

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
COORDENAÇÃO DO CURSO SUPERIOR EM ENGENHARIA ELÉTRICA



TIAGO DALTRO DUARTE

# Sistema Integrado de Monitoramento de Consumo de Água Residencial

João Pessoa-PB

Abril, 2023

TIAGO DALTRO DUARTE

# **Sistema Integrado de Monitoramento de Consumo de Água Residencial**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Superior em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como parte dos requisitos para a obtenção do grau de Engenheiro Eletricista.

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Coordenação do curso superior em Engenharia Elétrica

Orientador: Lincoln Machado de Araújo

João Pessoa-PB

Abril, 2023

Dados Internacionais de Catalogação na Publicação (CIP)  
Biblioteca Nilo Peçanha do IFPB, *campus* João Pessoa

D812s Duarte, Tiago Daltro.

Sistema integrado de monitoramento de consumo de água  
residencial / Tiago Daltro Duarte. – 2023.

82 f. : il.

TCC (Graduação - Curso Superior de Bacharelado em  
Engenharia Elétrica) - Instituto Federal de Educação da Paraíba  
/ Unidade Acadêmica de Processos Industriais, 2023.

Orientação : Prof<sup>o</sup> D.r Lincoln Machado de Araújo.

1. Monitoramento remoto de consumo de água. 2. Siste-  
mas embarcados. 3. Aplicação web. 4. Computação em nuvem.  
5. Microcontrolador ESP32. I. Título.

CDU 004.771:628.1(043)

Lucrecia Camilo de Lima  
Bibliotecária - CRB 15/132

TIAGO DALTRO DUARTE

## Sistema Integrado de Monitoramento de Consumo de Água Residencial

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como parte dos requisitos para a obtenção do grau de Engenheiro Eletricista.

Aprovado pela banca examinadora em: 27/04/2023

### BANCA EXAMINADORA

Documento assinado digitalmente  
 LINCOLN MACHADO DE ARAUJO  
Data: 11/05/2023 16:57:31-0300  
Verifique em <https://validar.iti.gov.br>

---

Lincoln Machado de Araújo, D.Sc – IFPB  
Orientador

Documento assinado digitalmente  
 PATRIC LACOUTH DA SILVA  
Data: 11/05/2023 15:24:51-0300  
Verifique em <https://validar.iti.gov.br>

---

Membro da Banca  
Patric Lacouth da Silva, D.Sc. – IFPB



---

Membro da Banca  
Thiago de Carvalho Batista, D.Sc.– IFPB

João Pessoa  
2023

# AGRADECIMENTOS

A Deus pelo amor e misericórdia.

Aos meus pais por todos os anos de luta e dedicação a mim e a meu irmão.

Um agradecimento especial ao meu irmão, Gabriel Daltro Duarte, por sempre estar comigo nos momentos mais difíceis da minha vida e no curso de Engenharia Elétrica sendo meu professor em todos os momentos que precisei.

A meu amigo e orientador, professor Lincoln, por todo o suporte durante a execução do trabalho e pelas ótimas conversas e orientações brilhantes.

Aos meus queridos amigos Cláudio, Adriano e Luana que estiveram sempre comigo nos momentos difíceis do curso, auxiliando.

Em especial, ao meu querido amigo Kerven Maciel, por sempre me auxiliar, ensinar e estar presente em tantos momentos difíceis que passei durante o estágio e no Curso de Engenharia Elétrica.

A todos os ótimos professores que me ensinaram fascinantes conhecimentos com amor.

Ao IFPB por toda sua ótima infraestrutura.

*“O amor é paciente, é bondoso; o amor não é invejoso, não é arrogante, não se ensoberbece, não é ambicioso, não busca os seus próprios interesses, não se irrita, não guarda ressentimento pelo mal sofrido, não se alegra com a injustiça, mas regozija-se com a verdade; tudo desculpa, tudo crê, tudo espera, tudo suporta.”*

*(São Paulo de Tarso)*

# RESUMO

Com o avanço tecnológico, tornou-se possível a criação e inovação de ideias relacionadas a melhorar a eficiência de várias áreas do trabalho. Como sabemos, a medição de água residencial é feita por hidrômetros e a leitura desses equipamentos são feitas uma vez por mês por leituristas, entregando a fatura ao cliente para ser paga. Assim, seria um grande avanço para a concessionária de distribuição de água, adotar um sistema de monitoramento do consumo de água em tempo real. Este trabalho descreve o desenvolvimento de um sistema integrado para monitoramento do consumo de água, com o objetivo de proporcionar uma melhora significativa na velocidade das medições. O sistema integrado consiste em um dispositivo medidor de consumo de água, que deve ser instalado nas residências dos clientes da concessionária, e uma aplicação web capaz de gerenciar e apresentar esses dados de consumo. Além disso, são apresentados os dados pessoais de cada cliente e o valor a ser pago mensalmente pelo consumo de água em litros. Para a construção do dispositivo de medição foram utilizados o microcontrolador ESP32 que possui conectividade WI-FI e o sensor de vazão YF-S201. As medições de consumo realizadas pelo dispositivo são enviadas para o banco de dados em tempo real da nuvem Firebase, que hospeda os dados de consumo de todas as residências até que sejam lidos pela aplicação web desenvolvida. A aplicação foi implementada com a linguagem de programação Python e o framework Django. Após implementação, ela foi hospedada nos serviços de computação em nuvem Amazon Web Services (AWS). Para a realização dos testes, o dispositivo desenvolvido foi instalado em um fluxo de água e verificou-se na aplicação web os dados de consumo e a geração das faturas referentes ao consumo medido. Com base nos resultados obtidos, conclui-se que o sistema desenvolvido é capaz de monitorar remotamente o consumo de água, gerenciar os dados pessoais dos clientes e computar suas faturas.

**Palavras-chaves:** Monitoramento Remoto de Consumo de água; Sistemas Embarcados; Aplicação Web; Computação em Nuvem; ESP32; YF-S201;

# ABSTRACT

With technological advances, it has become possible to create and innovate ideas related to improving the efficiency of various areas of work. As we know, residential water is measured using hydrometers and these devices are read once a month by meter readers, delivering the invoice to the customer to be paid. Thus, it would be a great step forward for the water distribution concessionaire to adopt a real-time water consumption monitoring system. This work describes the development of an integrated system for monitoring water consumption, with the objective of providing a significant improvement in the speed of measurements. The integrated system consists of a water consumption measuring device, which must be installed in the concessionaire's customers' homes, and a web application capable of managing and presenting this consumption data. In addition, the personal data of each customer and the amount to be paid monthly for water consumption in liters are presented. For the construction of the measuring device, the ESP32 microcontroller was used, which has WI-FI connectivity and the flow sensor YF-S201. Consumption measurements performed by the device are sent to the Firebase cloud real-time database, which hosts the consumption data of all homes until they are read by the developed web application. The application was implemented with the Python programming language and the Django framework. After implementation, it was hosted on Amazon Web Services (AWS) cloud computing services. To carry out the tests, the developed device was installed in a stream of water and the consumption data and the generation of invoices referring to the measured consumption were verified in the web application. Based on the results obtained, it is concluded that the developed system is able to remotely monitor water consumption, manage customers' personal data and compute their invoices.

**Key-words:** Remote Monitoring of Water Consumption; Embedded systems; Web application; Cloud computing; ESP32; YF-S201;

# LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma do Projeto . . . . .	2
Figura 2 – Arquitetura do Projeto . . . . .	4
Figura 3 – Sistema de Supervisão e Controle . . . . .	7
Figura 4 – Sistema de Telemetria . . . . .	8
Figura 5 – Partes de um hidrômetro . . . . .	8
Figura 6 – Esquemático sem a Presença de um Campo Magnético . . . . .	9
Figura 7 – Esquemático com a Presença de um Campo Magnético . . . . .	9
Figura 8 – Componentes de um Microcontrolador . . . . .	12
Figura 9 – Diagrama Básico de um Microcontrolador . . . . .	13
Figura 10 – Microcontrolador INTEL 8051 . . . . .	13
Figura 11 – Processo de Interrupção . . . . .	14
Figura 12 – Arquitetura cliente-servidor . . . . .	16
Figura 13 – Organização da Aplicação em Camadas . . . . .	17
Figura 14 – Comunicação Cliente Servidor . . . . .	18
Figura 15 – Notação do Diagrama Entidade-Relacionamento . . . . .	26
Figura 16 – Projeto Criado no Firebase - Google Cloud . . . . .	27
Figura 17 – Modelo DOIT Esp32 DevKit v1 . . . . .	29
Figura 18 – Diagrama de pinagem da placa de desenvolvimento ESP32 38 pinos . . . . .	30
Figura 19 – Sensor de Fluxo de Água de Efeito Hall - YF-S201 . . . . .	31
Figura 20 – Ferramenta Utilizada para Construção do Software de Medição . . . . .	32
Figura 21 – Protótipo Utilizado para a Medição . . . . .	33
Figura 22 – Componentes Utilizados para Montagem do Circuitos e Testes . . . . .	34
Figura 23 – Esquemático do Circuito Utilizado no Protótipo . . . . .	35
Figura 24 – URLs Adicionais para Gerenciadores de Placas . . . . .	36
Figura 25 – Pacote ESP32 para Instalação . . . . .	36
Figura 26 – Verificação da Placa DOIT ESP32 DEVKIT V1 no Arduino IDE . . . . .	37
Figura 27 – Fluxograma do Algoritmo do Firmware do Dispositivo de Medição de Consumo . . . . .	39
Figura 28 – Visual Studio Code . . . . .	40
Figura 29 – Ilustração da Ferramenta DB Browser for SQLite . . . . .	41
Figura 30 – Ilustração da Ferramenta brModelo . . . . .	42
Figura 31 – Fluxograma Model-Template-View do Django . . . . .	43
Figura 32 – Modelo Conceitual do Banco de Dados do Projeto . . . . .	45
Figura 33 – Estrutura Tarifária Residencial - CAGEPA . . . . .	47
Figura 34 – Hospedando a aplicação Web no AWS Elastic Beanstalk . . . . .	49
Figura 35 – Funcionamento do Protótipo de Medição com a Contagem em 0 litros . . . . .	50

Figura 36 – Funcionamento do Protótipo de Medição com a Contagem de 4 litros . . .	51
Figura 37 – Dados do consumo persistidos no Firebase . . . . .	51
Figura 38 – Página de Login para Acesso do Administrador . . . . .	52
Figura 39 – Cadastro do Administrador . . . . .	52
Figura 40 – Cadastro do Administrador . . . . .	53
Figura 41 – Interface Web Finalizada . . . . .	53
Figura 42 – Opções Disponibilizadas pela Aba de Cliente . . . . .	54
Figura 43 – Opções Disponibilizadas pela Aba de Endereço . . . . .	54
Figura 44 – Ilustração das Opções Disponibilizadas pela Aba de Fatura . . . . .	55
Figura 45 – Ilustração do Cadastro de um Clientes . . . . .	55
Figura 46 – Ilustração da Lista de Clientes . . . . .	56
Figura 47 – Página de Editar Cliente . . . . .	56
Figura 48 – Página dos Dados do Cliente Cadastrado no Sistema . . . . .	57
Figura 49 – Página de Exclusão de um Cliente no Sistema . . . . .	57
Figura 50 – Página do Cadastro de um Novo Endereço . . . . .	58
Figura 51 – Página para Busca do Endereço para Edição . . . . .	58
Figura 52 – Página dos Dados do Endereço do Cliente . . . . .	59
Figura 53 – Página de Busca do Endereço para Exclusão . . . . .	59
Figura 54 – Página de Exclusão do Endereço . . . . .	60
Figura 55 – Página de Lista de Faturas . . . . .	60
Figura 56 – Página de Lista de Faturas com Algumas Faturas . . . . .	61

# LISTA DE TABELAS

Tabela 1 – Métodos Internos de Solicitações HTTP . . . . .	19
Tabela 2 – Grupos de Respostas de Código de Status . . . . .	20
Tabela 3 – Especificações da DOIT ESP32 DEVKIT V1 . . . . .	30
Tabela 4 – Especificações do Sensor de Vazão YF-S201 . . . . .	31
Tabela 5 – Componentes Utilizados no Circuito do Protótipo . . . . .	35

# SUMÁRIO

	<b>Resumo</b> . . . . .	<b>I</b>
	<b>Abstract</b> . . . . .	<b>II</b>
	<b>Lista de ilustrações</b> . . . . .	<b>III</b>
	<b>Lista de tabelas</b> . . . . .	<b>V</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>1</b>
<b>1.1</b>	<b>Objetivo e Justificativa</b> . . . . .	<b>1</b>
<b>1.2</b>	<b>Levantamento Bibliográfico</b> . . . . .	<b>2</b>
<b>1.3</b>	<b>Organização do Trabalho</b> . . . . .	<b>4</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>5</b>
<b>2.1</b>	<b>Automação</b> . . . . .	<b>5</b>
<b>2.2</b>	<b>Sistemas Supervisórios</b> . . . . .	<b>6</b>
<b>2.3</b>	<b>Sistema de Medição Digital de Água</b> . . . . .	<b>7</b>
2.3.1	HIDRÔMETROS . . . . .	8
2.3.1.1	HIDRÔMETROS COM SENSOR DE EFEITO HALL . . . . .	9
<b>2.4</b>	<b>Sistemas Embarcados</b> . . . . .	<b>10</b>
2.4.1	MICROCONTROLADORES . . . . .	12
2.4.2	INTERRUPÇÕES NO MICROCONTROLADOR . . . . .	13
2.4.3	SOFTWARE PARA SISTEMAS EMBARCADOS . . . . .	14
<b>2.5</b>	<b>Aplicações Web</b> . . . . .	<b>15</b>
2.5.1	ARQUITETURA DE APLICAÇÕES WEB . . . . .	17
2.5.1.1	PROTOCOLO HTTP . . . . .	18
2.5.1.2	MÉTODOS HTTP . . . . .	19
2.5.2	COMPUTAÇÃO EM NUVEM . . . . .	20
2.5.2.1	DESENVOLVIMENTO DE APLICAÇÕES COM COMPUTAÇÃO EM NUVEM . . . . .	22
<b>2.6</b>	<b>Banco de Dados</b> . . . . .	<b>24</b>
2.6.1	BANCO DE DADOS RELACIONAIS . . . . .	24
2.6.1.1	ENTIDADES, ATRIBUTOS E CONJUNTOS DE ENTIDADES . . . . .	25
2.6.1.2	CARDINALIDADE . . . . .	26
2.6.1.3	FIREBASE . . . . .	27
<b>3</b>	<b>METODOLOGIA</b> . . . . .	<b>29</b>
<b>3.1</b>	<b>Montagem do Dispositivo de Medição</b> . . . . .	<b>29</b>

3.1.1	DISPOSITIVOS, FERRAMENTAS E LINGUAGENS UTILIZADAS . . . . .	29
3.1.1.1	O MICROCONTROLADOR ESP32 . . . . .	29
3.1.1.2	SENSOR DE FLUXO DE ÁGUA YF-S201 . . . . .	30
3.1.1.3	ARDUÍNO IDE . . . . .	31
3.1.1.4	LINGUAGEM C++ . . . . .	32
3.1.2	MONTAGEM DO DISPOSITIVO DE MEDIÇÃO DE CONSUMO DE ÁGUA	33
3.1.3	FIRMWARE DO DISPOSITIVO DE MEDIÇÃO DE VOLUME . . . . .	35
3.1.3.1	PREPARANDO O AMBIENTE DE DESENVOLVIMENTO . . . . .	35
3.1.3.2	ALGORITMO DO FIRMWARE DO DISPOSITIVO DE MEDIÇÃO DE CONSUMO .	37
<b>3.2</b>	<b>Desenvolvimento da Aplicação Web . . . . .</b>	<b>40</b>
3.2.1	FERRAMENTAS E LINGUAGENS UTILIZADAS . . . . .	40
3.2.1.1	VISUAL STUDIO CODE . . . . .	40
3.2.1.2	DB BROWSER FOR SQLITE . . . . .	40
3.2.1.3	BRMODELO . . . . .	41
3.2.1.4	FRAMEWORK WEB PYTHON - DJANGO . . . . .	42
3.2.1.5	HTML E CSS . . . . .	44
3.2.2	BANCO DE DADOS . . . . .	45
3.2.2.1	BANCO DE DADOS SQLITE . . . . .	46
3.2.3	PREPARANDO O AMBIENTE DE DESENVOLVIMENTO . . . . .	46
3.2.4	FUNCIONALIDADES DA APLICAÇÃO WEB . . . . .	46
3.2.4.1	CÁLCULOS DA FATURA . . . . .	47
<b>3.3</b>	<b>Implantação da aplicação web no AWS . . . . .</b>	<b>49</b>
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>50</b>
<b>4.1</b>	<b>Resultados Obtidos . . . . .</b>	<b>50</b>
4.1.1	TESTE E FUNCIONAMENTO DO PROTÓTIPO DE MEDIÇÃO . . . . .	50
4.1.2	INTERFACE WEB . . . . .	52
4.1.2.1	LOGIN E CADASTRO DO ADMINISTRADOR . . . . .	52
4.1.2.2	PÁGINA PRINCIPAL DA INTERFACE WEB . . . . .	53
4.1.2.3	CADASTRO DE UM CLIENTE . . . . .	55
4.1.2.4	LISTAR CLIENTES . . . . .	55
4.1.2.5	CADASTRAR UM NOVO ENDEREÇO . . . . .	57
4.1.2.6	EDITAR ENDEREÇO . . . . .	58
4.1.2.7	LISTA DE FATURAS . . . . .	60
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>62</b>
<b>6</b>	<b>TRABALHOS FUTUROS . . . . .</b>	<b>63</b>
<b>7</b>	<b>ACESSO AOS CÓDIGOS DO PROJETO . . . . .</b>	<b>64</b>

**REFERÊNCIAS . . . . . 65**

# 1 INTRODUÇÃO

Com o avanço tecnológico, tornou-se possível a criação e inovação de ideias relacionadas a melhorar a eficiência de várias áreas de trabalho. Uma das áreas que pode ser muito beneficiada, é a área de medição de consumo de água.

Segundo o (IBGE, 2019) 85,5% dos domicílios contam com a Rede Geral de distribuição como a principal fonte de abastecimento de água. Todos esses domicílios são obrigados a utilizar os hidrômetros, para monitoramento do consumo de água. O hidrômetro é um dispositivo mecânico responsável pela medição do volume de água que atravessa determinada tubulação onde ele está instalado. Este aparelho utiliza um contador chamado relógio que fica atualizando o consumo de água da residência em tempo real.

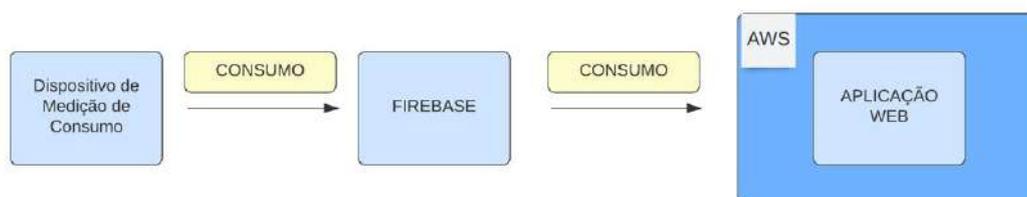
O monitoramento do consumo de água residencial é feito pelas concessionárias de forma manual. As concessionárias de abastecimento de água contratam funcionários para a função de leituristas e eles são responsáveis por verificar todas as residências para fazer a leitura dos hidrômetros instalados. A forma como a leitura é realizada em tempos atuais é ultrapassada e pode ocasionar falhas tanto para os consumidores quanto para as companhias de abastecimento, tais como impossibilidade de acesso ao hidrômetro em tempo hábil, leitura de forma incorreta gerando uma cobrança indevida e dificuldade para o consumidor em saber seu consumo diário. Além disso, segundo (CARVALHO, 2010), é comum que leituristas não consigam acesso ao hidrômetro por conta da ausência do responsável do domicílio, fazendo com que seja necessário o retorno do leiturista ao local ou a cobrança por meio de estimativas (JÚNIOR; CORAL, 2021).

## 1.1 Objetivo e Justificativa

Diante do exposto, este trabalho tem como objetivo desenvolver um protótipo para o monitoramento do consumo de água, onde será possível para um administrador de determinada concessionária visualizar os dados relativos ao consumo de água da residência dos clientes. Esses dados serão coletados utilizando um protótipo construído com um sensor de vazão, conectado a um circuito que integra um microcontrolador com conectividade Wi-Fi. O microcontrolador é responsável por transmitir os dados de consumo para o banco de dados em nuvem Firebase, que é uma plataforma desenvolvida pelo Google para aplicação de aplicativos móveis e da web. Integrado ao sistema, também foi construído uma aplicação web utilizando o framework Django da linguagem Python, para o processamento dos dados de consumo salvos no Firebase. A aplicação web, destinada ao uso das

concessionárias de água, poderá ser acessada por qualquer navegador instalado em um computador ou smartphone com acesso a internet. A plataforma disponibilizará os dados de consumo juntamente com os dados pessoais do cliente, assim como, seus endereços e listas de faturas referentes a cada endereço que os clientes possuírem. Após o término do desenvolvimento da aplicação web, ela será hospedada na Amazon Web Services(AWS) que é uma plataforma de serviços de computação em nuvem. Na Figura 1 é representado o esquemático do sistema integrado desenvolvido.

Figura 1 – Fluxograma do Projeto



Fonte: Autória Própria

## 1.2 Levantamento Bibliográfico

Para a elaboração do projeto, foi realizada uma pesquisa sobre trabalhos que propunham sistemas de monitoramento de consumo residencial em tempo real. Foram analisados diversos trabalhos de conclusão de cursos, artigos científicos, dissertações, sobre o tema e os conceitos necessários.

Serão apresentados abaixo, alguns dos trabalhos analisados para construção desse trabalho.

(SOUZA; NUNES; BIANCHINI, 2016) desenvolveram um sistema de monitoramento de consumo de água residencial. Para esse desenvolvimento foi utilizado um CPU de baixo custo, denominado Raspberry Pi. Onde esse sistema apresenta um monitoramento em tempo real, por meio de um firmware específico, desenvolvido para analisar os dados coletados através da captura desses dados utilizando um sensor de vazão de água. Também desenvolveram uma interface em linguagem Python, responsável por analisar todos os dados obtidos, coletados, e plotar gráficos, referentes ao consumo de água das últimas 24 horas, última semana, último mês ou algum intervalo definido pelo próprio usuário. Para finalizar, o grupo desenvolveu um adicional ao código do projeto com a funcionalidade de "twittar", em uma conta específica, informações relativas ao consumo de água observado pelo sistema como um todo.

