



TRABALHO DE CONCLUSÃO DE CURSO

**DISPOSITIVO DE AFERIÇÃO DE PESO
PARA OTIMIZAÇÃO DA COLETA
SELETIVA DE LIXO**

DAVID GONÇALVES MENDES

Brasília, Dezembro de 2023

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Ciência da Computação

TRABALHO DE CONCLUSÃO DE CURSO

**DISPOSITIVO DE AFERIÇÃO DE PESO
PARA OTIMIZAÇÃO DA COLETA
SELETIVA DE LIXO**

DAVID GONÇALVES MENDES

*Trabalho de conclusão de curso submetido ao Departamento de Ciência
da Computação como requisito parcial para obtenção
do grau de Engenheiro em Engenharia da Computação*

Banca Examinadora

Prof. Dr. Daniel Chaves Café, ENE/Universidade de Brasília
Orientador _____

Prof. Dr. José Edil Guimarães de Medeiros
Examinador Interno _____

Prof^ª. Dra. Simone Borges Simão Monteiro
Examinador interno _____

FICHA CATALOGRÁFICA

MENDES, DAVID GONÇALVES

DISPOSITIVO DE AFERIÇÃO DE PESO PARA OTIMIZAÇÃO DA COLETA SELETIVA DE LIXO [Distrito Federal] 2023.

xvi, 37 p., 210 x 297 mm (CIC/ENE/UnB, Engenheiro, Engenharia da Computação, 2023).

Trabalho de conclusão de curso - Universidade de Brasília.

Departamento de Ciência da Computação

1. IoT

2. ESP32

3. Sistema Web

4. Protocolo MQTT

I. CIC/ENE/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

MENDES, D. (2023). *DISPOSITIVO DE AFERIÇÃO DE PESO PARA OTIMIZAÇÃO DA COLETA SELETIVA DE LIXO*. Trabalho de conclusão de curso, Departamento de Ciência da computação, Universidade de Brasília, Brasília, DF, 37 p.

CESSÃO DE DIREITOS

AUTOR: DAVID GONÇALVES MENDES

TÍTULO: DISPOSITIVO DE AFERIÇÃO DE PESO PARA OTIMIZAÇÃO DA COLETA SELETIVA DE LIXO.

GRAU: Engenheiro em Engenharia da Computação ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Projeto Final de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Projeto Final de Graduação pode ser reproduzida sem autorização por escrito do autor.

DAVID GONÇALVES MENDES

Depto. de Ciência da Computação (CIC)

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Agradecimentos

Gostaria de agradecer, inicialmente, aos meus familiares. Aos meus pais, Rogéria Eulino Gonçalves e Dilcemar de Paiva Mendes por todo amor e suporte que me proporcionaram durante toda a vida para a minha formação, não somente acadêmica, mas também como pessoa. Ao meu irmão, Marcus Vinícius Gonçalves Mendes que sempre serviu de inspiração para mim, inclusive me influenciando a optar pelo desenvolvimento deste projeto na área de eletrônica. À minha irmã, Julia Gonçalves Mendes por estar sempre presente e ter me suportado e apoiado por todos esses anos. À minha tia Rosemary Eulino Gonçalves, por ter acreditado em mim, me proporcionado um bom computador quando ainda estava no começo o curso, o qual foi essencial para o bom andamento do meu processo de graduação.

Também gostaria de deixar meus agradecimentos à Struct, empresa júnior de engenharia de Computação, pela base que me proporcionou para atuar profissionalmente no mercado de trabalho, por todas as experiências inesquecíveis que pude vivenciar e por todas as pessoas incríveis com as quais pude conviver como membro e como diretor dessa instituição. Um agradecimento especial aos amigos Gabriel Cruz, Felipe Vaz, Arthur Mota, Artur Padovesi, Gabriel Xavier, Bernardo Lobo, Pedro Venzi, Nicolas Paulin, Pedro Ximenes, Luiz Henrique, Caetano Korilo e Arthur de Sá,.

Agradeço também à todos os meus amigos do curso que conheci ao longo do período de graduação. Em especial, ao meu amigo Gustavo Pereira Chaves, que esteve ao meu lado me ajudando em grande parte das disciplinas desde o primeiro semestre no qual ingressamos na UnB.

À Switch Dreams, empresa em que trabalho, por ter fornecido um domínio e acesso a um servidor para que fosse possível colocar a plataforma web no ar.

Ao meu orientador, Daniel Chaves Café, por toda a ajuda, paciência e suporte que me possibilitaram concluir o desenvolvimento deste projeto.

Do fundo do coração, muito obrigado a todos vocês!

DAVID GONÇALVES MENDES

RESUMO

A má disposição de lixo gerado, principalmente, pelos grandes centros urbanos, causa graves impactos para o meio ambiente. A coleta seletiva é uma política que visa a coleta e disposição dos resíduos de forma adequada e ao reaproveitamento do que for possível de inserir na cadeia de produção de novos materiais. No entanto existe um alto custo associado à adoção dessa medida, fazendo com que muitos lugares ainda optem por manter o descarte regular de lixo. O objetivo deste trabalho é desenvolver um dispositivo de pesagem, de baixo custo, integrado com uma plataforma web que forneça dados relacionados à variação de volume de lixo e que possam ser utilizados para otimizar o processo de recolhimento de resíduos e, conseqüentemente, reduzir alguns dos custos associados à implementação da política de coleta seletiva.

Para o desenvolvimento desse sistema, foi construído uma estrutura de pesagem utilizando-se de quatro células de carga na configuração de uma ponte de Wheatstone, um módulo amplificador HX711 e um microcontrolador ESP32, responsável pela submissão dos dados via protocolo MQTT. Os dados são então processados pelo framework Ruby on Rails e armazenados em um banco de dados relacional PostgreSQL, sendo posteriormente mostrados em uma página web construída utilizando a biblioteca React.

Para validar o funcionamento do sistema, foi utilizado um cesto de roupa suja de forma a simular o armazenamento e recolhimento de resíduos sólidos, devido às suas similaridades quanto ao comportamento, uma vez que ambos enchem gradualmente até serem recolhidos, sofrendo uma queda abrupta no volume armazenado.

A análise dos dados consistiu na construção de um gráfico com as medições do dispositivo, permitindo a observação da variação de peso sobre a balança ao longo do tempo, possibilitando o processamento dessas informações para calcular a taxa de variação média do sistema e a estimativa do tempo restante para que o sistema atinja sua máxima capacidade, permitindo o planejamento mais eficaz para o momento da próxima coleta.

Palavras-chave: IoT, ESP32, Sistema Web, Protocolo MQTT, Resíduos Sólidos, Coleta Seletiva.

ABSTRACT

The poor disposal of waste generated, mainly by large urban centers, causes serious impacts on the environment. Selective waste collection is a policy that aims to collect and dispose the waste appropriately and even reuse it, whenever it's insertion into the production chain of new materials is possible. However, there is a high cost associated with adopting this measure, which causes

many places to maintain regular waste disposal policies. This project consists of implementing a low-cost weighing device, integrated with a web platform that provides information related to the waste volume variation that can be used to optimize the waste collection process and, consequently, reduce some of the costs associated with the implementation of the selective waste collection policy.

To develop this system, a weighing structure was built using four load cells in the configuration of a Wheatstone bridge, an HX711 amplifier and an ESP32 microcontroller, responsible for submitting data via MQTT protocol. The data is then processed by the Ruby on Rails framework and stored in a PostgreSQL relational database, and is subsequently displayed on a web page built using the React library.

To validate the functioning of the system, a laundry basket was used to simulate the storage and collection of solid waste, due to their similarities, since both gradually fill up until they are collected, resulting in abrupt drop in the volume stored.

Data analysis consisted in the construction of a graph with the device measurements, allowing observation of weight variation on the scale over time, making it possible to process this information to calculate the average rate of measurements variation and be able to estimate the time remaining for the system to reach its maximum capacity, which allows a more effective planning for the next collection time.

Keywords: IoT, ESP32, Web System, MQTT Protocol, Garbage, Selective Collection.

SUMÁRIO

| | | |
|----------|--------------------------------|-----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | GERAÇÃO E TRATAMENTO DE LIXO | 1 |
| 1.2 | COLETA SELETIVA | 2 |
| 1.3 | ECONOMIA | 3 |
| 1.4 | OBJETIVOS | 3 |
| 1.4.1 | GERAL | 3 |
| 1.4.2 | ESPECÍFICOS | 4 |
| 1.5 | PROPOSTA DE SOLUÇÃO | 4 |
| 1.5.1 | ARQUITETURA | 5 |
| 2 | REVISÃO BIBLIOGRÁFICA | 7 |
| 2.1 | FUNDAMENTAÇÃO TEÓRICA | 7 |
| 2.1.1 | MICROCONTROLADORES | 7 |
| 2.1.2 | CÉLULAS DE CARGA | 9 |
| 2.1.3 | PROTOCOLO MQTT | 13 |
| 2.1.4 | APLICAÇÃO WEB | 15 |
| 2.2 | APLICAÇÕES SIMILARES | 18 |
| 3 | METODOLOGIA | 19 |
| 4 | RESULTADOS | 21 |
| 4.1 | MONTAGEM | 21 |
| 4.2 | SOFTWARE | 22 |
| 4.2.1 | ESP32 | 22 |
| 4.2.2 | DASHBOARD | 24 |
| 4.3 | VALIDAÇÃO | 25 |
| 4.4 | CONFIGURAÇÃO DO AMBIENTE | 25 |
| 4.5 | MEDIÇÕES | 26 |
| 4.5.1 | CÁLCULOS | 26 |
| 4.5.2 | DADOS OBTIDOS | 28 |
| 5 | CONCLUSÃO | 32 |
| 5.1 | ANÁLISE DOS RESULTADOS | 32 |
| 5.2 | PRÓXIMOS PASSOS | 32 |
| 5.2.1 | MELHORIA NA CAPTAÇÃO DOS DADOS | 32 |
| 5.2.2 | PERFORMANCE | 33 |
| 5.2.3 | ESCALABILIDADE | 33 |

| | | |
|--|---------------------------------|-----------|
| 5.2.4 | SEGURANÇA | 34 |
| 5.2.5 | MONITORAMENTO E AUTOMAÇÃO | 35 |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | | 36 |

1 INTRODUÇÃO

1.1 GERAÇÃO E TRATAMENTO DE LIXO

A geração de lixo é um processo que sempre esteve presente na sociedade, se agravando principalmente nas últimas décadas devido ao considerável crescimento dos núcleos urbanos e do fortalecimento da cultura do consumo de bens materiais. Com o aumento da população em todo o planeta, os impactos da má disposição de resíduos começaram a aparecer, como por exemplo o aquecimento global. Apesar disso, no Brasil, algumas providências quanto a esses impactos ambientais, só começaram a ser tomadas, de forma efetiva recentemente. Destaca-se a promulgação da lei de nº 12.305/2010, que instituiu a Política Nacional de Resíduos Sólidos no país, a qual determinava a extinção de lixões a céu aberto (Brasil 2010), devendo o lixo coletado ser mantido de forma apropriada.

