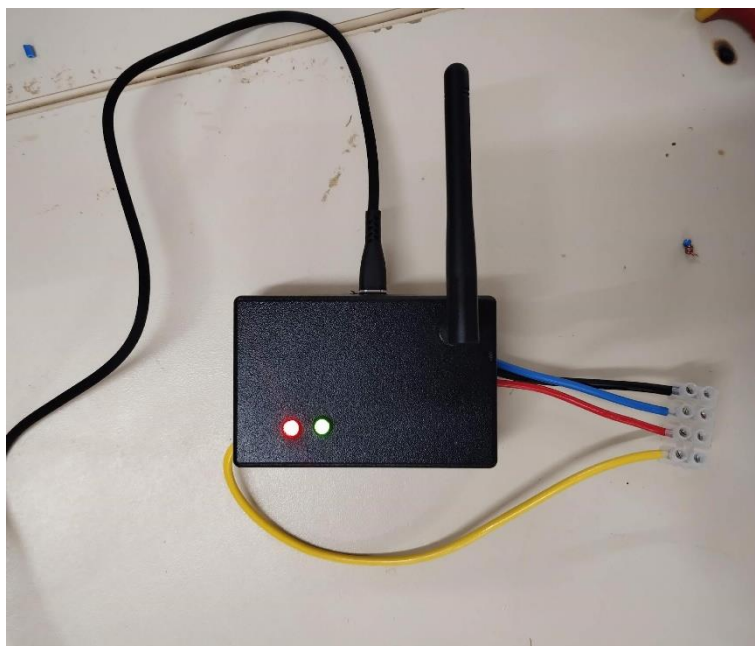
demonstrado na figura 19. Observe a existência de quatro cabos coloridos que se estendem a partir da caixa de plástico, esses cabos foram interligados aos cabos oriundos do sensor de nível, com o cabo azul conectado à chave de nível alta; o cabo vermelho, à chave de nível média; o cabo amarelo, à chave de nível baixa e o cabo preto, ao GND. Adicionalmente, um cabo micro-USB está conectado à porta correspondente no ESP 32, com o propósito de fornecer energia à placa de circuito impresso.

Figura 18: Placa do sistema embarcado



Fonte: Elaborado pelo autor, 2023

Figura 19: Sistema embarcado finalizado



Fonte: Elaborado pelo autor, 2023

5.1.3 Software do Sistema embarcado

O código⁴ do sistema embarcado foi desenvolvido para tratar três possíveis casos de leitura do sensor de nível, qualquer outro tipo de leitura é considerado como erro de leitura, como mostrado na tabela 1. Foi necessário utilizar as bibliotecas WiFi.h e HTTPClient.h para tornar possível a conexão entre o SE e o sistema de comunicação.

5.2 Aplicação Web

A aplicação web⁵ foi desenvolvida, utilizando o *framework Django* e a biblioteca *Selenium*. Ambos foram instalados, utilizando o gerenciador de pacotes *Python*, *pip*. Ao executar o comando "Django-admin createproject nome .", é gerada uma estrutura do *framework* que já funciona como uma aplicação *Web*. Dentro dessa estrutura, foi criado um aplicativo chamado "*sendmsg*", que contém uma função chamada "*sendmsg*" responsável pelo envio de mensagens via *Whatsapp Web*.

Dentro da função "*sendmsg*" e com o auxílio da biblioteca *Selenium*, são executados os seguintes passos:

- Abrir o navegador.
- Acessar URL⁶
- Identificar o botão de enviar mensagem.
- Executar a função de "*click*" no componente do botão.
- Fechar o navegador

5.3 Implantação

A implantação do sistema de monitoramento de reservatórios começou com a instalação do sensor de nível no reservatório principal do *Campus Bambuí*, conforme ilustrado na Figura 20. Essa etapa foi realizada uma semana antes da implantação completa do sistema, com o objetivo de realizar testes preliminares e identificar eventuais vazamentos ou problemas estruturais no sensor.

⁴ https://github.com/Luiz409/Codigo_ESP32

⁵ https://github.com/Luiz409/Aplicacao_Web_TCC

⁶ <https://web.whatsapp.com/send?phone=<numero>&text=<mensagem>>

Figura 20: Sensor de nível implantado



Fonte: Elaborado pelo autor, 2023

O SE foi instalado no almoxarifado do reservatório, conforme ilustrado na Figura 21. Essa escolha de localização foi estratégica, uma vez que a área próxima a uma janela oferecia não apenas uma fonte de alimentação elétrica disponível, mas também permitia que o SE estabelecesse uma conexão Wi-Fi de forma eficaz.

Figura 21: Sistema embarcado implantado



Fonte: Elaborado pelo autor, 2023

A implantação da aplicação *Web* foi realizada em uma máquina virtual hospedada no *datacenter* do *Campus Bambuí*. Essa máquina virtual dispõe de 8 GB de memória principal, 25 GB de memória secundário, quatro núcleos de processamento e opera com um sistema operacional baseado em *Linux*. Além disso, ela é equipada com uma interface gráfica, que foi o ambiente utilizado para a implantação completa da aplicação *Web*. O acesso à máquina virtual foi feito, utilizando um computador *desktop* via protocolo RDP (*Remote Desktop Protocol*).