(DUTRA, 2019) desenvolveu um aplicativo mobile para a plataforma iOS, para exibição e acompanhamento online, com acesso remoto do consumo residencial de um

usuário. Foi desenvolvido um hidrômetro digital, a partir da utilização de um controlador ESP826612-E, juntamente com um sensor de vazão YF-S201. Onde o microcontrolador ESP826612-E lê os dados advindos do sensor e disponibiliza os mesmos via Wi-Fi para serem armazenados na plataforma Firebase, que é um banco de dados online da empresa Google. Dessa forma, o aplicativo constituído pelas linguagens Swift 4.0 e IDE Xcode 10.1, coleta os dados do consumo que estão disponibilizados no Firebase e exibi esses dados de consumo ao usuário em uma interface.

(VENDEMIATTI, 2020) desenvolveu um medidor remoto e automático para geração de dados que fosse possível detectar alterações em padrões de consumo residenciais e industriais, descobrindo assim, vazamentos com maior agilidade. Dessa forma, para construção desse medidor, foi utilizado sensores de vazão e pressão, um circuito de condicionamento para conexão entre os sensores e o microcontrolador com conectividade Wi-Fi. Também foi desenvolvida uma API em PHP para receber os dados externamente do microcontrolador via Wi-Fi, utilizando a requisição HTTP, e dessa forma, persistir os dados no banco de dados que foi criado, com ajuda de um navegador Web. O sistema desenvolvido foi instalado em pontos consumidores para coleta de dados reais de consumo. Com os dados coletados, foi possível estabelecer padrões de utilização do consumo, colaborando para a detecção de ocorrências de vazamentos em determinados pontos, assim como, alterações nos padrões de consumo dos consumidores. Essas alterações ficam bem mais evidentes devido a mudança na rotina do consumo da água.

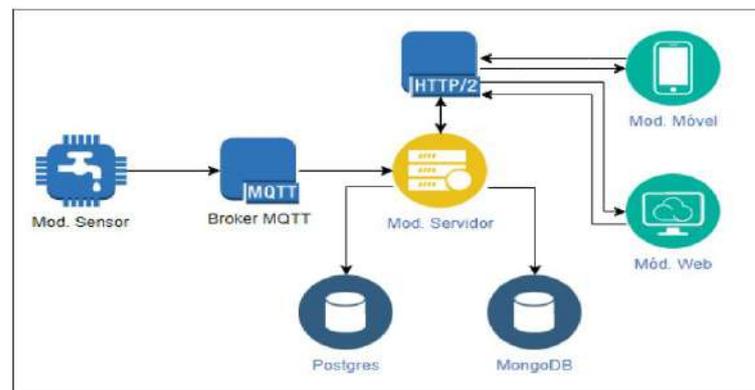
(AMERICO; IZIDORO, 2022) desenvolveu um sistema eletrônico capaz de monitorar o consumo de água de uma residência, empresa, ou outros estabelecimentos. Visou, encontrar vazamentos ou desvios de água nas tubulações, através da medição do consumo, feita por um sensor de fluxo YF-S201, enviando dados através de um microcontrolador ESP 32, via Wi-Fi e disponibilizando as informações para uma aplicação Mobile, na qual o usuário pode ter acesso a todo o sistema de supervisão. Esta aplicação é conectada em nuvem do Blynk, assim como toda a aplicação desenvolvida no dispositivo também. Os dados visualizados na aplicação no dispositivo, são: o percentual de bateria do dispositivo, o consumo de água em ( $\text{m}^3/\text{h}$ ) e o valor em reais ( $\text{R}\$/\text{m}^3$ ) do consumo.

(JÚNIOR; CORAL, 2021) desenvolveu um protótipo utilizando conceitos de IoT para monitorar o consumo de água residencial e industrial, utilizando um sistema web criado para gerenciamento e acompanhamento dos dispositivos e consumidores. Também desenvolveu um sistema móvel para o consumidor acompanhar o monitoramento. O projeto em si, foi dividido em quatro etapas pelo autor: sensor, servidor, sistema web e móvel. O módulo sensor YF-B2 utilizado é responsável pela leitura dos dados, já o servidor processa os dados, gerencia os usuários, protótipos de medição e de armazenamento em banco de dados em tempo real. O módulo Web é destinado as companhias de distribuição de recursos hídricos para gerenciar os consumidores e por fim, o módulo móvel que é um

aplicativo destinado aos consumidores para monitoramento dos protótipos vinculados ao consumidor, podendo ver de forma minuciosa as medições informadas.

(MELO, 2018) desenvolveu um sistema de monitoramento do consumo de água onde foi utilizado o protocolo de comunicação MQTT (Message Queue Telemetry Transport), e o microcontrolador NodeMCU Dev Kit v1.0, além do sensor de vazão YF-S201B. Também foi organizada uma plataforma chamada ThingSpeak para exibição dos dados. A plataforma com serviços de análise voltada para IoT e integrada ao MATLAB, que permite visualizar, analisar e processar fluxos de dados na nuvem. Nessa plataforma, é possível os usuários projetarem e construir aplicações IoT sem a necessidade de configurar servidores ou desenvolver aplicações web. É importante destacar que todas as etapas desse trabalho, segundo (MELO, 2018), foram implementadas e o protótipo apresentou resultados satisfatórios. Na Figura 2 é apresentado a arquitetura do projeto.

Figura 2 – Arquitetura do Projeto



Fonte: (JÚNIOR; CORAL, 2021)

### 1.3 Organização do Trabalho

Este trabalho está organizado em 7 capítulos, incluindo o corrente capítulo, a Introdução. No capítulo 2 é realizada a fundamentação teórica do trabalho. São apresentados neste capítulo todos os conceitos e definições necessários para a construção do projeto. No capítulo 3 é descrita de forma detalhada a metodologia empregada no trabalho para alcance dos objetivos. No capítulo 4 são apresentados os resultados obtidos em cada etapa da metodologia adotada. No capítulo 5 encontram-se as conclusões do trabalho e no capítulo 6 serão apresentadas melhorias no trabalho desenvolvido que podem se converter em trabalhos futuros. Por fim, é apresentado o capítulo 7 para a visualização dos códigos construídos do projeto e logo após, as referências bibliográficas utilizadas no desenvolvimento do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta Capítulo, é apresentado um levantamento bibliográfico sobre o tema proposto: Sistema Integrado de Monitoramento de Consumo de Água Residencial. Para isso, destacam-se alguns trabalhos já realizados para o entendimento geral do tema, com ideias e possíveis soluções já obtidas, que foram cruciais para o desenvolvimento deste trabalho. Abordando alguns conceitos sobre, automação, desenvolvimento e funcionamento dos sistemas supervisórios, e os sistemas de medição de água. Além disso, são expostos alguns conceitos importantes sobre internet das coisas(IoT), computação em nuvem, sistemas embarcados, aplicações web e banco de dados.

### 2.1 Automação

A automação de processos é algo cada vez mais comum na indústria, com linhas de montagem inteiras operando muitas vezes com muito pouca ou até mesmo nenhuma intervenção humana. Entretanto, a automação está, principalmente nos últimos anos, indo além da aplicação industrial. Hoje em dia já são encontradas no mercado diversas soluções em automação para fins residenciais, por exemplo. E esta disseminação e ampliação do campo de aplicação em soluções automatizadas possui forte tendência de crescimento(SOUZA; NUNES; BIANCHINI, 2016).

A automação residencial, também conhecida por Domótica, termo originado da junção das palavras Domus (casa em latim) e a palavra robótica (controle automatizado de algo), é a tecnologia que integra os dispositivos eletrônicos e eletromecânicos de uma residência, possibilitando o controle de forma automática de tarefas que antes dependiam unicamente dos moradores, incluindo iluminação, sistema de som, climatização, controles de acesso, dentre outros (COSTA, 2020).

A Expansão da domótica está relacionado com as inúmeras vantagens que a mesma possibilita e pela sua constante modernização, que além de visar o conforto, eficiência energética e acessibilidade, proporciona um conceito cada vez mais moderno de moradia. Esses pontos geram grande aceitação dos clientes e difusão do mercado da automação residencial. A domótica tem como característica a evolução dos meios de comunicação, principalmente para a transmissão sem fio, que enseja a interatividade dos diversos equipamentos presentes nas residências sem a necessidade de cabeamentos, para que através dessa centralização possibilitem o controle, supervisão e reprogramação de maneira simples, para atender as necessidades do usuário, podendo, por exemplo, ser manipulada através de um smartphone ou um computador (SANTOS; JUNIOR, 2019).

Com o desenvolvimento da eletrônica ao longo dos anos, houve um aumento significativo de benefícios relacionados a automação residencial, bem como a redução dos preços de algumas tecnologias no mercado. Dessa forma, esses fatores tornaram a automação residencial mais acessível a grande parte da população. Esses benefícios, como controle de iluminação residencial, controle dos portões da residencial, controle por comando de voz relacionados a alguns dispositivos existentes na residência, acessibilidade, conforto e eficiência energética, trazem para a população, mais conforto, praticidade e comodidade.

## 2.2 Sistemas Supervisórios

Um Sistema de supervisão Scada(Supervisory Control and Data Acquisition - Sistemas de Supervisão e Aquisição de Dados) é um software que utiliza tecnologia de informação e de comunicação para o monitoramento e controle, coletando e organizando dados, para apresentar ao operador através da interface homem máquina (IHM) (SANTOS; MARINATO; SANTOS, 2021).

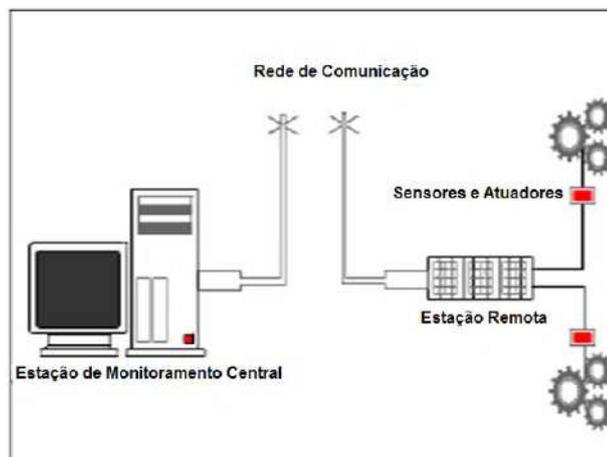
Já para (MORAES; CASTRUCCI, 2007), um Sistema Supervisório é um sistema digital de monitoração e operação que gerencia variáveis envolvidas em todo um processo. Todas as informações obtidas e coletadas são continuamente atualizadas e podem ser guardadas em um banco de dados local ou remoto para fins de registro histórico e posterior uso, como desenvolvimento de relatórios, planilhas, gráficos e diversas outras funções, variando de acordo com o interesse do usuário. (SOUZA; NUNES; BIANCHINI, 2016).

Os primeiros sistemas SCADA, basicamente telemétricos, permitiam informar periodicamente o estado corrente do processo industrial, monitorando sinais representativos de medidas e estados de dispositivos, através de um painel de lâmpadas e indicadores, sem que houvesse qualquer interface aplicacional com o operador.

O sistema SCADA, é uma tecnologia que permite esse monitoramento, rastreando a informação do processo da instalação física do sistema. As informações são coletadas por equipamentos coletores de dados, manipulando e analisando, depois armazenando e permitindo que o usuário tenha acesso a esses dados. O sistema SCADA é composto tipicamente por estações terminais mestre, unidades remotas como, por exemplo, o CLP (Controlador Lógico Programável), RTU (Remote Terminal Unit - Unidade Terminal Remota) e IED (Intelligent Electronic Device - Dispositivo Eletrônico Inteligente). Temos também os dispositivos de campo, que são os sensores e atuadores. Além disso, temos toda a tecnologia de comunicação embarcada juntamente com o software, para a interconexão desses componentes (SANTOS; MARINATO; SANTOS, 2021).

Uma representação do sistema SCADA pode ser visualizado na Figura 3 abaixo.

Figura 3 – Sistema de Supervisão e Controle



Fonte: (AZEVEDO; AZEVEDO, 2011)

### 2.3 Sistema de Medição Digital de Água

É necessária a medição do volume de água que circula em todo o sistema de distribuição e consumo, e para isto devem ser empregados medidores apropriados para tal função. Tais medidores sofreram diversas modificações desde o início da necessidade de medição, ainda na época dos sumérios, até os dispositivos disponíveis na atualidade, que embarcam diferentes tecnologias (VENDEMIATTI, 2020).

Gradativamente, várias técnicas de controle inteligente vêm sendo aplicada em diversas esferas na sociedade, utilizando-se várias áreas do conhecimento para melhorar o controle e monitoramento de recursos naturais, uma preocupação cada vez mais presente nos dias atuais. Estas técnicas tiveram um desenvolvimento bastante significativo quando a tecnologia de processamento se tornou mais eficiente e acessível (SOUZA; NUNES; BIANCHINI, 2016).

Conceitua-se telemetria como o processo de medição remoto da variação das características dos objetos monitorados (por exemplo, fluxo de água em um hidrômetro) que transfere os resultados coletados para uma estação distante, onde serão analisados e consistidos. A arquitetura de telemetria é composta por três subsistemas: o Elemento de Medição Remota (EMR), que engloba os medidores e os componentes de transmissão; a rede de telecomunicações utilizada, podendo ser a internet, rede de celular, rede telefônica ou outras; e o sistema de monitoramento e gerenciamento da informação, onde são analisadas todas as informações obtidas remotamente (BRANDÃO, 2022).

A Figura 4 representa um sistema básico de telemetria, onde temos o módulo EMR mencionado anteriormente, a rede de telecomunicações e o sistema de monitoramento.

Figura 4 – Sistema de Telemetria



Fonte: (SILVA et al., 2007)

O EMR (Elemento de Medição Remota) faz a medição dos dados desejados para que sejam enviados através do subsistema de transmissão (SILVA et al., 2007). Os dados são imediatamente enviados pela rede de telecomunicações utilizada, podendo ser a internet, Wi-Fi, rede de celular, rede telefônica ou outras; para uma central de monitoramento, com o objetivo de otimizar recursos e racionalizar custos.

### 2.3.1 Hidrômetros

O hidrômetro é um instrumento de medição de água muito utilizado pelas empresas de saneamento básico para medir o consumo de água dos seus clientes, permitindo a emissão das contas de acordo com o volume e a distribuição de água. (DUTRA, 2019)

A classificação dos hidrômetros como sendo para uso residencial ou industrial está mais relacionada ao porte do instrumento em si do que a outras mudanças e características de funcionamento. Já a classificação quanto ao princípio de funcionamento utilizado possui certa relevância, visto que dependendo da aplicação visada para o instrumento, certos tipos de hidrômetros terão melhor resposta do que outros. (SOUZA; NUNES; BIANCHINI, 2016)

Na Figura 5 é representado um esquema das partes fundamentais de um hidrômetro.

Figura 5 – Partes de um hidrômetro



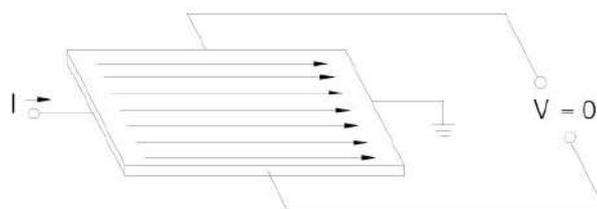
Fonte: (LTDA, 2019)

### 2.3.1.1 Hidrômetros com Sensor de Efeito Hall

O sensor de efeito hall recebe este nome por ser baseado no efeito Hall descoberto em 1879 por Edwin Hall. Quando um condutor que transporta corrente elétrica é colocado em um campo magnético, uma tensão será gerada perpendicularmente à corrente e ao campo. Esse princípio é conhecido como efeito Hall.

A Figura 6 ilustra o princípio básico do efeito Hall. Ele mostra uma fina folha de material semiconductor (elemento Hall) através da qual uma corrente passa. As conexões de saída são perpendiculares à direção da corrente. Quando nenhum campo magnético está presente, a distribuição de corrente é uniforme e nenhuma diferença de potencial é observada na saída (HONEYWELL, 2016).

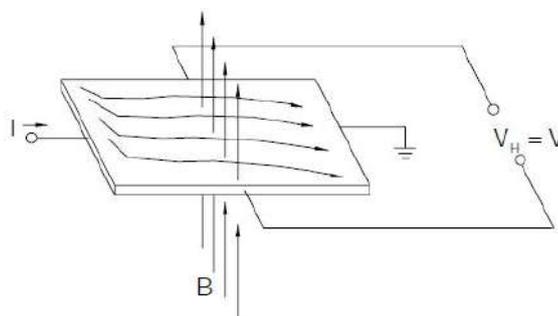
Figura 6 – Esquemático sem a Presença de um Campo Magnético



Fonte: (HONEYWELL, 2016)

Quando um campo magnético perpendicular está presente conforme ilustrado na Figura 7, o fluxo de corrente é distorcido. A distribuição resultante provoca o aparecimento de uma tensão entre os terminais de saída. Esta tensão é chamada Tensão Hall (HONEYWELL, 2016).

Figura 7 – Esquemático com a Presença de um Campo Magnético



Fonte: (HONEYWELL, 2016)

Portanto, o sensor de efeito Hall é composto por duas partes, uma parte móvel com unidades de material ferro magnético (imã) e a outra parte é um sensor magnético feito de um semiconductor capaz de detectar a presença de campos magnéticos. Quando a peça composta por imãs em sua superfície se desloca e passa pelo sensor magnético, gera

um pulso nos terminais de saída do sensor. Logo, se o sensor de efeito Hall for conectado a uma pequena turbina pela qual passa água, é possível se determinar o volume de água que passa pela turbina através da contagem de pulsos gerados pelo sensor, uma vez que existe uma relação direta entre o número de pulsos gerados pelo sensor e do número de rotações da turbina em torno de seu eixo (ROMERO, 2019).

## 2.4 Sistemas Embarcados

A indústria eletrônica tem crescido nos últimos anos a uma taxa impressionante e um dos principais motivos para tal crescimento é a incorporação de sistemas eletrônicos numa grande variedade de produtos tais como automóveis, eletrodomésticos e equipamentos de comunicação pessoal. Sistemas de computação estão presentes em todo lugar e não é surpresa que anualmente são produzidos milhões de sistemas destinados a computadores pessoais (desktop), estações de trabalho, servidores e computadores de grande porte. O que pode surpreender, no entanto, é que bilhões de sistemas são produzidos anualmente para as mais diferentes propostas; tais sistemas estão embutidos em equipamentos eletrônicos maiores e executam repetidamente uma função específica de forma transparente para o usuário do equipamento. Como resultado da introdução de sistemas eletrônicos em aplicações tradicionais temos produtos mais eficientes, de melhor qualidade e mais baratos. Dentre os componentes eletrônicos mais utilizados temos os componentes digitais que permitem algum tipo de computação tais como microprocessadores e microcontroladores (BARROS; CAVALCANTE, 2010).

O termo embarcado tem sua origem no fim da década de 1960, nesse período existia apenas um simples programa de controle de telefones, sendo adaptado para outros dispositivos. Posteriormente, com o aumento dos microprocessadores tornou-se possível desenvolver software específico para os variados tipos de processador. Na década de 1970, começavam a surgir bibliotecas de códigos direcionados para sistemas embarcados com processadores específicos. (PONT, 2020)

Um sistema Embarcado, é definido pelo (IEEE, 1990 apud ZURITA, 2007) como "um sistema computacional que faz parte de um sistema maior e implementa alguns dos requerimentos deste sistema". Esta definição estabelecida há mais de duas décadas continua válida, embora a revolução experimentada por esse segmento durante estes últimos anos tenha impulsionado alguns autores a complementarem-na. Nesta linha, Steve Heath o define como sendo "um sistema baseado em um microprocessador, que é projetado para controlar uma função ou uma gama de funções, e não para ser programado pelo usuário final como ocorre com os PCs"(ZURITA, 2007).

Os sistemas embarcados(SE) estão aplicados em diversas atividades humanas devido ao baixo custo e maior facilidade de aplicação se comparado com outros sistemas. São

encontrados em telefones celulares, sistemas de controle dos automóveis e aviões, eletrodomésticos e na domótica. São sistemas que possuem uma capacidade de processamento de informações vinda de um software que está sendo executado internamente nessa unidade (PONT, 2020).

Os sistemas embarcados possuem algumas características comuns que são definidas a seguir, como por exemplo, funcionalidade única, restrições de projeto mais rígidas e os sistemas reativos em tempo real. Esse conjunto de características são citadas em (BARROS; CAVALCANTE, 2010).

- **Funcionalidade única:** Usualmente um sistema embarcado executa somente um programa repetidamente. Por exemplo um pager é sempre um pager, enquanto um computador pessoal pode executar variedades de programas.
- **Restrições de projeto mais rígidas:** Todos os sistemas de computação possuem em geral alguma restrição de projeto a ser satisfeita, como por exemplo custo, tamanho, desempenho, potência dissipada, etc. Nos sistemas embarcados, no entanto, estas restrições são normalmente mais rígidas, por exemplo o custo de um sistema não pode ser muito alto para não onerar o custo do equipamento, o tempo de resposta deve permitir em várias aplicações processamento em tempo real e devem dissipar pouca potência para permitir uma maior duração da bateria ou não necessitar de um sistema de refrigeração.
- **Sistemas reativos de tempo real:** Muitos sistemas embarcados devem reagir a mudanças no ambiente e devem fornecer resultados em tempo real. Por exemplo um piloto automático (cruise controller) continuamente monitora e reage a velocidade e aos sensores de freio. Ele deve computar a aceleração e desaceleração repetidamente num intervalo de tempo. Caso haja um retardo o controle do carro pode ser perdido.

Para entender como funcionam esses sistemas embarcados e a necessidade de utilizá-los, é preciso conhecer a forma como trabalham os processadores. Os processadores podem ser divididos em duas categorias: microprocessadores e microcontroladores. Os Microprocessadores são circuitos integrados capazes de executar instruções a este submetidas, agem sob o controle de um programa armazenado na memória, executando operações aritméticas, lógica booleana, além de controlar a entrada e saída de dados, permitindo a comunicação com outros dispositivos e periféricos. Os microprocessadores estão presentes em sua quase totalidade em computadores. Como os microprocessadores precisam necessariamente de outros dispositivos para seu funcionamento, tais como memória e controladores de entrada e saída de dados, o mercado criou uma necessidade de um sistema de processamento, que já incorporasse os dispositivos necessários para seu funcionamento, assim, os micro-controladores foram criados (PEREIRA; CARVALHO; MORAES, 2014).

