



**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA**

RODRIGO ALENCAR DE SOUZA

**DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA PARA
MONITORAMENTO CONTÍNUO DA URINA HUMANA**

Manaus
2022

RODRIGO ALENCAR DE SOUZA

**DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA PARA
MONITORAMENTO CONTÍNUO DA URINA HUMANA**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para a obtenção do título de Engenheiro em Eletrônica.



Orientador: Fábio de Sousa Cardoso, Prof. Dr.

Manaus
2022

*Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia – EST*

Reitor:

André Luiz Nunes Zogahib, Dr.

Vice-Reitor:

Kátia do Nascimento Couceiro, Dr.

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo, Me.

Coordenador do Curso de Engenharia Eletrônica:

Bruno da Gama Monteiro MSc.

Banca Avaliadora composta por:

Data da defesa: 21/10/2022.

Prof. Fábio de Sousa Cardoso, Dr. (Orientador)

Prof. Angilberto Muniz Ferreira Sobrinho, Dr. (Avaliador 1)

Prof. Bruno da Gama Monteiro MSc. (Avaliador 2)

CIP – Catalogação na Publicação

Souza, Rodrigo Alencar

Desenvolvimento de protótipo de sistema para monitoramento contínuo da urina humana / Rodrigo Alencar de Souza; [orientado por] Prof. Fábio de Sousa Cardoso – Manaus: 2022.
55 f. p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica). Universidade do Estado do Amazonas, 2022.

1. Urina humana, 2. Sensores.
3. ESP32. 4. Banco de dados. 5. Bluetooth.
- I. Sousa Cardoso, Fábio.

RODRIGO ALENCAR DE SOUZA

PROTÓTIPO DE SISTEMA PARA MONITORAMENTO CONTÍNUO DA URINA HUMANA

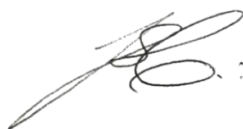
Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletrônico.

Nota obtida: 8,8 (Oito pontos e oito décimos)

Aprovada em: 21/10/2022.

Área de concentração: Sistemas Embarcados

BANCA EXAMINADORA



Orientador: Fábio de Sousa Cardoso, Dr.



Avaliador: Angilberto Muniz Ferreira Sobrinho, Dr.



Avaliador: Bruno da Gama Monteiro, Me.

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, aos meus pais e irmãos, por todo o suporte durante toda a trajetória na universidade, e a todos os meus familiares, amigos e professores que direta ou indiretamente me apoiaram nesta caminhada. Meu muito obrigado.

AGRADECIMENTOS

A Deus por me manter saudável e firme em meus objetivos. À minha família que sempre me apoiou e confiou em minhas escolhas. Aos meus amigos e a todos os professores por compartilharem seus conhecimentos e ensinamentos.

RESUMO

Este documento apresenta o projeto de desenvolvimento de um protótipo de sistema de para monitoramento contínuo da urina humana. Segundo a Sociedade Brasileira de Patologia Clínica 10 milhões de pessoas são afetadas pela disfunção renal e que mais de 70% dos pacientes que iniciam a diálise descobrem a doença quando os rins já estão gravemente comprometidos. Dentre os vários problemas que interferem na saúde humana, diversos deles estão relacionados à urina, como por exemplo, desidratação, infecções urinárias, diabetes, cálculos renais e etc. O presente trabalho busca a detecção de variações da urina através de uma plataforma ESP32 e de sensores (pH e reconhecimento de cores) acoplados nesta plataforma com o intuito de monitorar as características dessa urina e informar ao usuário através do smartphone conectado via Bluetooth com o ESP32 e indicar ao usuário se está tudo nos conformes ou não, assim como também enviar os valores captados pelos sensores para um banco de dados e com isso ser possível acompanhar a variação do que foi monitorado no decorrer do tempo. Com isso será possível prever uma complicação de saúde precocemente ou concomitantemente ao problema, auxiliando o usuário a procurar ajuda médica quando necessário.

Palavras-chaves: Urina humana, sensores, ESP32, Banco de dados, Bluetooth.

ABSTRACT

This document presents the project to develop a prototype system for continuous monitoring of human urine. According to the Brazilian Society of Clinical Pathology 10 million people are affected by renal dysfunction and that more than 70% of patients who start dialysis discover the disease when the kidneys are already severely compromised. Among the various problems that interfere with human health, several of them are related to urine, such as dehydration, urinary infections, diabetes, kidney stones and so on. The present work seeks the detection of urine variations through an ESP32 platform and sensors (pH and color recognition) coupled to this platform in order to monitor the characteristics of this urine and inform the user through the smartphone connected via Bluetooth with the ESP32 and indicate to the user if everything is in compliance or not, as well as send the values captured by the sensors to a database and with this it is possible to follow the variation of what was monitored over time. With this, it will be possible to predict a health complication early or concomitantly with the problem, helping the user to seek medical help when necessary.

Keywords: Human urine, Sensors, ESP32, Database, Bluetooth.

LISTA DE FIGURAS

Figura 1 - Relação cor da urina com a desidratação.	16
Figura 2 - Diagrama em blocos do ESP32	17
Figura 3 - Sensor de pH PH4502C	19
Figura 4 - Sensor de cores	20
Figura 5 - Conversor de nível lógico	21
Figura 6 - Conversor de nível lógico esquematizado.....	22
Figura 7- Símbolo do sistema Android.	23
Figura 8 - Exemplo de banco de dados	25
Figura 9 - Símbolo do MySQL.....	26
Figura 10 - Banco de dados MySQL.	27
Figura 11 - Diagrama Funcional do Protótipo.....	29
Figura 12 - Pagina de download Arduino IDE.....	30
Figura 13 - Tela Inicial do Fritzing.....	31
Figura 14 - Link para habilitar a biblioteca do ESP32 no IDE Arduino.....	32
Figura 15 - Biblioteca do ESP32 no IDE Arduino	32
Figura 16 - Circuito interno de alimentação do ESP32.....	33
Figura 17 - Pinagem GPIO do módulo ESP32.	33
Figura 18 - Esquemático das conexões – Sensores, Conversor Lógico e Módulo ESP32.....	34
Figura 19 - Erro de overflow de armazenamento.	35
Figura 20 - Calibragem do sensor de pH.....	35
Figura 21 - Tensão lida no multímetro digital após a calibração.	36
Figura 22 - Papel com as cores aproximadas das urinas	36
Figura 23 - Calibração do sensor de cores	37
Figura 24 - Valores das cores lidos pela entrada do ESP32.....	37
Figura 25 - Protótipo Montado.	38
Figura 26 - App utilizado para a comunicação Bluetooth.....	40
Figura 27 - Aba devices do aplicativo Serial Bluetooth Therminal.	41
Figura 28 - Página inicial do Clever Cloud após login.....	42
Figura 29 - Funcionalidades do site Clever Cloud.	42
Figura 30 - Credenciais do banco de dados.....	43
Figura 31 - Painel de configurações do banco de dados MySQL.....	43

Figura 32 - Menu de Configuração das variáveis do banco de dados.....	44
Figura 33 - Banco de dados com variáveis configuradas.....	44
Figura 34 - Amostras de urina.	45
Figura 35 - Resultado exame de urina (EAS) feito em laboratório..	46
Figura 36 - Visualização dos dados através do banco de dados.....	47
Figura 37 - Visualização dos dados no aplicativo Serial Bluetooth Terminal	47

SUMÁRIO

INTRODUÇÃO	13
1 REFERENCIAL TEÓRICO	15
1.1 URINA	15
1.1.1 <i>Cor</i>	15
1.1.2 <i>PH urinário</i>	15
1.2 DESIDRATAÇÃO.....	16
1.3 MICROCONTROLADOR.....	17
1.3.1 <i>Módulo ESP32</i>	17
1.4 WI-FI.....	18
1.5 BLUETOOTH.....	18
1.6 SENSORES	18
1.6.1 <i>Sensor de pH</i>	18
1.6.2 <i>Sensor de cores</i>	20
1.7 CONVERSOR DE NÍVEL LÓGICO	21
1.8 SISTEMA OPERACIONAL ANDROID	22
1.9 BANCO DE DADOS.....	23
1.9.1 <i>Aplicações de bancos de dados</i>	24
1.9.2 <i>MySQL</i>	25
2 MATERIAIS E MÉTODOS	28
3 IMPLEMENTAÇÃO DO PROJETO	30
3.1 INSTALAÇÃO DOS <i>SOFTWARES</i> UTILIZADOS	30
3.1.1 <i>Arduino IDE</i>	30
3.1.2 <i>Fritzing</i>	31
3.2 INSTALAÇÃO DAS BIBLIOTECAS NO IDE ARDUINO.....	32
3.3 IMPLEMENTAÇÃO DO DISPOSITIVO DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO.....	33
3.4 FIRMWARES DOS DISPOSITIVOS DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO	34
3.5 CALIBRAÇÃO DOS SENSORES.....	35
3.6 MONTAGEM DO PROTÓTIPO	38
3.6.1 <i>Seleção de Componentes e Sistemas</i>	38
3.6.2 <i>Aquisição de Materiais</i>	39
3.6.3 <i>Comissionamento</i>	39
3.7 CONFIGURAÇÃO DO SMARTPHONE PARA A COMUNICAÇÃO BLUETOOTH COM O ESP32	40
3.8 CONFIGURAÇÃO DO BANCO DE DADOS MYSQL NO CLEVER CLOUD	41
4 RESULTADOS OBTIDOS.....	45
4.1 AMOSTRAS.....	45
4.2 RESULTADO DO SISTEMA DE MONITORAMENTO.....	45
4.3 VISUALIZAÇÃO DOS DADOS OBTIDOS.....	47
CONCLUSÃO	48
REFERÊNCIAS	50

**APÊNDICE A - CÓDIGO USADO NO MICROCONTROLADOR PARA A
TRANSMISSÃO DE DADOS VIA BLUETOOTH..... 52**

**APÊNDICE B - CÓDIGO USADO NO MICROCONTROLADOR PARA A
TRANSMISSÃO DE DADOS VIA BLUETOOTH..... 55**

INTRODUÇÃO

A medicina laboratorial teve sua origem a partir de uma análise de urina; há referências sobre urina nos desenhos feitos por nossos primeiros ancestrais e em hieróglifos egípcios. Os médicos da antiguidade baseavam-se, na maioria das vezes, apenas na análise da urina do paciente para obter um diagnóstico; este era baseado na observação da turvação, odor, volume, cor e até presença ou não de açúcar na urina (STRASINGER, 2000).

Dados da Sociedade Brasileira de Nefrologia indicam que o número de doentes renais no Brasil dobrou na última década. Estima-se que 10 milhões de brasileiros sofram de alguma disfunção renal. Atualmente, entre 90 mil e 100 mil pessoas passam por diálise no país. Além disso, 70% dos pacientes que iniciam a diálise descobrem a doença quando os rins já estão gravemente comprometidos. Sem um diagnóstico preciso, a maioria dos pacientes morre sem sequer ter acesso à diálise (NORONHA, 2013).

A medicina e a tecnologia são áreas muito interligadas, assim como a tecnologia vai evoluindo com o tempo, os equipamentos médicos, métodos, análises vão acompanhando essas mudanças, aumentando a capacidade de detecção de problemas relacionados à saúde e a eficácia dos processos, tais como, processos de teste, análise, divulgação do resultado, etc.

A evolução dos microcontroladores tornou possível um aumento significativo em seu processamento dando mais potência ao microcontrolador associado a um tamanho mais reduzido que em microcontroladores mais antigos, além disso, podemos citar o fato de estes dispositivos se tornarem mais conectáveis com outros tipos de dispositivos eletrônicos, graças as tecnologias atuais como o Bluetooth, Wi-Fi, como também sendo capaz de suportar mais equipamentos de entrada.

Este projeto de pesquisa tem como objetivo final propor um protótipo de sistema que monitore a urina humana, sendo composto pelos sensores de pH e de cores, conectados à plataforma de *hardware* ESP32, que realizará a aquisição e processamento dos dados referentes a essas características presentes na urina e esses dados serão informados ao usuário através de um aplicativo no smartphone Android, como também, em uma aplicação WEB se está tudo em conformidade com sua urina e acompanhar como andam estes parâmetros no decorrer do tempo, detectando, por exemplo, uma desidratação que com o decorrer do tempo pode se tornar algo mais grave.

Este trabalho se justifica que pelo uso do sistema será possível monitorar de forma contínua as características da urina indicando possíveis problemas de imediato, como uma

possível desidratação ou infecção, assim auxiliando o usuário do sistema a evitar condições que podem trazer ou agravar problemas de saúde com o passar do tempo.

O trabalho abordará conhecimentos aprendidos em diversas disciplinas que foram ministradas no decorrer do curso de Engenharia Eletrônica, as quais compreendem: Construção Eletrônica, Sensores e Instrumentação Eletrônica, Microprocessadores e Microcontroladores, Projeto de Sistemas Microprocessados e Processamento Digital de Sinais.

O presente trabalho se divide em 4 capítulos: Referencial teórico, Métodos e Materiais, implementação do projeto e resultados obtidos. No primeiro capítulo, apresenta uma revisão bibliográfica sobre os principais conceitos e tecnologias que serviram de base para o desenvolvimento do projeto. Neste capítulo, são explicados os parâmetros a serem medidos, além de uma explicação detalhada do microcontrolador ESP32 e do conversor de nível lógico. Como também os protocolos de comunicação utilizados que são Bluetooth e Wi-fi e os *softwares* utilizados no desenvolvimento dos códigos.

No segundo capítulo são mostrados os métodos utilizados para se obter o resultado sendo dividido em 7 etapas (instalação dos *softwares* utilizados, instalação das bibliotecas no IDE Arduino, *firmware* do dispositivo de aquisição de dados e comunicação, calibração dos sensores, montagem do protótipo, configuração do smartphone para a comunicação *Bluetooth* com o ESP32, configuração do banco de dados MySQL no Clever Cloud) utilizadas para o desenvolvimento do protótipo.