Os resíduos gerados podem ter diferentes destinos. Frequentemente, são descartados de forma irregular em áreas popularmente conhecidas como lixões, ou vazadouros, nas quais não existe um tratamento adequado do lixo. Como consequência, quando em contato com a água da chuva, os diferentes tipos de resíduos se misturam com o chorume do lixo orgânico e podem chegar a contaminar o solo e o lençol freático da região. Além disso, nesses ambientes são gerados gases tóxicos, como o metano, que contribuem para o efeito estufa, agravando o aquecimento global. Outra consequência do descarte em lixões é a ameaça à saúde da população local, uma vez que esse ambiente atrai animais que podem agir como vetores de proliferação de doenças.

Por conta desses problemas, o lixão a céu aberto se tornou ilegal no Brasil, devendo ser erradicados pelos municípios do país até 2024. Apesar disso, como noticiado pela rede globo, cerca de um terço dos municípios ainda não acabou completamente o uso de lixões, no entanto, como afirma o presidente da Confederação Nacional de Municípios, Paulo Ziulkoski, não há recursos financeiros suficiente nas prefeituras para abolir essa prática (Globo 2023).

Como principal alternativa aos lixões, os aterros sanitários são áreas com uma estrutura mais adequada para o armazenamento dos resíduos, contendo um sistema de drenagem de gases (que podem ser reaproveitados para a geração de energia) e de chorume. Os aterros também são construídos com uma base constituída de uma camada impermeável para evitar o vazamento de líquidos para o solo. Antes de serem encaminhados para os aterros, o lixo pode passar por um processo de incineração, no qual é queimado em temperaturas elevadas a fim de reduzir seu volume, podendo também gerar, como subproduto, energia térmica (posteriormente convertida em elétrica) e materiais passíveis de reaproveitamento econômico.

A reciclagem é um dos processos mais desejáveis para o tratamento dos resíduos sólidos, uma vez que consiste na reutilização desses materiais já descartados, na cadeia de produção de novos materiais. Já para a matéria orgânica descartada, o processo amplamente utilizado é conhecido

como compostagem, em que esses resíduos são utilizados como matéria-prima para a produção de adubo natural, substituto de produtos químicos utilizados na agricultura. A compostagem também contribui para a redução do aquecimento global, já que é esse tipo de resíduo que, ao entrar em decomposição, gera os gases tóxicos que agravam o efeito estufa.

1.2 COLETA SELETIVA

A coleta seletiva de lixo pode ser definida como um processo no qual resíduos descartados são separados para o recolhimento de acordo com suas características, podendo ser classificados principalmente como plástico, papel, metal, vidro, orgânico e rejeito. Essa separação é de extrema importância, pois facilita todos os processos de tratamento citados acima. Na ocorrência da coleta regular de lixo, muitos materiais perdem seu potencial de reciclagem devido à mistura de diferentes tipos de resíduos.

Apesar da importância da coleta seletiva, ao verificar os dados de pesquisas relacionadas ao assunto, não são encontradas muitas informações confiáveis para mensurar exatamente o estágio no qual cada um dos municípios brasileiros se encontram. Como descrito por Leonardo e Elimar, há vários fatores que influenciam uma baixa confiabilidade nas pesquisas (Conke e Nascimento 2018). Alguns desses fatores são a definição da amostragem, a qual deve ser feita preferencialmente de forma aleatória; a mensuração dos dados, que deve consistir em perguntas quanto ao estágio atual da coleta seletiva ao invés de perguntas binárias sobre a presença dessa ação; e o agente de comunicação, que não deve ser um agente único (como a prefeitura dos municípios), visto que pode resultar em respostas enviesadas, não representando a situação real da coleta seletiva na região.

Após analisar de forma crítica o resultado de diversas pesquisas, foi verificado que em 2014, a coleta seletiva estaria presente em menos de metade dos municípios do país, além de ainda haver muito desperdício, com cerca de 10% dos resíduos recicláveis sendo bem reaproveitados. Mesmo anos depois, não foi percebida uma melhora nessa situação. Como reportado pela rede Globo, de mais de 80 milhões de toneladas de resíduos gerados no país em 2022, cerca de apenas 4% foram reaproveitados (Globo 2023). Um dos fatores que influencia esses resultados é a falta de infraestrutura (como balanças de pesagem), o que resulta em uma falta de dados palpáveis para mensurar regularmente e eficientemente o estado da coleta seletiva.

Em um estudo de caso realizado por Bravo et. al., também foi mostrado que a sociedade deve colaborar e estar em sintonia para que a coleta seletiva ocorra de forma eficiente (Bravo et al. 2018). Muitas vezes, a coleta é atrapalhada pela falta de conhecimento das pessoas quanto ao destino de cada tipo de resíduo. Os benefícios trazidos pela coleta seletiva é um conhecimento que deveria ser construído desde a base escolar dos indivíduos.

1.3 ECONOMIA

Do ponto de vista econômico, é necessário considerar o custo relacionado à coleta e ao tratamento de resíduos. Em um estudo de caso realizado na cidade de Palmas, TO, por Waldecy Rodrigues e Willian Cardoso Santana, foi adotado o Método de Valoração de Contingente (MVC) a fim de verificar a viabilidade econômica da implantação da coleta seletiva na cidade (Rodrigues e Santana 2012). Nesse estudo, foi apurada uma pesquisa sobre a Disposição a Pagar da população pelo uso da coleta seletiva em Palmas, o que foi utilizada como métrica para estimar os benefícios econômicos dessa atividade, sendo posteriormente comparada com o real custo de sua implantação.

Uma questão interessante levantada nessa pesquisa, foi do quanto a educação dos indivíduos é importante para o processo da conscientização em relação à coleta seletiva. Foi apurado que, em Palmas, para cada ano de escolaridade dos contribuintes de imóveis residenciais e não residenciais, sua disposição a pagar pela coleta seletiva tem um aumento de cerca de 0,05% e 0,10%, respectivamente.

Por fim, o resultado obtido foi de que a disposição a pagar total da população sobre o sistema de coleta de resíduos urbanos seria em torno de 1 milhão de reais, enquanto que o custo real desse sistema é estimado em 9,6 milhões de reais, o que resulta em uma relação de aproximadamente 11%. Assim, concluiu-se que a implementação da coleta seletiva não seria viável do ponto de vista puramente econômico. No entanto, foram levantados alguns questionamentos quanto aos demais benefícios dessa prática em relação ao acréscimo dos custos e quanto à possibilidade de adoção de um sistema mais eficaz de coleta e tratamento dos resíduos sólidos.

A fim de diminuir o custo associado à coleta seletiva e otimizar esse processo, seria ideal definir as rotas de forma que os locais com maior volume de resíduos acumulados fossem priorizados. No entanto, a quantidade de resíduos gerados e suportados varia a depender do local, além de que não necessariamente existirá uma constância no volume gerado a cada dia. Sendo assim, como é possível saber exatamente o momento ideal para realizar a coleta de forma que não hajam viagens a mais do que o necessário e, conseqüentemente, desperdício de recursos?

1.4 OBJETIVOS

1.4.1 Geral

O objetivo geral deste trabalho é criar um dispositivo de pesagem que seja capaz de aferir o volume de resíduos sólidos armazenados sobre ele e enviar essa informação para uma plataforma que possibilite a visualização da variação do volume e que forneça uma estimativa para a realização da próxima coleta. Para que o objetivo geral seja atingido, é necessário dividi-lo em pequenos objetivos específicos a serem cumpridas.

1.4.2 Específicos

O primeiro objetivo específico consiste no mapeamento e comparação entre os componentes e ferramentas existentes que podem ser utilizados para o desenvolvimento do dispositivo. Em seguida, o próximo objetivo consiste na construção do dispositivo em si, isto é, a estrutura da balança com todos os equipamentos eletrônicos e mecânicos necessários já definidos. Com a estrutura montada, o próximo objetivo é realizar a sua calibragem, levantando as variáveis necessárias para o seu bom funcionamento, ou seja, garantir que as medições de peso estão sendo devidamente efetuadas e retornando os valores corretos. O objetivo seguinte é o desenvolvimento do software responsável por fazer com que todo o sistema que envolve o microcontrolador funcione, efetuando as leituras de carga e as enviando pela rede para que possam ser acessadas por uma aplicação externa. Com os dados de leitura sendo propriamente enviados, o próximo objetivo consiste no desenvolvimento de um processo que seja capaz de recebê-los e armazená-los em banco de dados para que não sejam perdidos e possam ser utilizados posteriormente. Por fim, o último objetivo envolve o desenvolvimento de um software que disponibilize os dados de medições, em conjunto com algumas estatísticas, em uma interface de usuário (UI) para o usuário final do sistema.

A implementação desse sistema em vários pontos de coleta é importante pois com ele será possível otimizar dinamicamente as rotas de recolhimento de resíduos sólidos, evitando trajetos desnecessários e, conseqüentemente, o custo associado à adoção da política de coleta seletiva.

1.5 PROPOSTA DE SOLUÇÃO

Para projetar uma solução, será tomado como ponto de observação o contexto da coleta de resíduos dentro da Universidade de Brasília, na qual a coleta ainda não é otimizada considerando a quantidade de lixo gerada. Nesse ano, a SeMA (Secretaria do Meio Ambiente da UnB) está promovendo campanhas de coleta seletiva de lixo, visando a separação dos resíduos entre rejeito, orgânico e reciclável. Além disso, também foi aberto um edital para a realização da coleta dos resíduos sólidos recicláveis nos campi de planaltina (FUP) e do gama (FGA).

Com isso em mente, a proposta desse projeto consiste na criação de um mecanismo que seja capaz de medir, em tempo real, o volume de lixo reciclável gerado pelos departamentos da universidade. O sistema deve tornar essas informações disponíveis para que o agente responsável pela coleta encontre o momento ideal para o recolhimento dos resíduos, otimizando assim esse processo. Dessa forma, espera-se que a utilização desse mecanismo viabilize uma redução parcial dos custos associados à implementação e manutenção da coleta de lixo seletiva nos centros urbanos, tornando-se uma opção para que municípios que ainda não detêm dessa prática tenham mais condições financeiras de aplicá-la.

Para isso, inicialmente será realizada a construção de um aparelho (Hardware) capaz de medir a massa de lixo que foi gerada e que está sendo armazenada em determinado local. Além disso,

é importante que este aparelho esteja conectado à uma conexão sem fio a qual o possibilite se comunicar com um software, desenvolvido para a exibição dos dados de massa coletados pelo aparelho. Nesse software, deve ser possível acompanhar, ao longo do tempo, qual a massa de lixo está atualmente sendo armazenada no local, além de poder disponibilizar outras informações úteis, tais como a média de crescimento e o valor máximo medido/suportado, que possibilitariam a realização de previsões quanto à data em que se tornará necessária a realização da próxima coleta dos resíduos.

1.5.1 Arquitetura

A solução proposta, ilustrada na Figura 1.1, diz respeito a um sistema de pesagem no qual os resíduos sólidos armazenados em lixeiras da universidade (A) serão previamente coletados (B) e armazenados sobre uma balança (C) construída a partir de células de carga e um microcontrolador (D), também responsável por enviar os dados coletados regularmente, via rede sem fio. Os resultados obtidos serão então armazenados em um banco de dados (E) integrado com uma aplicação web. Esta consiste em uma plataforma que irá exibir as informações relevantes de forma acessível e intuitiva para o usuário, possibilitando a visualização da quantidade de lixo a ser coletado e de um histórico envolvendo essa informação (F). Com isso, será possível monitorar o volume de resíduos gerados e coletá-los apenas quando necessário (G), isto é, quando próximo de atingir o limite de capacidade de armazenamento.