Em condições usuais, o SE estará com apenas o LED verde acionado, indicando que o ESP32 está funcionando, mas não há nenhum problema, e a aplicação *Web* sendo executada continuamente na máquina virtual. Já quando for detectado que o nível está abaixando, o processo descrito na Figura 14 é executado.

5.4 Validação do sistema

Com sistema de monitoramento implantado, foi realizada uma simulação de falta de água, no dia 15 de setembro de 2023, onde foi desligada a bomba d'água que alimenta o reservatório principal. O desligamento da bomba foi feito quando o reservatório estava com sua capacidade máxima e demorou cerca de 20 minutos até que o nível de água atingisse a primeira chave de nível. Quando a chave foi aberta, o sistema demorou cerca de três minutos para avisar todos os responsáveis de manutenção de que o nível estava abaixando. Em conversa com os responsáveis de manutenção, eles alegaram que, após o recebimento do alerta, eles iriam demorar cerca de 10 minutos para desligarem as bombas d'água que retiram a água do reservatório principal.

Considerando que em dias cotidianos, caso haja problema com o abastecimento do reservatório principal, este levará aproximadamente 60 minutos para esvaziar-se a ponto de permitir a entrada de ar nas tubulações, como descrito pelos responsáveis de manutenção e, considerando que o sistema de monitoramento de reservatório juntamente à ação dos responsáveis de manutenção, levará cerca de 13 minutos para evitar que o ar entre pelas tubulações, fica comprovada a eficácia da solução.

6 CONCLUSÃO

O desenvolvimento do sistema de monitoramento de reservatórios demonstrou ser uma solução eficaz em evitar a entrada de ar pelas tubulações de águas da rede de abastecimento do *Campus*. Ao longo deste trabalho, foram alcançados todos os objetivos, desde o desenvolvimento do SE medidor de nível, com o microcontrolador ESP 32, até a criação da aplicação *Web*, utilizando o *framework Django* e a automação de mensagens via *WhatsApp Web*, para ser utilizado como sistema de comunicação.

O objetivo de integrar essas partes do sistema foi alcançado ao utilizar o protocolo HTTP e a rede local de *Internet* do *Campus*, isso permitiu um monitoramento constante do nível de água no reservatório, com a capacidade de transmitir informações e enviar alertas aos responsáveis pela manutenção. Isso torna-se crucial em situações em que a água pode se esgotar rapidamente.

Os resultados da simulação de falta de água demonstraram que o sistema é capaz de detectar e responder, de forma eficaz, as situações de emergência. Em questão de minutos, os alertas foram enviados aos responsáveis, que puderam tomar medidas preventivas, evitando que o reservatório esvaziasse completamente, impedindo a entrada de ar nas tubulações. Além disso, a implementação do sistema em uma máquina virtual, no *datacenter* do *Campus*, garantiu a disponibilidade contínua da aplicação *Web*. Para futuros projetos, recomenda-se o desenvolvimento de um armazenamento do histórico.

O sistema de monitoramento de reservatórios é uma solução que combina diferentes tecnologias para garantir o abastecimento de água de forma eficaz e confiável. Sua eficácia foi comprovada em simulações, destacando seu valor e importância na infraestrutura do *Campus*. A combinação de *hardware* e *software*, juntamente à integração de diferentes tecnologias, torna esse sistema uma contribuição significativa para a gestão de recursos hídricos e para a manutenção eficiente do sistema de abastecimento do *Campus*.