### 2.4.1 Microcontroladores

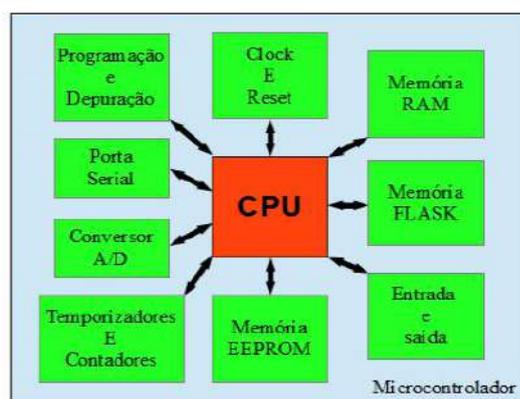
Os microcontroladores podem ser encontrados em praticamente todos os dispositivos eletrônicos digitais que nos cercam: teclado do computador, dentro do monitor, disco rígido, relógio de pulso, rádio relógio, máquinas de lavar, forno de micro-ondas, telefones, etc. Certamente eles foram tão ou mais importantes para a revolução dos produtos eletrônicos que os computadores. Eles permitiram a evolução de equipamentos que a anos não evoluíam, como os motores a combustão, que agora com o novo controle eletrônico podem funcionar com sistema bicomustível e poluindo menos (FUTIDA; ROMERO, 2008).

O microcontrolador é um pequeno computador, formado por um único circuito integrado, que reúne um núcleo de processador, memórias voláteis e não voláteis e diversos periféricos de entrada e saída de dados. Além disso, o microcontrolador possui algumas vantagens ao ser utilizado, como por exemplo, a economia, ao ser levantado o fator financeiro do projeto, o tamanho do microcontrolador, que acaba sendo um benefício na montagem e organização do projeto, a diminuição drástica do consumo de energia, sabendo que o microcontrolador consome pouca energia, e uma das principais características é que o sistema pode utilizar o modo de espera, diminuindo ainda mais o consumo de energia.

Um microcontrolador pode incorporar uma grande variedade de dispositivos como: conversores analógico-digitais (ADC) e digital-analógicos (DAC), temporizadores, contadores, interfaces seriais, memória de instruções e/ou dados, controladores de interrupção, geradores de *clock*, controladores de DMA, etc. Por isso, é comum que não seja desenvolvido apenas um, mas sim uma família de microcontroladores, cada um deles apresentando um conjunto diferente de dispositivos, frequência de *clock*, potência consumida, faixa de temperatura suportada, encapsulamento e preços compatíveis com estas facilidades (BARROS; CAVALCANTE, 2010).

A figura 8 ilustra alguns componentes de um Microcontrolador.

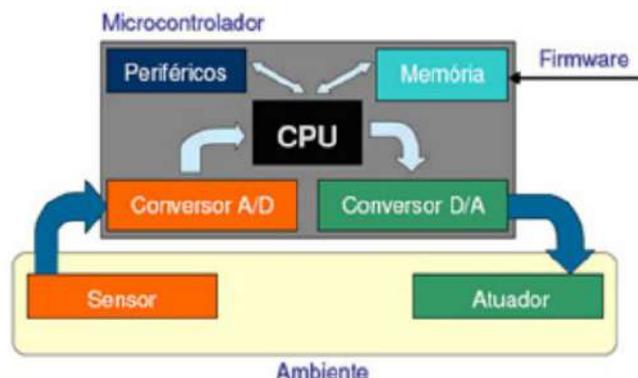
Figura 8 – Componentes de um Microcontrolador



Fonte: (KERSCHBAUMER, 2021)

Na Figura 9 é ilustrado um sistema embarcado controlando uma determinada tarefa de monitoração do ambiente.

Figura 9 – Diagrama Básico de um Microcontrolador



Fonte: (PEREIRA; CARVALHO; MORAES, 2014)

Os microcontroladores surgiram como uma evolução natural dos circuitos digitais devido ao aumento da complexidade dos mesmos. Chegou-se a um ponto em que era mais simples, mais barato e mais compacto, substituir a lógica das portas digitais por um conjunto de processador e software. O primeiro microcontrolador foi lançado pela empresa INTEL em 1977 e recebeu a sigla "8048". Com a sua posterior evolução, deu origem à família "8051", representado na Figura 10 abaixo. Esse chip pode ser programado em linguagem C e possui um poderoso conjunto de instruções (PENIDO; TRINDADE, 2013).

Figura 10 – Microcontrolador INTEL 8051



Fonte: (PENIDO; TRINDADE, 2013)

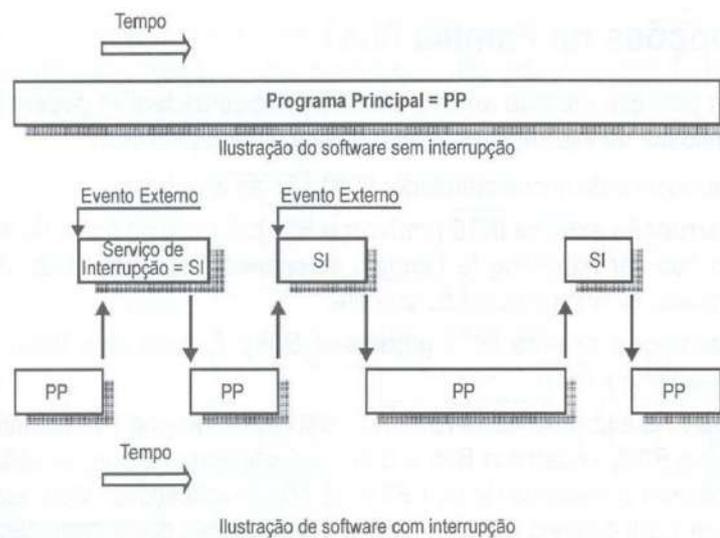
## 2.4.2 Interrupções no Microcontrolador

As Interrupções são extremamente utilizadas nos estudos e aplicações dos microcontroladores e microprocessadores. Em computadores pessoais, PCs, quando uma tecla do teclado é pressionada, é gerada uma interrupção para o microprocessador, requisitando o tratamento desta. Dessa forma, o PC não é obrigado a verificar constantemente o teclado em busca de uma tecla. O teclado possui um microcontrolador dedicado para tratar

e transmitir de maneira serial a informação sobre as teclas que estão sendo pressionadas. Na outra ponta, na placa-mãe do PC, possui um controlador responsável por armazenar as teclas pressionadas e gerar a interrupção. A interrupção é um meio usado para avisar a unidade principal de processamento do sistema, que uma função mais importante deve ser executada em um dado momento.

Ilustra-se na Figura 11 o fluxo de um software sem interrupção e do software com interrupção.

Figura 11 – Processo de Interrupção



Fonte: (NICOLSI, 2007)

Segundo (NICOLSI, 2007), interrupção é um evento externo ou interno que obriga o microprocessador a suspender suas atividades temporariamente, para atender a este evento que o interrompeu. Em resumo, é uma ocorrência que faz o microprocessador parar a sua rotina e se desviar para outro ponto do software, em que se localiza o serviço de interrupção que foi gerado pela ocorrência. É importantíssimo na estratégia de programação do microcontrolador. (É como uma sub-rotina na programação tradicional só que dispara por evento externo.) Após o tratamento da interrupção estar completo, o microprocessador desvia o fluxo de execução para onde estava antes de ter sido interrompido.

### 2.4.3 Software para Sistemas Embarcados

Várias linguagens podem ser usadas para o desenvolvimento de software de sistemas embarcados. Algumas vezes é interessante usar mais de uma. Por exemplo, o projetista pode usar C na maior parte do sistema e usar *Assembly* em regiões críticas, onde o tempo de execução deve ser controlado em detalhes. Segundo uma pesquisa realizada recentemente pela [www.8052.com](http://www.8052.com), 49% dos usuários de processadores compatíveis com o

8051 usam *Assembly*, 33% usam C, 5% usam Basic, 3% usam Pascal e 9% usam outras linguagens. O interesse por linguagens orientadas a objeto vem crescendo dia a dia. Estas linguagens permitem um ciclo de desenvolvimento mais rápido com o uso de melhores métodos de estruturação e modularização, bem como a reutilização de objetos. Entretanto, há ainda alguns obstáculos a serem vencidos para que sejam mais utilizadas em sistemas embarcados. Linguagens orientadas a objetos criam e destroem objetos dinamicamente, dificultando o controle do tamanho necessário de memória e tempo de execução dos programas (BARROS; CAVALCANTE, 2010).

O software de um sistema embarcado é diferente de um software de aplicação, devido, principalmente, à forte interconexão com os dispositivos de hardware periféricos, às restrições impostas pela aplicação na qual ele está inserido (requisitos de tempo real, por exemplo) e ao modo como ele é desenvolvido. Em um computador pessoal o hardware é controlado por um complexo sistema operacional, que gerencia memória, vídeo, teclado e meios de comunicação e o software desenvolvido que precisa apenas se deter a problemas referentes à aplicação. O software de um sistema embarcado, por outro lado, acessa diretamente o hardware e deverá ser escrito de forma a se adaptar aos recursos de hardware disponíveis, visando executar uma tarefa bastante específica. (GOMES, 2010)

O uso dos softwares embarcados na indústria não é mais uma questão restrita aos setores de Pesquisa e Desenvolvimento (P&D), mas parte integrante e essencial das estratégias de diferenciação competitiva dos seus produtos, segundo (TAURION, 2005 apud PEREIRA; CARVALHO; MORAES, 2014). A importância da tecnologia de sistemas embarcados no setor industrial, pode contribuir para a diferenciação dos equipamentos e ainda tornar o equipamento mais competitivo, pois já se observa que uma parcela significativa da diferenciação entre os produtos se baseia na maior oferta de funcionalidades suportadas por esta tecnologia de sistemas embarcados. A revolução digital tem mudado e vai continuar mudando a dinâmica de muitas indústrias. Na indústria de eletroeletrônicos se vê claramente a digitalização substituindo o mundo analógico. Torna-se cada vez mais importante melhorar a qualidade dos softwares embarcados, pois estes se tornam cada vez mais complexos e os produtos físicos se tornam cada vez mais dependentes dos softwares integrados. (PEREIRA; CARVALHO; MORAES, 2014).

## 2.5 Aplicações Web

Segundo (MACHADO, 2019), um sistema baseado na web diz respeito a um software que se usa através da internet e tendo um web browser (Chrome, Firefox, Internet Explorer) como interface. Não é necessário instalar nenhum software extra ou se preocupar com atualizações. Um exemplo são ferramentas de email como Gmail ou Hotmail.

Aplicações Web estão cada dia mais presentes e seu desenvolvimento representa

boa parte da produção de organizações desenvolvedoras de software bem como de mídia em geral. (GONÇALVES et al., 2005)

Segundo (MILETTO; BERTAGNOLLI, 2014), o desenvolvimento de um sistema computacional não é uma atividade trivial, pois envolve analisar e compreender determinado problema. Quando o software é desenvolvido para a plataforma Web, vários aspectos são incorporados de modo que ele possa ser acessado de forma remota e segura por meio de um navegador.

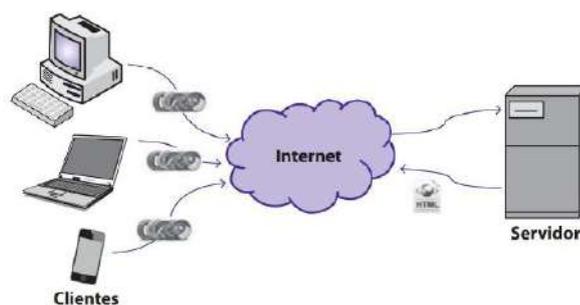
Um sistema de folha de pagamento, por exemplo, pode ser considerado como uma aplicação que é executada em um computador local(não necessitando de conexão em rede). Para isso, todos os dados e recursos necessários para o seu funcionamento já estão disponíveis nesse computador. Assim, ele se concentra no processamento de dados, gerando informações condizentes com os dados de entrada.

Ao considerar um sistema de vendas online, é necessário observar que, em primeiro lugar, ele deve estar sempre disponível, ser executado a partir de um computador remoto (denominado servidor) e ser acessado pelos usuários através de seus computadores(clientes) por meio de um navegador(browser).

Analisando o sistema de folha de pagamento e o de vendas online, é possível observar que há grande diferença entre eles. O primeiro é centralizado, o que significa dizer que tudo é executado e processado em um único computador. Já o segundo sistema é considerado distribuído, ou seja, parte do processamento ocorre em um computador e parte em outro, denominado servidor. Esse tipo de sistema é organizado como um sistema cliente-servidor, como ilustrado na Figura 12.

A partir da Figura 12, é possível deduzir que o usuário, por meio da máquina cliente em que está executando o navegador, solicita informações a uma aplicação que está rodando em um servidor remoto, através da internet. Esse servidor analisa a requisição enviada e responde com dados em forma de páginas Web.

Figura 12 – Arquitetura cliente-servidor



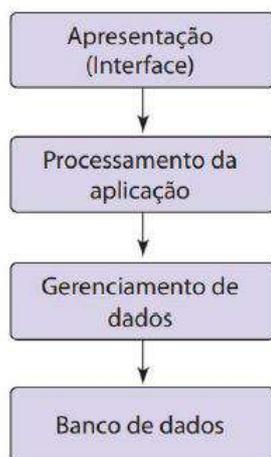
Fonte: (MILETTO; BERTAGNOLLI, 2014)

Tratando-se de uma aplicação Web segundo (MILETTO; BERTAGNOLLI, 2014), é necessário levar em conta vários aspectos para que se possa, de fato, utilizá-la. Esses aspectos incluem o conhecimento dos requisitos exigidos pela aplicação, como:

- Quais tecnologias podem ser utilizadas.
- Como é realizado o processamento do sistema.
- Quais são o perfil do usuário e suas tarefas na interface gráfica da aplicação.
- Como armazenar os dados.

É importante saber, neste momento, que um sistema Web, para ser considerado robusto e consistente, normalmente separa esses aspectos em quatro partes diferentes.

Figura 13 – Organização da Aplicação em Camadas



Fonte: (MILETTO; BERTAGNOLLI, 2014)

A figura 13 ilustra um modelo de separação dessas camadas. Essa separação está relacionada com as boas práticas de desenvolvimento de software, em que um dos objetivos é permitir a implementação de alterações e atualizações em cada uma das partes sem prejudicar as demais.

### 2.5.1 Arquitetura de Aplicações Web

Em uma aplicação Web com arquitetura cliente-servidor, o servidor web é um programa de computador responsável pelo processamento da aplicação, gerenciamento dos dados e do banco de dados, enquanto os clientes são responsáveis pela apresentação dos dados aos usuários da aplicação.

Geralmente os clientes são os navegadores instalados no sistema operacional, como por exemplo, o Chrome, Mozilla e etc. Os cliques dos usuários enquanto eles navegam em páginas webs por meio dos navegadores, são transformados, pelos próprios navegadores em solicitações do Protocolo HTTP. Essas solicitações são processadas pelo servidor web que responde, por meio do protocolo HTTP, documentos HTML que contém os dados solicitados pelos usuários. Ao receber esses documentos HTML, os exibem para os usuários.

### 2.5.1.1 Protocolo HTTP

O HTTP é o protocolo mais utilizado em toda a Internet, parte desse resultado vem do seu modo de operação. O método de requisição/resposta foi bem aceito por todos, pois permite o consumo de recursos e conteúdo sob demanda. Muitos autores também atribuem que pelo fato do HTTP ser executado sobre o TCP se obtém esse resultado, pois com ele nenhuma das aplicações, cliente ou servidor, precisa se preocupar com mensagens perdidas, mensagens duplicadas, mensagens longas ou confirmações(SILVA, 2021).

As paginas webs são constituídas por documentos ou arquivos de vários formatos e essas paginas são construídas com HTML (HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto) que é a linguagem padrão para a programação de paginas web. A pagina construída em linguagem HTML é interpretada por um browser ou navegador, que em HTTP é considerado como cliente. As paginas ou textos HTML também são considerados como documentos pelo protocolo HTTP. Paginas web se encontram em um servidor web, onde podem estar várias outras páginas. O cliente ou usuário acessa essa pagina através de um navegador que interpreta o código HTML e constrói a pagina a ser visualizada por ele. Uma explicação mais simplificada do funcionamento seria a seguinte: o cliente faz uma requisição de uma pagina ao servidor através do protocolo HTTP e o servidor responde ao cliente com a pagina solicitada, como pode ser apresentado na Figura 14 (LIMA; PETRICA, 2010).

Figura 14 – Comunicação Cliente Servidor



Fonte: (KUROSE; ROSS, 2006)

### 2.5.1.2 Métodos HTTP

O Hypertext Transfer Protocol (HTTP) é um protocolo utilizado para envio de textos, mídias e outros dados pela Internet. Ele utiliza o modelo requisição/resposta onde um cliente estabelece comunicação com um servidor, realiza uma solicitação e aguarda até que o servidor retorne sua resposta (SILVA, 2021).

As mensagens de requisição e resposta do protocolo HTTP seguem um padrão, uma requisição é constituída de uma linha informando o método a ser executado em seguida de um cabeçalho que pode ter mais de uma linha e por ultimo o corpo da mensagem caso seja necessário (LIMA; PETRICA, 2010).

Segundo (TANENBAUM, 2003), embora o HTTP tenha sido projetado para utilização na Web, ele foi criado de modo mais geral que o necessário, visando às futuras aplicações orientadas a objetos. Por essa razão, são aceitas operações chamadas métodos, diferentes da simples solicitação de uma página da Web. Cada solicitação consiste em uma ou mais linhas de texto ASCII, sendo a primeira palavra da primeira linha o nome do método solicitado. Para acessar objetos gerais, também podem estar disponíveis métodos adicionais específicos de objetos. Os nomes diferenciam letras maiúsculas de minúsculas; portanto, GET é um método válido, mas get não é.

O método GET solicita ao servidor que envie a página (ou objeto, no caso mais genérico; na prática, apenas um arquivo). A página é codificada em MIME(Extensões Multi função para Mensagens de Internet) de forma adequada. A grande maioria das solicitações a servidores da Web tem a forma de métodos GET. O método HEAD solicita apenas o cabeçalho da mensagem, sem a página propriamente dita. Esse método pode ser usado para se obter a data da última modificação feita na página, para reunir informações destinadas à indexação, ou apenas para testar a validade de um URL. O método POST transporta um URL, porém em vez de substituir os dados existentes, os novos dados são "anexados" a ele, em um sentido mais genérico. Dessa forma, definimos que POST é o Método usado para envio de dados ou formulário HTML ao servidor. Na Tabela 1 é ilustrado alguns métodos para solicitações do protocolo HTTP.

Tabela 1 – Métodos Internos de Solicitações HTTP

Método	Descrição
GET	Solicita a leitura de uma página Web
HEAD	Solicita a leitura de um cabeçalho de página da Web
PUT	Solicita o armazenamento de uma página da Web
POST	Acrescenta a um recurso(por exemplo, uma página da Web)
DELETE	Remove a página da Web
TRACE	Ecoa a solicitação recebida
CONNECT	Reservado para uso futuro
OPTIONS	Consulta certas opções

Fonte: (TANENBAUM, 2003)

Toda solicitação obtém uma resposta que consiste em uma linha de status e, possivelmente, informações adicionais (por exemplo, uma página da Web ou parte dela). A linha de status contém um código de status de três dígitos informando se a solicitação foi atendida e, se não foi, por que não. O primeiro dígito é usado para dividir as respostas em cinco grupos importantes, como mostra a Tabela 2. Os códigos 1xx raramente são usados na prática. Os códigos 2xx significam que a solicitação foi tratada com sucesso, e que o conteúdo (se houver) está sendo retornado. Os códigos 3xx informam ao cliente que ele deve procurar em outro lugar, usando um URL diferente ou seu próprio cache. Os códigos 4xx significam que a solicitação falhou devido a um erro do cliente, como uma solicitação inválida ou uma página inexistente. Finalmente, os erros 5xx significam que o próprio servidor tem um problema, seja causado por um erro em seu código ou por uma sobrecarga temporária (TANENBAUM, 2003).

Tabela 2 – Grupos de Respostas de Código de Status

<b>Código</b>	<b>Significado</b>	<b>Exemplos</b>
1xx	Informação	100 = o servidor concorda em lidar com a solicitação do cliente
2xx	Sucesso	200 = solicitação bem-sucedida; 204 = nenhum conteúdo presente
3xx	Redirecionamento	301 = página movida; 304 = página em cache ainda válida
4xx	Erro do cliente	403 = página proibida; 404 = página não encontrada
5xx	Erro do servidor	500 = erro interno do servidor; 503 = tente novamente mais tarde

Fonte: (TANENBAUM, 2003)

## 2.5.2 Computação em Nuvem

Segundo (SOFTWARE, 2015), a Computação em Nuvem está sendo responsável por uma das maiores revoluções ocorridas nos últimos anos na área de Tecnologia da Informação. Os impactos dessa transformação têm crescido e se acelerado, na medida em que a nuvem oferece cada vez mais serviços, com mais segurança, com maiores recursos e com custos cada vez mais atraentes e competitivos. É uma indústria que definitivamente muda o modo de fazer as coisas na área de TI e que, embora ainda jovem, já proporciona resultados consolidados, fazendo com que sua adoção seja uma opção segura.

O fundamental a ser entendido na Computação em Nuvem é que não é uma revolução tecnológica encerrada em si mesma. Além de mudar o modo como se produzem os serviços de TI das empresas, ela potencializa a mudança do modo como a empresa oferece

seus produtos e serviços, atinge seus novos clientes, acompanha os clientes existentes e pratica o seu marketing no dia a dia, tanto para a captação de novos clientes quanto para a manutenção dos existentes.

Além disso, a Computação em Nuvem permite que organizações de qualquer porte tenham acesso a recursos que antes só estavam disponíveis para grandes empresas, por exigirem elevados investimentos, e agora podem ser pagos sob demanda. Isso muda as condições de competitividade nos mercados, criando oportunidades ímpares de crescimento acelerado sem exigir a antecipação de grandes investimentos na área de infraestrutura de tecnologia. Nesse sentido estamos num novo mundo, onde TI é fundamental para qualquer negócio e, por outro lado, qualquer negócio pode ter sofisticados serviços de TI. Isso muda as relações de força entre as empresas, em qualquer setor.