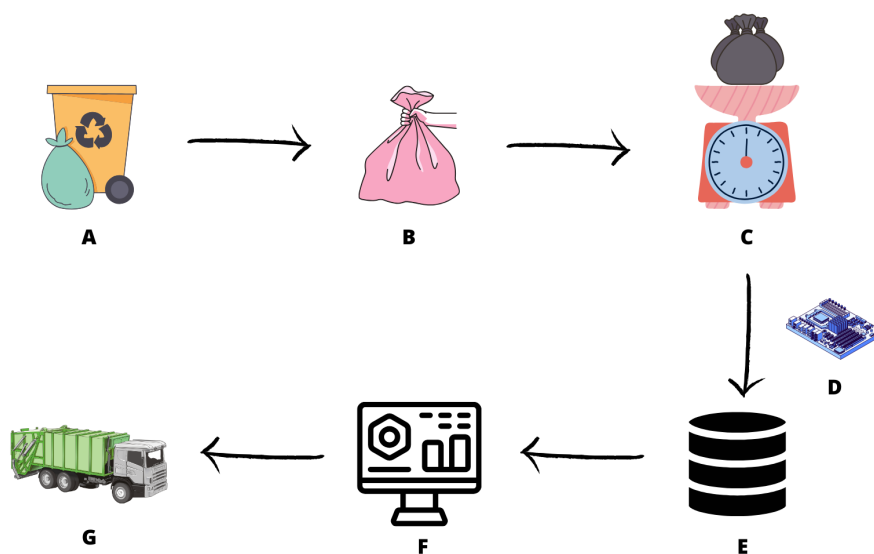


Figura 1.1: Ciclo da coleta de lixo

No próximo capítulo, será apresentada a fundamentação teórica, contendo os principais conceitos necessários para o entendimento do projeto a ser desenvolvido, assim como a apresentação de algumas aplicações similares. Em seguida, no capítulo *metodologia*, será descrito o passo a passo realizado para o desenvolvimento do sistema como um todo, cujos resultados serão apre-

sentado no capítulo seguinte. Por fim, no capítulo *conclusão*, será feita uma análise sucinta dos resultados e apontados alguns pontos de futuras melhorias para o projeto.

2 REVISÃO BIBLIOGRÁFICA

2.1 FUNDAMENTAÇÃO TEÓRICA

2.1.1 Microcontroladores

Um microcontrolador (MCU) é um pequeno computador com um único chip de circuito integrado (System on Chip, ou SoC), podendo conter uma ou mais CPUs, memória e periféricos. São projetados principalmente para o desenvolvimento de sistemas embarcados, que podem ser definidos como conjuntos de Hardware e Software projetados para a realização de uma tarefa específica (Souza 2022). Dos mais conhecidos no mercado é possível citar: AVR, ESP32, Raspberry Pi Pico e MSP430.

Os AVR são microcontroladores desenvolvidos pela empresa Atmel e são bastante populares por estarem presentes dentro das placas de Arduino, que é uma plataforma de desenvolvimento de sistemas eletrônicos. O Arduino é amplamente utilizado, principalmente pelo fato de ter sido projetado para ser de baixo custo e de fácil uso e aprendizagem, até para quem não possui qualquer conhecimento em eletrônica e programação (Kashif 2022). Faz o uso de uma versão simplificada da linguagem C/C++ e possui uma ampla variedade de códigos *open-source* desenvolvidos pela comunidade, o que resulta em uma vastidão de bibliotecas já prontas para aplicações comuns. Por outro lado, os microcontroladores AVR não têm uma performance muito boa quando comparados com os concorrentes, não possuem suporte nativo as comunicações Wi-Fi e Bluetooth e nem a algumas linguagens de programação famosas (como Java e Python). Além disso, têm uma baixa capacidade de memória no geral, tendo alguns modelos com memória flash de 32KB, o que possibilita o carregamento de códigos da ordem de apenas algumas centenas de linhas.

Assim como o AVR, o MSP430 é um microcontrolador de baixo custo e de fácil aprendizagem, apesar de não ser tão popular, sendo inclusive utilizado, no momento de escrita desse texto, para a capacitação dos estudantes de sistemas microprocessados na Universidade de Brasília. Desenvolvido pela empresa americana Texas Instruments, o MSP30 tem como principal diferencial o baixo consumo de energia e é uma boa opção para quem está começando a aprender sobre eletrônica e sistemas embarcados, principalmente por conta de sua excelente documentação. A empresa também disponibiliza de um ambiente de desenvolvimento integrado (IDE) amigável para o usuário que facilita o desenvolvimento e depuração dos códigos desenvolvidos. Em contrapartida, as placas da família MSP430 possuem algumas das mesmas desvantagens do Arduino, como baixa memória flash e poder de processamento.

Diferentemente dos citados acima, as placas Raspberry Pi surgiram inicialmente como pequenos computadores voltados para o ensino de noções básicas de computação, dispondo de vasta gama de recursos como Wi-Fi, Bluetooth, saída HDMI, USB, entre outros. Após algum tempo, surgiram modelos mais voltados para o mercado de microcontroladores, como é o caso do Rasp-

berry Pi Pico, o qual possui 2 núcleos com arquitetura ARM de 32 bits e clock de 133MHz. O Raspberry Pi Pico possui também suporte à linguagem Python e uma memória bastante robusta, com 264KB de memória RAM e 2MB de memória flash (Olujinmi 2022). Em contrapartida, esse modelo também não tem Wi-Fi e Bluetooth embutidos, além de apresentar uma maior complexidade na implementação de sistemas em C/C++, quando comparado aos microcontroladores já citados.

Por fim, o ESP32 é um microcontrolador de baixo custo desenvolvido e mantido pela empresa Espressif Systems. Possui embutido o suporte às tecnologias de Wi-Fi e Bluetooth, clock de até 240MHz, memória Ram de 520KB e flash de 4MB. Apesar de, em modo ativo, apresentar, um consumo energético maior quando comparado ao Raspberry Pi Pico, o ESP32 apresenta modos de baixo consumo bem eficientes. Em modo de hibernação, por exemplo, pode reduzir esse consumo de 240 mA para 5 μA , representando uma redução de cerca de 99,998%. A família ESP32 apresenta uma variação de linhas, cada uma com algumas particularidades em relação à arquitetura e funcionalidades disponíveis. Um dos principais pontos negativos desse microcontrolador é a sua curva de aprendizado, sendo um MCU mais complicado de se programar se comparado aos concorrentes do mercado.

2.1.1.1 ESP32-C3

O microprocessador do ESP32-C3 é construído na arquitetura RISC-V de 32 bits, de núcleo único, de até 160 MHz. Possui também recursos de memórias cache (16 KB), SRAM (estática, 400KB), ROM (Somente leitura, 384KB) e RTC (comunicação em tempo real, funciona no modo de repouso de energia; 8KB), além de uma interface JTAG, definida pela norma IEEE 1149.1 como um padrão utilizado para uma ampla variedade de testes em placas de circuito impresso, facilitando a detecção de falhas nesses sistemas (Guimarães, Farias e Bandeira). Esse MCU também está integrado com uma variedade de periféricos, tais como I2C, I2S, UART, LED PWM, entre outros. Seu diagrama de blocos completo pode ser visualizado na Figura 2.1.

A programação do microcontrolador pode ser realizada a partir do carregamento, na memória, de código em linguagem binária compatível com a sua arquitetura. Para a geração deste código, a própria Espressif disponibiliza ferramentas e bibliotecas auxiliares que permitem a compilação de código em linguagem de alto nível (C/C++), traduzindo-o em linguagem de máquina. Informações sobre a configuração do ambiente de desenvolvimento, assim como os recursos disponíveis para cada uma das linhas do ESP32, estão bem documentadas e podem ser encontradas no guia de programação da ESP-IDF mantido pela Espressif.

Para esse projeto, foi optado pelo uso do ESP32-C3, principalmente por sua relação de custo-benefício e por já possuir suporte integrado à tecnologia de Wi-Fi, de suma importância para o desenvolvimento do sistema proposto.

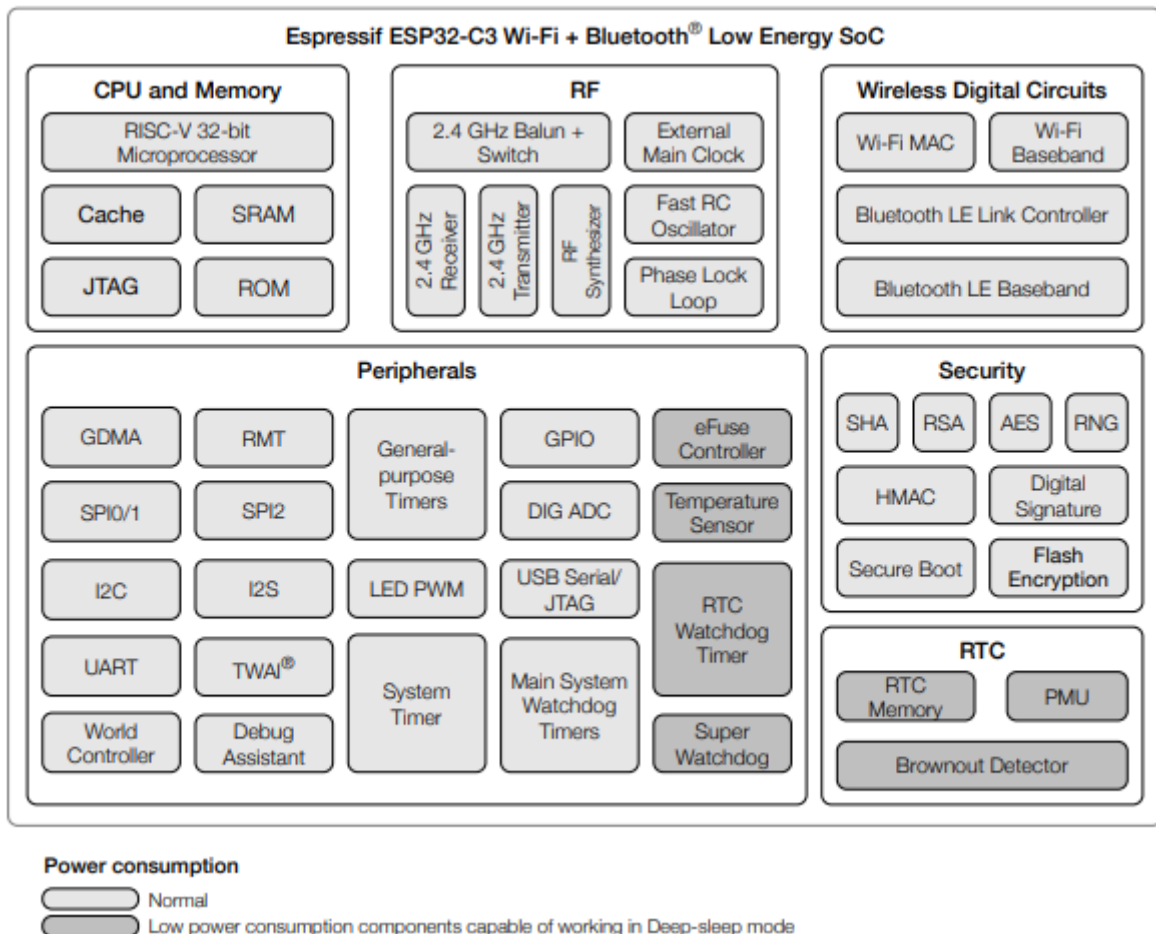


Figura 2.1: Diagrama de blocos ESP32C3

2.1.2 Células de carga

Uma célula de carga é um transdutor capaz de converter uma entrada de força, tal como tensão, compressão e pressão em um sinal elétrico (contributors 2023). Ela é projetada de tal forma que, conforme a força aplicada à célula de carga aumenta, o sinal muda proporcionalmente. O tipo mais utilizado é a célula de carga de extensômetro, que consiste em um fio disposto sobre um suporte flexível de tal forma que, ao ser deformado, gera uma variação linear na resistência do fio. Pela segunda lei de Ohm, a resistência de um condutor é diretamente proporcional à seu comprimento e inversamente proporcional à sua área, como mostrado na Equação 2.1, em que L e A são o comprimento e a área transversal do condutor, respectivamente, e ρ é a resistividade do material.

$$R = \frac{\rho L}{A} \quad (2.1)$$

Ao ser aplicada uma força de compressão sobre o fio, que ocorre ao adicionar mais peso sobre a célula de carga, ele será encurtado, fazendo com que sua área transversal aumente e que seu comprimento diminua, resultando em uma diminuição na resistência. O oposto ocorre quando o

fio é tensionado, aumentando a sua resistência, como mostrado na figura 2.2. Para esse projeto, foi utilizada uma célula de carga de 50 kg de capacidade, cujo diagrama pode ser visto da Figura 2.3.