No terceiro capítulo, implementação do projeto, é abordada a execução das etapas citadas no capítulo de métodos. Mostra-se a preparação do ambiente de desenvolvimento, o desenvolvimento do *firmware*, a confecção do protótipo e todos os testes executados, , configuração do banco de dados e aplicativo Bluetooth no smartphone Android.

No quarto capítulo, resultados obtidos, mostram-se os resultados dos experimentos realizados na implementação, realizou-se análises baseadas no referencial teórico e na comparação com resultados válidos, como também análises de confiabilidade e integridade das transmissões.

E por fim é apresentada a conclusão que retorna ao que foi proposto mostrando que o protótipo consegue realizar com eficácia a aquisição dos dados, o monitoramento e a transmissão, atendendo assim às exigências do que foi proposto anteriormente.

1 REFERÊNCIAL TEÓRICO

1.1 URINA

A urina é um fluido biológico formado a partir da depuração sanguínea pelos rins a através da filtração glomerular; sua fácil obtenção permite a investigação de muitas funções metabólicas do organismo; trata-se, na realidade de um ultrafiltrado do plasma, do qual foram reabsorvidos glicose, aminoácidos, água e outras substâncias essenciais ao metabolismo do organismo. É constituída por uréia e outras substâncias químicas orgânicas e inorgânicas dissolvidas em água. Podem ocorrer grandes variações na concentração dessas substâncias, devido à influências de fatores como ingestão alimentar, atividade física, metabolismo orgânico, função endócrina e até mesmo posição do corpo (STRASINGER, 2000) (GUYTON, 2008).

1.1.1 Cor

A urina normal é tipicamente transparente e amarela à inspeção visual. Amostras de urina concentrada podem apresentar uma coloração âmbar, ao passo que amostras diluídas se mostram amarelo claras. No entanto, a coloração da urina nunca deve ser utilizada para a determinação da gravidade específica. Existem inúmeros pigmentos endógenos e exógenos que podem levar a uma coloração anormal da urina, sendo as cores mais observadas a vermelha, a marrom e a preta, em casos de hematúria, hemoglobinúria e mioglobinúria, além da bilirrubina (DALMOLIN, 2011).

1.1.2 PH urinário

O pH urinário não reflete necessariamente o pH sanguíneo, e pode ser influenciado pela dieta, alimentação recente, infecção bacteriana e tempo de armazenamento/retenção. Estudos demonstram que as tiras reagentes para detecção do pH não são acuradas em comparação aos pHmetros. Para manter o pH sanguíneo constante, entre 7,35 e 7,45 haverá alteração do pH urinário. Animais que ingerem carne e cereais apresentam o pH urinário entre 5,5 e 7,0 devido à presença de fosfatos de sódio e cálcio, enquanto que herbívoros, com dieta mais alcalina, apresentam pH urinário fisiológico entre 7,0 e 8,5 devido à presença de bicarbonato de cálcio solúvel. Existem várias causas de urina alcalina. A alimentação pode resultar em uma urina alcalina transitória, que devido à secreção de ácido gástrico causa uma

relativa alcalose. A infecção do trato urinário com bactérias produtoras de urease (*Proteus sp.* e *Staphylococcus sp.*, por exemplo) pode acarretar na conversão de ureia em amônia, alcalinizando a urina. Atraso de processamento, má conservação da amostra também constituem causas de urina com elevado pH. Da mesma forma, uma redução do pH urinário possui várias causas (DALMOLIN, 2011).

1.2 DESIDRATAÇÃO

A desidratação ocorre quando o corpo usa ou perde mais líquido do que o ingerido causando dificuldades para o organismo realizar suas funções normais. A cor e clareza de urina conforme figura 1, a densidade da urina e a presença de cetonas na urina podem ajudar a indicar o grau de desidratação. Dentre as complicações causadas pela desidratação pode-se citar: convulsões, edema cerebral, choque por baixo volume de sangue, insuficiência renal que pode ser potencialmente fatal ocorre quando os rins já não são capazes de remover o excesso de líquidos e resíduos do sangue e etc. (MINHAVIDA, 2021).

Figura 1 – Relação cor da urina com a desidratação.



Fonte: (PRÓ-RIM, 2019, p1.).

1.3 MICROCONTROLADOR

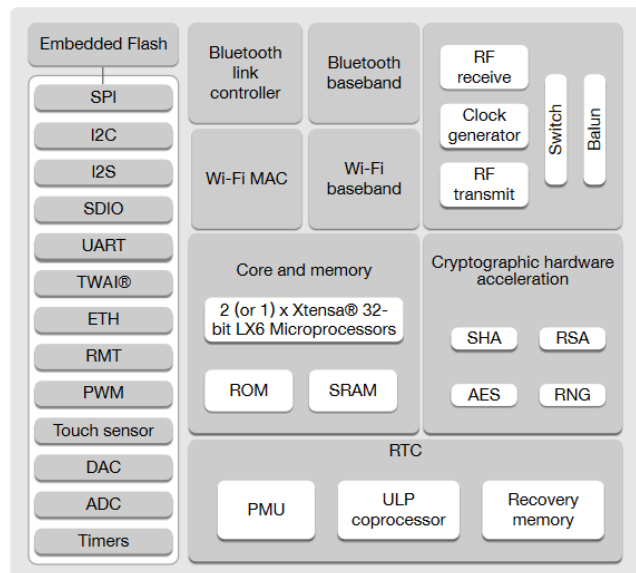
Um microcontrolador é um componente que tem, num único chip, além de uma *CPU*, elementos, tais como, memórias *ROM* e *RAM*, temporizadores, contadores, canais de comunicação e conversores analógico-digitais (SILVA JÚNIOR, 1988).

1.3.1 Módulo ESP32

O ESP32 é um SoC (*System on Chip*) que oferece conectividade (Wi-Fi, Bluetooth, com frequência de 2,4 GHz), poder computacional (*CPU* + memórias), Entradas/Saídas (*I/Os*), Relógio em tempo real (*RTC*), suporte à diversas comunicações (*I2C*, *I2S*, *SPI* e etc.), como também suporte à operação *Low Power* e blocos de *hardware* dedicados à segurança (BERTOLETI, 2019).

O ESP32 por conter módulo Wi-Fi e o módulo Bluetooth já embarcados ao seu chip é uma opção poderosa e completa para o objetivo deste trabalho. A figura 2 detalha as funcionalidades do ESP32.

Figura 2 – Diagrama em blocos do ESP32.



Fonte: (ESPRESSIF SYSTEMS, 2021, p.12).

1.4 WI-FI

É uma rede de comunicação sem fio lançada em 1997 baseada no padrões de transmissão e codificação IEEE 802.11, sendo bastante popular, pois está presente nos mais diversos lugares, fazendo parte do cotidiano de casas, escritórios, indústrias, lojas comerciais e até espaços públicos das cidades. O Wi-Fi possui algumas vantagens, como alcance de conexão e vazão, o que o torna adequado para navegação na Internet em dispositivos móveis, como smartphones e tablets. A principal desvantagem do Wi-Fi é o maior consumo de energia, quando comparado com outras tecnologias de comunicação sem fio (SANTOS *et al.*, 2016).

1.5 BLUETOOTH

O Bluetooth é o protocolo de comunicação proposto pela Ericsson para substituir a comunicação RS-232 e é uma das principais tecnologias de rede sem fio para *PANs – Personal Area Networks*, que é utilizada em smartphones, headsets, PCs e outros dispositivos. O Bluetooth se divide em dois grupos: Bluetooth clássico e o *Bluetooth Low Energy (BLE)*, versão 4.0 ou superior. As versões mais antigas do Bluetooth foram focadas em aumentar a taxa de comunicação tornando o protocolo mais complexo e, por consequência, não otimizado para dispositivos com limitações energéticas. Ao contrário das versões anteriores, o *BLE* possui especificação voltada para baixo consumo de energia, permitindo dispositivos que usam baterias do tamanho de moedas (SANTOS *et al.*, 2016).

1.6 SENSORES

A palavra sensor pode ser definida como “aquilo que sente”. Na eletrônica, um sensor é conhecido como qualquer componente ou circuito eletrônico que permita a análise de uma determinada condição do ambiente, podendo ela ser algo simples como temperatura ou luminosidade; uma medida um pouco mais complexa como a rotação de um motor ou a distância de um carro até algum obstáculo próximo (PATSKO, 2006).

1.6.1 Sensor de pH

O pH, potencial hidrogeniônico, é uma escala logarítmica que indica a acidez ou alcalinidade de uma solução aquosa em uma escala de 0 a 14. Ele pode ser medido de diversas formas como através de papel de tornassol, com o uso de indicadores químicos (métodos

colorimétricos) e através de pHmetros (medidores de pH). O sensor de pH que consiste em um eletrodo para mergulhar na solução em que se deseja saber o pH e um módulo eletrônico de leitura (PH-4502C) conforme figura 3 que se conecta à porta de entrada e envia os dados da leitura (LOPES, 2019).

Figura 3 – Sensor de pH.



Fonte: (AUTOCORE ROBÓTICA, 2021, p.1).

Especificações:

Tensão de aquecimento: $5 \pm 0.2V$ (AC/DC);

Corrente de trabalho: 5-10mA;

Faixa de temperatura: 0~60°C;

Tempo de resposta: 5s;

Tempo de sedimentação: 60s;

Saída: Analógica;

Faixa de medição: 0,00 ~ 14,00 pH;

Zero pontos: 7 ± 0.5 ph;

Erro alcalino: 0.2pH;

Temperatura de Operação: -10~50°C (Temperatura Nominal 20°C);

Comprimento do cabo: 90cm;

Conector: Plugue BNC;

Dimensões: 66mm Largura x 33mm Profundidade x 20mm Altura (Módulo);

Tamanho: 160mm Largura x 26mm Profundidade x 26mm Altura (Sensor);

Peso: 82g (AUTOCORE ROBÓTICA, 2021).

1.6.2 Sensor de cores

O sensor de cores (figura 4) funciona emitindo uma luz branca que atinge a superfície analisada e retorna de forma refletida. Este reflexo é processado pelo sensor que gera um padrão RGB (*Red, Green, Blue*) ou vermelho, verde e azul, caso o valor esteja na variação que predefinido na memória do sensor, ele emite uma combinação de sinais discretos ou um sinal via comunicação informando que o objeto analisado está dentro da especificação de cor (SILVEIRA, 2016).

Figura 4 – Sensor de cores.



Fonte: (ELETROGATE, 2021, p.1).

Especificações:

Alta resolução para conversão de intensidade de luz para frequência;

Frequência de saída programável;

Comunica-se diretamente com um microcontrolador;

Tensão de operação: 2,7V a 5,5V;

Função Power Down;

Margem de erro típica: 0,2% a 50kHz;

Coefficiente de temperatura estável 200 ppm /° C (ELETROGATE, 2021).

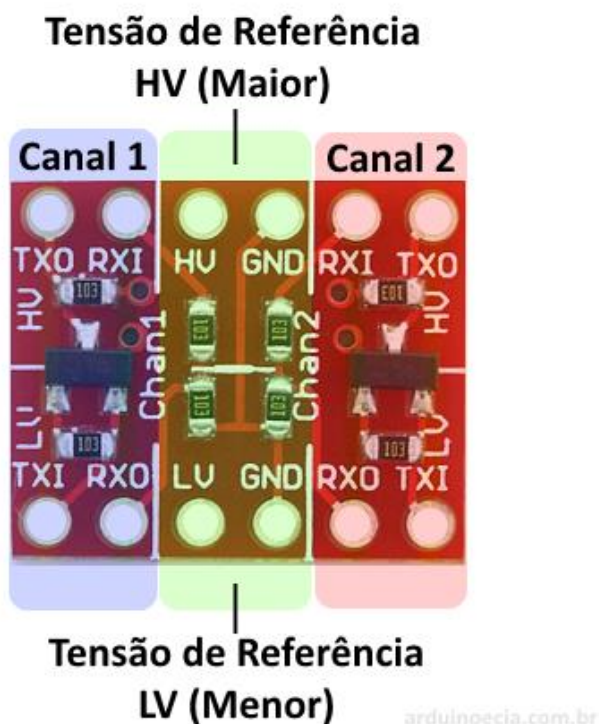
1.7 CONVERSOR DE NÍVEL LÓGICO

O Conversor de Nível Lógico (figura 5) é um módulo eletrônico de grande utilidade e aplicabilidade na montagem de projetos que envolvem microcontroladores, já que muitas vezes a tensão de nível lógico vem a ser diferente da oferecida pelas portas do Arduino, por exemplo. É um pequeno módulo capaz de diminuir sinais de 5V para 3,3V ou elevar tensões de nível lógico de 3,3V para 5V, e isso de forma totalmente segura, possuindo a capacidade de trabalhar com tensões menores, entre 2,8V e 1,8V. Por ser bidirecional, o Conversor de Nível Lógico conta com dois pares de pinos de nível elevado para o rebaixado, e vice-versa, possuindo duas entradas e saídas em cada lado, o que possibilita trabalhar de forma independente.

A placa apresenta internamente dois circuitos: um divisor de tensão comum, que vai diminuir a tensão de 5V para 3,3V, e um circuito baseado em MOSFET, que vai fazer a conversão em ambos os sentidos (menor para o maior e maior para o menor).

Temos uma disposição dos dois canais nas laterais da placa, e na parte central temos os conectores de tensão de referência, cuja ligação é obrigatória para que a placa funcione. A figura 6 demonstra o funcionamento do dispositivo de forma esquematizada (USAINFO, 2022).