REFERÊNCIAS

- BEGA, E. A. **Instrumentação industrial**. 3. ed.: Editora Interciência, 2006.
- BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. Novatec Editora, 2014. v. 2.
- CUNHA, A. F. O que são sistemas embarcados. **Saber Eletrônica**, v. 43, n. 414, p. 1–6, 2007. Disponível em: https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf. Acesso em: 22 mai. 2023.
- EICOS. **Sensor de Nível LA16M-40**. 2023. Disponível em: <http://www.eicos.com.br/sensor-de-nivel/montagem-lateral/LA16M-40/>. Acesso em: 10 mai. 2023.
- ESPRESSIF. ESP32 Datasheet, 2021. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 22 mai. 2023.
- FIGUEIREDO, C. R. *et al.* **Elimin-Ar: emprego de equipamentos eliminadores e bloqueadores de ar em ramais prediais de sistemas públicos de abastecimento de água: relatório - R4**. Brasília, 2017. Disponível em: <https://drive.google.com/file/d/1mg0HRruED11byVckwKfiCLrHpSCI2c8i/view>. Acesso em: 12 abr. 2023.
- GALVÃO, C. P. Monitoramento do nível de água em reservatórios residenciais. **Tópicos especiais de Engenharia Civil**, v. 1, p. 8, 2022. Disponível em: https://web.archive.org/web/20221020024622id_/https://poisson.com.br/livros/individuais/Topicos_Engenharia_Civil/volume1/Topicos-especiais_de_Engenharia_Civil_Vol1.pdf#page=8. Acesso em: 22 mai. 2023.
- GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. 1. ed. Porto Alegre: Plageder, 2009. Coordenado pela Universidade Aberta do Brasil – UAB/UFRGS e pelo Curso de Graduação Tecnológica – Planejamento e Gestão para o Desenvolvimento Rural da SEAD/UFRGS. Disponível em: <http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>. Acesso em: 10 mai. 2023.
- HOLOVATY, A.; WILLISON, S. *et al.* Django web framework. **URL <http://www.djangoproject.com>**, 2003.
- HONORATO, W. A.; REIS, R. S. WhatsApp: uma nova ferramenta para o ensino. *In: IV Simpósio de Desenvolvimento, Tecnologias e Sociedade*, p. 1–6, 2014. **Anais [...]**
- KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet - Uma abordagem top-down**. 8. ed.: Editora Pearson, 2021.
- LIMA, J. J. F. de. Sistema embarcado IOT aplicado ao contexto de crises hídricas. **Conjecturas**, v. 22, n. 6, p. 443–457, 2022. Disponível em: <https://www.conjecturas.org/index.php/edicoes/article/view/1080>. Acesso em: 27 mai. 2023.
- MATTOS, A. N. de. **Telemetria e conceitos relacionados**. 2004.
- PRAVALIKA, V.; PRASAD, C. R. *et al.* Internet of things based home monitoring and device

control using Esp32. **International Journal of Recent Technology and Engineering**, v. 8, 1S4, p. 58–62, 2019

ROSE, K.; ELDRIDGE, S.; CHAPIN, L. The internet of things: An overview. **The Internet Society (ISOC)**, v. 80, p. 1–50, 2015. Disponível em: <https://link.springer.com/article/10.1007/s10796-014-9492-7>. Acesso em: 27 jul. 2023.

SCRUM, Desenvolvimento Ágil. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 25 set. 2023.

SILVA, R. J. G. d. *et al.* **Sistema de automação do controle de volume de água em reservatório de abastecimento**. 2021. Trabalho de Conclusão de Curso. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/23748>. Acesso em: 22 mai. 2023.

SOUSA, A. V. d. L. Desenvolvimento de um programa em python para automação da geração de relatórios de acompanhamento de obras. Centro de Engenharias-CE, 2021. Disponível em: <https://repositorio.ufersa.edu.br/handle/prefix/7596>. Acesso em: 24 mai. 2023.

SYSTEMS, E. **ESP HTTP Client**. 2023. Disponível em: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/protocols/esp_http_client.html. Acesso em: 12 mai. 2023.

TANENBAUM, A. S. **Redes de Computadores**. 5. ed. São Paulo, Brasil: Editora Pearson, 2011.

TSCHOFENIG, H. *et al.* **Architectural Considerations in Smart Object Networking**. Copyright © 2015 IETF Trust and the persons identified as the document authors. Mar. 2015. Disponível em: <https://www.rfceditor.org/rfc/pdf/rfc7452.txt.pdf>. Acesso em: 27 jul. 2023.

VIRGENS, G. d. V. L. *et al.* **Projeto de sistema de controle e monitoramento de reservatórios de água**. 2020. Trabalho de Conclusão de Curso. Disponível em: <http://riut.utfpr.edu.br/jspui/handle/1/26575>. Acesso em: 22 mai. 2023.

WAZLAWICK, R. S. **Metodologia de pesquisa para ciência da computação**. Elsevier, 2009. v. 2. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/63429158/Metodologia_de_Pesquisa_para_Ciencia_da_Computacao_by_Raul_Sidnei_Wazlawick_z-lib.org.epub20200526-102268-f3or0k-libre.pdf?1590493476=&response-content-disposition=inline%5C%3B+filename%5C%3DMetodologia_de_Pesquisa_para_Ciencia_da.pdf&Expires=1684847748&Signature=ZpQxNR4yxNQbXPHnXU5ai17wpFOAkCxduH0iai1smF-JR80opF4TSXomyNdJPTt6eeHv92Iv6KaEgJTRRT082hv2BAsJIYEKLhHLSvws7LusHEFApucqQiyuvY5YJTBPUFi1~EguQgzExrpQ1vGAfZwKVYapfiMrWslWmr~w04FOb8t9W3p3Wt3xc~7dTnYonR3FQ7FIC~QKhQCQ3gdDZMPBcV-3jApm4aG0wyZlvpcqCGk6nbvFp-ZAcoaffwx0AOMrQ0UMaaORREgxYItOBT7i1wlj~6DxYEAiwPPEGxktCjKiLC027NgUoU~AkB5fQVQEu3E6ij7Ww6BxN0hlg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. Acesso em: 10 mai. 2023.