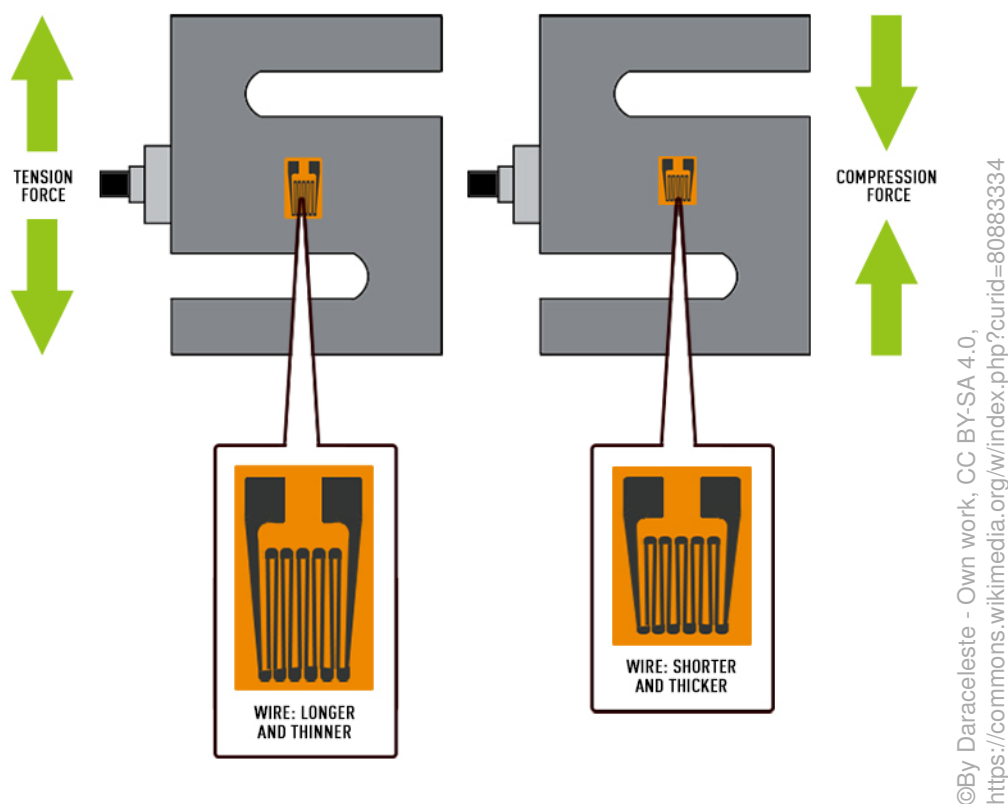


Figura 2.2: Funcionamento de célula de carga de extensômetro

2.1.2.1 Ponte de Wheatstone

Como a resistência medida por uma única célula de carga é bem baixa, há uma dificuldade elevada de apurar as mudanças com precisão. Portanto, uma técnica amplamente adotada é a *Ponte de Wheatstone*, que consiste na montagem de uma configuração composta por 4 células de carga idênticas conectadas a uma tensão conhecida de entrada, como mostrado na figura 2.4. Sendo V_0 a tensão de saída do circuito, esse valor será nulo caso o sistema esteja balanceado e caso a resistência de um único resistor mude, essa tensão também será alterada. Usando como base a Lei de Ohm, que diz que $U = R \cdot I$, (sendo R a resistência, U a tensão e I a corrente), é possível analisar o circuito e obter a seguinte relação:

$$V_0 = \left(\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) \cdot V_{EX} \quad (2.2)$$

Em que R_N representa cada um dos resistores do circuito e V_{EX} a tensão de entrada do sistema. A partir da oscilação do valor medido em V_0 , é possível calcular o peso que está causando

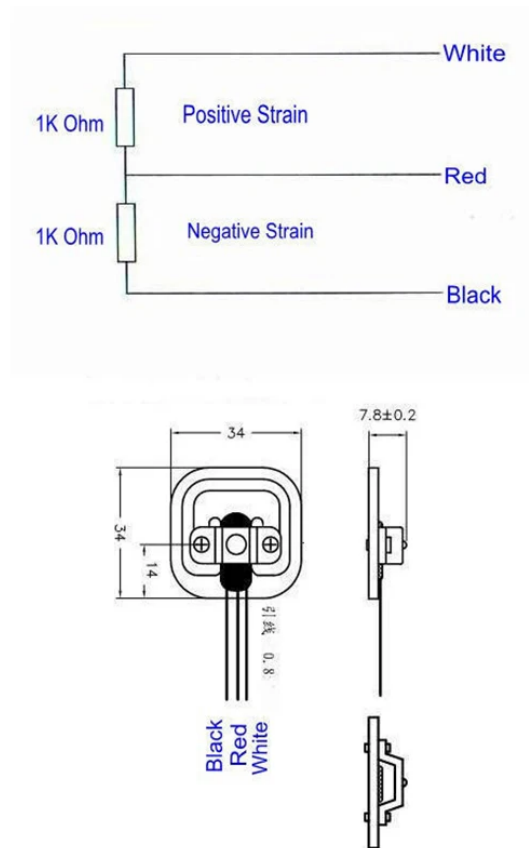
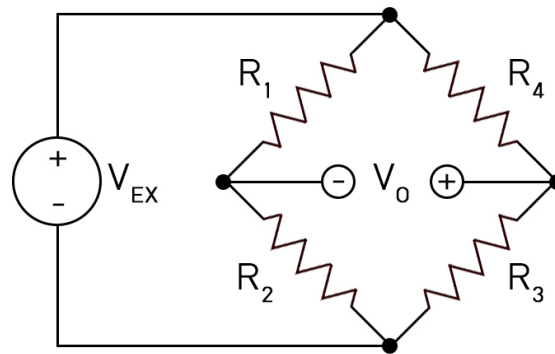


Figura 2.3: Diagrama célula de carga de 50Kg

essa variação. Dependendo da célula de carga utilizada (ou até de outros fatores externos como a temperatura do local), mesmo para uma tensão de entrada constante e uma mesma configuração de ponte de Wheatstone, ainda é possível que V_0 varie de forma diferente para diferentes projetos. Por conta disso, não é possível definir com exatidão uma relação entre peso e tensão para todas as células de carga, tornando-se necessária a realização de testes empíricos fazendo-se do uso de massas conhecidas a fim de efetivamente calibrar esse sistema visando obter medidas confiáveis do sistema de pesagem.

Existem 3 configurações possíveis para uma ponte de Wheatstone. No primeiro caso, apenas uma única resistência será variável e as demais resistências serão determinadas de acordo com Equação 2.2 para o caso em que $V_0 = 0$ e o transdutor não está sofrendo forças de tensão ou compressão (resultando na relação $R_2R_3 = R_1R_4$). No segundo caso, utiliza-se 2 resistências variáveis, que devem ser dispostas do mesmo lado (R1 e R2 ou R3 e R4) mas em sentidos opostos, para que a variação em uma das resistências não anule a outra. Na última configuração, todas as resistências são variáveis, sendo dispostas seguindo a mesma lógica do caso anterior, tomando-se os devidos cuidados com o a orientação das células de cargas para que as variações de resistência se somem (ao invés de se anularem), consequentemente aumentando a sensibilidade do sistema.



©By Daraceleste - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=80883335>

Figura 2.4: Configuração Ponte de Wheatstone

2.1.2.2 HX711

O HX711 é um conversor analógico digital com precisão de 24 bits projetado para aplicações de pesagem utilizando células de carga. Seu funcionamento consiste em converter alterações nos valores de resistência para dados digitais, com o objetivo de amplificar os sinais medidos e, assim, aumentar a precisão do sistema. A integração entre esse módulo e o microcontrolador é relativamente simples e seus sinais de controle são conduzidos pelos pinos sem a necessidade de programar os registradores internos do chip. A fonte de alimentação regulada disponibilizada pelo chip é capaz de fornecer energia o suficiente para o sensor externo e para o conversor AD da placa, não sendo preciso o uso de fontes de energia adicionais na placa. As Figuras 2.5 e 2.6 representam o circuito integrado desse módulo e uma imagem do módulo em si, respectivamente.

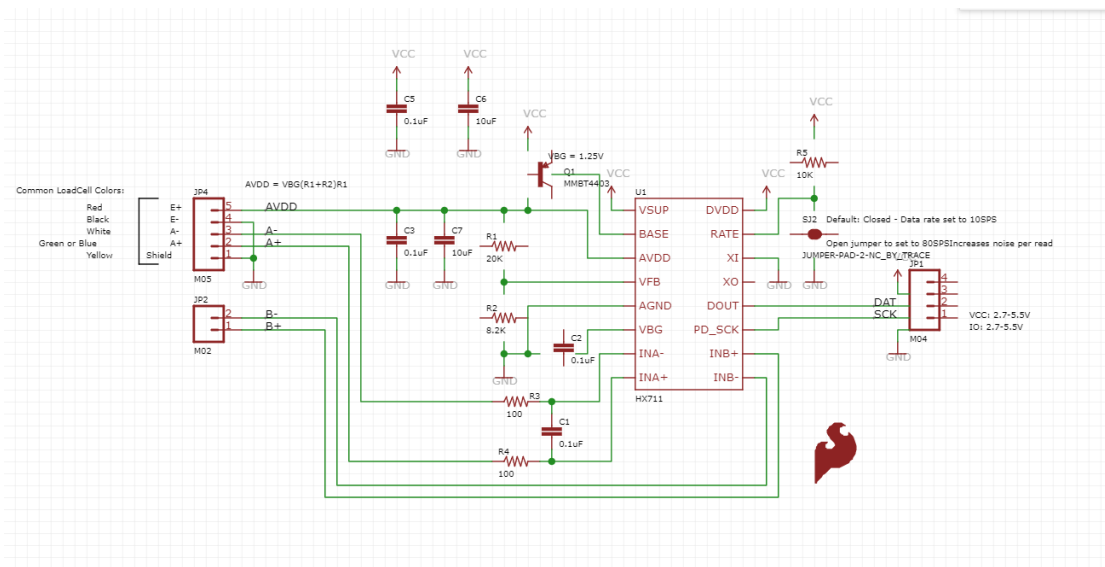


Figura 2.5: Circuito do módulo HX711

A comunicação deste módulo com o microcontrolador é feita através de uma interface serial síncrona, em que cada bit dos dados é transmitido por um único pino (DT) de forma sequencial.