Figura 5 – Conversor de nível lógico



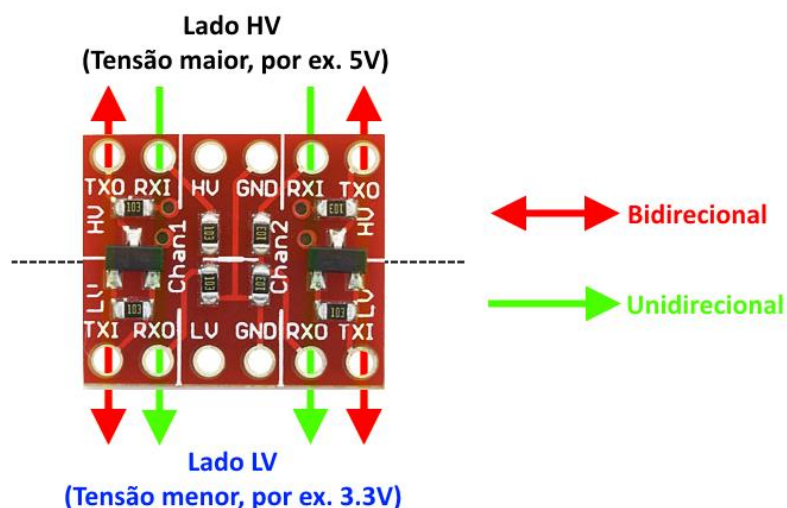
Fonte: (ARDUINOECIA, 2015, p1.).

Em cada canal temos 2 entradas e 2 saídas, que funcionam da seguinte maneira:

- RXI e RXO (RX *Input* e RX *Output*) – Pinos do divisor de tensão. Esses pinos convertem o sinal de 5V para 3,3V, e não funcionam no sentido inverso.

- TXI e TXO (TX *Input* e TX *Output*) – Pinos do circuito MOSFET. Esses pinos convertem o sinal de 5V para 3,3V e de 3,3V para 5V, se necessário (ARDUINOECIA, 2015).

Figura 6 – Conversor de nível lógico esquematizado.



arduinoecia.com.br

Fonte: (ARDUINOECIA, 2015, p1.).

1.8 SISTEMA OPERACIONAL ANDROID

O sistema Android (símbolo exibido na figura 7) foi desenvolvido pela Android, Inc., a qual foi adquirida pelo Google em 2005. Em 2007, foi formada a Open Handset Alliance™ para desenvolver, manter e evoluir o Android, impulsionando a inovação na tecnologia móvel, melhorando a experiência do usuário e reduzindo os custos. É um sistema operacional baseado no kernel do Linux. Apesar de ter sido desenvolvido inicialmente para smartphones, hoje é usado em diversas outras aplicações como tablets, netbooks, relógios, etc.

Mesmo baseado no kernel do Linux, existe pouca coisa em comum com distribuições Linux convencionais (embarcadas ou não), lembrando que um sistema embarcado (ou sistema embutido) é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. O Android é uma máquina virtual Java rodando sobre o kernel do Linux, dando suporte para o desenvolvimento de aplicações Java através de um conjunto de bibliotecas e serviços.

Muitas características presentes no Android, tais como gráficos 3D e suporte a banco de dados nativo, também estão disponíveis em outras plataformas móveis. Porém, apenas no

Android há um componente que permite exibir e manipular um mapa do Google Maps, serviço de mapas do Google, dentro de uma aplicação. Somente no Android todos os aplicativos são criados igualmente. Ou seja, nele não há distinção entre aplicativos que são nativos e os demais. Isso possibilita uma grande customização do sistema operacional, permitindo a substituição completa de aplicativos nativos por outros, criados por terceiros. Além disto, todos os aplicativos têm acesso as mesmas funcionalidades.

Atualmente, os dispositivos Android incluem smartphones, tablets, e-readers, robôs, motores a jato, satélites da NASA, consoles de jogos, geladeiras, televisões, câmeras, equipamentos voltados à saúde, relógios inteligentes (*smartwatches*), sistemas automotivos de “*infotainment*” de bordo para controlar rádio, GPS, ligações telefônicas, termostato, etc. (FERNANDES, 2012).

Figura 7 – Símbolo do sistema Android.



Fonte: (LOGOMARCAS, 2022, p1.).

1.9 BANCO DE DADOS

Um banco de dados (BD) é uma coleção de dados relacionados. Por dados, entende-se fatos conhecidos que podem ser armazenados e têm um significado implícito. Por exemplo, considere os nomes, telefones e endereços das pessoas que você conhece. Atualmente, esses dados estão armazenados nos celulares, que têm o seu próprio software para gerenciar esses dados. Um banco de dados tem as seguintes propriedades implícitas:

- representa algum aspecto do mundo real;
- é uma coleção de dados logicamente coerente;
- é projetado, construído e populado para um propósito específico.

Bancos de dados podem ser de qualquer tamanho ou complexidade. Por exemplo, a lista de contatos mencionada anteriormente pode conter algumas centenas de registros, cada qual com uma estrutura simples. Em contrapartida, um exemplo de banco de dados grande e complexo é o da Amazon.com, que contém dados de 60 milhões de usuários ativos, milhões de livros, jogos, aparelhos eletrônicos e outros itens. Esse banco de dados ocupa mais de 42 terabytes e é armazenado em centenas de computadores.

Um sistema de gerenciamento de banco, bancos de dados, (ou bases de dados), são conjuntos de dados com uma estrutura regular que organizam informação. Um banco de dados normalmente agrupa informações utilizadas para um mesmo fim.

Um banco de dados é usualmente mantido e acessado por meio de um software conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Muitas vezes o termo banco de dados é usado como sinônimo de SGDB.

O modelo de dados mais adotado hoje em dia é o modelo relacional, onde as estruturas têm a forma de tabelas, compostas por linhas e colunas.

Resumindo, um banco de dados é uma coleção de dados relacionados. Entende-se por dado, toda a informação que pode ser armazenada e que apresenta algum significado implícito dentro do contexto ao qual ele se aplica. Por exemplo, num sistema bancário, uma pessoa é identificada pelo seu CPF (cliente). Em um sistema escolar a pessoa é identificada pelo seu número de matrícula(aluno). Além disso, os dados que serão armazenados em cada situação podem diferir consideravelmente (BRASIL, [s.d.]).

1.9.1 Aplicações de bancos de dados

Bancos de dados são usados em muitas aplicações, enquanto atravessando virtualmente a gama inteira de software de computador. Bancos de dados são o método preferido de armazenamento para aplicações multiusuárias grandes ao qual a coordenação entre muitos usuários é necessária. Na figura 8 há um exemplo de banco de dados real.

Um Aplicativo de Banco de dados é um tipo de software exclusivo para gerenciar um banco de dados. Aplicativos de banco de dados abrangem uma vasta variedade de necessidades e objetivos, de pequenas ferramentas como uma agenda, até complexos sistemas empresariais para desempenhar tarefas como a contabilidade. O termo "Aplicativo de Banco de dados" usualmente se refere a softwares que oferecem uma interface para o banco de dados. O software que gerencia os dados é geralmente chamado de sistema gerenciador de banco de dados (SGBD) ou (se for embarcado) de "*database engine*". Exemplos de aplicativos de banco de dados são

Microsoft Visual FoxPro, Microsoft Access, dBASE, FileMaker , HyperCard, MySQL, PostgreSQL (BRASIL, [s.d.]).

Figura 8 – Exemplo de banco de dados

Exemplo de um Banco de Dados

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

Fonte: (NETO, 2008, p1.).

1.9.2 MySQL

O MySQL (símbolo exibido na figura 9) foi originalmente desenvolvido pela empresa sueca TCX, que necessitava de um servidor de banco de dados que operasse com grandes escalas de dados rapidamente sem exigir caríssimas plataformas de hardware. No início eles utilizavam o mSQL, mas depois de alguns testes chegaram à conclusão que o mSQL não era rápido nem flexível o suficiente para as necessidades existentes. A figura 10 mostra um exemplo de um banco de dados MySQL.

As principais características do MySQL são:

- O servidor de banco de dados MySQL é extremamente rápido, confiável, e fácil de usar. O Servidor MySQL também tem um conjunto de recursos muito práticos desenvolvidos com a cooperação dos próprios usuários.

- O Servidor MySQL foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes, e tem sido usado em ambientes de produção de alta demanda por vários anos de maneira bem sucedida. Apesar de estar em constante desenvolvimento, o Servidor MySQL oferece hoje um rico e proveitoso conjunto de funções. A conectividade, velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

- MySQL é um Sistema de Gerenciamento de Bancos de Dados relacional. Um banco de dados relacional armazena dados em tabelas separadas em vez de colocar todos os dados em um só local. Isso proporciona velocidade e flexibilidade. SQL é a linguagem padrão mais comum usada para acessar bancos de dados e é definida pelo Padrão ANSI/ISO SQL. (O padrão SQL vem evoluindo desde 1986 e existem diversas versões disponibilizadas).

- MySQL é um software cujo código fonte é aberto. Código fonte aberto significa dizer que é possível para qualquer um usar e modificar o programa. Qualquer pessoa pode fazer download do MySQL pela Internet e usá-lo sem pagar nada (o MySQL só é cobrado em alguns poucos casos). Se você quiser pode estudar o código fonte e alterá-lo para adequá-lo às suas necessidades. O MySQL usa a GPL (*GNU General Public License*).

- Licença Pública Geral GNU) <http://www.fsf.org/licenses>, para definir o que você pode e não pode fazer com o software em diferentes situações. Se você sentir desconforto com a GPL ou precisar embutir o MySQL em uma aplicação comercial você pode adquirir a versão comercial licenciada.

- O MySQL suporta diferentes plataformas, tais como: Windows, Linux, FreeBSD, Unix, entre outros.

- O MySQL possui suporte a múltiplos processadores. O Programa de Banco de Dados MySQL é um sistema cliente/ servidor que consiste de um servidor SQL multi-tarefa que suporta acessos diferentes, diversos programas clientes e bibliotecas, ferramentas administrativas e *API's* que são interfaces de programação (QUESADA et al., 2001).

Figura 9 – Símbolo do MySQL.



Fonte: (PISA, 2012, p1.).

Figura 10 – Banco de dados MySQL.

The screenshot displays the phpMyAdmin interface for a MySQL database named 'test'. The left sidebar shows a tree view of the database structure, including tables like 'wp_commentmeta', 'wp_comments', and 'wp_gmp_icons'. The main area shows a list of tables with columns for 'Table', 'Action', 'Rows', 'Type', and 'Collation'. A blue arrow points to the 'Filters' section above the table list.

Table	Action	Rows	Type	Collation
wp_commentmeta	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode
wp_comments	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode
wp_gmp_icons	★ Browse Structure Search Insert Empty Drop	47	InnoDB	utf8_general_ci
wp_gmp_maps	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
wp_gmp_markers	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
wp_gmp_marker_groups	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
wp_gmp_marker_groups_relation	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
wp_gmp_membership_presets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
wp_gmp_modules	★ Browse Structure Search Insert Empty Drop	14	InnoDB	utf8_general_ci
wp_gmp_modules_type	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci
wp_gmp_options	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci
wp_gmp_options_categories	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci
wp_gmp_usage_stat	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
wp_links	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode
wp_options	★ Browse Structure Search Insert Empty Drop	385	InnoDB	utf8mb4_unicode
wp_postmeta	★ Browse Structure Search Insert Empty Drop	445	InnoDB	utf8mb4_unicode
wp_posts	★ Browse Structure Search Insert Empty Drop	98	InnoDB	utf8mb4_unicode
wp_sm_sessions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode
wp_ta_link_clicks	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode
wp_term_taxonomy	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode

Fonte: (KINSTA, 2022, p1.).

2 MATERIAIS E MÉTODOS

Neste capítulo é mostrado o desenvolvimento do projeto desde a pesquisa teórica em conjunto com os testes iniciais dos sensores em laboratório, passando pela calibração dos sensores e montagem do protótipo, realização da comunicação do ESP32 com os protocolos Bluetooth e Wi-Fi, e por último a criação do sistema e sua integração, assim estabelecendo a interface do sistema de monitoramento.

O sistema é monitorado através de dois sensores, são eles: sensor de cores TSC-230 e sensor de pH PH4502C. Os mesmos serão acoplados ao módulo ESP32 através de fios e o sensor de pH será acoplado também a um conversor lógico, os sensores entrarão em contato com a urina, para que os parâmetros possam ser medidos. Em seguida, os valores de leitura são injetados nas entradas analógicas de um ESP32 que realizará a aquisição dos dados. Posteriormente esses dados serão transmitidos pelo ESP32 através dos protocolos Bluetooth e Wi-fi, sendo recebidos por uma aplicação Bluetooth, como também em um banco de dados.

Os valores discretos dos parâmetros serão exibidos através da tela de um smartphone e em um banco de dados MySQL. O sistema foi desenvolvido nas seguintes etapas.

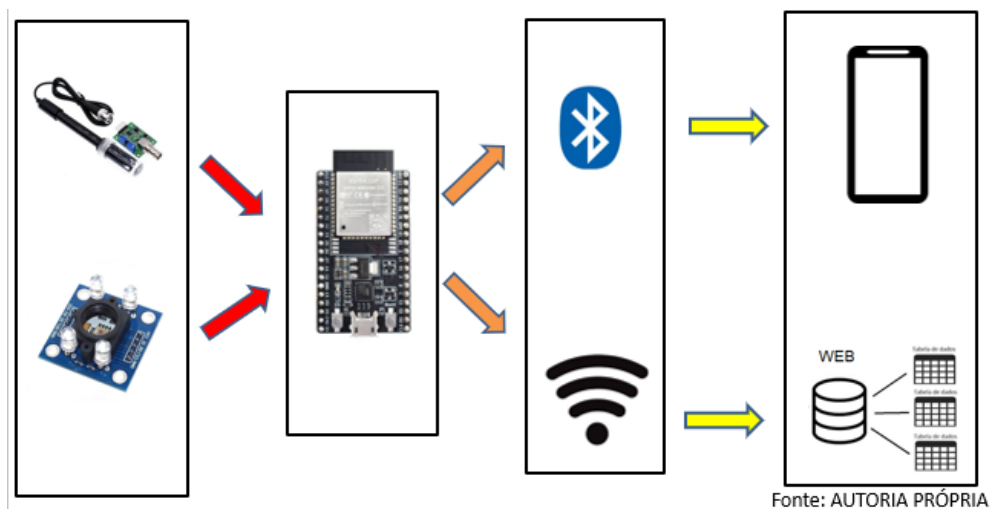
Na primeira etapa, foi realizada uma revisão de literatura nas áreas de microcontroladores e microprocessadores, processamento de sinais, química, fisiologia, sensores, redes sem fio, armazenamento e criação de banco de dados com PHP, SQL e HTML.