Segundo (SOFTWARE, 2015) as principais vantagens da utilização da computação em nuvem são:

- **Custo:** Como você só paga pelo que realmente usa, o custo de uma infraestrutura na nuvem é na maioria das vezes menor do que o custo de uma infraestrutura convencional, em que você compra as máquinas para a sua empresa. O sistema não precisa funcionar de madrugada? Basta programar o desligamento das máquinas e deixar de pagar por elas nesse período. O volume de acessos está baixo? Troque a máquina que está usando por uma menor e pague menos. Simples assim. No jargão da Computação em Nuvem, esse modelo dinâmico de alocação e liberação de recursos é conhecido pelo termo pagamento pelo uso (“pay-per-use”), uma vez que só se paga pelo que é efetivamente utilizado.
- **Agilidade:** Provedores de infraestrutura convencionais demoram dias, ou até semanas, para entregar novas máquinas, discos ou qualquer outro recurso para seus clientes. Observe que isso acontece independentemente de a infraestrutura ser própria, instalada em um data center local ao negócio, ou terceirizada, rodando em um provedor de hospedagem. Na Computação em Nuvem, o processo de alocação (ou diminuição) de novos recursos demora apenas alguns minutos. Basta escolher a máquina desejada, seu sistema operacional e parametrizar alguns dados: pronto, a nova máquina está no ar. Sem qualquer investimento.
- **Flexibilidade:** Não precisa mais de uma máquina? É só desligar. Precisa de uma máquina maior? É só aumentar o tamanho. Mais disco? É só definir um novo tamanho para seu disco. Mais memória dinâmica? É só definir quanto é preciso. A qualquer momento. Instantaneamente.
- **Alta Disponibilidade:** Para que um ambiente computacional convencional garanta alta disponibilidade é necessário duplicá-lo, criando um ambiente de contingência

que é acionado em caso de falha do principal. Claro, essa duplicação de recursos implica em desperdícios, uma vez que o ambiente de contingência fica ocioso a maior parte do tempo em condições normais. Mesmo não estando em operação, a infraestrutura de contingência demandou investimentos e está gerando despesas de depreciação, ocupando espaço, requerendo segurança e supervisão – ou seja, há geração de custos mesmo sem a infraestrutura estar no ar. Somente quando algo falha é que a infraestrutura adicional passa a ser utilizada. Com a Computação em Nuvem, a garantia de alta disponibilidade passou a ser muito mais acessível: o ambiente de contingência pode ser previamente configurado, mas mantido em formato reduzido e de baixo custo. Quando necessário, esse ambiente pode ser aumentado em poucos minutos, assumindo uma configuração semelhante ao do ambiente principal. A alta disponibilidade na nuvem tem custos expressivamente menores que quaisquer outras soluções, dentro da empresa ou em provedores onde se paga o aluguel de máquinas.

### 2.5.2.1 Desenvolvimento de Aplicações com Computação em Nuvem

Segundo (MUNHOZ, 2012), a computação em nuvem traz novamente a tona uma ideia de centralização onde vários *datacenters* são criados e mantidos por grandes empresas e corporações por todo o mundo. Contudo Neto e Freitas 2011, destacam que os últimos 40 anos mudaram um cenário centralizado com mainframes, onde todas as aplicações e dados eram locais e distribuídas por meios de redes internas quando aplicativos *desktops* utilizavam a mesma base de dados. Após as aplicações, passaram a ser utilizadas via *browser*, de uma forma também interna até chegarmos ao nosso cenário atual.

Dentre as varias formas de computação em Nuvem e seus *datacenters*, alguns são destaques por suas popularidades. São estas as mais populares: Google, Amazon, e Microsoft. Onde seus hardwares “feitos sob - demanda”, central de energia e resfriamento de alta tecnologia, juntamente a softwares de controle, provem aos clientes formas rápidas de adicionar ou subtrair maquinas do seu meio computacional.

### **Amazon Web Services(AWS)**

Segundo (VERAS, 2013), a Amazon Web Services(AWS) é a principal oferta de arquitetura do tipo computação em nuvem (*cloud computing*) da atualidade. Esta arquitetura permite às empresas o acesso a serviços de infraestrutura na forma *on demand*. A ideia é que esta forma de aquisição de serviços de infraestrutura de TI, conhecida como serviços de infraestrutura de nuvem, reduza custos, minimize os riscos do negócio e maximize as oportunidades. Do ponto de vista teórico, o serviço tradicional de *datacenter*, quando comparado ao serviço de nuvem, é pouco eficaz, considerando que o uso e a capacidade dos recursos não podem ser otimizados como no modelo baseado em nuvem.

As empresas podem, ao utilizar a arquitetura AWS:

- Continuar a utilizar aplicações existentes baseadas em sistemas operacionais, banco de dados, arquitetura de software e linguagens de programação já comuns. Estas aplicações devem ser movidas para a nuvem AWS.
- Desenvolver novas aplicações que desde o início aproveitam os recursos da nuvem AWS e que podem até misturar-se com arquiteturas "legadas" para servir a diferentes modelos de negócio.

A proposta da Amazon para a AWS é fornecer serviços baseados em nuvem com flexibilidade, efetividade, escalabilidade, elasticidade e segurança. O (IDC, 2012 apud VERAS, 2013) recentemente relatou a experiência de onze empresas de médio e pequeno porte com a utilização da arquitetura AWS. Os resultados obtidos por elas apontam para uma redução de custos, aumento da produtividade em relação à de manter a infraestrutura de TI interna, redução do *downtime* e ganhos expressivos de produtividade.

### **Microsoft Azure**

A plataforma de serviços Azure da Microsoft é um grupo de tecnologia em nuvem que pode ser usado tanto para aplicativos em nuvem quanto para aplicativos locais. Seus aplicativos podem ser executados em uma variedade de sistemas. De acordo com Chapell 2008, cada componente da tecnologia Azure tem sua própria função, ou seja: Windows Azure, plataforma baseada em ambiente Windows para executar aplicativos e armazenar dados remotamente nos servidores da Microsoft. Microsoft .Net Service, responsável por disponibilizar aplicativos baseados em nuvens e locais. Microsoft SQL Services, fornece serviços de dados em nuvens baseado em SQL Server. Live Service, através do live framework, fornece acesso aos dados a partir de aplicativos live da Microsoft e outros. Esta é uma plataforma simples de se entender, pois seu objetivo é executar aplicativos Windows e armazenar seus dados nas nuvens. Seus maiores benefícios são de ao invés de comprar diferentes máquinas físicas e instalar seus aplicativos, uma organização na nuvem pode fazer isto pra ela. E elas pagarão apenas pelos aplicativos e dados armazenados ao invés de servidores (MUNHOZ, 2012).

### **Google Application Engine**

O Google foi uma das empresas pioneiras a fornecerem a plataforma computação em nuvens para desenvolvimento de aplicativos. Todavia (JUNIOR, 2011 apud MUNHOZ, 2012), demonstra que o Google Application Engine ou Google GAE é um serviço que diferente do conceito original, onde todo o ambiente responsável pelo ciclo de desenvolvimento e publicação do software é local, neste caso se obtém suas ferramentas disponibilizadas via web. Por meio de um ambiente *desktops* completo através de plugins para o Eclipse IDE chamado Googlipse para criar as aplicações (MUNHOZ, 2012).

Segundo (SANDERSON, 2013), o Google App Engine é um serviço de hospedagem de aplicativos da Web. Por "aplicativo da Web", queremos dizer um aplicativo ou serviço acessado pela Web, geralmente com um navegador da Web com carrinhos de compras, sites de redes sociais, jogos multiplayer, aplicativos móveis, aplicativos de pesquisa, gerenciamento de projetos, colaboração, publicação e todas as outras coisas que estamos descobrindo são bons usos para a Web. App Engine pode servir conteúdo tradicional de site também, como documentos e imagens, mas o ambiente é especialmente projetado para aplicações dinâmicas em tempo real.

Em particular, o Google App Engine foi projetado para hospedar aplicativos com muitos usuários simultâneos. Quando um aplicativo pode atender a muitos usuários simultâneos sem degradar desempenho, nós dizemos que ele é escalável. Aplicativos escritos para o App Engine escalam automaticamente. À medida que mais pessoas usam o aplicativo, o App Engine aloca mais recursos para o aplicativo e gerencia o uso desses recursos. O próprio aplicativo não precisa saber qualquer coisa sobre os recursos que está usando.

Ao contrário da hospedagem web tradicional ou servidores autogerenciados, com o Google App Engine, você paga apenas pelos recursos que utilizar. Estes recursos são medidos até gigabytes. Os recursos faturados incluem uso de CPU, armazenamento por mês, largura de banda de entrada e saída, e vários recursos específicos dos serviços do App Engine. Para ajudá-lo a começar, cada desenvolvedor recebe uma certa quantidade de recursos de graça, o suficiente para pequenas aplicações com pouco tráfego.

## 2.6 Banco de Dados

Segundo (RAMAKRISHNAN; GEHRKE, 2011), um banco de dados é uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas. Por exemplo, um banco de dados de uma universidade poderia conter informações sobre: Entidades como alunos, professores, cursos e turmas. E ainda, relacionamentos entre as entidades, como a matrícula dos alunos nos cursos, cursos ministrados pelos professores e o uso de salas por cursos.

Um sistema de gerenciamento de banco de dados, ou SGBD, é um software projetado para auxiliar a manutenção e utilização de vastos conjuntos de dados. A necessidade de tais sistemas, assim como o seu uso, tem crescido rapidamente. A alternativa para não se usar um SGBD é armazenar os dados em arquivos e escrever código específico do aplicativo para gerenciá-los.

### 2.6.1 Banco de Dados Relacionais

Segundo (ALVES, 2015), a maioria dos sistemas de gerenciamento de bancos de dados atualmente em uso se enquadra no tipo relacional. Um banco de dados relacional

se caracteriza pelo fato de organizar os dados em tabelas(ou relações), formadas por linhas e colunas. Assim, essas tabelas são similares a conjuntos de elementos ou objetos, uma vez que relacionam as informações referentes a um mesmo assunto de modo organizado.

Da mesma forma que na matemática, podem ser efetuadas operações entre dois ou mais conjuntos, como, por exemplo, obter os elementos que são comuns a ambos os conjuntos (tabelas/relações) num banco de dados relacional, também é possível executar certas operações com essas tabelas, como ligar duas ou mais por meio de campos comuns em ambas. Quando uma operação de consulta é executada, o resultado é um conjunto de registros que pode ser tratado como uma tabela virtual(que só existe enquanto a consulta está ativa). Isso significa que não há comandos para efetuar uma navegação pelos registros, do tipo MOVE ou SKIP, que permitem posicionar o ponteiro de registro no próximo registro da tabela.

Segundo (RAMAKRISHNAN; GEHRKE, 2011), atualmente, o modelo relacional é de longe o modelo de dados dominante e é a base dos SGBDs líderes do mercado, incluindo a família DB2 da IBM, o Informix, o Oracle, o Sybase, o Access e o SQLServer, da Microsoft, o FoxBASE e o Paradox. Os sistemas de banco de dados relacional são onipresentes no mercado e representa um negócio de bilhões de dólares.

O modelo relacional com sua representação tabular simples permite que até usuários iniciantes entendam o conteúdo de um banco de dados e possibilita o uso de linguagens de alto nível simples para consultar os dados. As principais vantagens do modelo relacional em relação aos modelos de dados mais antigos são sua representação de dados simples e a facilidade com que consultas complexas podem ser expressas.

### 2.6.1.1 Entidades, Atributos e Conjuntos de Entidades

Segundo (RAMAKRISHNAN; GEHRKE, 2011), uma entidade é um objeto do mundo real distinguível de outros objetos. Exemplos incluem: o brinquedo Dragonzord Verde, o departamento de brinquedos, o gerente do departamento de brinquedos, o endereço residencial do gerente do departamento de brinquedos. Normalmente, é útil identificar uma coleção de entidades semelhantes. Tal coleção é chamada de conjunto de entidades. Observe que os conjuntos de entidades não precisam ser disjuntos: o conjunto de funcionários do departamento de brinquedos e o conjunto de funcionários do departamento de utensílios podem ambos conter o funcionário John Doe (que pode trabalhar em ambos os departamentos). Poderíamos também definir um conjunto de entidades chamado Funcionários que contém os conjuntos de funcionários dos dois departamentos: brinquedos e de utensílios. Uma entidade é descrita utilizando-se um conjunto de atributos. Todas as entidades em um determinado conjunto de entidades têm os mesmos atributos; isto é o que queremos dizer com semelhantes. Nossa escolha de atributos reflete o nível de detalhes com os quais desejamos representar as informações sobre as entidades. Por exemplo,

o conjunto de entidade Funcionários poderia usar nome, código de pessoa física(CPF) e vaga de estacionamento como atributos.

### 2.6.1.2 Cardinalidade

Segundo (ARAÚJO, 2021), o Modelo Entidade-Relacionamento é um modelo de alto nível, independente do SGBD (Sistemas Gerenciadores de Bancos de Dados), que representa o problema a ser modelado. A notação que será utilizada para a representação deste modelo é o DER (Diagrama Entidade-Relacionamento), exemplificado na Figura 15, onde os retângulos representam as entidades e os losangos representam os relacionamentos entre estas entidades. Entidades ainda são descritas através de atributos e devem possuir uma chave primária (ou Primary Key - atributo ou conjunto de atributos que identificam unicamente uma instância em uma entidade, e que não podem receber um valor nulo). A Figura 15 representa que uma instância da Entidade A está associada a zero(opcional) ou mais instâncias da Entidade B. Por outro lado, uma instância da Entidade B está associada a uma (obrigatoriedade), e somente uma, instância da Entidade A. A este par de elementos chama-se cardinalidade, onde o primeiro elemento indica a participação (opcional ou obrigatório) do relacionamento, enquanto o segundo representa o grau do relacionamento (um ou muitos).

Figura 15 – Notação do Diagrama Entidade-Relacionamento



Fonte: (ARAÚJO, 2021)

Relacionamentos ainda podem ter atributos, quando algum dado precisa ser associado à ligação das duas instâncias envolvidas. Relacionamentos são descritos através da cardinalidade, que indica como as instâncias das entidades se relacionam. Os tipos utilizados na modelagem são:

- Um-para-um (1:1): uma instância em “A” está associada com no máximo uma instância em “B”, e uma instância em “B” está associada com no máximo uma instância em “A”;
- Um-para-muitos (1:n): uma instância em “A” está associada a qualquer número de instâncias em “B”, e uma instância em “B”, todavia, pode estar associado a no máximo uma instância em “A”;

- Muitos-para-muitos (n:n): uma instância em “A” está associada a qualquer número de instâncias em “B” e vice-versa. Alguns autores preferem chamar esta cardinalidade de m:n, por considerar que podem representar valores diferentes.

### 2.6.1.3 Firebase

Um exemplo de banco de dados em tempo real é o Firebase. Segundo (FIREBASE, 2011), o Realtime Database é um banco de dados original do Firebase hospedado na nuvem. Ele é a solução eficiente e de baixa latência para aplicativos móveis que exigem estados sincronizados entre clientes em tempo real. Os dados são armazenados como JSON e sincronizados em tempo real para cada cliente conectado. Quando você cria aplicativos multiplataforma com plataformas Apple, Android e SDKs JavaScript, todos os seus clientes compartilham uma instância do Realtime Database e recebem automaticamente atualizações com os dados mais recentes. Na Figura 16 é apresentado a página do Firebase após a conta estar logada.

Figura 16 – Projeto Criado no Firebase - Google Cloud



Fonte: (FIREBASE, 2011)

O Firebase Realtime Database permite criar aplicativos avançados e colaborativos, permitindo acesso seguro ao banco de dados diretamente do código do lado do cliente. Os dados são mantidos localmente e, mesmo offline, os eventos em tempo real continuam a ser acionados, proporcionando ao usuário final uma experiência responsiva. Quando o dispositivo recupera a conexão, o Realtime Database sincroniza as alterações de dados locais com as atualizações remotas que ocorreram enquanto o cliente estava offline, mesclando quaisquer conflitos automaticamente. O Realtime Database fornece uma linguagem de regras flexível e baseada em expressões, chamada Firebase Realtime Database Security

Rules, para definir como seus dados devem ser estruturados e quando os dados podem ser lidos ou gravados.

Quando integrado ao Firebase Authentication, os desenvolvedores podem definir quem tem acesso a quais dados e como eles podem acessá-los. O Realtime Database é um banco de dados NoSQL e, como tal, possui otimizações e funcionalidades diferentes em comparação com um banco de dados relacional. A API do Realtime Database foi projetada para permitir apenas operações que podem ser executadas rapidamente. Isso permite que você crie uma ótima experiência em tempo real que pode atender a milhões de usuários sem comprometer a capacidade de resposta. Por isso, é importante pensar em como os usuários precisam acessar seus dados e estruturá-los adequadamente.

## 3 METODOLOGIA

Nesta Capítulo será apresentado todo o desenvolvimento prático do sistema de monitoramento de consumo residencial.

### 3.1 Montagem do Dispositivo de Medição

#### 3.1.1 Dispositivos, Ferramentas e Linguagens Utilizadas

##### 3.1.1.1 O Microcontrolador ESP32

Após a obtenção do sinal do sensor de medição de vazão, é necessário uma MCU com capacidade de transmitir os dados processados via Wi-Fi.

O microcontrolador ESP32, tem sido cada vez mais utilizado para fins acadêmicos e industriais, especialmente com aplicações que necessitam de um acompanhamento em tempo real, ainda a uma combinação de baixo custo e alto desempenho quando comparados a outros Microcontroladores disponíveis no mercado. (COSTA, 2020)

No desenvolvimento deste protótipo foi utilizado o modelo DOIT Esp32 DevKit v1, como apresentado na Figura 17. Este modelo é uma das placas de desenvolvimento criadas pelo DOIT para avaliar o módulo ESP-WROOM-32. É baseado no microcontrolador ESP32 que possui suporte Wi-Fi, Bluetooth, Ethernet e Low Power, tudo em um único chip (ZERYNTH, 2015). Seu chip principal é o ESP32 com suporte ao padrão de comunicação IEEE 802.11 e comunicação Bluetooth v4.2 BR/EDR e Bluetooth Low Energy (BLE).

Figura 17 – Modelo DOIT Esp32 DevKit v1



Fonte: (ZERYNTH, 2015)

Na Tabela 3 é apresentado as especificações da placa DOIT ESP32 DEVKIT V1.

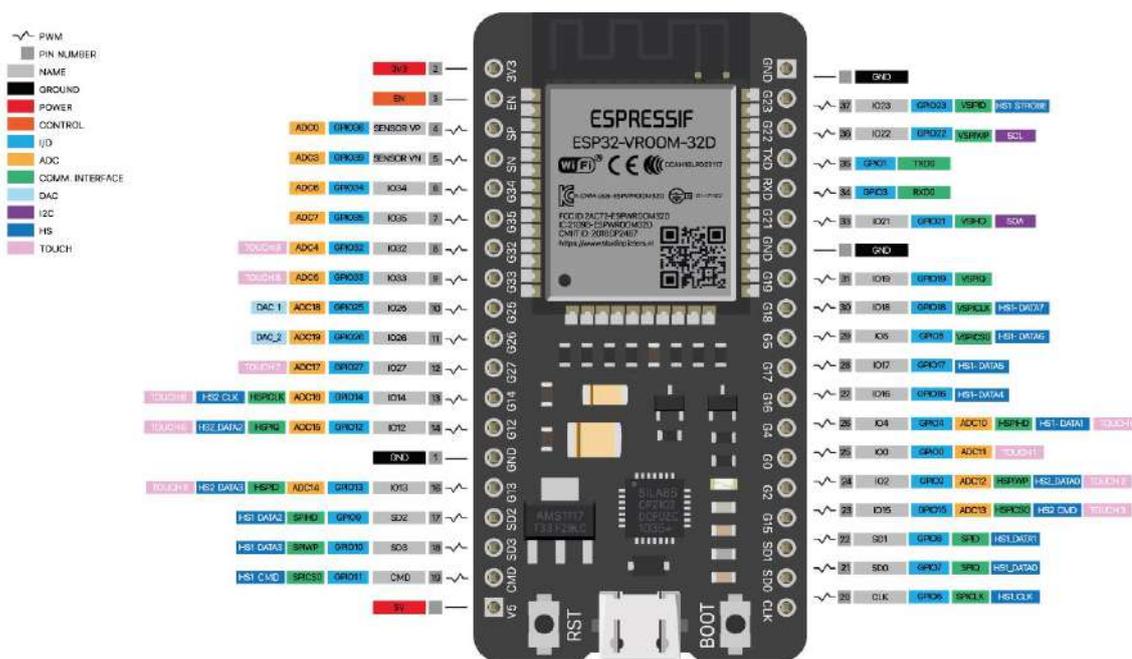
Tabela 3 – Especificações da DOIT ESP32 DEVKIT V1

Nome Comercial	DOIT ESP32 DEVKIT V1
Microcontrolador	Xtensa® Dual-Core 32-bit LX6
Tensão operacional	3,3V
Tensão de entrada	5-12V
Pinos de E/S digitais (DIO)	32
Pinos de entrada analógica (ADC)	16
Pinos de saídas analógicas (DAC)	2
Memória Flash	4 MB
RAM	520 KB
Velocidade do clock	240MHz
Wi-Fi	IEEE 802.11 b/g/n/e/i

Fonte: (ZERYNTH, 2015)

Na Figura 18 é representado o diagrama de pinagem da placa de desenvolvimento DOIT Esp32 DevKit v1 38 pinos.

Figura 18 – Diagrama de pinagem da placa de desenvolvimento ESP32 38 pinos



Fonte: (APRENDIZAGEM, 2020)

### 3.1.1.2 Sensor de Fluxo de Água YF-S201

Para que ocorra a medição do volume de água, o sensor deve ficar em série com o fluxo de água, de maneira que o fluido atravessasse sua parte cilíndrica. Ao fazer isto, o fluido provoca um movimento de rotação em uma hélice interna ao sensor. Nesta hélice, está acoplado um ímã e quando a hélice faz a rotação devido a passagem da água, o sensor de efeito hall identifica um campo magnético e produz uma quantidade de pulsos de tensão

proporcional ao volume de água que está atravessando a parte cilíndrica do sensor. O sensor YF-S201 é ilustrado na Figura 19. Ele possui três fios: vermelho (alimentação 5-24VDC), preto (terra) e amarelo (saída de pulso de efeito Hall). Segundo o fabricante do sensor, o sensor pode apresentar um erro de até 10%. Também é estabelecido pelo fabricante que a vazão mínima para o funcionamento correto do sensor é de 1L/minuto e a vazão máxima é de 30L/minutos.

Figura 19 – Sensor de Fluxo de Água de Efeito Hall - YF-S201



Fonte: (DUTRA, 2019)

Na Tabela 4 é apresentado as especificações do sensor de vazão YF-S201.