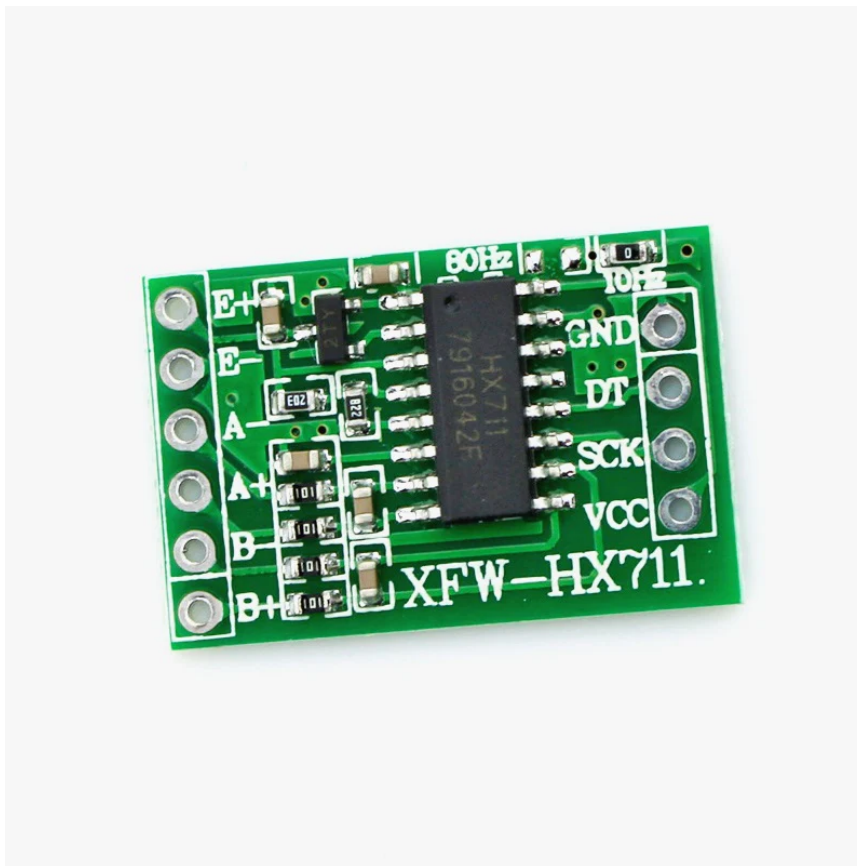


Figura 2.6: Módulo HX711

Também contém um pino de entrada (SCK) para temporização desse processo, sendo necessário para permitir que a sincronização entre o microcontrolador e o HX711.

Como descrito em seu datasheet, o funcionamento da interface serial desse módulo é da seguinte forma: quando os dados ainda não estão prontos para leitura, os pinos DT e SCK ficam em alto e em baixo nível, respectivamente (Semiconductor). Quando os dados estão prontos para serem lidos, DT fica em baixo nível. São então aplicados de 25 a 27 pulsos de clock de alto nível no pino SCK, em que cada pulso desloca um bit de dados do pino de DT, começando a partir do mais significativo. O 25º pulso coloca DT de volta em alto nível, como pode ser visto na Figura. O ganho e o canal da próxima conversão podem ser controladas a partir da quantidade de pulsos na conversão anterior, podendo ser 25 (canal A com ganho de 128), 26 (canal B com ganho de 32) ou 27 (canal A com ganho de 64).

2.1.3 Protocolo MQTT

Como descrito pela *Amazon Web Services*, plataforma de serviços de computação em nuvem da Amazon, MQTT, ou *Message Queue Telemetry Transport*, é um protocolo de mensagens de rede de comunicação máquina para máquina que funciona sobre o protocolo TCP/IP, sendo comumente utilizado para aplicações de Internet das Coisas (IoT) (AWS). Como grande parte dos

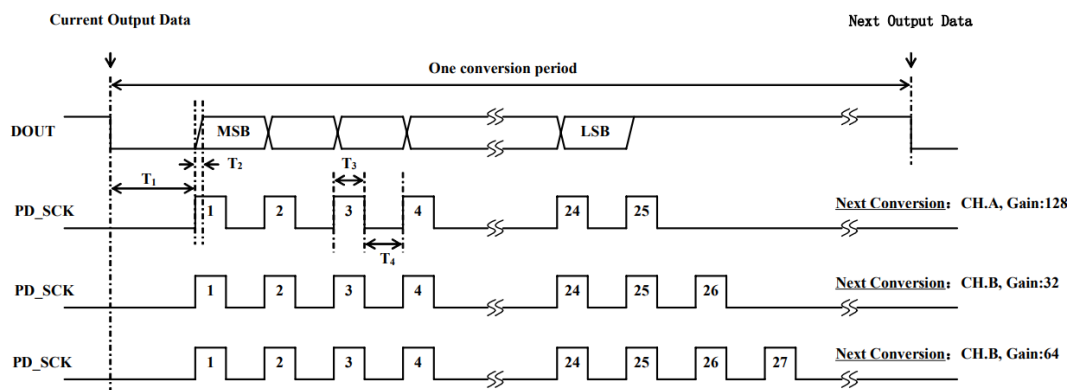


Figura 2.7: Interface de comunicação serial do módulo HX711

dispositivos nesse meio apresentam uma limitação de recursos, o MQTT é a principal opção de protocolo de comunicação adotada, uma vez que não requer muitos recursos e utiliza cabeçalhos pequenos, os quais contribuem para a otimização da largura de banda da rede em que estiver operando. Também é possível afirmar que esse protocolo é altamente escalável, permitindo sua integração com inúmeros dispositivos, além de apresentar uma boa confiabilidade.

O Protocolo MQTT funciona com base no modelo de publicação/assinatura, no qual as mensagens são enviadas para um agente que fica responsável por filtrar as mensagens dos remetentes e encaminhá-las corretamente para os destinatários. Dessa forma, o remetente e o destinatário não possuem conhecimento da localização de rede um do outro, não apresentam conectividade de rede simultaneamente e nem precisam estar sincronizados para enviar ou receber as mensagens.

Os principais componentes do protocolo são: os clientes (podem ser publicador ou destinatário), que são os dispositivos que se comunicam via MQTT na rede; o agente (ou *broker*), que é o sistema responsável pelo gerenciamento das mensagens e pela autorização dos clientes; e a conexão MQTT, que se inicia com o envio de uma mensagem *CONNECT* solicitando conexão ao *broker*, que é confirmada através de uma mensagem *CONNACK* (*connect acknowledgement*).

Após estabelecer conexão com o agente, o cliente é livre para publicar e receber mensagens, podendo inclusive especificar um tópico no qual essas mensagens serão enviadas/recebidas. Os tópicos são uma forma de o agente organizar as mensagens por meio de palavras chaves, que funcionam de forma hierárquica. Por exemplo, um cliente conectado pode enviar uma mensagem para um tópico chamado *unb/ft*. Assim, qualquer cliente interessado que tenha se inscrito nesse tópico em particular irá receber essa mensagem. Também é possível utilizar "coringas" para receber as mensagens de todos os subtópicos de um determinado tópico, sem precisar se inscrever em um tópico específico.

Um fator importante de ser considerado quanto à utilização do protocolo é que o MQTT não apresenta, por padrão, uma forma de lidar com questões de segurança e confiabilidade das mensagens trafegadas. Como comentado em um paper publicado por Dan Dinculeană e Xiaochun Cheng, muitos dos agentes não lidam com essa questão, sendo necessária a implementação de al-

um algoritmo complementar de criptografia para proteger essas informações, sendo alguns deles comparados experimentalmente ao longo do texto (Dinculeană e Cheng 2019). Este algoritmo pode ser implementado tanto de ponta a ponta, sendo implementado apenas nos clientes, quanto de agente-cliente, em que o *broker* fica responsável por decifrar as mensagens recebidas de um cliente e novamente as cifrá-las para enviar aos demais clientes, o que acaba exigindo um poder computacional maior por parte do agente MQTT.

2.1.4 Aplicação web

Uma aplicação web trata-se de um sistema de software que executa e funciona com base na infraestrutura da internet (Sacramento 2022). Com a crescente demanda por sistemas IoT (Internet das coisas), aplicações desse tipo são frequentemente utilizadas para a integração de objetos do cotidiano à rede. Existem diversas ferramentas que podem ser utilizadas para o desenvolvimento web, assim como sistemas já prontos, projetados especificamente para o uso em aplicações de IoT. Nos parágrafos que se seguem, serão apresentadas algumas dessas ferramentas e sistemas.

2.1.4.1 Frameworks

Uma opção de dashboard já existente é fornecida pela empresa norte americana de software aberto Adafruit. Possui diversos componentes prontos para serem integrados ao sistema, alguns que permitem até a interação remota com o dispositivo. Para aplicações simples, essa dashboard é uma ótima opção de sistema já pronto que permite uma fácil integração com dispositivos de captação de dados. No entanto, possui algumas limitações, como uma quantidade máxima de dados que podem ser captados por minuto e histórico limitado à 30 dias, isto no plano gratuito. Além disso, por se tratar de um software montado de forma genérica para funcionar com diversas aplicações IoT, existe um limite do quanto é possível personalizá-lo para um caso de uso mais específico.

O desenvolvimento de uma aplicação web completa para a comunicação com dispositivos na rede exige um maior conhecimento sobre os protocolos utilizados e um maior trabalho de implementação. Em contrapartida, permite uma maior customização e controle sobre os dados e como eles serão exibidos para o usuário final. Existem múltiplos frameworks que podem ser utilizados para desenvolvimento web, como é o caso do flask, django, ruby on rails, node, react, entre outros.

Flask e Django são frameworks amplamente utilizados, principalmente por serem baseados na linguagem de programação de alto nível *python*, que é atualmente a mais popular no mundo nos últimos anos, de acordo com o índice de popularidade de linguagens de programação (Carbonelle 2023). De modo geral, ambos funcionam bem para o desenvolvimento de uma aplicação web voltada para integração com um sistema IoT. No entanto, Django apresenta uma robustez um pouco superior em seus recursos. Em contrapartida, Flask é mais simples de se desenvolver, sendo ideal para softwares mais simples implementados por equipes pequenas (Galante 2019).

Node.js e React também são boas opções, liderando o ranking de popularidade entre os frameworks de desenvolvimento web em julho de 2023, ambos com mais de 40% de usuários (Vailshery 2023). São comumente utilizadas em conjunto e a sua popularidade é justificada por serem baseados em javascript, que é a linguagem de programação que executa em todos os navegadores de internet. O Node é um software que permite a execução de códigos javascript fora dos navegadores, enquanto que o React é uma biblioteca que tem como foco a criação da interface visual das páginas web.