Na segunda etapa ocorreu a aquisição dos componentes e montagem do protótipo utilizando a placa de desenvolvimento do ESP32 ligada diretamente ao módulo que conterá os sensores de pH (PH-4502C) e de cores (TSC230) para a configuração inicial, desenvolvimento do *firmware* a ser embarcado no microcontrolador e testes dos sensores para averiguar o correto funcionamento dos equipamentos.

Na terceira etapa aconteceu a configuração de uma aplicação em um smartphone Android, como também a configuração de uma aplicação *WEB* com banco de dados para o monitoramento dos dados captados pelos sensores processados pelo módulo microcontrolado ESP32 e transmitidos via Bluetooth e Wi-Fi.

Na quarta e última etapa realizou-se o acoplamento da placa de desenvolvimento ESP32 juntamente com os sensores em uma simulação do objeto alvo no laboratório da Escola Superior de Tecnologia (EST), que é uma unidade da Universidade Estadual do Amazonas, os dados serão coletados, analisados e validados para averiguar a eficácia do protótipo, como também interpretação dos resultados finais. A figura 11 mostra as entradas, processamento e saída dos dados.

Figura 11 – Diagrama Funcional do Protótipo.



Fonte: Autoria própria.

Durante a montagem do protótipo de sistema de monitoramento foram utilizados os materiais listados abaixo:

20 *jumper wires* fêmea-fêmea 12cm.

1 módulo ESP32.

1 sensor de cores TSC-230.

1 sensor de pH PH4502C.

1 conversor lógico de 5V para 3.3V de 8 canais.

1 cabo micro-USB.

3 IMPLEMENTAÇÃO DO PROJETO

Este capítulo apresenta o processo de desenvolvimento do projeto. Foram aplicadas diversas tecnologias e utilizadas diferentes plataformas de programação.

São apresentados neste capítulo os seguintes tópicos:

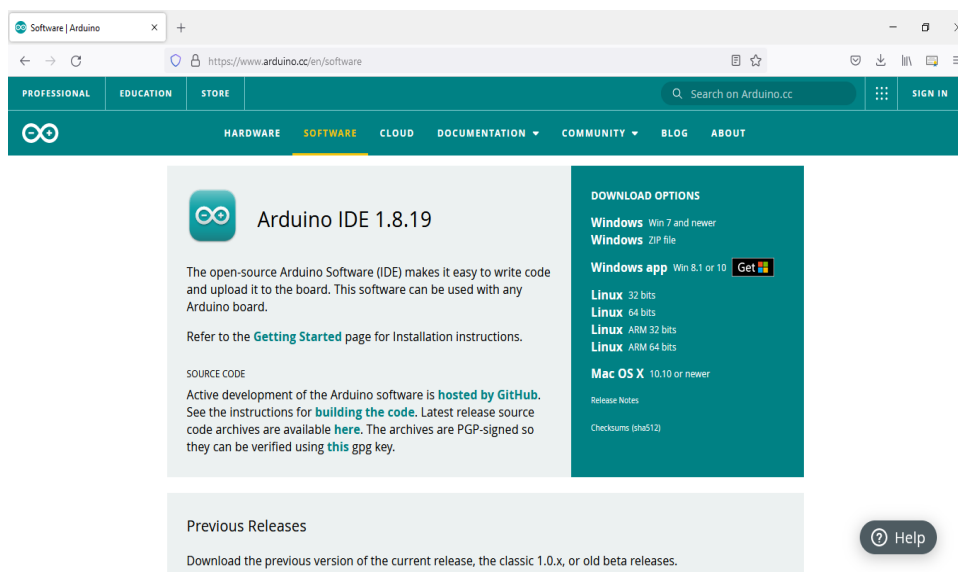
- a. instalação dos *softwares* utilizados;
- b. instalação das bibliotecas no IDE Arduino;
- c. *firmware* do dispositivo de aquisição de dados e comunicação;
- d. calibração dos sensores;
- e. montagem do protótipo;
- f. configuração do smartphone para a comunicação *Bluetooth* com o ESP32;
- g. configuração do banco de dados MySQL no Clever Cloud.

3.1 INSTALAÇÃO DOS *SOFTWARES* UTILIZADOS

3.1.1 Arduino IDE

Para a implementação do projeto foi necessário escrever um código na plataforma Arduino que utiliza a linguagem C e C++ e tendo compatibilidade com o microcontrolador ESP32. O *download* da versão 1.8.19 pode ser realizado pela URL: <https://www.arduino.cc/en/software> (figura 12).

Figura 12 – Pagina de download Arduino IDE.



Fonte: Autoria própria.

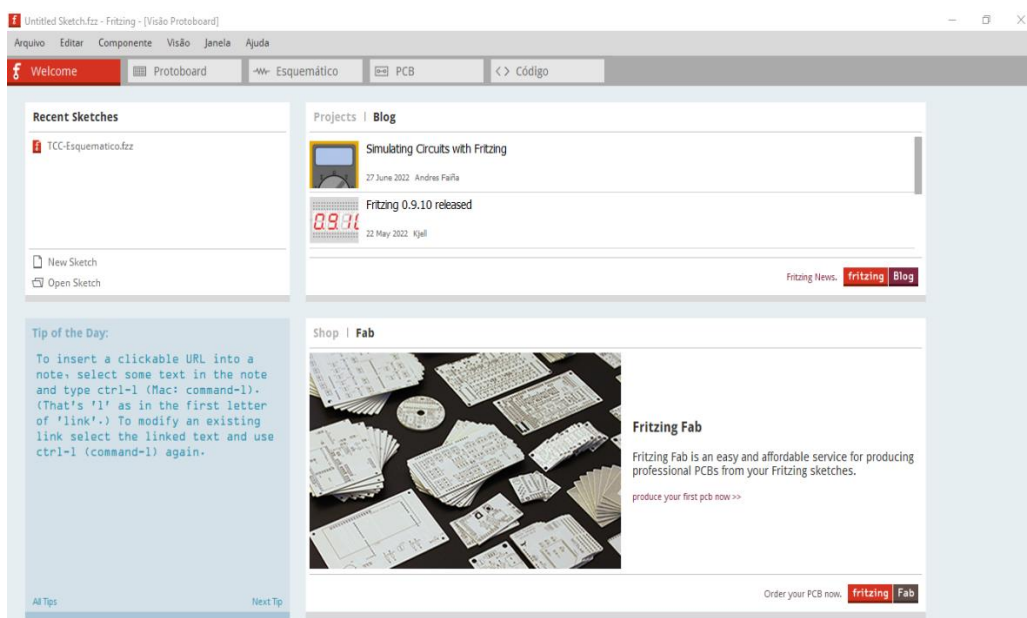
O *download* e a instalação são processos bem simples de serem realizados. Seguindo somente os passos da tela que irão ser mostrados durante a instalação, sendo possível concluí-la com sucesso. Após estes procedimentos de *download* e instalação, é possível ter acesso ao ambiente de desenvolvimento. Após a instalação é necessário realizar a configuração para a compatibilidade com o microcontrolador ESP32.

3.1.2 Fritzing

É possível fazer o *download* do Fritzing pela URL (<https://fritzing.org/download/>), para qualquer sistema operacional, seja Windows, Mac ou Linux. Os procedimentos a seguir são para instalação do *software*:

Para instalar o Fritzing, executa-se o arquivo de instalação baixado, seguindo as instruções apresentadas. O instalador pode solicitar direitos do administrador ("UAC") durante a instalação. Deve-se confirmar esta solicitação clicando em "sim". Isso permitirá que o instalador baixe bibliotecas adicionais da Microsoft. A figura 13 demonstra o Fritzing já instalado.

Figura 13 – Tela Inicial do Fritzing.



Fonte: Autoria própria.

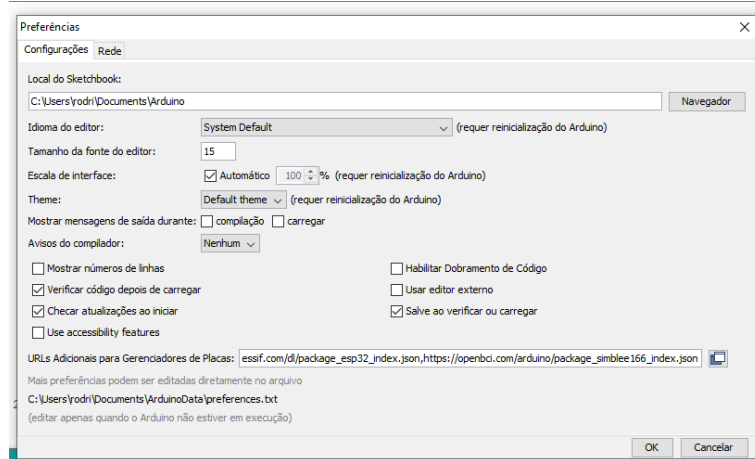
3.2 INSTALAÇÃO DAS BIBLIOTECAS NO IDE ARDUINO

Para se ter acesso a funções e códigos específicos é necessário a instalação de bibliotecas que facilitam e tornam mais completa o Arduino IDE para se instalar a biblioteca para habilitar o uso do ESP32 com suas próprias funcionalidades e código é preciso utilizar um link na aba arquivos, e logo após na aba preferências dentro do Arduino IDE que permitirá que a Arduino IDE disponibilize a utilização do ESP32 dentro do seu gerenciamento de bibliotecas conforme figura 14.

Link utilizado:

<https://dl.espressif.com/dl/package_esp32_index.json,https://openbci.com/arduino/package_simblee166_index.json>.

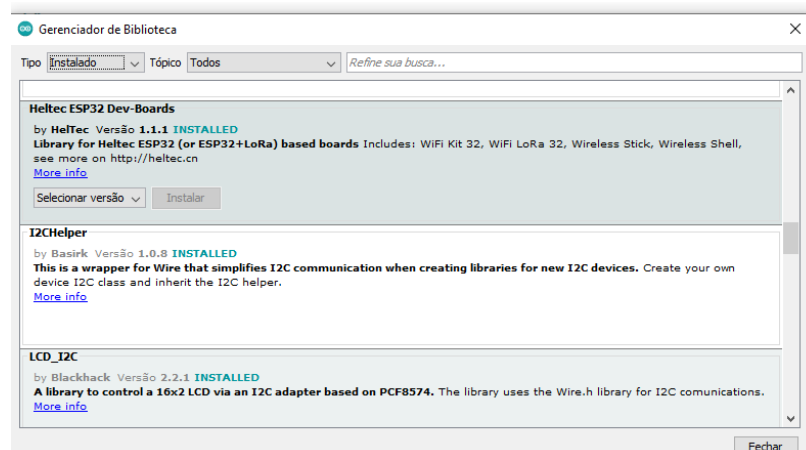
Figura 14 – Link para habilitar a biblioteca do ESP32 no IDE Arduino.



Fonte: Autoria própria.

Após entrar na tela de gerenciar bibliotecas, basta somente procurar pela biblioteca ESP32 e clicar no ícone instalar conforme figura 15.

Figura 15 – Biblioteca do ESP32 no IDE Arduino.

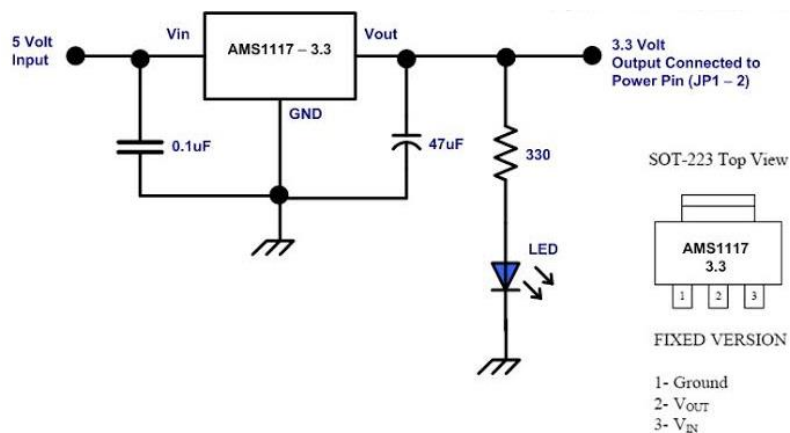


Fonte: Autoria própria.

3.3 IMPLEMENTAÇÃO DO DISPOSITIVO DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO

O módulo ESP 32 tem o regulador AMS1117 de 3,3V (corrente max 1A). No *datasheet* do ESP32 é especificado que a corrente fornecida para alimentação de 3,3V deve ser no máximo de 1 A. Portanto, nunca se deve conectar um outro módulo ou dispositivo que consuma mais do que 1 A, para não sobrecarregar o seu regulador interno. A figura 16 demonstra o regulador junto dos seus componentes auxiliares.