Tabela 4 – Especificações do Sensor de Vazão YF-S201

Nome Comercial	Sensor de Vazão YF-S201
Modelo	YF-S201
Tipo do Sensor	Efeito Hall
Tensão de trabalho	5 a 24V DC (tensão de trabalho min testada 4,5V)
Consumo máximo de corrente	15mA a 5V
Taxa de fluxo de trabalho	1 a 30 litros/minuto
Pulsos por litro	450
Comprimento do cabo	15cm

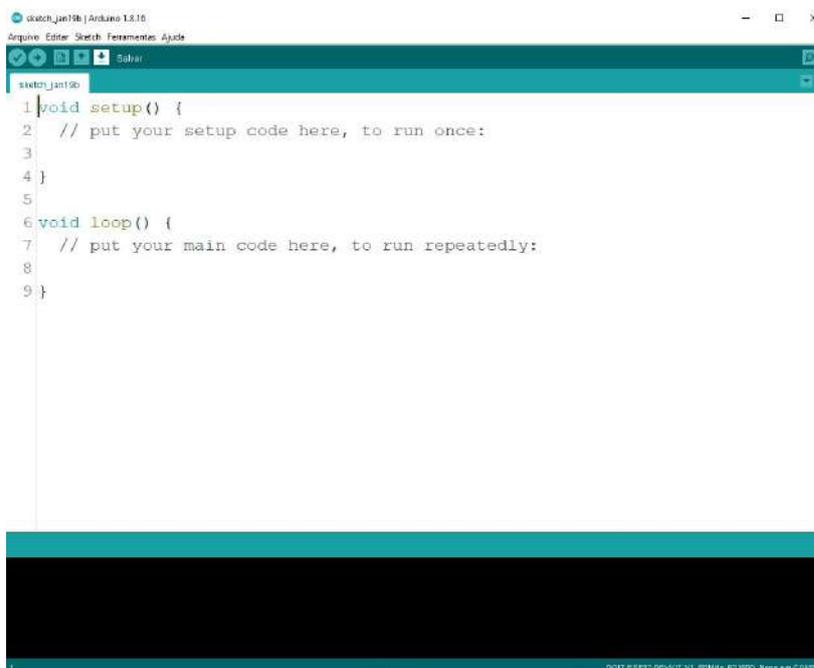
### 3.1.1.3 Arduíno IDE

O software Arduíno IDE como apresentado na Figura 20, foi utilizado na escrita do código para a medição realizada pelo dispositivo de medição.

Segundo (FEZARI; DAHOUD, 2018), IDE significa “Integrated Development Environment”. Arduíno IDE é um software desenvolvido pela empresa Arduíno e distribuído através da plataforma Arduino.cc. Ele é usado principalmente para editar, compilar e carregar o código nas placas da família Arduíno. O Arduíno IDE também é compatível com alguns outros microcontroladores que não pertencem a família Arduíno. Essa compatibilidade é possível pois os fabricantes desses microcontroladores desenvolveram

as ferramentas de software necessárias para tornar a compatibilidade possível. Um exemplo de microcontrolador que se encaixa nessa situação é a ESP32, um microcontrolador no qual foi utilizado para a construção do dispositivo de medição de consumo de água apresentado neste trabalho.

Figura 20 – Ferramenta Utilizada para Construção do Software de Medição

A screenshot of the Arduino IDE (version 1.8.16) interface. The window title is "sketch\_jun19@ Arduino 1.8.16". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The toolbar contains icons for "Salvar", "Executar", "Monitor", "Serial", "Verificar", "Enviar", and "Ajuda". The main text area shows the following code:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

The status bar at the bottom indicates "PORT: ESP32 DEVKIT V1, COM16, 115200, Nove em C/USP".

Fonte: Autoria Própria

O Arduíno IDE é de código aberto e está disponível para instalar e começar a compilar o código em qualquer lugar, tornando a compilação de código facilmente utilizada até mesmo por iniciantes. Está disponível para sistemas operacionais como MAC, Windows, Linux e é executado na plataforma Java que vem com funções e comandos embutidos que desempenham um papel vital para depurar, editar e compilar o código no ambiente.

#### 3.1.1.4 Linguagem C++

C++ é uma linguagem de programação criada por Bjarne Stroustrup no início da década de 1980. Com base em C e em Simula, Hubbard(2003) acrescenta que é atualmente uma das linguagens mais populares para programação orientada a objetos. Foi padronizada em 1998 pelo AmericanNationalStandardsInstitute (ANSI) e pela InternationalStandardsOrganization(ISO). Possui o mecanismo classe/objeto, permite herança simples e herança múltipla e sobrecarga de operadores e funções (JUNIOR; VIRTUOSO; MARTINS, 2012).

Segundo (MANSSOUR, 1999), Um programa em C++ consiste em um ou mais arquivos. Um arquivo é uma porção de texto contendo código fonte em C++ e comandos do

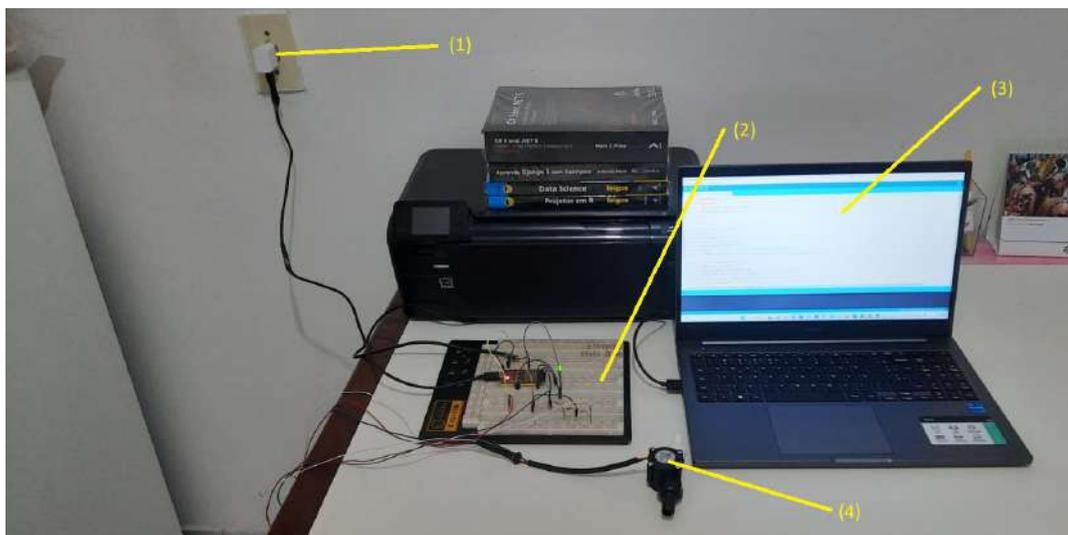
pré-processador. Em outras palavras, ele corresponde a um arquivo fonte em um sistema tradicional. Um arquivo é conceitualmente traduzido em diversas fases, onde a primeira corresponde ao pré-processamento, que realiza a inclusão do arquivo e a substituição de macros, e resulta em uma seqüência de símbolos. A extensão dos nomes dos arquivos fonte em C++, normalmente, é ".cpp". Entretanto, algumas implementações de C++ utilizam ".c", ".C", ".cp" ou ".cxx".

Existem cinco espécies de símbolos em C++: identificadores, palavras-chave, literais, operadores e outros separadores. Brancos, tabulações horizontal e vertical, novas linhas, avanços de formulário e comentários (coletivamente, "espaço em branco"), são ignorados, exceto pelo fato de servirem para separar símbolos. Algum espaço em branco é necessário para separar identificadores, palavras-chave e constantes que de outro modo ficariam adjacentes.

### 3.1.2 Montagem do Dispositivo de Medição de Consumo de Água

O protótipo do dispositivo de medição do consumo de água é apresentado na Figura 21. Foi utilizado, um carregador da marca SAMSUNG como fonte de tensão de 9,0V-1,57A ou 5,0V-2,0A (1) para alimentação do microcontrolador ASP32 utilizada no projeto, um protoboard (2) para a montagem do circuito, um notebook (3) para compilação, visualização do código elaborado e o monitoramento das medidas de consumo de água e número de pulsos apresentados pelo sensor. E por último, um sensor de vazão, modelo YF-S201 (4) para medição do consumo de água, de acordo com o número de pulsos estimados pelo sensor.

Figura 21 – Protótipo Utilizado para a Medição

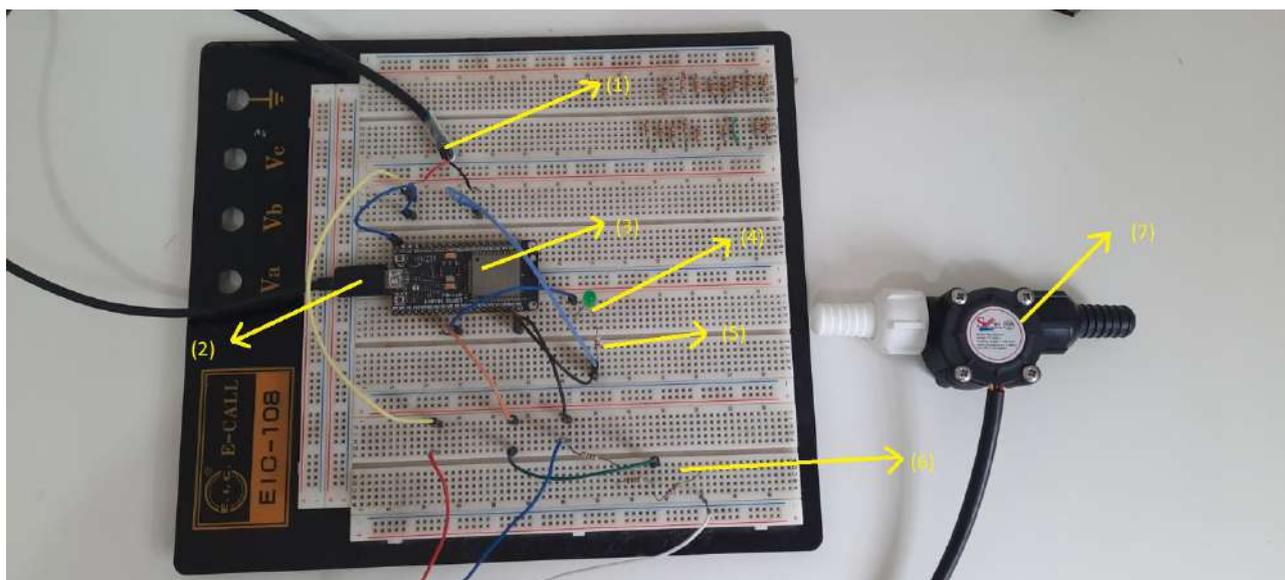


Fonte: Autoria Própria

Na Figura 22 é possível visualizar como foi organizado a montagem do circuito

e as ligações através de jumpers entre a fonte de tensão (1), o microcontrolador ESP32 (3), o LED (4) que é utilizado com a finalidade de informar se a conexão entre o ESP32 e o roteador é estabelecida ou não, o resistor de  $470\ \Omega$  (5), os três resistores de  $1\ \text{M}\Omega$  (6), e o sensor de vazão (7). Além disso, é destacado o cabo USB (2) responsável pela comunicação entre a ASP32 e o notebook.

Figura 22 – Componentes Utilizados para Montagem do Circuitos e Testes



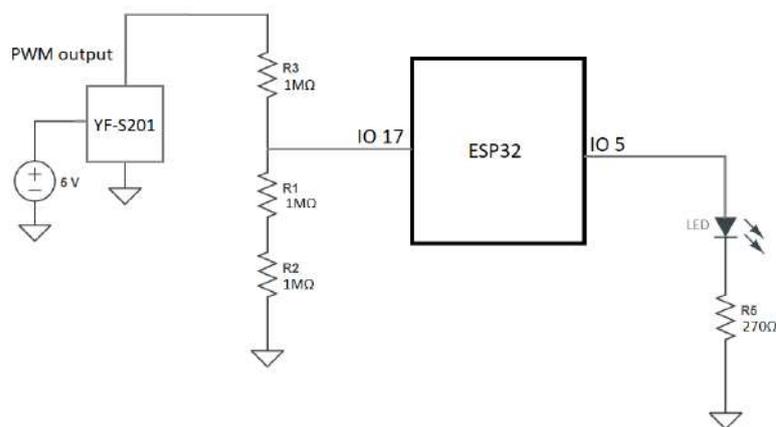
Fonte: Autoria Própria

Na Figura 23 é possível visualizar o esquemático do circuito eletrônico do protótipo. O sensor de vazão utilizado, YF-S201, é alimentado com uma tensão de 5V. Quando um fluxo de água passa pelo sensor, ele dispara pulsos de tensão com amplitude de 5V, que serão contados pela ESP32, devido ao seu sensor de efeito hall magnético integrado, como explicado na Seção 2.3.1.1. Foi colocado um divisor de tensão entre a saída do sensor de vazão e o pino de entrada IO17 da ESP32. O divisor de tensão, composto por 3 resistores de  $1\ \text{M}\Omega$ , reduz a amplitude dos pulsos de tensão de 5 volts para 3.3 volts que é a tensão operacional do pino IO17 da ESP32.

No pino IO5 é ligado um LED que possui a função de indicar se a conexão entre a ESP32 e o roteador WIFI está estabelecida. Se a luz do LED estiver ligada, a conexão da ESP32 com a internet, por meio da rede WIFI, está estabelecida. Caso o LED não esteja ligada, a conexão não foi estabelecida.

Na Tabela 5 são apresentados os componentes utilizados para a construção do circuito do protótipo.

Figura 23 – Esquemático do Circuito Utilizado no Protótipo



Fonte: Autoria Própria

Tabela 5 – Componentes Utilizados no Circuito do Protótipo

Componentes	Quantidade
Fonte de tensão	1
ESP32	1
LED	1
Resistor 470 $\Omega$	1
Resistor 1 $M\Omega$	3
Sensor de Vazão	1
Jumpers	8
Protoboard	1

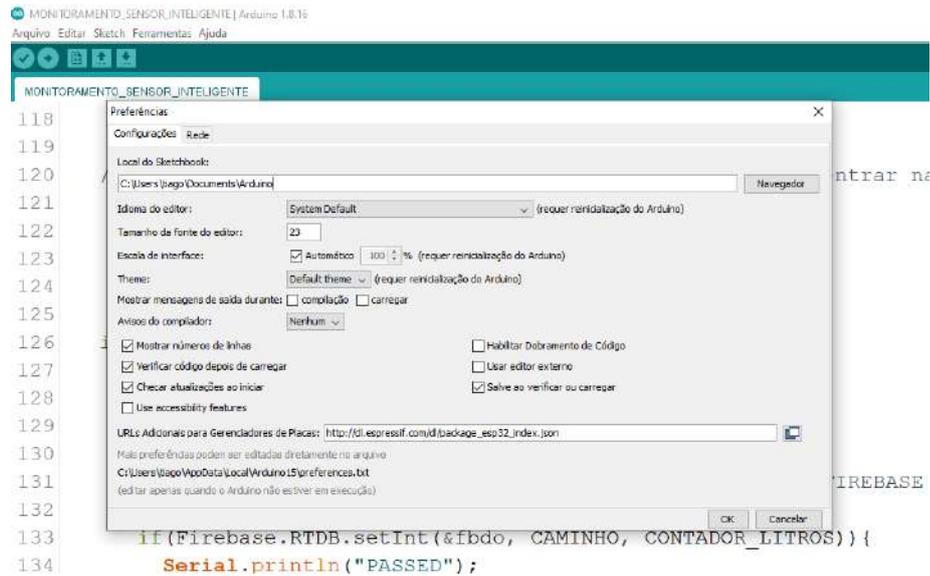
Fonte: Autoria Própria

### 3.1.3 Firmware do Dispositivo de Medição de Volume

#### 3.1.3.1 Preparando o Ambiente de Desenvolvimento

Antes de iniciar a elaboração do código, foi necessário atualizar a Arduino IDE, pois versões antigas não permitem o uso do ESP32. Com o Arduino IDE aberto, para instalação da placa ESP32, foi selecionado a opção arquivo e depois preferências, e na aba de configurações, em preferências, na opção, URLs adicionais para Gerenciadores de Placa, foi adicionado o link <[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)> como apresentado na Figura 24.

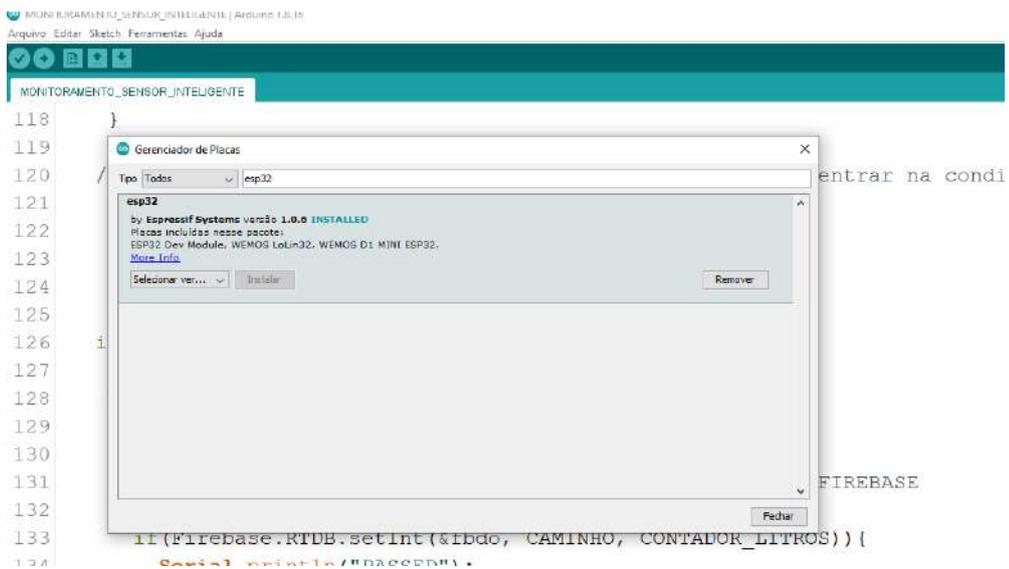
Figura 24 – URLs Adicionais para Gerenciadores de Placas



Fonte: Autoria Própria

Continuando a configuração e instalação dos drives para uso, clicando em Ferramentas, Placas, e depois Gerenciador de Placas, foi digitado na barra de pesquisa esp32, para então ser instalado o pacote ESP32, da Espressif Systems na versão 1.0.6, como ilustrado na Figura 25.

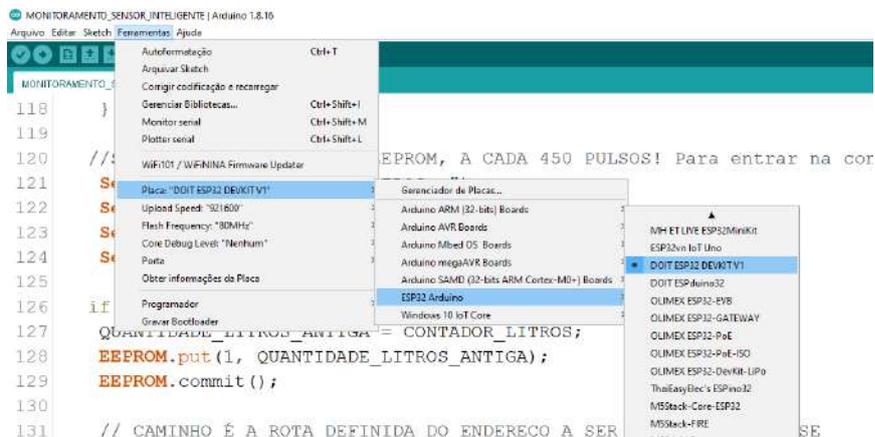
Figura 25 – Pacote ESP32 para Instalação



Fonte: Autoria Própria

Para verificar se a instalação da Placa ESP32 DEVKIT V1 no Arduino IDE havia ocorrido de forma correta, foi selecionado a opção ferramentas e depois a opção placa, onde foi constatado a placa DOIT ESP32 DEVKIT V1, como apresentado na Figura 26.

Figura 26 – Verificação da Placa DOIT ESP32 DEVKIT V1 no Arduino IDE



Fonte: Autoria Própria

### 3.1.3.2 Algoritmo do firmware do dispositivo de medição de consumo

Sabe-se, de acordo com a Tabela 4, que o sensor de vazão YF-S201 emite pulsos devido a quantidade de água que passa através dele. Então, o algoritmo de medição de vazão, consiste em contar a quantidade de pulsos que o sensor emite e, quando a contagem de número de pulsos for igual a 450 pulsos, sabe-se que 1 litro de água foi consumido.

Para se contar a quantidade de pulsos, foi configurado uma interrupção externa no pino IO17 da ESP32, pino este, que está conectado a saída do sensor YF-S201, de acordo com o circuito montado na Figura 23. Essa interrupção é sensível a borda de subida, e sempre que ocorre um pulso, esse pulso é detectado pela ESP32 e assim uma função de tratamento de interrupção é chamada. Dentro dessa função de interrupção ocorre a contagem deste pulso incrementando um contador que armazena o número de pulsos. Uma vez que esse contador, alcance o número de 450 pulsos, sabe-se que 1 litro de água passou pelo sensor de vazão YF-S201. Assim, ao chegar em 450 pulsos, a função de tratamento de interrupção incrementa um segundo contador, este referente ao total de litros consumidos. A cada incremento desse segundo contador de litros, esse novo valor também é persistido na memória EEPROM da ESP32, para que o valor não seja perdido em caso de uma falta de energia ou desligamento da ESP32.

O algoritmo do firmware do dispositivo de medição define que para questões de falta de energia e desligamento da esp32, seja verificado o primeiro endereço de memória EEPROM da esp32 ao ser iniciada novamente, e lá contendo um valor numérico estabelecido, isso indicará que já foi salvo um valor de consumo em litros nas outras posições da memória EEPROM da esp32, e esse valor em litros será coletado, e a execução do dispositivo de medição de consumo continuará a contagem de litros de onde havia parado. Caso, seja a primeira utilização do dispositivo de medição de consumo, esse valor numérico é salvo no primeiro endereço da esp32 e nos próximos oito endereços é salvo o valor zero.

Essa pequena lógica, também funciona para prevenir gravações desnecessárias, vendo que a memória EEPROM das esp32 possui um limite de gravações.

Dessa forma, a cada novo litro consumido, o consumo total é enviado para o banco de dados do Firebase, por meio de uma conexão WiFi. Esta conexão com a internet, por meio da interface de Rede WiFi é estabelecida assim que o dispositivo de consumo de água é ligado. No firmware do dispositivo de medição são estabelecidas condições que verificam constantemente se a conexão de WiFi foi interrompida. Para alertar se a conexão WiFi foi estabelecida ou interrompida, existe um LED no dispositivo de medição de consumo, onde se o LED estiver aceso, a conexão foi estabelecida corretamente, mas se o LED estiver apagado, a conexão foi perdida.