Ruby on Rails é um framework que funciona com base na linguagem de programação de alto nível *ruby*, sendo ao mesmo tempo robusta, de simples implementação e com padrões bem definidos. Também apresenta uma grande comunidade ativa e, conseqüentemente, diversas bibliotecas de código aberto prontas para uso.

Inertia é uma ferramenta que permite o desenvolvimento de aplicações javascript como um sistema monolítico, ou seja, possibilita a criação de uma aplicação com os frameworks mais modernos (como React, Vue e Svelte) sem a necessidade de construir uma API para servi-los, o que diminuiria a performance da aplicação. Desta forma, é viável o desenvolvimento de uma interface em React dentro de frameworks monolíticos, como Rails e Django.

Para esse projeto, por questões de familiaridade com as ferramentas e simplicidade do software a ser desenvolvido, optou-se pelo uso do framework Ruby on Rails em conjunto com o banco de dados relacional *PostgreSQL* para o armazenamento e processamentos dos dados de medições. Já para a disponibilização das informações para os usuários, foi utilizado o React para o desenvolvimento da interface de usuário e o Inertia para permitir uma fácil integração do sistema rails com essa UI.

2.1.4.2 Deploy

Como é o caso da maioria das aplicações web, é interessante a realização do *deploy* do sistema, ou seja, o processo de disponibilizá-lo em ambiente de produção para que seja executado continuamente e possa ser acessado pela grande maioria dos usuários com acesso à internet. Normalmente esse processo é feito em servidores na nuvem, visto que possuem uma infraestrutura mais apropriada, sendo mais fáceis de escalar e apresentando baixo risco de inatividade. Ao ser configurada para executar em um servidor (ou mesmo localmente), a aplicação pode ser acessada tanto diretamente pelo IP da máquina quanto pelo domínio. O acesso por um domínio, apesar de mais custoso, é preferível, uma vez que é muito mais difícil para os usuários de gravarem um endereço IP para poderem acessar uma aplicação do que um nome de domínio, além de que, a depender da infraestrutura do servidor utilizado para hospedar a aplicação, é possível que seu IP não seja fixo, tornando ainda mais difícil o acesso correto à plataforma.

A principal ferramenta utilizada pela maioria dos usuários para realizar o deploy é o *Docker*, que se trata de um software de código aberto que permite o gerenciamento de contêineres, que é um conjunto de um ou mais processos organizados isoladamente do sistema (Hat 2022).

Diferentemente de máquinas virtuais, nas quais diferentes sistemas operacionais compartilham recursos de Hardware de uma mesma máquina, os contêineres compartilham o mesmo kernel de um sistema operacional e isola seus processos internos do resto do sistema. A sua utilização é importante pois possibilita a portabilidade e consistência da aplicação em diferentes ambientes, garantindo a sua execução independentemente da máquina em que estão executando.

A utilização do docker para o processo do deploy de aplicações web consiste na escrita de um arquivo *Dockerfile*, que deve conter todos os passos para a execução do programa em uma nova máquina, incluindo a instalação das dependências, bibliotecas e a cópia dos arquivos necessários. A partir da *Dockerfile*, será gerada uma *imagem*, que será um arquivo com as instruções necessárias para a criação e execução de um contêiner docker. Normalmente essas imagens geradas são enviadas e armazenadas em uma plataforma chamada *Docker Hub*, repositório público de imagens docker, no qual podem ser acessadas para serem instaladas e utilizadas em diferentes máquinas. Dependendo da complexidade da aplicação, pode ser necessário mais de um contêiner para executá-la por completo. Nesse caso, uma ferramenta amplamente utilizada em conjunto com o docker é o *Docker Compose*, que tem como propósito justamente gerenciar em conjunto todos os contêineres necessários para a aplicação.

Em ambientes mais complexos, que exigem monitoramento e gerenciamento do ciclo de vida contêineres, torna-se importante o uso de ferramentas mais apropriadas para essa tarefa. O *Kubernetes* consiste uma plataforma de software para a orquestração de contêineres que permite o planejamento e a automação das tarefas necessárias para gerenciar arquiteturas baseadas em contêineres, assim como diversas funcionalidades que permitem otimizar o processo de execução das aplicações que dependentes desses contêineres(IBM).

Aprofundando ainda mais no processo de deploy, *ArgoCD* é uma ferramenta de GitOps, abordagem que consiste na combinação do sistema de controle de versões git com pipelines de integração contínua que permite a automação do processo de deploy. Seu funcionamento consiste no monitoramento do repositório Git no qual os arquivos de kubernetes da aplicação são mantidos e verificam se a versão declarada nele difere da que está em execução e, se for o caso, a atualiza, colocando no ar a versão mais recente da aplicação.

Neste projeto, optou-se pela realização do deploy utilizando as ferramentas Docker, kubernetes e ArgoCD em um servidor da empresa *DigitalOcean*. Essas ferramentas foram selecionadas por conta de conhecimento e prévio acesso à uma infraestrutura bem configurada e estabelecida com essas ferramentas, fornecida pela empresa de desenvolvimento de sites e aplicativos *Switch Dreams*, fundada por estudantes do curso de Engenharia de Computação da Universidade de Brasília. Devido à sua alta complexidade, não é muito recomendável o uso dessas ferramentas na ausência de um conhecimento sólido a respeito de *devops*. Um deploy muito mais simples utilizando somente o Docker com o Docker Compose também funcionaria bem.

2.2 APLICAÇÕES SIMILARES

Em um paper publicado pelos estudantes Matheus Monteiro, Francisco de Caldas, Pedro Moura, Vinícius Costa, Gustavo da Luz, Luís Carvalho e Rafael de Sousa do departamento de engenharia elétrica da Universidade de Brasília, foi mostrado o desenvolvimento de um produto que tinha como propósito medir o volume ocupado nos contêineres de lixo da cidade (Monteiro et al. 2021). Na solução proposta, foi utilizado um microcontrolador ESP32 em conjunto com um sensor de ultrassom para medir a distância até o lixo dentro do contêiner e assim calcular o volume ocupado. Além disso, também era desejável que a solução fosse auto-suficiente energeticamente, portanto também foi integrado ao sistema uma bateria recarregável por uma placa solar. Todas as informações sobre volume de lixo e nível da bateria, em cada contêiner, eram então disponibilizadas em uma plataforma web para que pudessem ser monitoradas de forma recorrente.

Em um projeto desenvolvido por Luiz Augusto Fidalgo, foi descrito um sistema projetado com o intuito de medir em tempo real o estoque de talheres e pratos em um complexo gastronômico, com o objetivo de aumentar a eficiência na gestão e distribuição desses objetos (Costa 2022). Nessa solução, também foi utilizado um ESP32 e a medição em tempo real foi feita por meio de quatro células de carga configurando uma ponte de Wheatstone completa. Os dados coletados eram então disponibilizados para o serviço IoT Hub na plataforma Microsoft Azure, o qual foi integrado com um aplicativo que dispunha dessas informações para os usuários.

Apesar de não ter sido aplicada no contexto da coleta de lixo, a solução descrita acima se aproxima bastante da que será desenvolvida no decorrer desse projeto, tendo como principais diferenças o banco de dados utilizado, a aplicação final que proverá os dados ao usuário e a dispensabilidade da operação em baixo consumo de energia, visto que os pontos de armazenamento de resíduos serão, a princípio, fixos e com acesso à rede elétrica.

3 METODOLOGIA

Para validação do funcionamento do sistema, foi utilizado um método experimental aplicado à um contexto similar à coleta de resíduos sólidos, em que o dispositivo de medição de peso foi aplicado sob um cesto de roupa suja em uma residência familiar de 3 pessoas. Dessa forma, os dados obtidos foram analisados de forma quantitativa, construindo-se um gráfico para avaliar as variações nas medições de peso, calculando-se a taxa de crescimento média do sistema e estimando o tempo restante para o atingimento da máxima capacidade do sistema.

Durante o mapeamento e seleção dos componentes e ferramentas a serem utilizados, foram efetuadas algumas pesquisas comparativas entre elas, sendo levados em consideração principalmente fatores como custo, funcionalidades, conhecimento prévio e acessibilidade. Por fim, foi optado pelo uso de células de carga de 50 kg de capacidade dispostas em uma ponte de Wheatstone para realizar a medição da variação de peso, como mostrado na Figura 3.1; módulo HX711 como conversor AD e amplificador de sinal para as variações de tensão medidas da ponte de Wheatstone; uso do microcontrolador ESP32-C3 para a leitura dos dados e conexão com a rede Wi-Fi; protocolo MQTT para o envio e recepção de dados; banco de dados relacional PostgreSQL para armazenar os valores medidos; e por fim o framework Ruby on Rails em conjunto com a biblioteca React para a construção da página web responsável por exibir os dados em uma interface para o usuário.

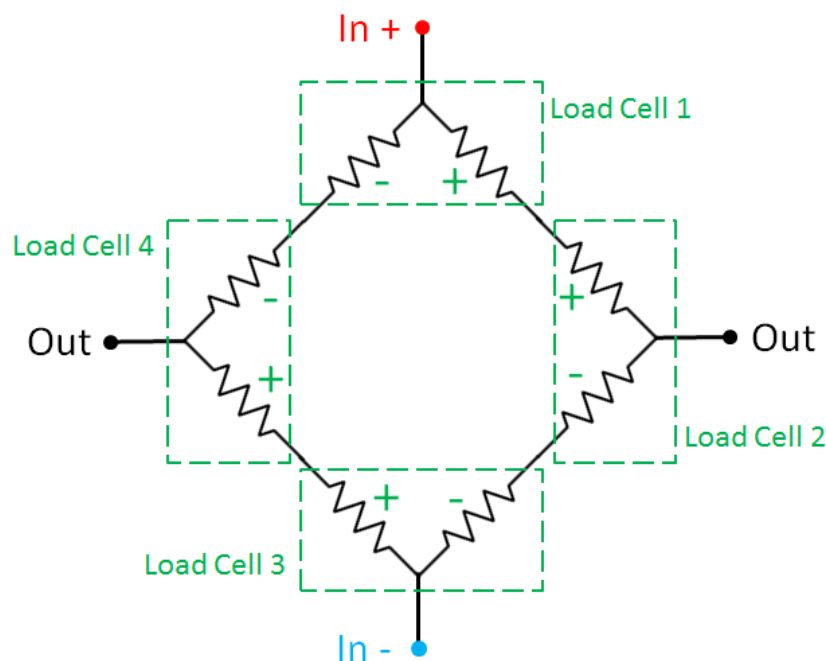


Figura 3.1: Ponte de Wheatstone formada por 4 células de carga

Já definidas as ferramentas, o desenvolvimento do sistema inicia-se com a montagem do dis-

positivo, sendo efetuada a conexão dos fios das células de carga entre si e com o conversor AD, que por sua vez foi ligado ao microcontrolador através de uma *proto-board* e alguns *jumpers* para auxiliar na conexão entre os pinos. Em seguida, as células de carga são coladas sobre uma placa com uma superfície lisa, sendo de suma importância deixar um espaço entre a superfície da placa e as células de carga, para permitir que estas possam flexionar livremente, não afetando o funcionamento de seus respectivos extensômetros. Para esse propósito, foram utilizados alguns pedaços de uma fita dupla face, aplicados nas bordas das células de carga. Para concluir a construção do dispositivo, uma nova placa de superfície lisa foi disposta sobre a face oposta das células de carga, que serviria como a plataforma de pesagem da balança.