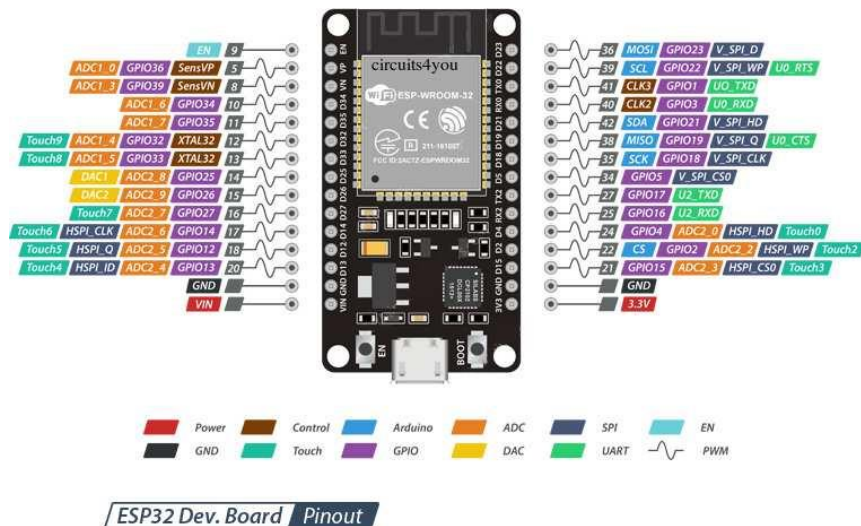
Figura 16 – Circuito interno de alimentação do ESP32.



Fonte: (ADVANCED MONOLITHIC SYSTEMS, [s.d.], p1.).

Os pinos de *General Purpose Input Output* (GPIO) demonstrado na figura 17 são portas que servem como interface entre o microcontrolador e periféricos, como sensores, atuadores ou até outros circuitos integrados e microcontroladores.

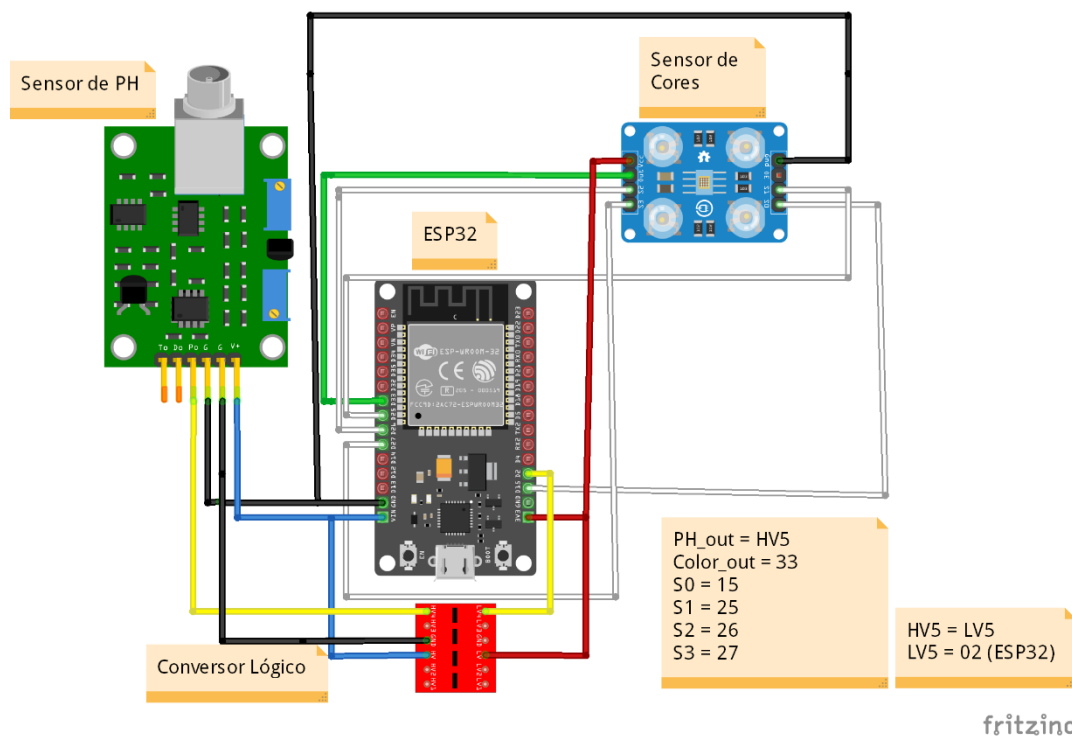
Figura 17 – Pinagem GPIO do módulo ESP 32.



Fonte: (LOBODAROBÓTICA, 2021, p1.).

O esquemático detalhado do protótipo (figura 18) foi elaborado no software Fritzing, afim de garantir o correto funcionamento dos componentes utilizados no protótipo. Os pinos utilizados foram os pinos GPIO 2 e 4, o pino de alimentação (5 Volts), o pino de alimentação de 3,3 Volts, o GND e os pinos de 5 Volts para 3,3 Volts do conversor lógico (HV5 e LV5).

Figura 18 - Esquemático das conexões – Sensores, Conversor Lógico e Módulo ESP32.



Fonte: Autoria própria.

3.4 FIRMWARES DOS DISPOSITIVOS DE AQUISIÇÃO DE DADOS E COMUNICAÇÃO

Após a escolha do microcontrolador ESP 32 foi iniciado o desenvolvimento do *firmware*. Primeiramente foi realizada a configuração do conversor ADC, no qual permite ter uma precisão considerável devido a utilização do ESP 32 utilizando um range de 0 a 4095 que varia de acordo com a tensão de entrada na porta do dispositivo. Em seguida, são realizadas as leituras dos sensores. Após a leitura analógica, este valor será em uma variável, e a partir daí, é tratado para poder ser enviado através do Bluetooth e banco de dados MySQL, sendo também necessário realizar a configuração do banco de dados e do app Bluetooth Serial para o recebimento dos dados. Devido à limitação de memória do microcontrolador ESP32 foi preciso utilizar dois códigos distintos um para a comunicação Bluetooth e outro para a comunicação Wi-Fi. Utilizando-as em um único código ocorre um *overflow* (conforme figura

19) , ou seja , um programa maior do que a memória do dispositivo.

Figura 19 – Erro de *overflow* de armazenamento.

```

Erro compilando para a placa ESP32 Dev Module
O sketch usa 1520202 bytes (115%) de espaço de armazenamento para programas. O máximo são 1310720 bytes.
  
```

Fonte: Autoria própria.

3.5 CALIBRAÇÃO DOS SENSORES

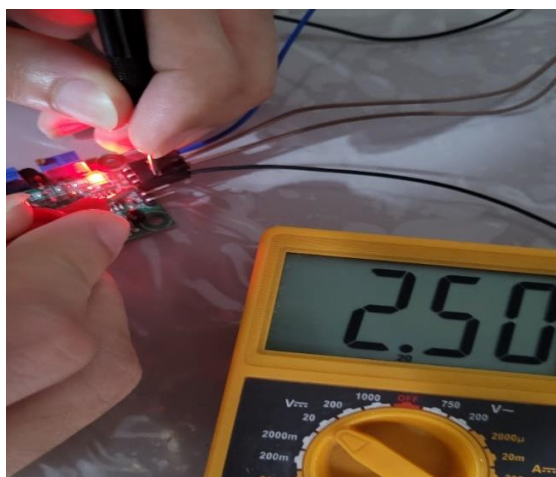
Para que fosse possível medir e avaliar os dados com precisão foi necessário realizar a calibração dos sensores utilizados neste trabalho, com os sensores calibrados os dados foram obtidos com mais confiabilidade e integridade. Para a configuração do sensor de pH foi necessário o uso de um multímetro para medir a tensão entre os terminais do GND e a entrada do sensor (5V) para constatar que a tensão de entrada fornecida pela porta USB estava de acordo com o recomendável para o pleno funcionamento do referido sensor. Posteriormente a verificação da tensão de entrada foi realizada a medição da tensão de saída do sensor, sendo efetuado para isso um curto no conector BNC do sensor (conforme figura 20) e a variação de tensão obtida através do resistor variável do próprio sensor com uma chave de fenda até alcançar 2,5 Volts (conforme figura 21) visualizado no display do multímetro digital. Após essa calibração foi criada uma variável de ajuste no código para ajustar a diferença na conversão de 5 para 3,3 Volts do conversor lógico.

Figura 20 – Calibragem do sensor de pH.



Fonte: Autoria própria.

Figura 21 – Tensão lida no multímetro digital após a calibração.



Fonte: Autoria própria.

O sensor de cores TSC-230 foi calibrado através de papéis impressos com as tonalidades aproximadas das urinas e a leitura dos espectros de cores lidos pelo sensor em RGB e tons de branco (conforme figuras 22 e 23). Através desses valores foi possível separar as cores pelos parâmetros que distoam entre si, realizou-se essa separação por meio de condicionais postas no código. Na figura 24 esses valores são exibidos no Arduino IDE através do monitor serial.

Figura 22 – Papel com as cores aproximadas das urinas.



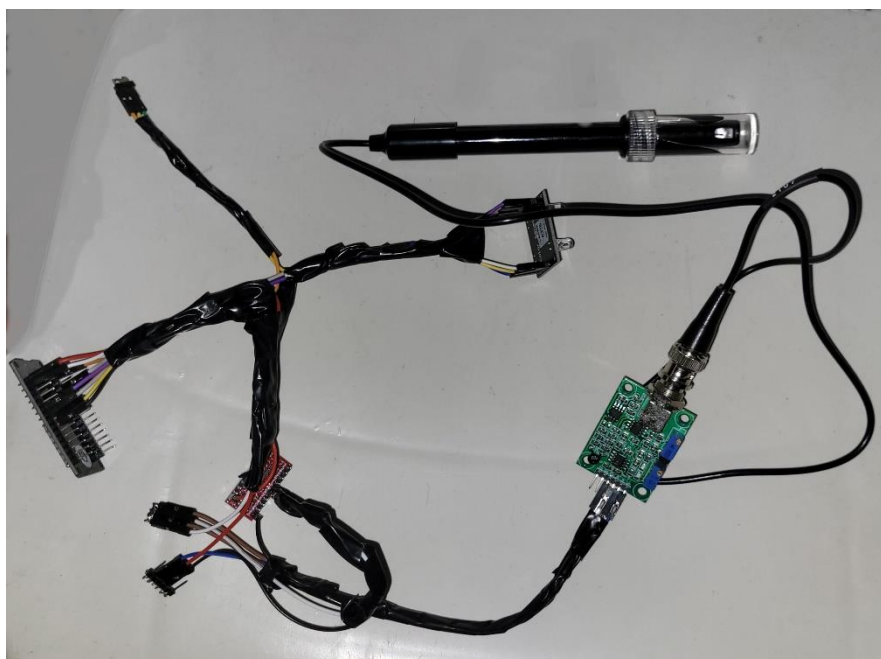
Fonte: Autoria própria.

3.6 MONTAGEM DO PROTÓTIPO

Com a aquisição dos materiais, a montagem dos componentes foi executada com facilidade, uma vez que as conexões foram realizadas conforme as especificações dos respectivos *datasheets* disponibilizados por cada fabricante. Colocou-se fita isolante para dar mais estabilidade e prevenir maus-contatos nos fios com o uso do protótipo (conforme figura 25).

A integração dos sensores com o microcontrolador ESP32 foi realizada através do desenvolvimento de um código, utilizando a linguagem de programação C e o *software* Arduino IDE. Após o desenvolvimento dos *firmwares* foi necessário fazer a configuração na plataforma Clever Cloud e o app Serial Bluetooth para Android, sendo também necessário configurar a rede Wi-Fi utilizada para a comunicação com o banco de dados na plataforma Clever Cloud que fornecerá a possibilidade de visualização e acompanhamento das informações no banco de dados MySQL.

Figura 25 – Protótipo Montado.



Fonte: A autoria própria.

3.6.1 Seleção de Componentes e Sistemas

Foi de extrema importância a pesquisa, identificação e seleção de componentes e sistemas que compõem o modelo. Para isso, a pesquisa avançou sobre uma gama de sensores

capazes de medir as características físicas e químicas da urina.

A principal dificuldade encontrada, foi o fato de existirem poucos sensores no mercado da cidade e o custo para a aquisição dos materiais, dificultando e limitando assim a escolha dos mesmos, como também, a falta de informações sólidas para o correto uso dos sensores atrelado ao fato de que o código utilizando as duas comunicações ao mesmo tempo causa uma extrapolação da memória do microcontrolador ESP32. Apesar das dificuldades e limitações encontradas, foi possível obter um resultado satisfatório com os sensores adquiridos.

3.6.2 Aquisição de Materiais

Os sensores selecionados foram comprados fora da cidade, pois não foi possível encontra-los nas lojas de componentes eletrônicos da cidade. Todos os sensores foram comprados no *site* da Shopee de revendedores nacionais e os outros componentes foram adquiridos na loja Smartproject localizada na própria cidade de Manaus. O tempo de chegada desses materiais até a cidade de Manaus foi de aproximadamente um mês.

Os custos dos materiais utilizados para construção do protótipo na moeda Real (R\$) estão descritos conforme a tabela 1 abaixo:

Tabela 1 - Custos do protótipo.

Componente	Quantidade	Preço
Módulo ESP32	1	R\$75,00
<i>jumper wires</i> fêmea-fêmea 12cm	20	R\$10,00
Sensor de pH	1	R\$180,00
Sensor de cores	1	R\$85,00
Conversor de nível lógico	1	R\$15,00
Total:		R\$365,00

Fonte: Autoria própria.

3.6.3 Comissionamento

Por fim, foi realizada a etapa de comissionamento, onde foram testados e avaliados os pontos mais relevantes do projeto, tais como: testes de continuidade elétrica, teste de integridade de componentes, testes de comunicação, teste de leitura de dados, calibração e validação das leituras de dados e teste de estabilidade de medição.

Nos testes de continuidade elétrica foi verificado a integridade dos circuitos através da medição de continuidade, e foram realizadas medições de tensão, de forma a eliminar a possibilidade de curto, de fuga de corrente e de perda de tensão e o correto funcionamento do dispositivo.

Nos testes de integridade de componentes foi verificado o estado geral de cada elemento do sistema, observando possíveis danos físicos, presença de umidade, e avarias tanto no sensor, quanto no microcontrolador.

Testes de comunicação foram realizados com objetivo de verificar primeiramente a conexão e envio de dados entre os microcontroladores ESP32, e em seguida com a rede Wi-Fi. Em seguida, foram verificados o correto envio de dados por parte do microcontrolador, e a correta recepção dos mesmos dados no banco de dados MySQL. Neste ponto o ESP32 se mostrou perfeitamente compatível com os sensores, podendo transmitir as informações via protocolo Wi-Fi e Bluetooth com velocidade aceitável e de forma praticamente instantânea.