O banco de dados hospedado no Firebase armazena o consumo de água enviado por cada dispositivo de medição de consumo. Para se criar um banco de dados no Firebase, foi necessário a criação de um projeto de banco de dados em tempo real, por meio da plataforma web disponibilizada pela empresa Google, proprietária do Firebase. A plataforma Firebase foi escolhida para ser utilizada neste projeto devido a sua funcionalidade de estabelecer uma comunicação assíncrona entre os dispositivos de medição de consumo de água e a aplicação Web, facilitando a utilização da aplicação web, já que para a aplicação web será possível consultar os dados de consumo de água sobre demanda.

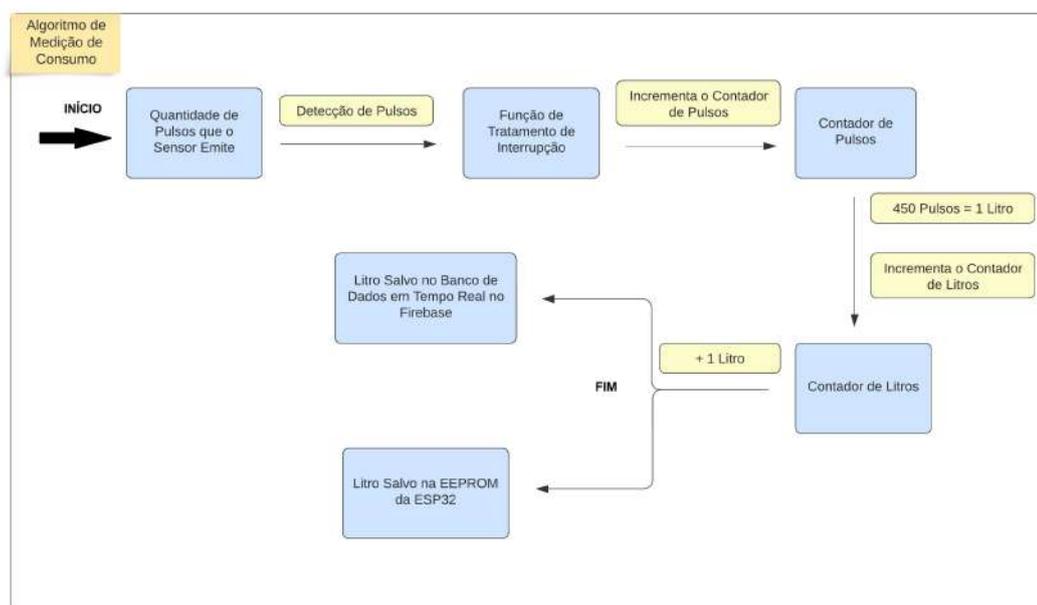
Após a criação do projeto, a plataforma web forneceu um identificador único para o banco de dados também criado, e este identificador único, consiste em uma sequência de caracteres. Todos os dispositivos de medição fazem uso desse identificador único na hora de enviar os dados de consumo, pois é através desse identificador único que a plataforma web do Firebase saberá em qual banco de dados, os dados de consumo serão persistidos. Por este motivo, este identificador único está gravado no firmware do dispositivo de medição de consumo.

Dessa forma, é importante destacar que todos os dispositivos fazem uso do mesmo banco de dados no Firebase e, portanto, compartilham o mesmo identificador único. Então, dentro do banco de dados, o consumo medido por cada dispositivo é identificado pelo endereço do estabelecimento no qual o dispositivo está instalado. O endereço é representado pelo CEP e pelo número do estabelecimento. CEP e número são concatenados, formando um identificador único para o consumo medido por cada dispositivo.

Para estabelecer a conexão entre a ESP32 e o Firebase, o firmware do dispositivo de medição de consumo utiliza as bibliotecas Wi-Fi Arduino e a biblioteca Firebase Arduino Client Library para ESP32. Esta última, responsável por disponibilizar funções para trabalhar com banco de dados em tempo real no Firebase.

Na figura 27 é apresentado o fluxograma contendo fluxograma do algoritmo do firmware do dispositivo de medição de consumo.

Figura 27 – Fluxograma do Algoritmo do Firmware do Dispositivo de Medição de Consumo



Fonte: Autoria Própria

## 3.2 Desenvolvimento da Aplicação Web

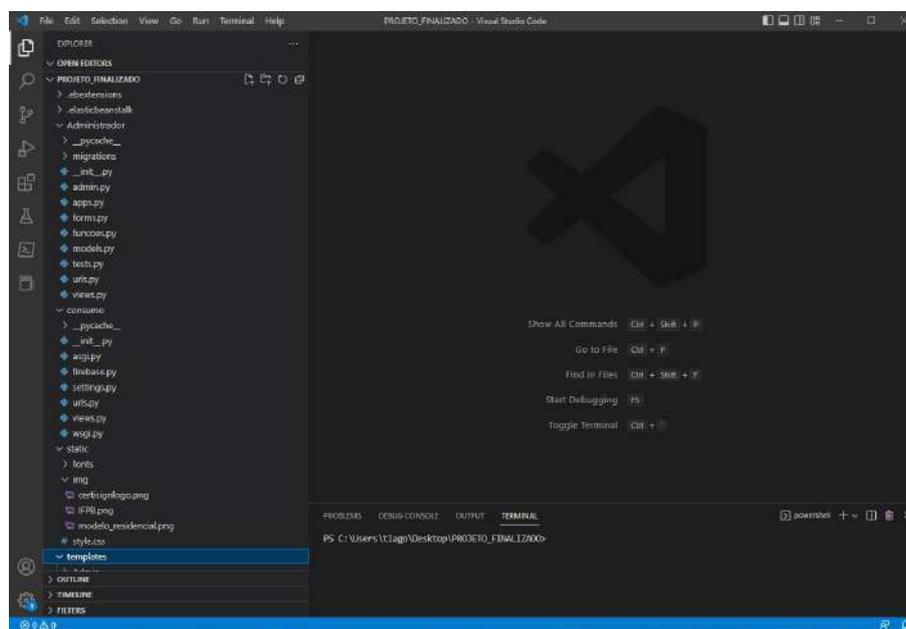
### 3.2.1 Ferramentas e Linguagens Utilizadas

O papel da plataforma web é fornecer a concessionária de água uma maneira de gerenciar os clientes, visualizar o consumo e contabilizar a fatura de cada cliente. Dessa forma, buscou-se utilizar algumas ferramentas e linguagens fundamentais para o desenvolvimento desta aplicação. Essas ferramentas e linguagens serão detalhadas a seguir.

#### 3.2.1.1 Visual Studio Code

O visual Studio Code, como ilustrado na Figura 28, é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele vem com suporte integrado para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens e tempos de execução (como C++, C#, Java, Python, PHP, Go, .NET) (MICROSOFT, 2023). Este editor foi utilizado em toda o desenvolvimento do projeto da plataforma web.

Figura 28 – Visual Studio Code



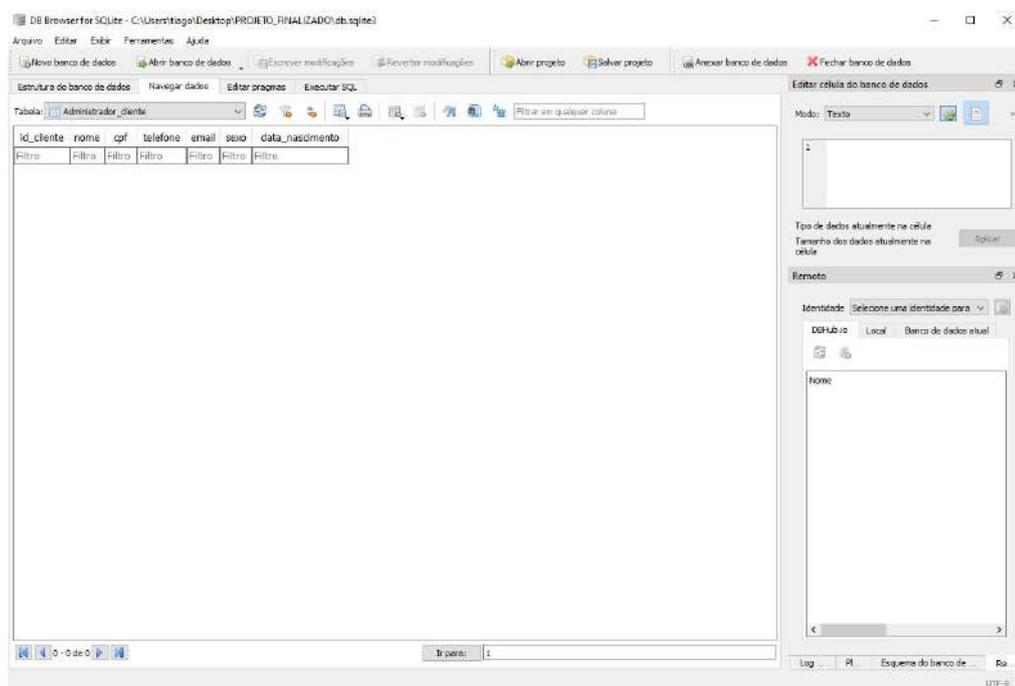
Fonte: Autoria Própria

#### 3.2.1.2 DB Browser for SQLite

O DB Browser for SQLite (DB4S) é uma ferramenta visual de código aberto de alta qualidade para criar, projetar e editar arquivos de banco de dados compatíveis com SQLite. O DB4S é para usuários e desenvolvedores que desejam criar, pesquisar e edi-

tar bancos de dados. O DB4S usa uma interface familiar semelhante a uma planilha e comandos SQL complicados não precisam ser aprendidos (DIGITALOCEAN, 2023).

Figura 29 – Ilustração da Ferramenta DB Browser for SQLite



Fonte: Autoria Própria

Esta ferramenta foi utilizada em todas as etapas do projeto para visualização da base de dados quando necessário, como por exemplo, pode ser observado na Figura 29, onde é possível visualizar as informações da tabela de clientes salva no banco de dados (arquivo db.sqlite3). O arquivo db.sqlite3 é um arquivo binário criado pelo framework web Django, para funcionar como base de dados. E por ser binário, sua visualização é dificultada dentro do Visual Studio Code. Dessa forma, para ter certeza que os dados haviam sido armazenados corretamente no banco de dados, utilizou-se o DB Browser for SQLite(DB4S) para uma melhor visualização de todos os dados armazenados, referentes aos clientes, endereços, faturas e administrador.

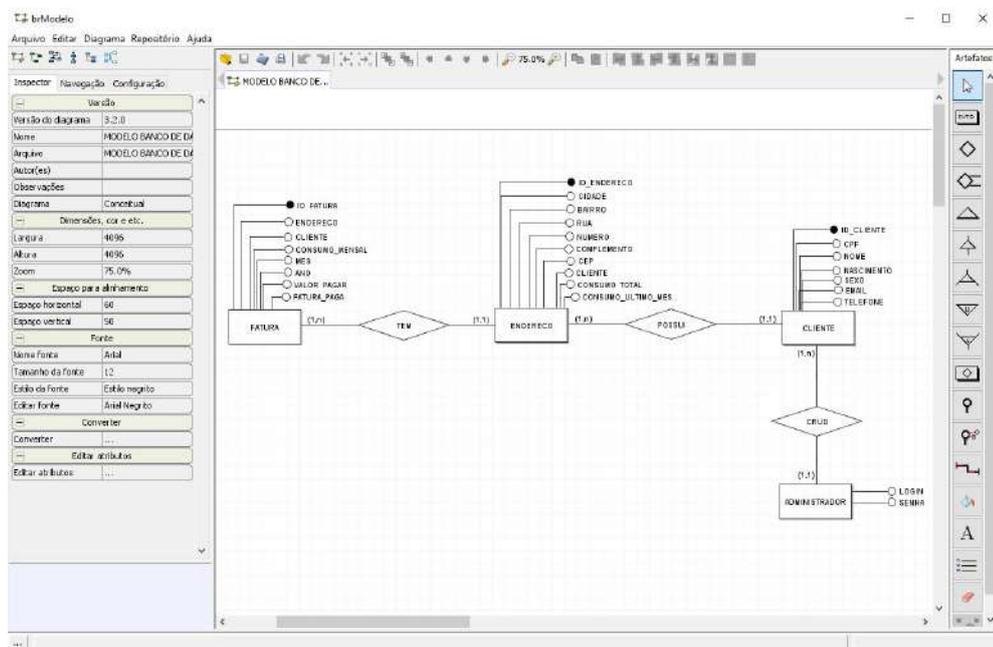
### 3.2.1.3 BrModelo

Segundo (RAMOS; MENNA, 2011), o brModelo é uma ferramenta de apoio ao projeto de um Banco de Dados relacional desenvolvida pelo GBD/UFSC. Ela consiste em um aplicativo que permite a definição de modelagens conceituais e lógicas para BDs relacionais com facilidade e independência de SGBD. A principal finalidade do desenvolvimento da brModelo foi a de oferecer uma ferramenta voltada para o aprendizado de modelagem de BDs relacionais.

A interface da brModelo é composta basicamente por uma representação da página (semelhante a editores de texto) utilizada para a criação do desenho da modelagem. Uma barra vertical à esquerda contém todos os elementos envolvidos no modelo em edição no momento (conceitual ou lógico). Estes elementos podem ser selecionados e inseridos no desenho da modelagem utilizada. Na parte direita da tela existe um inspetor de elementos, que exhibe as características de um ou mais elementos selecionados na modelagem e possibilita a edição destas características. A brModelo possui ainda uma barra de ferramentas e uma barra de menu, através das quais é possível acessar funções comuns a muitos sistemas, como funções de gerenciamento de arquivos e gerenciamento de conteúdo, bem como outras funções relativas ao processo de modelagem de BD, como por exemplo, o mapeamento entre os modelos.

Na Figura 30 é ilustrado um exemplo do uso do brModelo com a modelagem do modelo conceitual do banco de dados deste projeto.

Figura 30 – Ilustração da Ferramenta brModelo



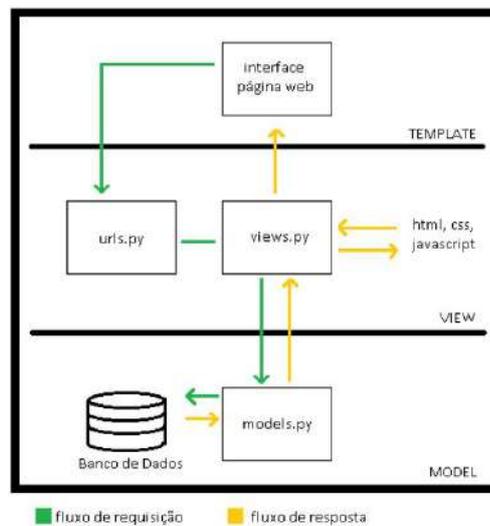
Fonte: Autoria Própria

### 3.2.1.4 Framework web Python - Django

Django é um framework para desenvolvimento rápido para web, escrito em Python, que utiliza o padrão model-template-view como apresentado na Figura 31. Incentiva um desenvolvimento rápido e um design limpo e pragmático, ao mesmo tempo que proporciona uma curva de aprendizado relativamente suave. Isso o torna atraente tanto para os programadores iniciantes como para os mais experientes (MELÉ, 2020).

Segundo (TELES; CALÇADA; VERAS, 2020), o Django também ajuda a evitar erros de segurança comuns e proporciona escalabilidade aos sistemas, ou seja, consegue oferecer capacidade de expansão de um sistema sem perda do seu desempenho, como é o que acontece por exemplo com Mozilla Firefox, Pinterest e Instagram. Muitas empresas escolhem o Django por sua extrema versatilidade, sendo ele utilizado para criação de sistemas que vão desde gerenciamento de conteúdo até plataformas científicas.

Figura 31 – Fluxograma Model-Template-View do Django



Fonte: (TELES; CALÇADA; VERAS, 2020)

Na utilização da estrutura MTV (Model-Template-View), o framework Django gerencia a maior parte da comunicação entre requisições HTTP. Quando é criado um projeto em Django temos um diretório inicial de pastas e arquivos, onde estão os arquivos `models.py`, `views.py` e `urls.py`, responsáveis pelo fluxo MTV (TELES; CALÇADA; VERAS, 2020 apud FOUNDATION, 2023).

De acordo com (TELES; CALÇADA; VERAS, 2020 apud FOUNDATION, 2023) é possível observar as camadas MTV e o funcionamento do fluxo entre elas:

- **Model:** Django já vem com uma solução para mapeamento objeto-relacional (ORM, do inglês Object-Relational Mapping) no qual o esquema do banco de dados é descrito em código Python. Neste caso, abstrações em Python podem ser usadas para criar consultas complexas sem que seja necessário realizar ações diretas no banco.
- **Template:** Esta é a camada de apresentação, onde as informações são visualizadas pelos usuários. Um template consiste de partes estáticas do arquivo HTML de saída e de partes com uma sintaxe especial que descrevem como o conteúdo dinâmico será apresentado. O Django tem um caminho de pesquisa para templates, o qual permite minimizar a redundância entre eles. Geralmente, uma view recupera dados

de acordo com os parâmetros de pesquisa, carrega um template e o renderiza com os dados recuperados.

- **View:** As views recebem a informação e o tipo da requisição (“POST” ou “GET”) do lado do cliente e, em seguida, formatam os dados para que sejam armazenados no banco através dos models da camada Model. As views também se comunicam via models com o banco para recuperar dados que são transferidos posteriormente aos templates, para a visualização do usuário. Cada view é responsável por fazer uma entre duas coisas: devolver um objeto `HTTPResponse` contendo o conteúdo para a página requisitada ou levantar uma exceção como `Http404`.

Toda a aplicação Web que compõe a solução de monitoramento de consumo de água apresentada neste trabalho foi desenvolvida com o framework web Django. Desde a criação e organização das tabelas do banco de dados até o desenvolvimento das funcionalidades da aplicação.

### 3.2.1.5 HTML E CSS

O HTML é uma linguagem de marcação utilizada na construção de páginas Web e os documentos HTML podem ser interpretados por navegadores.

Segundo (MEIRELES, 2010), HTML é uma linguagem derivada e completamente especificada a partir da linguagem SGML. No entanto, não é preciso sabê-la para usar HTML. Ela não é uma linguagem de programação como C, Python, Java, dentre outras que permite a criação de algoritmos, no entanto, é uma linguagem declarativa que tem por função organizar informações em um arquivo de texto, para posteriormente ser visualizado por um browser. A linguagem HTML serve para definir a aparência (formatação) dos documentos na web. Apesar de haverem outras linguagens usadas de forma concorrente, ela é ainda a base para as páginas web e interpretada por todos os navegadores disponíveis.

Na construção de uma página Web é usado um arquivo de texto simples. Mesmo que essa página exiba vários recursos, há por trás uma página de texto e outros arquivos separados que foram montados pelo navegador. O código é texto, mas não somente texto. Ele é todo marcado com HTML, que é responsável por definir toda a estrutura da página para que o navegador possa formatá-la e exibi-la de modo correto. O arquivo HTML tem seus elementos definidos usando "etiquetas de markup", ele é repleto de marcadores que se destacam através dos caracteres especiais: `<` e `>`. Essas etiquetas são responsáveis por dar ao browser as instruções necessárias sobre a estrutura do documento e sobre a forma como a página deve ser apresentada graficamente.

Para o desenvolvimento de um código em HTML, faz-se necessário somente dois tipos de software básicos: um editor de textos e um navegador. Um documento HTML é um arquivo de texto comum, escrito com o editor de textos e salvo com a extensão `.htm`

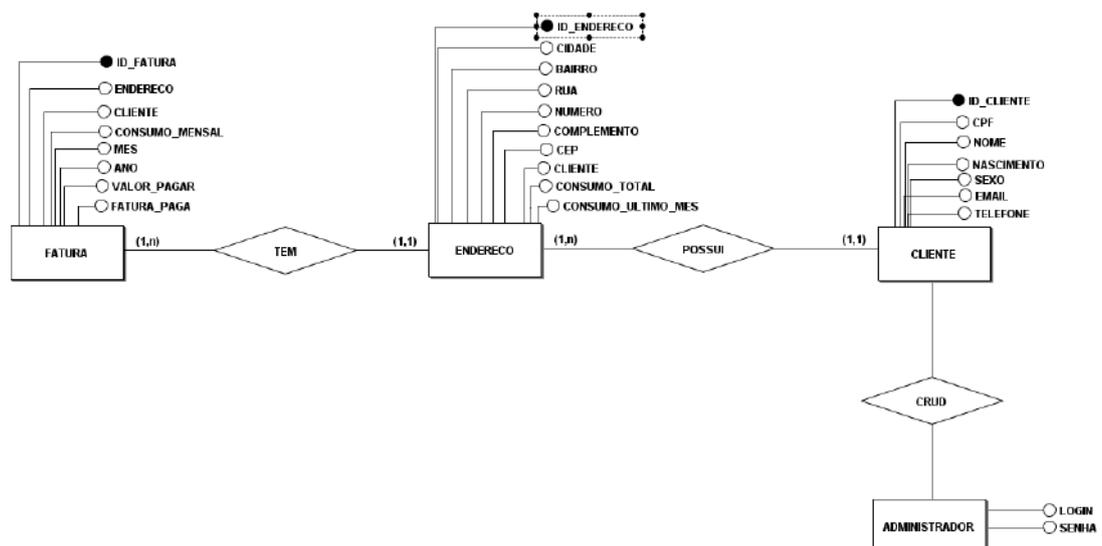
ou .html. Existem variados editores de textos que podem ser utilizados de acordo com a preferência do programador. Para usuários do sistema Windows, é aconselhável o uso do Bloco de Notas (Notepad).

Já o CSS é um mecanismo para adicionar estilos a uma página web, aplicado diretamente nas tags HTML. Assim, o HTML e o CSS foram utilizados no desenvolvimento das páginas web que apresentaram para os usuários da aplicação web os dados cadastrais dos clientes, o seu consumo e suas faturas.

### 3.2.2 Banco de Dados

Para começar a construção da aplicação web, foi necessário a elaboração de um modelo conceitual do banco de dados, como apresentado na Figura 32.

Figura 32 – Modelo Conceitual do Banco de Dados do Projeto



Fonte: Autória Própria

O banco de dados elaborado possui quatro entidades: a entidade cliente, fatura, endereço e administrador. A entidade administrador possui os atributos login e senha. Esta entidade representa os administradores da plataforma web desenvolvida que possuem acesso a todas as funcionalidades do sistema.

A entidade cliente possui os atributos id\_cliente, CPF, nome, nascimento, sexo, e-mail, telefone. Esta entidade representa os clientes da concessionária de água. Não há cardinalidade da entidade cliente em relação a entidade administrador, porém, ao relacionar a entidade cliente com a entidade endereço teremos uma cardinalidade de (1, n), ou seja, um cliente poderá ter no mínimo um endereço e no máximo n(vários) endereços.