O desenvolvimento do software de pesagem consiste na implementação de uma interface para a comunicação com o conversor AD, visando converter os bits recebidos (que representam a variação de tensão nos terminais da ponte Wheatstone) em valores inteiros de peso na ordem de grandeza desejada. Para isso, será necessário configurar a tara e utilizar massas conhecidas para efetuar a calibragem do dispositivo, permitindo converter corretamente os valores lidos. Esse mesmo software, deverá conter um módulo responsável pela conexão com a rede Wi-Fi e pelo envio desses valores para um agente MQTT.

A etapa seguinte consiste no desenvolvimento de todo o sistema web, composto por um processo responsável pela recepção dos valores das medições via MQTT e seu armazenamento em um banco de dados; um módulo para processar esses dados e efetuar os cálculos desejados sobre eles; e uma interface gráfica para dispor de todas as informações para o usuário final da plataforma. Nessa etapa também foram utilizados mecanismos de filtragem por software, nos quais foram calculadas as médias das medições em determinados períodos de tempo, reduzindo os ruídos apresentados pelo sistema.

4 RESULTADOS

4.1 MONTAGEM

Na primeira etapa do projeto, após definidos os componentes a serem utilizados, deve-se preparar cada um deles e testá-los, de preferência de forma isolada, afim de verificar se estão funcionando devidamente.

Após a validação do funcionamento do microcontrolador, a etapa seguinte consistiu na construção da estrutura de pesagem. Para isso, foram utilizadas duas placas com superfícies lisas de um material similar à cerâmica, quatro células de carga com capacidade de até 50 kg cada uma e uma fita dupla face. As células de carga são dispostas sobre uma das placas, sendo posicionadas como os vértices de um retângulo, visando a estabilidade do sistema, visto que seriam posteriormente conectadas na configuração de uma ponte de wheatstone, como mostrado na Figura 4.1. Para a fixação de cada célula, foi aplicada a fita dupla face entre a superfície oposta ao botão e a superfície da placa.

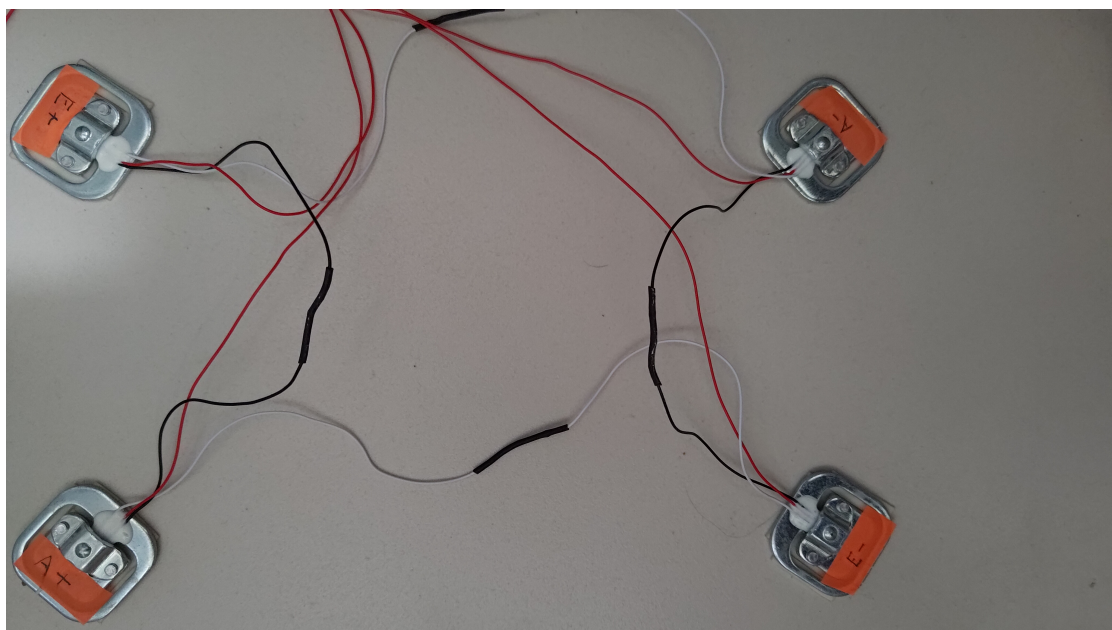


Figura 4.1: Células de carga na configuração de uma ponte de Wheatstone

O procedimento para a conexão entre os fios das células de carga consistia em cortá-los em um tamanho apropriado e desencapá-los, possibilitando o enrolamento de um fio no outro e a soldagem entre eles, visando estabelecer sua conexão. Também foi utilizada uma fita termo-retrátil para firmar a união dos fios e evitar problemas de mal contato. Finalizado esse procedimento, a segunda placa é então disposta sobre os botões das células de carga para atuar como a base da balança, podendo ser vista na Figura 4.2. Por fim, foi realizada a soldagem dos pinos do módulo amplificador HX711 e dos fios das células de carga em seus respectivos pinos de entrada do

módulo, como pode ser observado na Figura 4.3.



Figura 4.2: Balança vista de cima

4.2 SOFTWARE

4.2.1 ESP32

Para o desenvolvimento do software de leitura do HX711, foi utilizada uma biblioteca *open source* com suporte para a interface do ESP32. Seu uso é bem simples, inicialmente é necessário realizar um procedimento para calibrar a balança, que exige duas etapas. A primeira consiste na configuração da tara, em que é utilizado o método *tare* da biblioteca, o qual configura o deslocamento, que é o quanto que será subtraído do valor obtido nas leituras subsequentes. Esse deslocamento pode ser obtido e definido pelas funções *get_offset* e *set_offset*, respectivamente. A etapa seguinte consiste no uso de uma massa conhecida para verificar o resultado obtido a partir da saída de dados do módulo HX711. Com esse resultado, é possível calcular o *fator de calibração* da balança, que é dado pela relação:

$$fator = \frac{leitura}{massa} \quad (4.1)$$

O resultado obtido é então utilizado para configurar a escala da balança, de forma que as leituras subsequentes serão divididas por esse fator, resultando em valores que fazem sentido e que estarão na mesma ordem de grandeza que o valor de massa utilizado. Por exemplo, ao utilizar no cálculo uma massa conhecida de n gramas, o fator de calibração será tal que as leituras estarão na ordem de gramas.

Um fenômeno que pode ocorrer ao utilizar essas células de carga é o decaimento do resultado

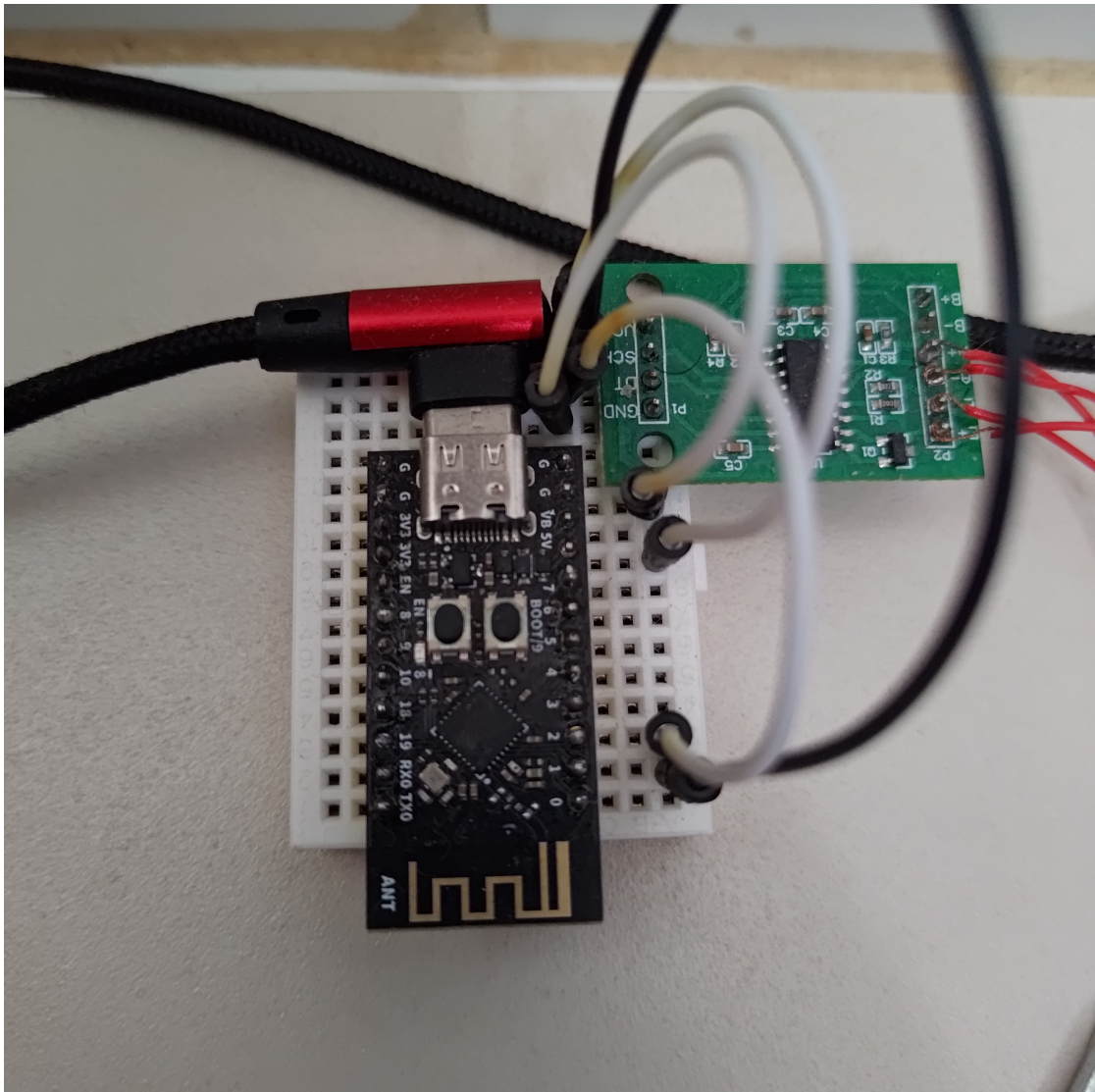


Figura 4.3: Conexão entre HX711 e o ESP32

obtido ao longo do tempo, cuja taxa é informada pelo próprio fabricante. Por exemplo, as células de carga de 50 kg utilizadas nesse projeto apresentam uma deriva de 0.1% da sua máxima amplitude por minuto, o que resulta em uma variação de até 50 gramas a cada minuto. Caso seja observada essa condição, é importante aguardar pela estabilização do sistema antes de realizar a leitura do valor que será utilizado para calcular o fator de calibração.

Para realizar a configuração da conexão WiFi do microcontrolador, foi implementado um módulo que tenta conectar à rede a partir das credenciais, as quais são configuradas como variáveis de ambiente através da ferramenta *menuconfig* disponibilizado pela esp-idf, até que uma conexão seja estabelecida. Essa conexão é necessária para o funcionamento do módulo MQTT, em que é realizada a configuração da url e a conexão com o agente responsável por gerenciar a comunicação mqtt, que no caso foi `mqtt://test.mosquitto.org`, além da implementação de um método para publicar mensagens em um tópico do broker.