3.7 CONFIGURAÇÃO DO SMARTPHONE PARA A COMUNICAÇÃO BLUETOOTH COM O ESP32

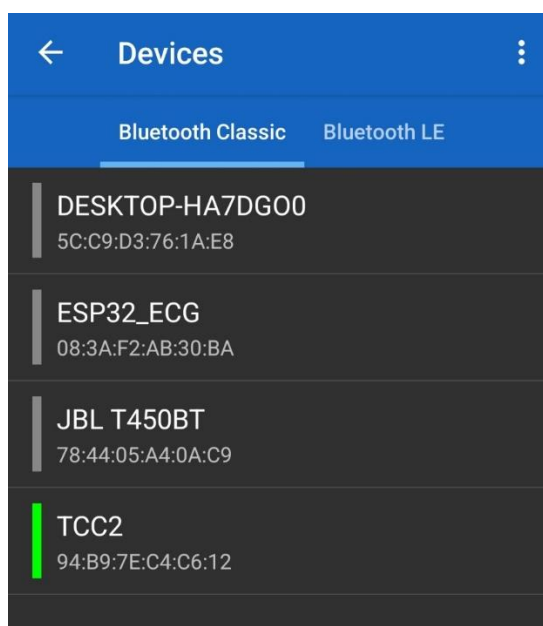
Para realizar a comunicação Bluetooth entre o smartphone e o ESP32 foi utilizado uma aplicação desenvolvida para Android que pode ser baixada e instalada pela loja de aplicativos Play Store gratuitamente conforme a figura 26, após baixar a aplicação é muito simples de se conectar ao módulo ESP32 e obter os dados. Ao iniciar o aplicativo liga-se o Bluetooth do smartphone e clica-se na opção devices dentro do aplicativo, será mostrado uma lista com os dispositivos detectados, posteriormente seleciona-se o nome do módulo ESP32 e o módulo estará conectado e transmitindo os dados (conforme a figura 27).

Figura 26 – App utilizado para a comunicação Bluetooth.



Fonte: Autoria própria.

Figura 27 – Aba devices do aplicativo Serial Bluetooth Terminal.

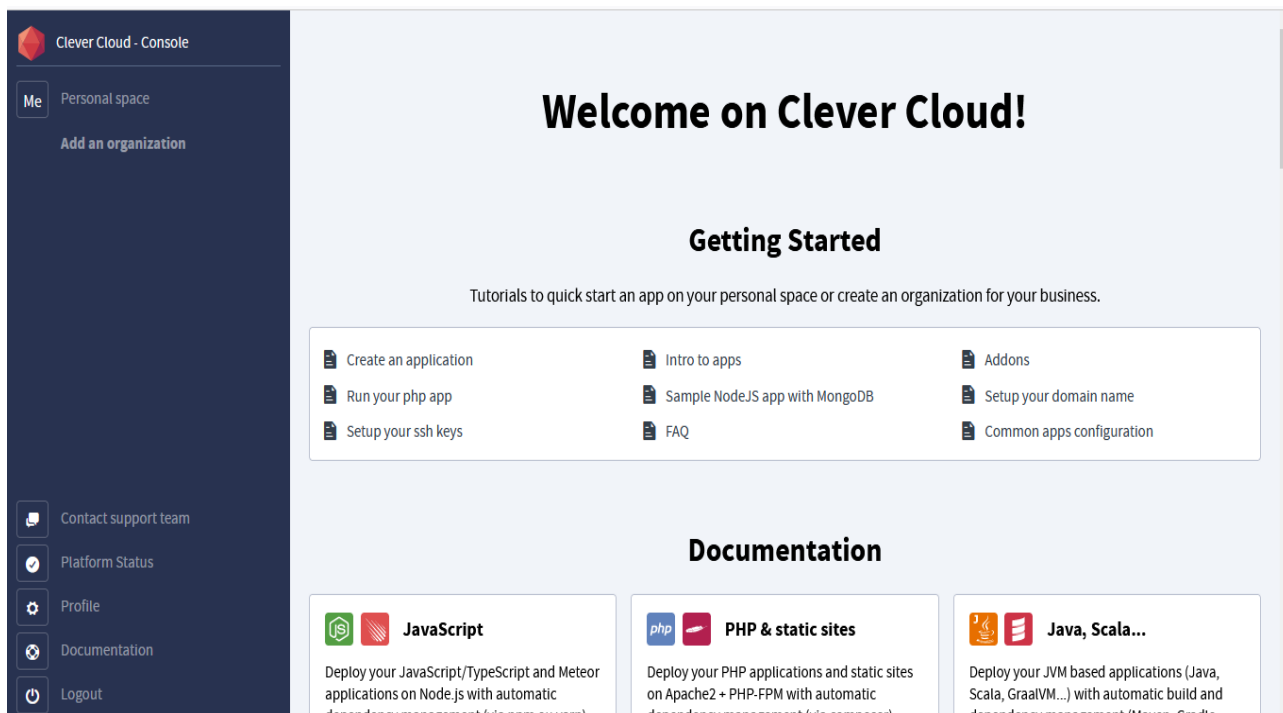


Fonte: Autoria própria.

3.8 CONFIGURAÇÃO DO BANCO DE DADOS MYSQL NO CLEVER CLOUD

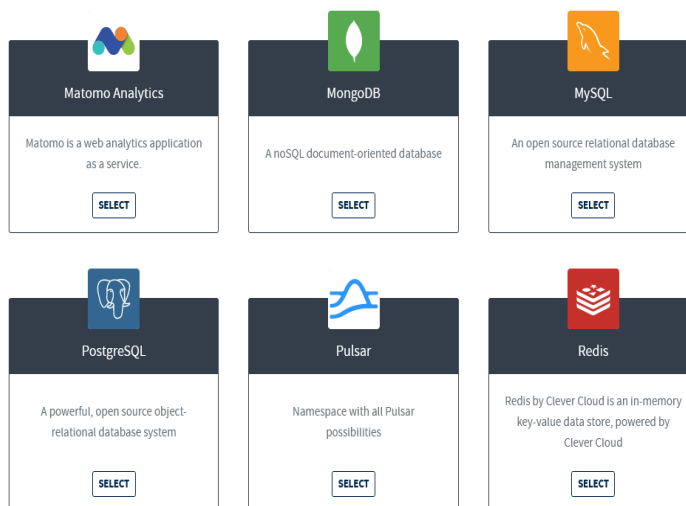
A comunicação com o banco de dados é efetuada por meio de um serviço hospedado na *web* que pode ser acessado pelo *link*: clevercloud.com, após acessar o site é necessário realizar o cadastro clicando-se em *sign up* e preenchendo os dados exigidos pelo cadastro. Após a realização do cadastro (figura 28) clica-se na aba *personal space* e em seguida clica-se em adicionar *add-on* e será exibido várias funcionalidades ao qual se seleciona a opção banco de dados MySQL conforme figura 29. Ao criar o banco de dados uma tela com as credenciais aparecerá, essas credenciais serão utilizadas no código do Arduino IDE para a conexão com o banco de dados conforme figura 30, na parte superior há uma aba chamada *PHPMysqlAdmin* que será utilizado para a criação de uma tabela, ao criar a tabela selecione o nome da variável e seu tipo (*char*, *float*, *int*, *current timestamp*) conforme figuras 31, 32 e 33. Com a criação do banco de dados e o endereçamento realizado no código do Arduino IDE o módulo ESP32 já estará apto para realizar o envio dos dados para a tabela do banco de dados.

Figura 28 – Página inicial do Clever Cloud após login.



Fonte: Autorial própria.

Figura 29 – Funcionalidades do site Clever Cloud.



Fonte: Autorial própria.

Figura 30 – Credenciais do banco de dados.

Database Credentials

Get credentials for manual connections to this database. [Export Environment Variables](#)

Host

Database Name

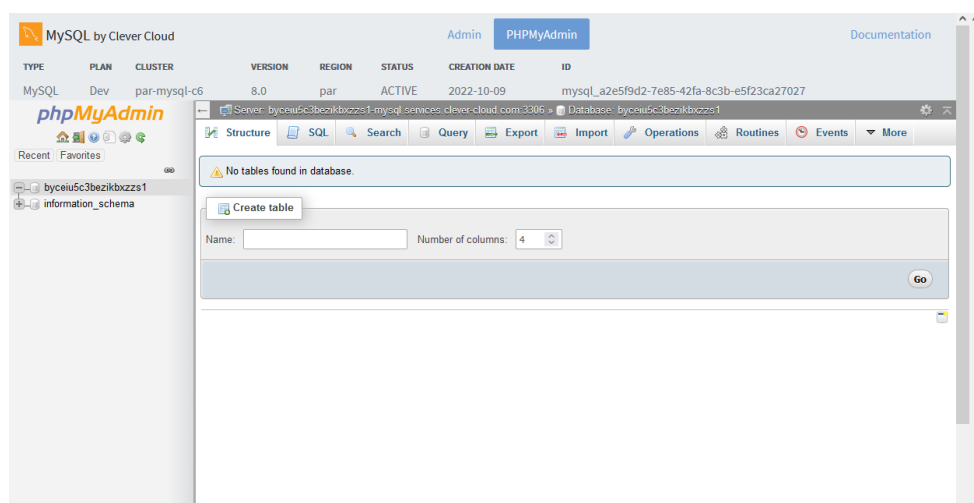
User

Password

Port

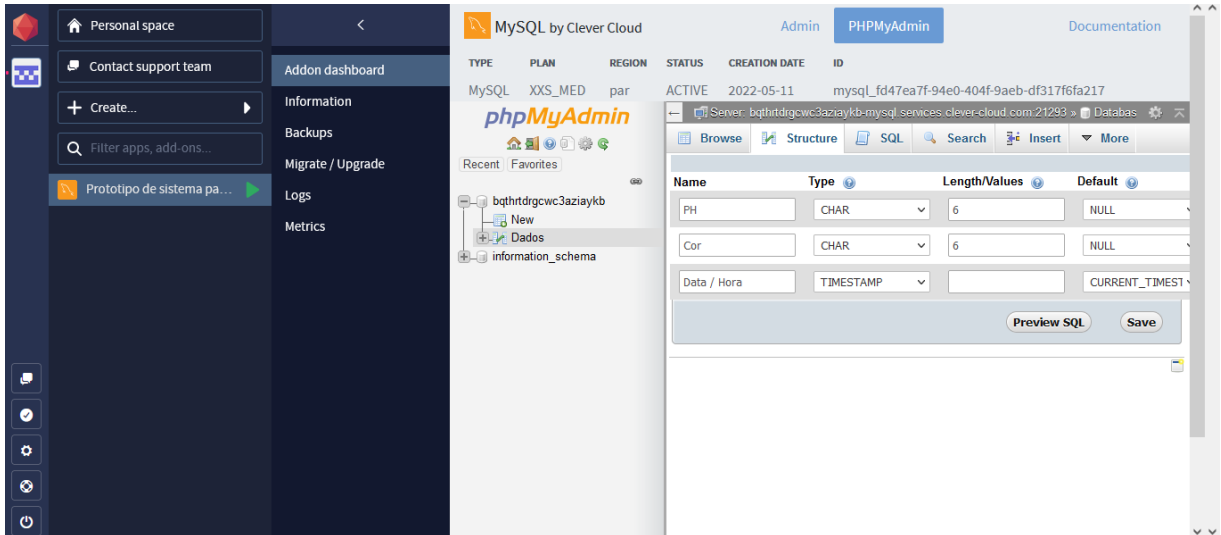
Fonte: Autoria própria.

Figura 31 – Painel de configurações do banco de dados MySQL.



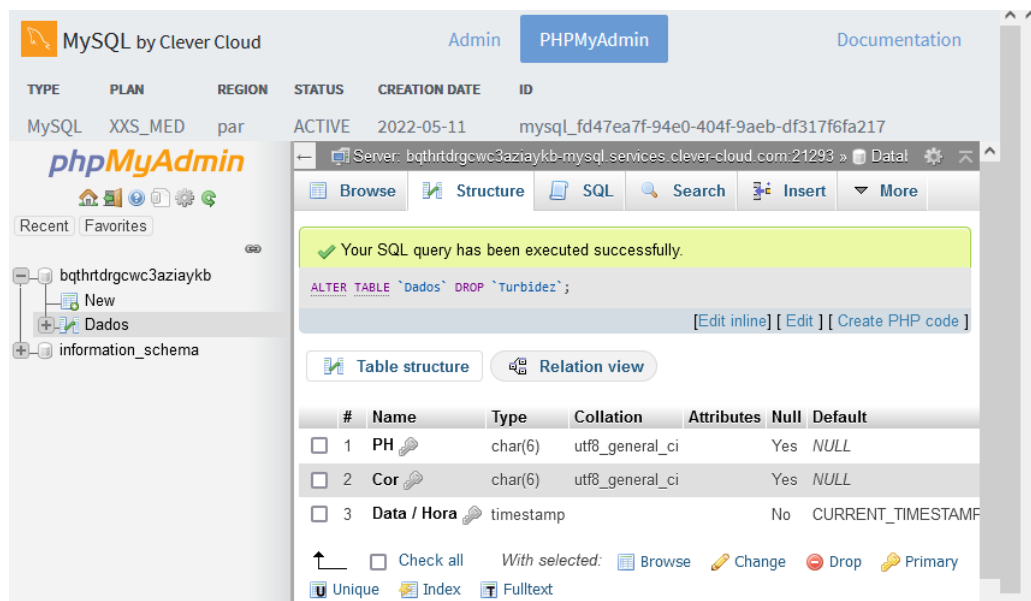
Fonte: Autoria própria.

Figura 32 – Menu de Configuração das variáveis do banco de dados.



Fonte: Autoria própria.