A entidade endereço possui os atributos id\_endereço, cidade, bairro, rua, número, complemento, cep, cliente, consumo\_total, consumo\_ultimo\_mes, e representa os en-

dereços dos clientes cadastrados. Abordando um pouco sobre cardinalidade da entidade endereço em relação a entidade cliente, temos uma cardinalidade de (1,1), ou seja, um endereço só poderá ter no mínimo um cliente e no máximo também um único cliente. No entanto, se observarmos a relação da entidade endereço com a entidade fatura, iremos obter uma cardinalidade de (1,n), onde, um endereço poderá ter no mínimo uma fatura e no máximo n(varias) faturas.

A entidade fatura possui os atributos `id_fatura`, `endereço`, `cliente`, `consumo_mensal`, `mes`, `ano`, `valor_pagar`, `fatura_paga`. A cardinalidade da entidade fatura em relação a entidade endereço é de (1, 1), ou seja, uma fatura só poderá possuir no mínimo um endereço, e no máximo, também um único endereço. O atributo `consumo_mensal` será o valor consumido de água por mês e o atributo `valor_pagar` será o valor a ser pago pelo cliente.

### 3.2.2.1 Banco de Dados SQLite

O SGBD utilizado na persistência das entidades manipuladas pela aplicação web foi o SQLite. O arquivo do banco de dados onde as entidades serão persistidas é criado juntamente com outros arquivos por padrão quando se cria um projeto no framework Django. O SQLite é distribuído juntamente com o Django, de modo que você não precisa instalar nada a mais para ter o banco de dados.

Segundo (FOUNDATION, 2001-2023), o SQLite é uma biblioteca C que fornece um banco de dados leve baseado em disco que não requer um processo de servidor separado e permite acessar o banco de dados usando uma variante não padrão da linguagem de consulta SQL. Alguns aplicativos podem usar o SQLite para armazenamento interno de dados. Também é possível prototipar um aplicativo usando SQLite e, em seguida, portar o código para um banco de dados maior, como PostgreSQL ou Oracle.

## 3.2.3 Preparando o Ambiente de Desenvolvimento

No ambiente de desenvolvimento da aplicação web foram instalados o Visual Studio Code, versões compatíveis da linguagem python e do framework Django. Além disso, foram instaladas extensões no Visual Studio Code para melhorar a experiência de codificação.

## 3.2.4 Funcionalidades da aplicação Web

Do ponto de vista das funcionalidades, a aplicação Web possui cinco grandes módulos, onde cada módulo agrupa um conjunto de funcionalidades relacionadas entre si. Os módulos são: Autenticação, Cadastro de Administrador, Cliente, Endereço e Fatura.

O Módulo de autenticação possui as funcionalidades de solicitar o nome do usuário e a senha de acesso ao usuário administrador, verificar estas informações de autenticação

e liberar o acesso a todos os demais módulos caso as credenciais fornecidas pelo administrador estejam corretas.

No módulo do Cadastro do Administrador, é possível para o administrador chefe cadastrado no sistema a partir da inicialização do sistema, cadastrar novos administradores.

No módulo Cliente é possível criar um novo cliente ou listar os clientes já criados. Ao listar os clientes, é possível editar o cliente já criado, visualizar os dados do cliente e até mesmo excluí-lo. Na visualização dos dados do cliente é destacado os dados pessoais do cliente e quais endereços o cliente tem posse. O administrador também possui o acesso ao consumo global de cada endereço de um dado cliente desde a sua primeira medição e o consumo do mês atual. Há também uma função de pesquisar o cliente pelo nome e limpar a pesquisa caso não se tenha mais interesse em pesquisar algo sobre o nome digitado.

Nas funcionalidades relacionadas ao endereço, o administrador poderá criar novos endereços para um cliente já existente. Se caso, o cliente já existir na aplicação, o CPF digitado na página de criação de endereço vincula o endereço criado ao cliente que possui o CPF informado. O administrador também poderá editar ou excluir um endereço já existente, basta apenas ter o CEP e o número da residência para realizar a busca do endereço e, em seguida, realizar uma das duas operações;

No módulo relacionado a fatura é possível listar as faturas que foram criadas ao longo dos meses pela aplicação web. Contendo os dados dos meses, anos, o consumo de cada mês e o nome do cliente a qual a fatura pertence, além do valor a ser pago pelo consumo de água mensal. Também há uma funcionalidade de buscar faturas, filtradas por CPF, mês e ano da fatura.

#### 3.2.4.1 Cálculos da Fatura

Os cálculos do consumo residencial foram implementados no código de acordo com a lógica da estrutura tarifária estabelecida pela CAGEPA, a Companhia de Água e Esgotos da Paraíba, como representado na Figura 33.

Figura 33 – Estrutura Tarifária Residencial - CAGEPA

Faixa de Consumo (m <sup>3</sup> /Eco/mês)	Valor Agua (R\$)	Valor Esgoto (R\$)
até 10m <sup>3</sup>	46,28	37,02
excedente de 10 - 20	5,97	4,78
excedente de 20 - 30	7,87	7,08
excedente de 30	10,69	10,69

Fonte: (PARAÍBA, 2023)

Esses dados tarifários e essa faixa de consumo, são utilizados para o fornecimento de água e coleta de esgotos sanitários para fins domésticos em economia de uso exclusivamente residencial. O valor em reais definido pelos cálculos no software, é disponibilizado em cada fatura gerada uma vez por mês, de acordo com o consumo de água da residência.

Para um dado endereço, no Firesabe está armazenado o consumo global de água desde endereço, contando a partir da data de instalação do dispositivo de medição de consumo na residência. Portanto, com o passar do tempo, este consumo global nunca decresce. Ele pode permanecer constante quando não há consumo de água ou pode crescer, caso ocorra consumo. As 23 horas e 59 minutos do último dia do mês a aplicação web gera a fatura de cada endereço baseado no volume de água consumido neste mês que chegou ao fim. Para determinar o valor do consumo mensal para um dado endereço, a aplicação web subtrai o consumo global de água da residência medido até o instante da geração da fatura, obtido do Firebase, do valor consumo global do mês anterior.

De acordo com a Figura 33, sabemos que até 10 m<sup>3</sup> de água consumidos o valor a ser pago é R\$ 46,28 reais para água e R\$ 37,02 para esgoto.

Dessa forma, temos:

$$Valor = 46,28 + 37,02 = R\$83,30 \quad (3.1)$$

Portanto, o valor a ser pago é R\$ 83,30 reais para até 10 m<sup>3</sup>, de acordo com a Equação 3.1, seguindo as regras de estrutura tarifária residencial da CAGEPA, apresentada na Figura 33 .

Para a situação em que o consumo de água ultrapassa 10 m<sup>3</sup>, há uma classificação de faixas de consumo excedentes, de acordo com a Figura 33. Dessa forma, para descobrir esse valor excedente em m<sup>3</sup>, primeiramente subtraímos o consumo atual do mês por 10000. O consumo atual é o valor em litros que foi coletado do Firebase e depois tratado pelo software, e o valor 10000 é a conversão de m<sup>3</sup> para litros, onde 10 m<sup>3</sup> é equivalente a 10000 litros. Após a subtração, dividimos este resultado por 1000 para encontrar o consumo em m<sup>3</sup> como representado na Equação 3.2. Assim, é possível identificar em qual faixa se enquadra melhor o consumo utilizando a estrutura tarifária residencial da CAGEPA.

$$Consumo = \frac{ConsumoAtual - 10000}{1000} = X m^3 \quad (3.2)$$

Para a faixa excedente entre 10 m<sup>3</sup> e 20 m<sup>3</sup> de acordo com as tarifas estabelecidas a serem pagas na Figura 33, temos:

$$valor = ((Consumo \times 5,97) + 46,28) + (37,02 + (Consumo \times 4,78)) \quad (3.3)$$

Para a faixa excedente entre 20 m<sup>3</sup> e 30 m<sup>3</sup>, temos:

$$valor = ((Consumo \times 7,87) + 46,28) + (37,02 + (Consumo \times 7,08)) \quad (3.4)$$

Para a faixa excedente de 30 m<sup>3</sup>, temos:

$$valor = ((Consumo \times 10,69) + 46,28) + (37,02 + (Consumo \times 10,69)) \quad (3.5)$$

### 3.3 Implantação da aplicação web no AWS

Após a implementação da aplicação web, ela foi hospedada no serviço AWS Elastic Beanstalk, que é um serviço de orquestração (configuração, gerenciamento e a coordenação automatizada de serviços, aplicações e sistemas de computador) oferecido pela Amazon Web Services (AWS), e que também permite balanceamento de carga, com o objetivo de hospedar aplicações web. O processo de hospedagem foi feito pelo terminal EB CLI, que é uma interface da linha de comando do AWS Elastic Beanstalk e fornece comandos interativos para simplificar a criação, atualização e o monitoramento de ambientes em um repositório local. Na Figura 34 é representado a aplicação hospedada.

Figura 34 – Hospedando a aplicação Web no AWS Elastic Beanstalk

```

C:\Windows\system32\cmd.exe - CALL C:\Users\tiago\.ebcli-virtual-env\Scripts\eb.exe init -p python-3.7 main
.ebcli-virtual-env C:\Users\tiago\Desktop\APLICACAO_WEB>eb status
environment details for: main-env
Application name: main
Region: us-west-2
Deployed Version: app-230416_200612724672
Environment ID: e-y3bg9tciq
Platform: arn:aws:elasticbeanstalk:us-west-2::platform/Python 3.7 running on 64bit Amazon Linux 2/3.5.1
Tier: WebServer-Standard-1.0
CNAME: main-env.eba-pjqwzak.us-west-2.elasticbeanstalk.com
Updated: 2023-04-16 23:08:41.719000+00:00
Status: Ready
Health: Green
!ert: An update to the EB CLI is available. See https://github.com/aws/aws-elastic-beanstalk-cli-setup to install the l
test version.

.ebcli-virtual-env C:\Users\tiago\Desktop\APLICACAO_WEB>eb deploy
reating application version archive "app-230416_202547740941".
loading main/app-230416_202547740941.zip to S3. This may take a while.
pload Complete.
023-04-16 23:25:51 INFO Environment update is starting.
023-04-16 23:25:55 INFO Deploying new version to instance(s).
023-04-16 23:26:01 INFO Instance deployment successfully generated a 'Procfile'.
023-04-16 23:26:11 INFO Instance deployment completed successfully.
023-04-16 23:26:19 INFO New application version was deployed to running EC2 instances.
023-04-16 23:26:19 INFO Environment update completed successfully.
!ert: An update to the EB CLI is available. See https://github.com/aws/aws-elastic-beanstalk-cli-setup to install the l
test version.

.ebcli-virtual-env C:\Users\tiago\Desktop\APLICACAO_WEB>

```

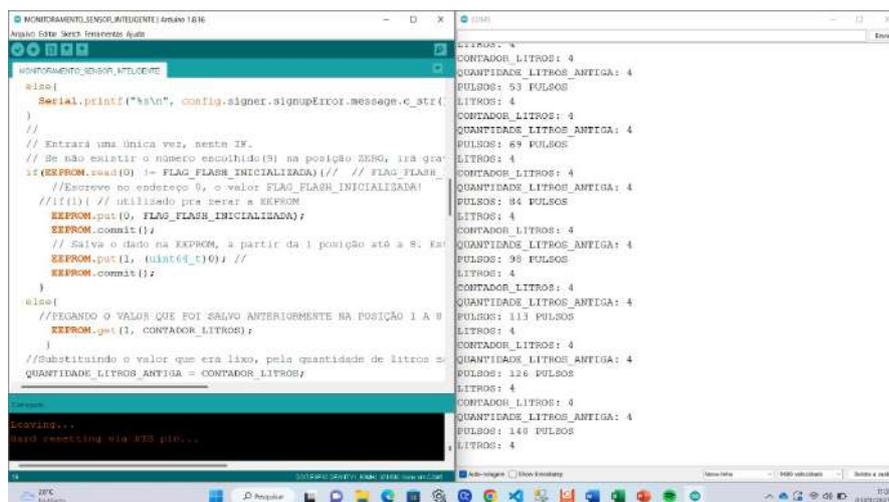
Fonte: Autória Própria



desligada rapidamente e, assim, foram contados apenas 3 pulsos, devido ao pequeno fluxo de água que passou através do sensor de vazão YF-S201. A mudança de 0 litro para 1 litro só ocorrerá quando a contagem de pulsos alcançar 450 pulsos.

Observando a Figura 36 é possível perceber que após ligar a torneira novamente e deixa-la ligada, a marca de 450 pulsos foi atingida quatro vezes, pois a quantidade de litros é 4, dessa forma, o contador de pulsos sempre é zerado ao atingir 450 pulsos, e começa novamente a contagem a partir 0 pulso.

Figura 36 – Funcionamento do Protótipo de Medição com a Contagem de 4 litros



Fonte: Autoria Própria

Na Figura 37 é apresentado o valor de consumo da residência em tempo real no Firebase.

Figura 37 – Dados do consumo persistidos no Firebase



Fonte: Autoria Própria

A cada novo litro, o protótipo de medição envia o valor de consumo via Wi-Fi

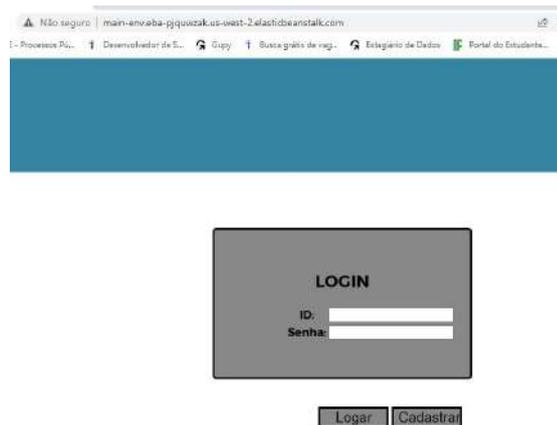
para a plataforma Firebase, e dessa forma, o valor é persistido no banco de dados criado nesta plataforma, no endereço composto por cep e número da residência. Este endereço foi adicionado no Firebase automaticamente assim que registrado pelo administrador na interface web.

## 4.1.2 Interface Web

### 4.1.2.1 Login e Cadastro do Administrador

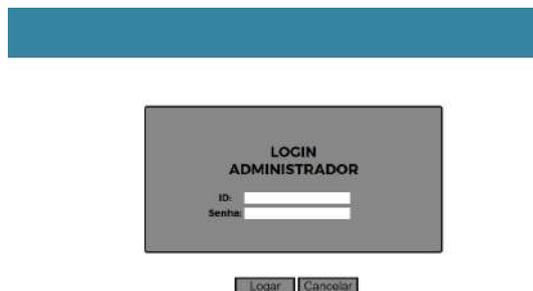
Na Figura 38 temos a primeira página da interface já hospedada no AWS Elastic Beanstalk, a página de login, onde é possível digitar o ID e a senha do administrador já cadastrado no sistema, para acessar a página principal da aplicação. O próprio sistema foi desenvolvido para já possuir um administrador chefe, criado quando o sistema é inicializado a primeira vez, capaz de criar outros administradores.

Figura 38 – Página de Login para Acesso do Administrador



Fonte: Autoria Própria

Figura 39 – Cadastro do Administrador



Fonte: Autoria Própria

Caso um administrador da concessionária não esteja cadastrado, é possível clicar no botão cadastrar e logo será apresentado a tela de login do administrador, mas, este acesso só estará disponível para o administrador chefe do sistema, onde ele será responsável por cadastrar outros administradores, como é apresentado na Figura 39.

Na tela de cadastro serão preenchidas informações referentes ao novo administrador, como apresentado na Figura 40. Assim, o administrador cadastrado poderá acessar a página principal com seu ID e senha que foram criados.

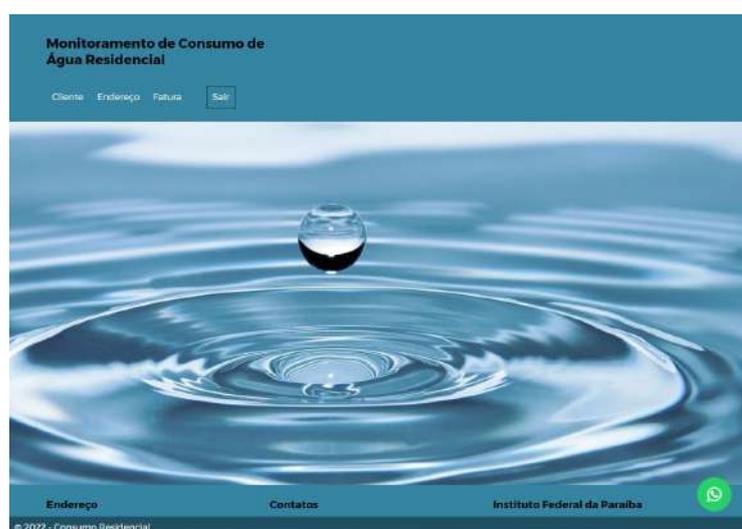
Figura 40 – Cadastro do Administrador

A imagem mostra um formulário de cadastro de administrador. No topo, há uma barra decorativa azul. Abaixo, um formulário cinza com o título "ADMINISTRADOR". O formulário contém campos de entrada para "Cpf", "Nome", "ID" e "Senha". Abaixo dos campos, há dois botões: "Salvar" e "Cancelar".

Fonte: Autoria Própria

#### 4.1.2.2 Página Principal da Interface Web

Figura 41 – Interface Web Finalizada

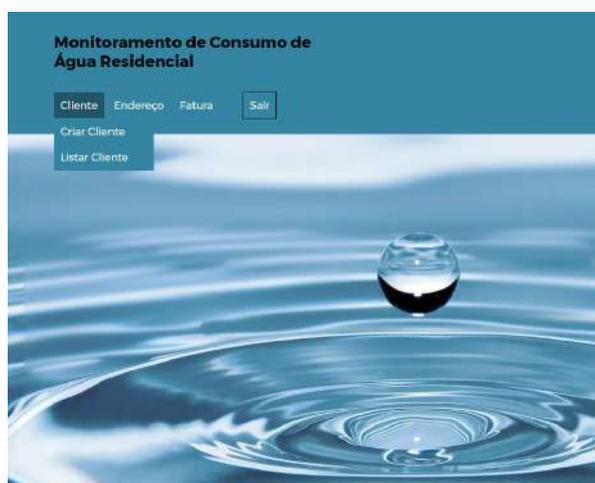


Fonte: Autoria Própria

Na Figura 41, é possível observar a página principal da interface web que foi criada com três opções de abas para uso: cliente, endereço, fatura.

A primeira aba como podemos observar na Figura 42, é referente ao cliente, onde o administrador poderá criar um novo cliente ou listar todos os clientes já criados.

Figura 42 – Opções Disponibilizadas pela Aba de Cliente



Fonte: Autoria Própria

A segunda aba ilustrada na Figura 43, é referente aos endereços. Nessa aba, o administrador poderá criar um novo endereço, editar um endereço já existente ou excluir um endereço.

Figura 43 – Opções Disponibilizadas pela Aba de Endereço



Fonte: Autoria Própria

A terceira aba ilustrada na Figura 44 é referente a fatura, nesta aba o administrador poderá listar as faturas referentes a cada endereço.

Figura 44 – Ilustração das Opções Disponibilizadas pela Aba de Fatura



Fonte: Autoria Própria

#### 4.1.2.3 Cadastro de um Cliente

Na página de cadastrar cliente, como pode ser visto na Figura 45, é possível, para o administrador, adicionar novos clientes ao sistema. Nesta página os dados dos clientes a serem cadastrados são, o nome do cliente, CPF, telefone, e-mail, sexo e data de nascimento.

Figura 45 – Ilustração do Cadastro de um Clientes



Fonte: Autoria Própria

#### 4.1.2.4 Listar Clientes

Na Figura 46, é possível visualizar a listagens dos clientes já cadastrados no sistema. Essa página apresenta algumas funcionalidades como a opção de pesquisa pelo primeiro nome ou nome completo do cliente.

Figura 46 – Ilustração da Lista de Clientes

**Lista de Clientes**

Pesquisa:

**Listagem**

- TIAGO DALTRO DUARTE -  -  -
- Gabriel Daltro Duarte -  -  -

© 2022 - Consumo Residencial

Fonte: Autoria Própria

A opção de editar os dados do cliente já cadastrado, como pode ser visto na Figura 47, permite que o administrador altere algum dos dados do cliente.

Figura 47 – Página de Editar Cliente

**Cadastrar Cliente**

Nome:

Cpf:

Telefone:

Email:

Sexo:

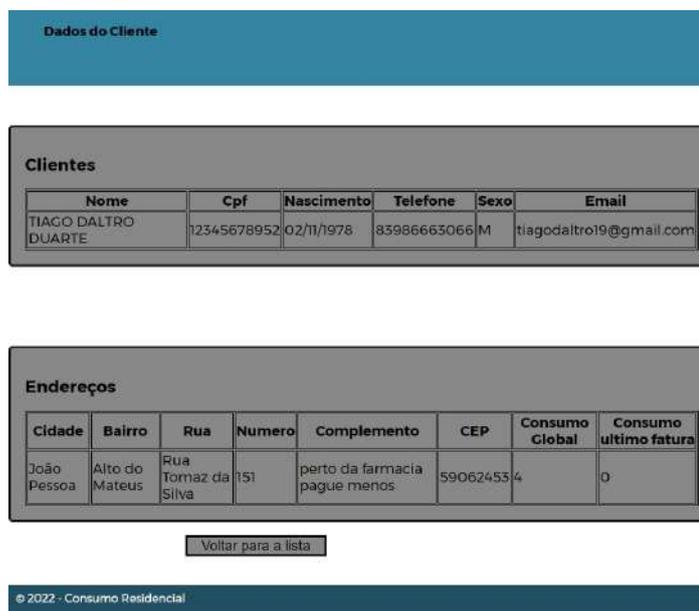
Data nascimento:

© 2022 - Consumo Residencial

Fonte: Autoria Própria

A opção de visualizar os dados do cliente como mostra a Figura 48, permite ao administrador observar todos os dados pessoais do cliente e seu(s) endereço(s). Além disso, é por essa página que o administrador consegue monitorar o consumo de água global e o consumo de água do último mês do cliente, referente a cada endereço seu.

Figura 48 – Página dos Dados do Cliente Cadastrado no Sistema



Fonte: Autoria Própria

Por último, a Figura 49 apresenta a funcionalidade de confirmação de exclusão do cliente do sistema.