Figura 33 – Banco de dados com variáveis configuradas.



Fonte: Autoria própria.

4 RESULTADOS OBTIDOS

Este capítulo mostra os resultados obtidos após a implementação do projeto. Importante frisar, a atenção que foi dedicada na coleta e registro de dados e dos custos envolvidos, uma vez que este seria apenas um primeiro passo para a implementação de um sistema mais robusto, e capaz de funcionar em um modelo de escala comercial.

4.1 AMOSTRAS

Para a verificação da confiabilidade dos sensores e para a aquisição dos dados para a análise foram coletadas 8 amostras de urina durante 2 dias que foram analisadas pelos aspectos de Ph e de cores, as amostras foram medidas pouco tempo após a sua coleta e armazenadas em copos de vidro na temperatura ambiente conforme a figura 34.

Figura 34 – Amostras de urina.



Fonte: Autoria própria.

4.2 RESULTADO DO SISTEMA DE MONITORAMENTO

Devido à análise das amostras foi gerado a tabela 2 com os dados de pH, cores e o status em que se encontra o nível de desidratação ao longo do dia.

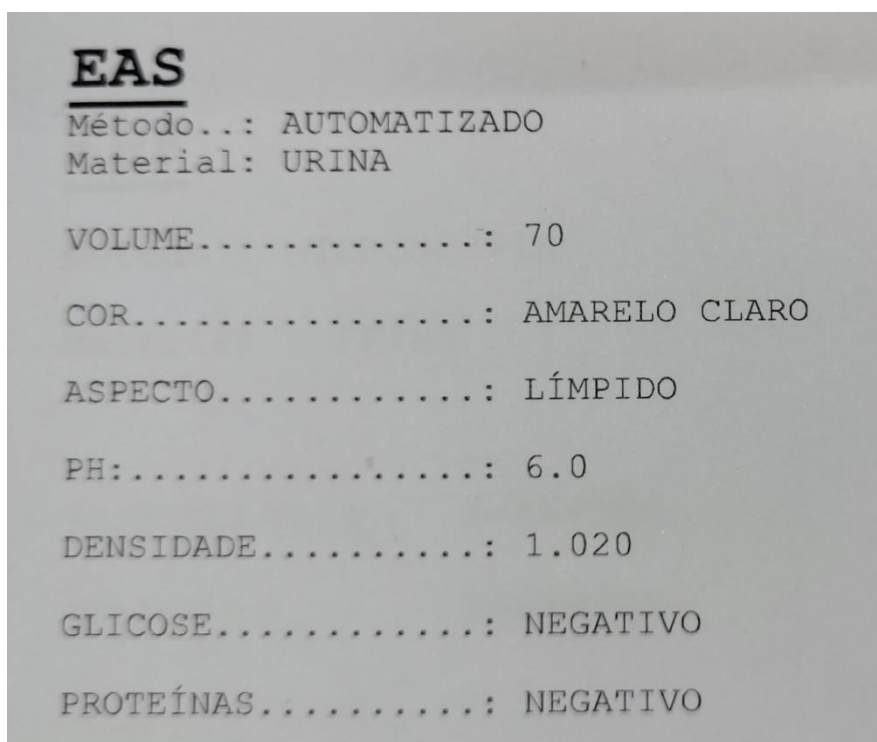
Tabela 2 – Resultado das análises das amostras de urina.

PH	Cor	Status
6.23	Amarelo Claro	Hidratado
6.34	Amarelo Claro	Hidratado
6.1	Amarelo Escuro	Desidratado
5.85	Amarelo Escuro	Desidratado
6.2	Amarelo Claro	Hidratado
6.45	Amarelo Escuro	Desidratado
6.37	Amarelo Claro	Hidratado
6.1	Amarelo Claro	Hidratado

Fonte: Autoria Própria.

Para se ter uma comparabilidade e uma confiabilidade maior nos dados a última amostra foi enviada para um laboratório em que foi realizado um exame de urina convencional (Elementos Anormais do Sedimento EAS) demonstrado na figura 35. Os resultados saíram muito próximos ao apresentado na tabela de resultados das análises das urinas.

Figura 35 – Resultado exame de urina (EAS) feito em laboratório.



```

EAS
Método...: AUTOMATIZADO
Material: URINA

VOLUME.....: 70

COR.....: AMARELO CLARO

ASPECTO.....: LÍMPIDO

PH:.....: 6.0

DENSIDADE.....: 1.020

GLICOSE.....: NEGATIVO

PROTEÍNAS.....: NEGATIVO

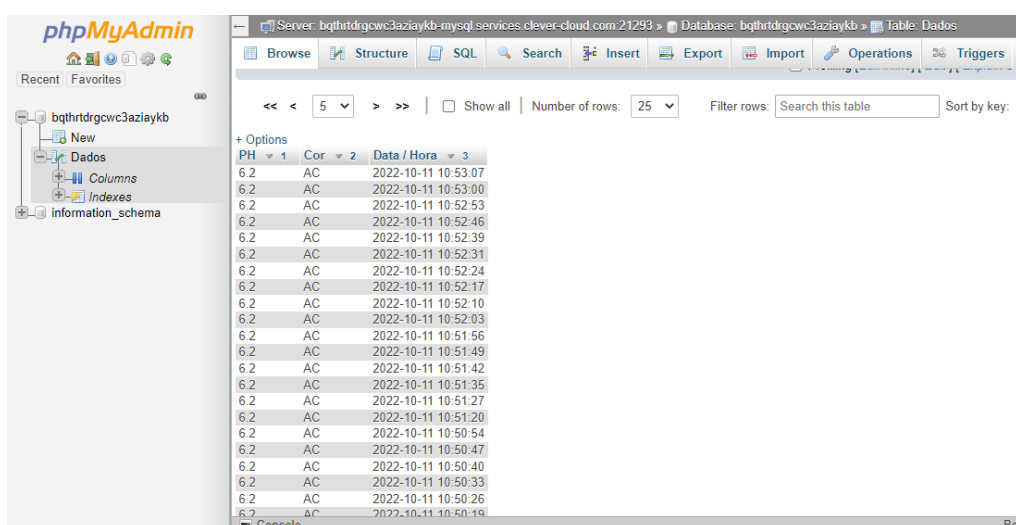
```

Fonte: Autoria Própria.

4.3 VISUALIZAÇÃO DOS DADOS OBTIDOS

Os dados obtidos das amostras foram visualizados através de um banco de dados na web ou pelo aplicativo Bluetooth constatando que as informações exibidas são confiáveis e relevantes não sendo afetadas pela transmissão, como também facilitando a coleta e visualização dos dados, sendo necessário apenas alimentar o módulo ESP32 para a aquisição e transmissão dos dados.

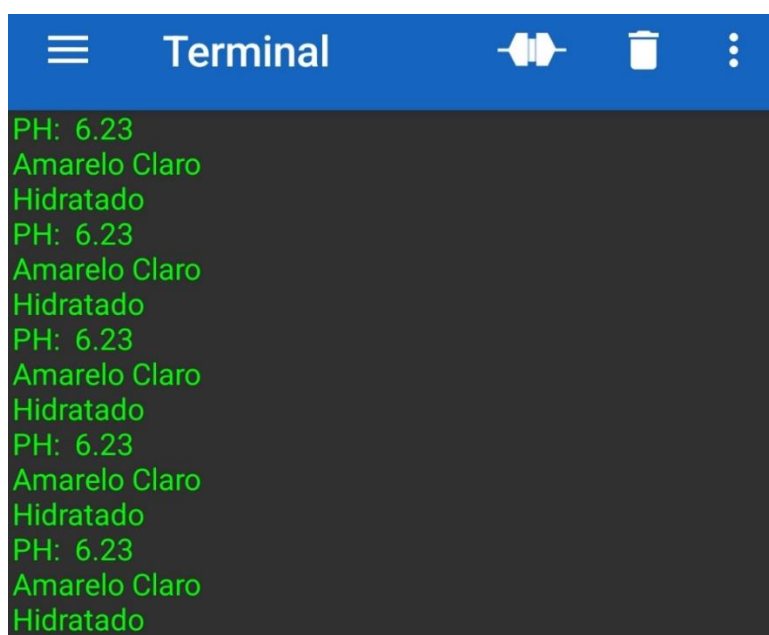
Figura 36 – Visualização dos dados através do banco de dados.



PH	Cor	Data / Hora
6.2	AC	2022-10-11 10:53:07
6.2	AC	2022-10-11 10:53:00
6.2	AC	2022-10-11 10:52:53
6.2	AC	2022-10-11 10:52:46
6.2	AC	2022-10-11 10:52:39
6.2	AC	2022-10-11 10:52:31
6.2	AC	2022-10-11 10:52:24
6.2	AC	2022-10-11 10:52:17
6.2	AC	2022-10-11 10:52:10
6.2	AC	2022-10-11 10:52:03
6.2	AC	2022-10-11 10:51:56
6.2	AC	2022-10-11 10:51:49
6.2	AC	2022-10-11 10:51:42
6.2	AC	2022-10-11 10:51:35
6.2	AC	2022-10-11 10:51:27
6.2	AC	2022-10-11 10:51:20
6.2	AC	2022-10-11 10:50:54
6.2	AC	2022-10-11 10:50:47
6.2	AC	2022-10-11 10:50:40
6.2	AC	2022-10-11 10:50:33
6.2	AC	2022-10-11 10:50:26
6.2	AC	2022-10-11 10:50:19

Fonte: Autoria própria.

Figura 37 – Visualização dos dados no aplicativo Serial Bluetooth Terminal.



```

Terminal
PH: 6.23
Amarelo Claro
Hidratado
PH: 6.23
Amarelo Claro
Hidratado
PH: 6.23
Amarelo Claro
Hidratado
PH: 6.23
Amarelo Claro
Hidratado
PH: 6.23
Amarelo Claro
Hidratado
PH: 6.23
Amarelo Claro
Hidratado

```

Fonte: Autoria própria.

CONCLUSÃO

No desenvolvimento da presente pesquisa foram realizadas revisões acerca de assuntos como: sensores para medição de pH e cores, técnicas de leitura de dados, transmissão de dados, protocolos de comunicação Bluetooth e Wi-Fi, armazenamento em banco de dados, tecnologias de gerenciamento de banco de dados.

Houve uma certa dificuldade para compra dos sensores, uma vez que ainda não existe uma popularização dos sensores utilizados neste trabalho, o que ocasiona uma dificuldade para realizar a aquisição de tais sensores na cidade, necessitando comprá-los de outros estados do território nacional.

A falta de uma base teórico-científica consolidada, ou estudos anteriores que pudessem auxiliar no desenvolvimento do protótipo são escassos. Assim, a maioria das soluções testadas para o desenvolvimento do protótipo foram elaboradas a partir de conhecimento teórico aplicado. No entanto, após a montagem do protótipo é possível perceber o bom funcionamento do sistema e a sua fácil interação com os protocolos de comunicação utilizados. Foram obtidos dados precisos com uma velocidade e qualidade eficiente de transmissão desses dados.

Quanto ao objetivo pretendido, se apresentou plenamente realizável, visto que, foi possível construir um sistema para o monitoramento contínuo da urina humana de boa qualidade e eficácia, apesar de certas limitações.

Esta pesquisa viabiliza que a solução seja utilizada em um estudo mais amplo, considerando a possibilidade de utilização de outros sensores, um dispositivo mais potente e com uma capacidade de armazenamento que suporte as duas comunicações ao mesmo tempo e de boa resposta na operação simultânea das duas comunicações, uma rede maior para comunicação dos dados para a web, podendo assim ter um sistema mais robusto e versátil, tornando assim, o monitoramento mais eficiente e adequado, informando assim com maior precisão a qualidade da urina ao longo do dia.

O próximo passo, que ainda pode ser desenvolvido em ambiente acadêmico, seria a ampliação da pesquisa para o aprimoramento do protótipo e futuramente a geração um produto, visto que o protótipo apresentou um desempenho satisfatório durante a execução dos testes e a transmissão dos dados.

Com a ampliação da pesquisa será possível reunir os recursos e a tecnologia necessários para levar a solução a um patamar que possa fazer a diferença na vida das pessoas e da saúde pública em geral, trazendo maior segurança, um tempo de ação mais rápido, das autoridades responsáveis, à possíveis complicações decorrentes da desidratação ao longo do tempo, e da falta de qualidade na dieta das pessoas, como por exemplo, a alta ingestão de sal.

REFERÊNCIAS

ADVANCED MONOLITHIC SYSTEMS. **AMS1117**. EUA: [s.d.]. Disponível em: <<http://www.advanced-monolithic.com/pdf/ds1117.pdf>>.

ARDUINOECIA. **Como funciona o conversor de nível lógico bidirecional**. [S.l.], 2015. Disponível em: <<https://www.arduinoecia.com.br/conversor-de-nivel-logico-33-5v-arduino/>>. Acesso em: 23 set. 2022.

AUTOCORE ROBÓTICA. **Sonda Eletrodo Sensor de PH com Módulo PH4502C**. [S.l.], 2021. Disponível em: <<https://www.autocorerobotica.com.br/sonda-eletrodo-sensor-de-ph-com-modulo-PH4502C>>. Acesso em: 7 out. 2021.

BERTOLETI, Pedro. **Projetos com ESP32 e LoRa**. 1.ed. São Paulo: Instituto Newton C Braga, 2019.

BRASIL, Laboratório Nacional de Computação Científica. **Apostila de Banco de dados**. Rio de Janeiro, [s.d.]. Disponível em: <https://www.lncc.br/~rogerio/poo/09%20-%20Banco%20de%20Dados/Apostila_de_BD.pdf>.

DALMOLIN, Magnun L. **A URINÁLISE NO DIAGNÓSTICO DE DOENÇAS RENAIS**. Rio Grande do Sul. Universidade Federal do Rio Grande do Sul: 2011.

ELETROGATE. **Módulo Sensor De Cor TCS230/TCS3200 - GY-31**. [S.l.], 2021. Disponível em: <<https://www.eletrogate.com/modulo-sensor-de-cor-gy-31-tcs230-tcs3200>>. Acesso em: 11 out. 2021.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**. [S.l.], 2021. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 02 out. 2021.

FERNANDES, Natalia Castro. **SISTEMA OPERACIONAL ANDROID**. Rio de Janeiro: Universidade Federal Fluminense, 2012.

GUYTON, AC: **Fisiologia Humana**, 10 ed. Rio de Janeiro: Editora Guanabara e Koogan, 2008.

KINSTA. **O Que é MySQL? Uma Explicação Simples para Quem Está Começando**. [S.l.]: 2022. Disponível em: <<https://kinsta.com/pt/base-de-conhecimento/o-que-e-mysql/>>. Acesso em: 22 set. 2022.

LOBODARÓBOTICA. **ESP32 Pinout: Saiba tudo sobre a ESP!**. [S.l.]: 2021. Disponível em: <<https://lobodarobotica.com/blog/esp32-pinout/>>. Acesso em: 18 set. 2022.

LOGOMARCAS. **Android Logo**. [S.l.], 2022. Disponível em: <<https://logosmarcas.net/android-logo/>>. Acesso em: 15 set. 2022.

LOPES, Yara Alves. **CONSTRUÇÃO DE SISTEMAS AUTOMATIZADOS PARA PASTEURIZAÇÃO DE LEITE E PRODUÇÃO DE IOGURTE NATURAL**. Trabalho de Conclusão de Curso apresentado a Universidade Federal Rural de Pernambuco – Unidade Acadêmica de Garanhuns, 2019.

MINHAVIDA. **Desidratação: sintomas, tratamentos e causas**. [S.l.], 2021. Disponível em: <<https://www.minhavidacom.br/saude/temas/desidratacao>>. Acesso em: 6 out. 2021.

NETO, Francisco Gonçalves Pereira. **Banco de Dados – Exemplo**. [S.l.]: 2008. Disponível em: <<http://e-reality-database.blogspot.com/2008/09/banco-de-dados-exemplo.html>>. Acesso em: 20 set. 2022.

NORONHA, Thais. **Sociedade Brasileira de Nefrologia alerta para sintomas que podem indicar doenças renais**. São Paulo, 2013. Disponível em: <<http://www.crfsp.org.br/noticias/4144-1403-dia-mundial-do-rim.html>>. Acesso em: 23 out. 2021.

PATSKO, Luís Fernando. **TUTORIAL Aplicações, Funcionamento e Utilização de Sensores**. [S.I]: Maxuell Bohn Instrumentação Eletrônica, 2006.

PISA, Pedro. **O que é e como usar o MySQL?**. [S.l.]: 2012. Disponível em: <<https://www.techtudo.com.br/noticias/2012/04/o-que-e-e-como-usar-o-mysql.ghtml>>. Acesso em: 21 set. 2022.

PRÓ-RIM. **A cor da urina pode revelar doenças**. [S.l.]: 2019. Disponível em: <<https://www.prorim.org.br/blog-artigos/a-cor-da-urina-pode-revelar-doencas/>>. Acesso em: 25 set. 2022.

QUESADA et al. **Apostila de MySQL**. São Paulo: 2001. Disponível em: <http://www.telecentros.sp.gov.br/saber/apostilas/antigas/apostila_sql.pdf>.

SANTOS, Bruno P.; SILVA, Lucas A. M.; CELES, Clayson S. F. S.; NETO, João B. Borges; PERES, Bruna S.; VIEIRA Marcos Augusto M.; VIEIRA, Luiz Filipe M.; GOUSSEVSKAIA, Olga N.; LOUREIRO, Antonio A. F. **Internet das Coisas: da Teoria à Prática**. Minas Gerais: Departamento de Ciência da Computação Universidade Federal de Minas Gerais (UFMG), 2016.

SILVA JÚNIOR, Vidal Pereira da Silva. **Microcontroladores**. São Paulo: Érica, 1988.

SILVEIRA, Cristiano Bertulucci. **12 Aplicações para o uso do Sensor de Cor**. [S.l.], 2016. Disponível em: <<https://www.citisystems.com.br/sensor-de-cor/>>. Acesso em: 11 out.2021.

STRASINGER, SK. **Uroanálise & Fluidos Biológicos**, 3 ed. São Paulo: Editorial Premier, 2000.

USAINFO. **Conversor de Nível Lógico 3,3 / 5V Bidirecional - CNL35**. [S.I], 2022. Disponível em: <<https://www.usainfo.com.br/conversores-de-sinal/conversor-de-nivel-logico-33-5v-bidirecional-cnl35-3741.html>>. Acesso em: 30 set. 2022.

APÊNDICE A - CÓDIGO USADO NO MICROCONTROLADOR PARA A TRANSMISSÃO DE DADOS VIA BLUETOOTH.

```

#include "BluetoothSerial.h"

// Bluetooth Serial object
BluetoothSerial SerialBT;

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and
enable it
#endif

String message = "";
char incomingChar;
// Variáveis auxiliares

// Pinos e variáveis do modulo de pH
const int potPin = 2;
float ph;
float Value = 0;
float Ajuste = 0.63;
float ph_ajustado;

//Pinos de conexao do modulo de cor
const int S0 = 15;
const int S1 = 25;
const int S2 = 26;
const int S3 = 27;
const int OUT = 33;

//cores do sensor de cor
int getR = 0;
int getG = 0;
int getB = 0;
int getN = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  // Nome do dispositivo Bluetooth
  SerialBT.begin("TCC2");
  pinMode(potPin,INPUT);
  delay(1000);

```

```

pinMode(S0,OUTPUT);
pinMode(S1,OUTPUT);
pinMode(S2,OUTPUT);
pinMode(S3,OUTPUT);
pinMode(OUT,INPUT);
// Escala de frequencia para 20%
digitalWrite(S0, HIGH);
digitalWrite(S1, LOW);

}

void readColor(){
  //Read red filter
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  delay(50);
  if(digitalRead(OUT) == HIGH){
    getR = pulseIn(OUT, LOW);
  } else {
    getR = pulseIn(OUT, HIGH); // duracao do pulso na saida
  }
  //Read blue filter
  digitalWrite(S3, HIGH);
  delay(50);
  if(digitalRead(OUT) == HIGH){
    getB = pulseIn(OUT, LOW);
  } else {
    getB = pulseIn(OUT, HIGH); // duracao do pulso na saida
  }
  //Read green filter
  digitalWrite(S2, HIGH);
  delay(50);
  if(digitalRead(OUT) == HIGH){
    getG = pulseIn(OUT, LOW);
  } else {
    getG = pulseIn(OUT, HIGH); // duracao do pulso na saida
  }
  //Read NO filter
  digitalWrite(S2, HIGH);
  digitalWrite(S3, LOW);
  delay(50);
  if(digitalRead(OUT) == HIGH){
    getN = pulseIn(OUT, LOW);
  } else {
    getN = pulseIn(OUT, HIGH); // duracao do pulso na saida
  }

  Serial.print("Leitura de cada filtro do sensor: ");
  Serial.print("[R,G,B,N]= [");

```

```

Serial.print(getR,DEC);
Serial.print(" ");
Serial.print(getG,DEC);
Serial.print(" ");
Serial.print(getB,DEC);
Serial.print(" ");
Serial.print(getN,DEC);
Serial.println("");

if (getR >= 30 and getG >= 50 and getB >= 65 and getN >= 12){
  char cor [] = "Amarelo Claro";
  SerialBT.print (cor);
  SerialBT.print ("Hidratado");
}

else if (getR >= 30 and getG >= 35 and getB >= 35 and getN >= 10){
  char cor [] = "Amarelo Escuro";
  SerialBT.print (cor);
  SerialBT.print ("Desidratado");
}
else {
  char cor [] = "Outra cor";
  SerialBT.print ("Possível doença ou não é urina");
}

}

void loop(){

  Value= analogRead(potPin);
  Serial.print ("Volts ADC: ");
  Serial.print(Value);
  Serial.print(" | ");
  float voltage=Value*(3.3/4095.0);
  ph=(3.3*voltage);
  ph_ajustado = ph + Ajuste;
  Serial.print ("Volts: ");
  Serial.print (voltage);
  Serial.print(" | ");
  Serial.print ("PH: ");
  Serial.println(ph_ajustado);
  SerialBT.print ("PH: ");
  SerialBT.println(ph_ajustado);
  if (ph_ajustado < 5.50 or ph_ajustado < 7.50){
    SerialBT.println ("PH Alterado");
  }
  delay(500);
  readColor();
  delay(5);
}

```

APÊNDICE B - CÓDIGO USADO NO MICROCONTROLADOR PARA A TRANSMISSÃO DE DADOS VIA WI-FI.

```

// Código para envio dos dados por Wi-Fi
#include <WiFi.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

#include <iostream>
#include <string>

IPAddress server_addr(185, 42, 117, 222); // O IP DO SERVIDOR DA CLEVER CLOUD
char user[] = "ux14smc97honvmaz"; // Usuario MySQL
char password[] = "E8thr1bGP2kiJLJbwrp"; // Senha MySQL
char ssid[] = "DomPenna_2G"; // Nome de rede Wifi
char pass[] = "dondinha01"; // Senha Wi-Fi
char INSERT_DATA[] = "INSERT INTO bqthrtgrgcwc3aziaykb.Dados (ph,cor) VALUES
('%s','%s')";
// EXEMPLO ACIMA: "INSERT INTO b3d3z8bggux9ngtdusfw.Dados (ID_SENSOR, PH)
VALUES (%d,%s)";

WiFiClient client;
MySQL_Connection conn(&client);
MySQL_Cursor* cursor;

// Pinos e variáveis do modulo de pH
const int potPin = 2;
float ph;
float Value = 0;
float Ajuste = 0.63;
float ph_ajustado;

//Pinos de conexao do modulo de cor
const int S0 = 15;
const int S1 = 25;
const int S2 = 26;
const int S3 = 27;
const int OUT = 33;
char cor ;

//cores do sensor de cor
int getR = 0;
int getG = 0;
int getB = 0;
int getN = 0;

```

```
void detectaCor();
```

```
void EnviaDados() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Sem conexão"); //
    WiFi.disconnect();
    delay(1000);
    WiFi.begin(ssid, pass);
    Serial.println();
    Serial.println("Conectando ao WiFi.");
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
    }
    Serial.println();
    Serial.println("Conectado a rede!"); //
  }
  char query[128];
  char DadosWifi[6];
  char ph2 [6];
  if (conn.connect(server_addr, 21293, user, password)) { //Conecta ao banco de dados
    delay(1000);
    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
    // Salvamento
    dtostrf(ph_ajustado, 1,2, ph2 );
    Serial.println (ph2);
    Serial.println (cor);

    sprintf(query, INSERT_DATA, ph2,cor);
    Serial.println (query);
    // Execute the query
    cur_mem->execute(query);
    // Note: since there are no results, we do not need to read any data
    // Deleting the cursor also frees up memory used
    delete cur_mem;
    Serial.println(); Serial.println("Data recorded."); //
  }
  else
    Serial.println(); Serial.println("Connection failed."); // Conexão não efetuada
  conn.close();
}

void setup() {
```



```

Serial.begin(115200);
pinMode(potPin,INPUT);
delay(1000);
pinMode(S0,OUTPUT);
pinMode(S1,OUTPUT);
pinMode(S2,OUTPUT);
pinMode(S3,OUTPUT);
pinMode(OUT,INPUT);
// Escala de frequencia para 20%
digitalWrite(S0, HIGH);
digitalWrite(S1, LOW);

}

void readColor(){
//Read red filter
digitalWrite(S2, LOW);
digitalWrite(S3, LOW);
delay(50);
if(digitalRead(OUT) == HIGH){
  getR = pulseIn(OUT, LOW);
} else {
  getR = pulseIn(OUT, HIGH); // duracao do pulso na saida
}
//Read blue filter
digitalWrite(S3, HIGH);
delay(50);
if(digitalRead(OUT) == HIGH){
  getB = pulseIn(OUT, LOW);
} else {
  getB = pulseIn(OUT, HIGH); // duracao do pulso na saida
}
//Read green filter
digitalWrite(S2, HIGH);
delay(50);
if(digitalRead(OUT) == HIGH){
  getG = pulseIn(OUT, LOW);
} else {
  getG = pulseIn(OUT, HIGH); // duracao do pulso na saida
}
//Read NO filter
digitalWrite(S2, HIGH);
digitalWrite(S3, LOW);
delay(50);
if(digitalRead(OUT) == HIGH){
  getN = pulseIn(OUT, LOW);
} else {
  getN = pulseIn(OUT, HIGH); // duracao do pulso na saida
}

```

```

}
```

```

Serial.print("Leitura de cada filtro do sensor: ");
Serial.print("[R,G,B,N]= [");
Serial.print(getR,DEC);
Serial.print(", ");
Serial.print(getG,DEC);
Serial.print(", ");
Serial.print(getB,DEC);
Serial.print(", ");
Serial.print(getN,DEC);
Serial.println("]");
```

```

if (getR >= 30 and getG >= 50 and getB >= 65 and getN >= 12){
  char cor [] = "AC";
```

```

}
```

```

else if (getR >= 30 and getG >= 35 and getB >= 35 and getN >= 10){
  char cor [] = "AE";
```

```

}
```

```

else {
  char cor [] = "OC";
}
```

```

}
```

```

void loop(){
```

```

  Value= analogRead(potPin);
  Serial.print ("Volts ADC: ");
  Serial.print(Value);
  Serial.print(" | ");
  float voltage=Value*(3.3/4095.0);
  ph=(3.3*voltage);
  ph_ajustado = ph + Ajuste;
  Serial.print ("Volts: ");
  Serial.print (voltage);
  Serial.print(" | ");
  Serial.print ("PH: ");
  Serial.println(ph_ajustado);
  delay(500);
  readColor();
  delay(5);
```

```
EnviaDatos();  
Serial.println();  
  
}
```