Figura 49 – Página de Exclusão de um Cliente no Sistema



Fonte: Autoria Própria

#### 4.1.2.5 Cadastrar um Novo Endereço

A Figura 50 mostra a página de cadastrar endereço. Nesta página, o administrador adiciona novos endereços ao sistema, de acordo com o CPF de cada cliente. Os dados do

endereço do cliente a serem cadastrados são: CFP do endereço, cidade, bairro, rua, número, complemento e CEP.

Figura 50 – Página do Cadastro de um Novo Endereço

Cadastrar Novo Endereço

CPF do Cliente:

Cidade:

Bairro:

Rua:

Número:

Complemento:

Cep:

Salvar Cancelar

© 2022 - Consumo Residencial

Fonte: Autoria Própria

#### 4.1.2.6 Editar Endereço

Na Figura 51 é ilustrado a página da interface web que faz a busca do endereço para edição, através do CEP e o número da residência.

Figura 51 – Página para Busca do Endereço para Edição

Buscar Endereço

CEP:

Número:

Buscar Cancelar

© 2022 - Consumo Residencial

Fonte: Autoria Própria

Ao selecionar buscar, a página de busca é direcionada para página dos dados do endereço do cliente. Dessa forma, fica a critério do administrador fazer alguma alteração nos dados ou não, como resentedado na Figura 52.

Figura 52 – Página dos Dados do Endereço do Cliente

**Editar Endereço**

CPF:	12345678952
CIDADE:	João Pessoa
BAIRRO:	Alto do Matheus
RUA:	Rua Tomaz da Silva
NUMERO:	151
COMPLEMENTO:	perto da farmacia pague me!
CEP:	59062453

Salvar Cancelar

© 2022 - Consumo Residencial

Fonte: Autoria Própria

Na Figura 53, é ilustrado a página de busca do endereço para exclusão. A busca é realizada através do CEP e número da residência. Após digitar os dados requisitados, o administrador poderá analisar os dados de endereço do cliente para uma possível exclusão.

Figura 53 – Página de Busca do Endereço para Exclusão

**Buscar Endereço**

CEP:	59062453
Numero:	151

Buscar Cancelar

© 2022 - Consumo Residencial

Fonte: Autoria Própria

Assim, como apresenta a Figura 54, será possível o administrador excluir os dados de algum endereço se necessário, apertando o botão excluir.

Figura 54 – Página de Exclusão do Endereço

Excluir Endereço

CPF: 12345678962  
CIDADE: João Pessoa  
BAIRRO: Alto do Matheus  
RUA: Rua Tomaz de Silva  
NUMERO: 151  
COMPLEMENTO: perto da farmácia pague mais  
CEP: 59062453

Excluir Cancelar

© 2022 - Consumo Residencial

Fonte: Autoria Própria

#### 4.1.2.7 Lista de Faturas

Na Figura 55 temos a página de lista de faturas, onde é possível para o administrador, observar o mês da fatura, o ano, o valor a pagar da fatura, o consumo mensal do cliente e o nome do cliente, referente a um endereço. Na página, ainda foi adicionado um campo para determinação sobre se a fatura foi paga ou não, mas esse campo foi relacionado a trabalhos futuros, por isso estar na ilustração apenas como modelo.

A página de lista de faturas também possui uma funcionalidade de pesquisar as faturas dos clientes, se dado o CPF, mês e ano da fatura requerida.

Figura 55 – Página de Lista de Faturas

Lista de Faturas

Buscar Fatura

CPF: Digite CPF do Cliente  
Mês: Digite o mês da Fatura  
Ano: Digite o ano da Fatura

Buscar Limpar Voltar

Faturas

Mês	Ano	Valor a Pagar	Fatura Paga	Consumo Mensal	Nome do Cliente
-----	-----	---------------	-------------	----------------	-----------------

© 2022 - Consumo Residencial

Fonte: Autoria Própria

Figura 56 – Página de Lista de Faturas com Algumas Faturas

Faturas					
Mês	Ano	Valor a Pagar	Fatura Paga	Consumo Mensal	Nome do Cliente
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	10	TIAGO DALTRO DUARTE
12	2022	76.25	False	20	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	15	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	44	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	5	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE
12	2022	76.25	False	0	TIAGO DALTRO DUARTE

Fonte: Autoria Própria

Como mencionado anteriormente, uma das funcionalidades do software desenvolvido é geração de faturas de cada endereço. Para teste dessa funcionalidade, decidiu-se alterar momentaneamente, a data de geração das das fatura, de maneira que as faturas fossem geradas de dois em dois minutos. Dessa forma, o mês e o ano das faturas geradas durante esse teste permaneceria o mesmo, mas os demais atributos das faturas poderiam variar de uma fatura para outra. A Figura 56 apresenta as faturas geradas durante esse teste. O valor do consumo mensal é dado em litros e se altera de uma fatura para a outra porque durante a execução do teste, o dispositivo de medição de consumo de água estava ligado e realizando medições de consumo. Esse teste foi realizado com apenas um cliente, o qual, estar registrado seu nome na coluna de nome do cliente.

## 5 CONCLUSÃO

Neste trabalho, foi apresentado o desenvolvimento de um sistema integrado para monitoramento de consumo de água residencial. Buscou-se projetar um sistema inteligente que proporciona uma melhora significativa no desempenho e velocidade das medições. O trabalho dos leituristas demanda muito tempo para verificação de determinada área de um bairro e mais tempo ainda para visitar todas as residências em todas as regiões de uma cidade, e esse sistema integrado criado possui a função de transmitir os dados de consumo em tempo real de forma rápida, prática e bem mais eficiente.

Com base nos dados obtidos com a construção do protótipo de medição integrado a aplicação web desenvolvida, houve um resultado satisfatório do monitoramento do consumo de água residencial, disponibilizado em tempo real a concessionária de água por meio da aplicação web, juntamente com todos os outros dados pessoais dos clientes, seus endereço e suas faturas que são administrados por um ou mais administradores da concessionária.

## 6 TRABALHOS FUTUROS

Diante dos resultados obtidos com este trabalho, são sugestões para trabalhos futuros:

- A criação de um mecanismo de cadastro do endereço do cliente manualmente, no dispositivo de medição de consumo, contendo rua e CEP, feito por um técnico da própria concessionária na hora da instalação do dispositivo de medição na residência. Atualmente este cadastro é feito na gravação do firmware do dispositivo de medição.
- Realizar a integração da aplicação web com alguma plataforma de pagamentos online ou com algum sistema bancário para que as faturas sejam convertidas em boletos e enviadas para o e-mail do cliente automaticamente todos os meses.
- Implementar o cálculo da fatura para clientes não residenciais permitindo assim que estes tipos de estabelecimento possam ser monitorados pelo sistema desenvolvido.
- Permitir que os clientes da concessionária também possam acessar a plataforma web, podendo fazer o monitoramento do seu consumo. Onde esse consumo poderá ser visualizado por meio de gráficos que disponibilizem o consumo de água durante o dia, mês e ano.
- Adicionar uma bateria capaz de fornecer a energia necessária para o protótipo funcionar por longos períodos de tempo, se possível, que ele tenha uma autonomia superior um ano. Atualmente o dispositivo de medição necessita de uma alimentação externa o que exige a instalação de cabos de alimentação até onde o dispositivo está instalado, sendo necessário a alteração da estrutura da residência.
- Adicionar ao dispositivo de medição de consumo a capacidade de se conectar a uma rede móvel, como por exemplo, 3G ou 4G, para ganhar autonomia em relação a rede WIFI de residência na qual está instalado;
- Adicionar ao sistema uma condição para identificar se a fatura do cliente já foi paga ou não. Na interface web foi representado esta coluna, apenas como uma demonstração da funcionalidade.
- Criar um design final para o dispositivo de medição de consumo de água, com materiais e vedação de resistência a água.
- Estabelecer a segurança de dados criptografando os dados na esp32 e na nuvem, disponibilizando esses dados unicamente para a concessionária de distribuição de água.

## 7 ACESSO AOS CÓDIGOS DO PROJETO

O código da aplicação web e o firmware do dispositivo de medição podem ser visualizados em: <https://github.com/Ti7801/>.

# REFERÊNCIAS

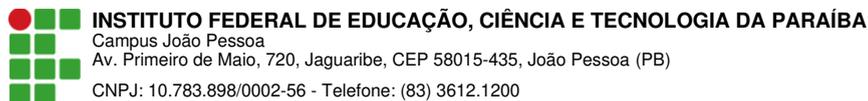
- ALVES, W. P. *Banco de Dados*. [S.l.]: Editora Érica, 2015. Citado na página 24.
- AMERICO, G. M.; IZIDORO, C. L. Sistema de baixo custo para monitoramento do consumo de Água. *Periódico Científico da UniSATC*, v. 7, n. 1, p. 292–317, 2022. Citado na página 3.
- APRENDIZAGEM, S. L. de Inovação e. *ESP32 – Pin Out*. 2020. Disponível em: <<https://www.studiopieters.nl/esp32-pinout/>>. Citado na página 30.
- ARAUJO, M. A. P. Modelagem de dados – teoria e prática. In: *MODELAGEM DE DADOS – TEORIA E PRÁTICA*. Valença, RJ: [s.n.], 2021. p. 33–66. Citado na página 26.
- AZEVEDO, C. L. de; AZEVEDO, D. K. de. *Monitoramento de Medidores de Água via Sistema Supervisório*. Monografia (Trabalho de Concluso de Curso) — Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná, CURITIBA, 2011. Citado na página 7.
- BARROS, E.; CAVALCANTE, S. Introdução aos sistemas embarcados. p. 1–36, 2010. Citado 4 vezes nas páginas 10, 11, 12 e 15.
- BRANDÃO, I. A. de P. *Desenvolvimento IoT de Micromedição Integrado com Cloud Computing para Monitoramento de Consumo de Água*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, Campina Grande - CG, 2022. Citado na página 7.
- CARVALHO, W. de F. *Medição individualizada de água em apartamentos*. Monografia (ESPECIALIZAÇÃO EM CONSTRUÇÃO CIVIL) — Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte – Minas Gerais, 2010. Citado na página 1.
- COSTA, A. A. A. da. *Desenvolvimento de Sistema de Automação Residencial Utilizando Microcontrolador*. Monografia (Trabalho de Concluso de Curso) — Departamento de Engenharias e Ciência da Computação, Universidade R. I. do A. Uruguai e das Missões, ERECHIM - RS, 2020. Citado 2 vezes nas páginas 5 e 29.
- DIGITALOCEAN, P. by. *DB Browser for SQLite*. 2023. Disponível em: <<https://sqlitebrowser.org/>>. Citado na página 41.
- DUTRA, M. M. *Desenvolvimento de um Sistema de Monitoramento de Baixo Custo e Aplicativo Móvel para Medição do Consumo de Água*. Monografia (Trabalho de Concluso de Curso) — Centro de Tecnologia-Curso de Engenharia de Controle e Automação, Universidade Federal de Santa Maria, Santa Maria-RS, 2019. Citado 3 vezes nas páginas 2, 8 e 31.
- FEZARI, M.; DAHOUD, A. A. Integrated development environment "ide"for arduino. 10 2018. Citado na página 31.

- FIREBASE. *Banco de dados em tempo real do Firebase*. 2011. Disponível em: <<https://firebase.google.com/docs/database?hl=pt-br>>. Citado na página 27.
- FOUNDATION, D. S. *Documentação do django*. 2023. Disponível em: <<https://docs.djangoproject.com/en/4.2/>>. Citado na página 43.
- FUNDATION, P. S. *sqlite3— Interface DB-API 2.0 para bancos de dados SQLite*. 2001–2023. Disponível em: <<https://docs.python.org/3/library/sqlite3.html>>. Citado na página 46.
- FUTIDA, I. T.; ROMERO, R. A. F. *Desenvolvimento mecânico e de Controle PWM para sistema robótico*. São Carlos, SP, Brasil., 2008. Citado na página 12.
- GOMES, H. V. *Metodologia de Projeto de Software Embarcado Voltada ao Teste*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010. Citado na página 15.
- GONÇALVES, R. F. et al. Uma proposta de processo de produção de aplicações web. *Revista Produção*, v. 15, n. 3, p. 376–389, 2005. Citado na página 16.
- HONEYWELL. *Hall Effect Sensing and Application*. 2016. Disponível em: <<http://denethor.wlu.ca/pc300/projects/sensors/hallbook.pdf>>. Citado na página 9.
- IBGE. *Conheça o Brasil - População DOMICÍLIOS BRASILEIROS*. 2019. Disponível em: <<https://educa.ibge.gov.br/jovens/conheca-o-brasil/populacao/21130-domicilios-brasileiros.html>>. Citado na página 1.
- IDC. *The Business Value of Amazon Web Services Accelerates Over Time*. [S.l.: s.n.], 2012. Citado na página 23.
- IEEE. *IEEE Standard Glossary of Software Engineering Terminology, Version 610.12-1990*. [S.l.: s.n.], 1990. Citado na página 10.
- JUNIOR, D. de C. V. *Computação em Nuvem: Desenvolvimento de Aplicações Empresariais ricas na internet, na Arquitetura Java, Google Web Toolkit e Google Application Engine*. Monografia (Trabalho de Concluso de Curso) — Universidade Federal do Paraná, Cornélio Procópio - PR., 2011. Citado na página 23.
- JUNIOR, M. A. C.; VIRTUOSO, G. H. F.; MARTINS, P. J. Propriedades desejáveis a uma linguagem de programação: Uma análise comparativa entre as linguagens c, c++ e java. In: . [S.l.: s.n.], 2012. p. 1–4. Citado na página 32.
- JÚNIOR, G. T.; CORAL, S. Sistema de monitoramento de consumo de Água utilizando telemetria. p. 1–18, 2021. Citado 3 vezes nas páginas 1, 3 e 4.
- KERSCHBAUMER, P. R. *Microcontroladores*. Santa Catarina - SC, 2021. Citado na página 12.
- KUROSE, J.; ROSS, K. W. *Redes de Computadores e a internet: Uma abordagem top-down*. [S.l.]: 3ª Edição, Editora Pearson, São Paulo– SP, 2006. Citado na página 18.
- LIMA, L.; PETRICA, E. Protocolo http. p. 1–11, 2010. Citado 2 vezes nas páginas 18 e 19.

- LTDA, A. I. *Hidrômetros – Recursos, Medição, Certificado E Modelos*. 2019. Disponível em: <<https://agroclique.com.br/hidrometros-recursos-medicao-certificado-e-modelos/>>. Citado na página 8.
- MACHADO, V. P. *Desenvolvimento para Web*. Teresina-PI, 2019. Citado na página 15.
- MANSSOUR, P. I. H. Linguagem de programação c++ (notas de aula). 4 1999. Citado na página 32.
- MEIRELES, A. B. de. *DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA GERENCIAMENTO DE RESERVA DE RECURSOS UTILIZANDO O FRAMEWORK DJANGO*. Monografia (Trabalho de Concluso de Curso) — UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI, PARNAÍBA - PI, 2010. Citado na página 44.
- MELO, R. C. da S. *Sistema de Monitoramento de Consumo de Água Utilizando o Protocolo de Comunicação MQTT*. Monografia (Trabalho de Concluso de Curso) — Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Santa Maria-RS, 2018. Citado na página 4.
- MELÉ, A. *Aprenda Django 3 com Exemplos*. [S.l.]: Novatec Editora Ltda, 2020. Citado na página 42.
- MICROSOFT. *Visual Studio Code*. 2023. Disponível em: <<https://code.visualstudio.com/>>. Citado na página 40.
- MILETTO, E. M.; BERTAGNOLLI, S. C. *Desenvolvimento de Software II*. [S.l.]: Bookman Companhia Editora, 2014. Citado 2 vezes nas páginas 16 e 17.
- MORAES, C. C.; CASTRUCCI, P. L. *Engenharia de automação industrial*. [S.l.]: Editora LTC, 2007. Citado na página 6.
- MUNHOZ, G. V. *COMPUTAÇÃO EM NUVEM, UTILIZANDO COMO ESTUDO DE CASO UM SISTEMA PARA CLUBE DE LAZER*. Monografia (Trabalho de Concluso de Curso) — Fundação Educacional do Município de Assis, Instituto Municipal de Ensino Superior de Assis - IMESA, Assis-SP, 2012. Citado 2 vezes nas páginas 22 e 23.
- NICOLOSI, D. E. C. *Microcontrolador 8051 Detalhado*. [S.l.]: Editora ÉRICA, 2007. Citado na página 14.
- PARAÍBA, C. de A. e E. da. *Estrutura tarifária do Consumo Residencial*. 2023. Disponível em: <<http://agenciavirtual.cagepa.pb.gov.br/gsan/exibirConsultarEstruturaTarifariaPortalCagepaAction.do>>. Citado na página 47.
- PENIDO, de C. C.; TRINDADE, R. S. *Microcontroladores*. 2013. Disponível em: <<https://www2.ifmg.edu.br/ceadop3/apostilas/microcontroladores>>. Citado na página 13.
- PEREIRA, L. A. M.; CARVALHO, T. B. F. R.; MORAES, M. H. Software embarcado, o crescimento e as novas tendências deste mercado. *Revista de Ciências Exatas e Tecnologia*, p. 1–10, 2014. Citado 3 vezes nas páginas 11, 13 e 15.

- PONT, I. A. D. *Sistemas Embarcados*. Monografia (Trabalho de Concluso de Curso) — Curso de Engenharia Mecatrônica, Instituto Federal de Santa Catarina - CAMPUS CRICIÚMA, Criciúma - SC, 2020. Citado 2 vezes nas páginas 10 e 11.
- RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de Gerenciamento de Banco de Dados*. [S.l.]: McGraw-Hill Interamericana Editores, 2011. Citado 2 vezes nas páginas 24 e 25.
- RAMOS, L. A.; MENNA, O. S. *Portabilização da Ferramenta de Modelagem de Banco de Dados Relacional brModelo*. Monografia (Trabalho de Concluso de Curso) — UNIVERSIDADE FEDERAL DE SANTA CATARINA CENTRO TECNOLÓGICO, DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA CURSO DE SISTEMAS DE INFORMAÇÃO, Florianópolis - SC, 2011. Citado na página 41.
- ROMERO, L. de A. W. *TÉCNICAS DE DESAGREGAÇÃO DE CONSUMO D'ÁGUA A PARTIR DA DINÂMICA DE VAZÃO EM RESIDÊNCIAS*. Dissertação (Mestrado) — Universidade Federal da Bahia, Escola Politécnica, Salvador - BA, 2019. Citado na página 10.
- SANDERSON, D. *Programming Google App Engine, Second Edition*. [S.l.]: Mike Loukides and Meghan Blanchette, 2013. Citado na página 24.
- SANTOS, F. T. dos; MARINATO, M. M.; SANTOS, W. C. dos. Utilização de um sistema supervisorio na usina fotovoltaica. p. 1–4, 2021. Citado na página 6.
- SANTOS, J. W.; JUNIOR, R. C. de L. *Sistema de Automação Residencial de Baixo Custo Controlado pelo Microcontrolador ASP32 e Monitorado Via Smartphone*. Monografia (Trabalho de Concluso de Curso) — Departamento de Engenharias e Ciência da Computação, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019. Citado na página 5.
- SILVA, A. E. et al. Sith: Sistema de telemetria para medição de consumo de Água. In: *Anais do V Workshop de Trabalhos de Iniciação Científica e Graduação Bahia, Alagoas e Sergipe da 7 Escola Regional de Computação Bahia, Alagoas e Sergipe*. Salvador, Brasil: [s.n.], 2007. p. 1–12. Citado na página 8.
- SILVA, A. V. F. e. *UMA ANÁLISE COMPARATIVA DAS VERSÕES DO PROTOCOLO HTTP: EVOLUÇÃO E PONTOS QUE AMPLIEM O USO DO HTTP/3*. Monografia (Trabalho de Concluso de Curso) — CENTRO UNIVERSITÁRIO CHRISTUS, SISTEMAS DE INFORMAÇÃO, Fortaleza, 2021. Citado 2 vezes nas páginas 18 e 19.
- SOFTWARE, O. *Computação em Nuvem*. [S.l.]: Edição realizada por Opus Software Com. e Repr. Ltda, 2015. Citado 2 vezes nas páginas 20 e 21.
- SOUZA, A. F. de; NUNES, G. M. O.; BIANCHINI, T. H. *Sistema Supervisorio para Monitoramento de Consumo de Água*. Monografia (Trabalho de Concluso de Curso) — Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná, Curitiba, 2016. Citado 5 vezes nas páginas 2, 5, 6, 7 e 8.
- TANENBAUM, A. S. *Redes de Computadores*. [S.l.]: Vrije Universiteit, 2003. Citado 2 vezes nas páginas 19 e 20.

- TAURION, C. *Software Embarcado, A nova onda da Informática*. 1ª ed. ed. [S.l.: s.n.], 2005. Citado na página 15.
- TELES, A. S.; CALÇADA, D. B.; VERAS, N. D. L. *LIVRO DE MINICURSO-SERCEMAPI 2020*. [S.l.]: Secretaria Regional SBC Piauí, 2020. Citado na página 43.
- VENDEMIATTI, C. *Sistema Remoto para Monitoramento do Consumo de Água em Tempo Real*. Dissertação (Mestrado) — Universidade Estadual Paulista, Sorocaba - SP, 2020. Citado 2 vezes nas páginas 3 e 7.
- VERAS, M. *Arquitetura de Nuvem(AWS): Amazon WEB Services*. [S.l.]: Brasport Livros e Multimídia Ltda., 2013. Citado 2 vezes nas páginas 22 e 23.
- ZERYNTH. *Zerynth Enabling IoT*. 2015. Disponível em: <[https://olddocs.zerynth.com/r2.6.2/official/board.zerynth.doit\\_esp32/docs/index.html](https://olddocs.zerynth.com/r2.6.2/official/board.zerynth.doit_esp32/docs/index.html)>. Citado 2 vezes nas páginas 29 e 30.
- ZURITA, M. E. P. V. Projeto de sistemas embarcados. In: *Projeto de Sistemas Embarcados- ResearchGate*. Teresina, Piauí: [s.n.], 2007. p. 1–10. Citado na página 10.



## Documento Digitalizado Restrito

### Trabalho de Conclusão de Curso

**Assunto:** Trabalho de Conclusão de Curso  
**Assinado por:** Tiago Daltro  
**Tipo do Documento:** Anexo  
**Situação:** Finalizado  
**Nível de Acesso:** Restrito  
**Hipótese Legal:** Informação Pessoal (Art. 31 da Lei no 12.527/2011)  
**Tipo do Conferência:** Cópia Simples

Documento assinado eletronicamente por:

- **Tiago Daltro Duarte, ALUNO (20151610457) DE BACHARELADO EM ENGENHARIA ELÉTRICA - JOÃO PESSOA**, em 12/05/2023 20:43:44.

Este documento foi armazenado no SUAP em 12/05/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 829515  
Código de Autenticação: 44f9c2521e