O programa é responsável por utilizar todos os módulos citados anteriormente em conjunto.

Inicialmente devem ser configurados os valores do fator de calibração (calculado por meio um método auxiliar) e a tara (obtida empiricamente ao executar o programa sem peso sobre a balança). Além desses valores, é necessário configurar os pinos do ESP32 que serão conectados ao amplificador HX711, sendo neste caso optados pelos pinos 1 e 2 para o SCK e DOUT, respectivamente. Com essas constantes configuradas, o loop principal consiste em armazenar em uma variável, a cada 5 segundos, os dados lidos do HX711 e, a cada 5 minutos, enviar uma mensagem com o valor da última medição via MQTT para o tópico *DGM/scale* do broker. O código completo desse software pode ser encontrado em <<https://github.com/DavidsonGM/scale-esp32>>.

4.2.2 Dashboard

Para poder acessar os dados coletados pelo sistema de pesagem, foi desenvolvida uma aplicação web utilizando-se do framework *Ruby on Rails*. Neste framework, existe uma biblioteca chamada *ruby-mqtt* que possibilita a configuração da aplicação para atuar como um cliente MQTT. Por padrão, o rails não executa, por tempo indeterminado, processos em background além do servidor web em si. No entanto, é possível configurar uma *thread* para executar em segundo plano, a qual deve se conectar ao mesmo *broker* cujas mensagens estão sendo enviadas pelo microcontrolador e se inscrever no tópico de interesse. A *thread* então processa cada mensagem recebida e armazena seu conteúdo no banco de dados associado à aplicação. O sistema gerenciador de banco de dados utilizado foi o *PostgreSQL* e contém uma única tabela, cujas colunas de interesse são o valor medido e a data de criação. A fim de otimizar a performance do sistema, também pode ser optado pela criação de uma tabela de cache com um campo de texto já no formato dos dados a serem lidos na dashboard.

Para o *front-end*, ou seja, a parte de visualização dos dados da aplicação, foi utilizada o *Inertia*, que viabiliza o desenvolvimento da dashboard em React dentro do próprio rails. Para a exibição dos dados de forma agradável e intuitiva ao usuário, existem diversas bibliotecas que podem ser utilizadas para a construção de gráficos, como o *react-apexcharts*, que é simples e intuitivo de ser utilizado, além de apresentar um interface gráfica amigável para o usuário (Charts). Todo o código da aplicação rails pode ser encontrado no seguinte repositório no github: <<https://github.com/DavidsonGM/scale-mqtt-client>>.

O processo de deploy do sistema consistiu na escrita de uma *Dockerfile* para a criação da imagem do contêiner, responsável por executar a aplicação rails, e enviá-la ao docker hub. Em seguida, foi realizada a escrita dos manifestos kubernetes (arquivos no formato YAML responsáveis pelo gerenciamento de seus recursos), os quais foram armazenados em um repositório git no github. Por já existir uma infraestrutura bem configurada integrando esse repositório ao ArgoCD, a etapa seguinte consistiu simplesmente na criação da aplicação dentro da dashboard da plataforma, em que bastava especificar o caminho, dentro do repositório, para os manifestos da aplicação e sincronizá-los para que o deploy fosse realizado. Também foi necessária a configuração de um domínio para essa etapa, uma vez que a infraestrutura do servidor na qual seria hospedada estava configurada de forma que não possibilitaria o acesso diretamente via IP.

Com isso, já é possível a integração do sistema completo, em que os dados medidos pela balança são enviados via MQTT para um tópico específico do *broker* e são lidos pela aplicação web, hospedada na *Digital Ocean*, a qual exibe esses dados em um gráfico representando o histórico dos valores medidos.

4.3 VALIDAÇÃO

Após a integração dos diferentes componentes, é importante realizar a validação do funcionamento do sistema completo. O teste utilizado para realizar essa verificação foi utilizar uma massa conhecida de 5 kg sobre a balança por um longo período de tempo e observar os resultados obtidos. Dessa forma, é esperado que, ao acessar à dashboard, seja possível visualizar um gráfico com uma linha com pouca variação ao longo do tempo, estabilizada com os valores peso iguais ao da massa utilizada.

A partir de uma análise desse teste inicial, podem ser necessários alguns ajustes nas constantes da balança (fator de calibração e tara), e até a implementação de algum mecanismo para tratar variações ou ruídos inesperados das medições, como o cálculo da média móvel dos valores que estão sendo medidos. Findados esse ajustes, a validação deve ser efetuada novamente até que o resultado obtido seja mais próximo do esperado, aumentando a confiabilidade do sistema.

Com a balança devidamente calibrada, a nova etapa de testes consiste em introduzir o sistema em um ambiente o mais próximo possível do contexto real no qual foi projetada para atuar (sob algum recipiente de armazenamento de resíduos sólidos) e observar o comportamento das medições ao longo do tempo. Para validar o funcionamento do sistema, é esperado que seja possível observar na dashboard um pequeno crescimento dos valores medidos ao longo do tempo, até que haja um decaimento expressivo por conta da retirada do lixo, antes de voltar a crescer. A figura 4.4 ilustra como poderia ser o comportamento das medições do sistema em um cenário ideal, isto é, com medições precisas e sem ruídos.

4.4 CONFIGURAÇÃO DO AMBIENTE

O processo de calibragem consistiu, inicialmente, chamar o método *tare* da biblioteca do HX711 sem nenhum peso sobre a balança para obter o *offset*, que é o valor que representa a tara, independentemente do fator de calibração. Em seguida, foi efetuado o cálculo da média dos resultados de múltiplas medições consecutivas na balança a partir de uma massa conhecida de 5kg. A partir dessa média, foi calculado o fator de calibração inicial. No entanto, notou-se que havia um pequeno decaimento nas medições ao longo do tempo, então a próxima etapa consistiu em manter essa massa sobre a balança e ajustar empiricamente o fator de calibração até que as medições se estabilizassem em torno de 5000 gramas.

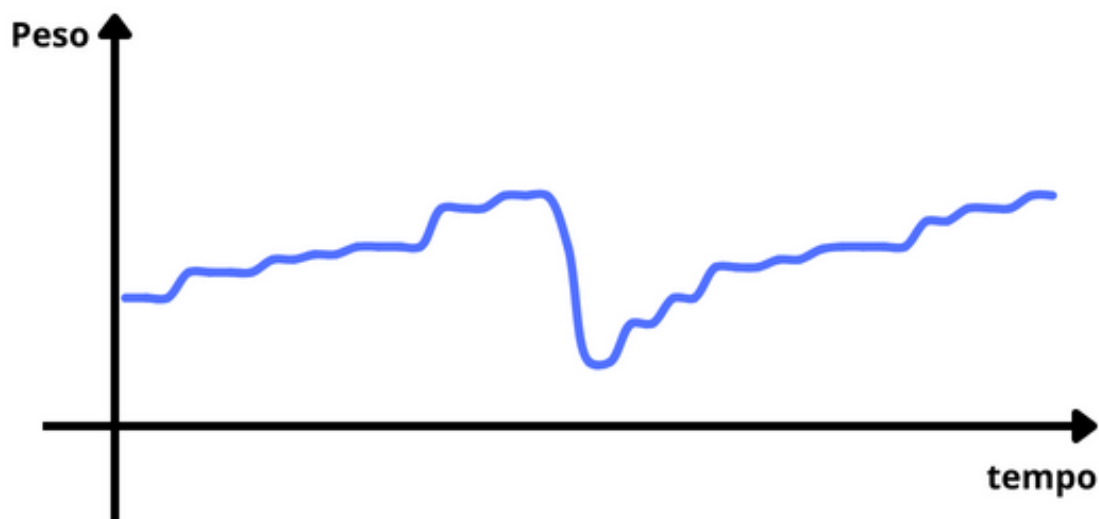


Figura 4.4: Relações de medições tempo x peso em um cenário ideal hipotético

Com o hardware propriamente calibrado, a validação comportamento do sistema como um todo foi efetuada a partir da utilização um cesto de roupa suja para a observação do comportamento das medições ao longo do tempo, uma vez que pode apresentar medidas de peso na mesma ordem que uma lixeira grande e apresenta um comportamento de variação similar de volume, aumentando lentamente até certo ponto, em que é esvaziado. A principal diferença entre esses dois ambientes é que o lixo sempre é recolhido por completo, enquanto que o cesto de roupa é, geralmente, descarregado parcialmente, visto que é comum a separação entre os diferentes tipos de roupas para a lavagem e que a capacidade das máquinas de lavar pode ser inferior a quantidade de roupas armazenadas no cesto. Dessa forma, existe uma baixa probabilidade de as medições apresentadas nesse teste atingirem um valor nulo, o que seria esperado após cada recolhimento no contexto do armazenamento de resíduos.

4.5 MEDIÇÕES

Os dados apurados pela balança são armazenados em um banco de dados e exibidos na dashboard, que apresenta um gráfico com a variação das medições ao longo do tempo e algumas estatísticas adicionais que visam a facilitar a observação do comportamento do sistema, as quais serão descritas mais detalhadamente nos parágrafos seguintes.

4.5.1 Cálculos

O valor máximo é dado em quilogramas e representa o pico das últimas medições. Optou-se por exibi-lo para representar o valor de máxima capacidade do recipiente, ou seja, aquele na qual é esperado que seja realizado uma coleta. Optou-se por não utilizar o máximo de todas as medições para que fosse possível automatizar uma possível troca entre recipientes de diferentes

capacidades e também para evitar que valores de "falsos picos" originados de falhas no sistema possam interferir diretamente no correto funcionamento da plataforma. O intervalo utilizado para aferir esse valor foi de uma semana.

A variação representa o percentual de quanto que a última medição cresceu (ou diminuiu) em relação à anterior. Seu cálculo é dado pela Equação 4.2, em que n é o número de medições.

$$var = \frac{m_n}{m_{n-1}} - 1 \quad (4.2)$$

A variação média é a taxa de crescimento média do sistema ao longo do tempo. Seu cálculo consiste em iterar sobre a lista de medições e tirar a média das variações entre cada um dos pontos. Como o principal objetivo dessa informação é possibilitar uma previsão quanto à quando o sistema deverá atingir sua máxima capacidade, foram desconsideradas variações negativas acima de 1 kg, visto que essas deveriam representar descarregamentos do recipiente. Decaimentos abaixo desse valor não foram descartados uma vez que tendem a representar ruídos no sistema que, caso não fossem consideradas no cálculo, resultariam em uma taxa de crescimento acima da esperada. Findado o seu cálculo, a média das variações é então multiplicada por $\frac{60}{intervalo_tempo}$, para que sua unidade seja sempre g/h. Assim como o valor máximo, o cálculo da variação média também leva em conta apenas as medições nos últimos dias, pois a geração de resíduos pode ser bem diferente a depender do período de tempo. No contexto das universidades, por exemplo, é esperado que o volume de resíduos sólidos gerados durante o período letivo seja consideravelmente maior do que o gerado durante o recesso.

A última informação exibida na interface é a estimativa para a próxima coleta, cujo cálculo é mostrado na Equação 4.3, em que t é o tempo total, em horas, para que o sistema atinja seu último valor de pico (max); m_n é o valor da última medição e $crescimento$ é a taxa de crescimento média do sistema. Caso a taxa de crescimento média seja negativa, não será possível estimar o tempo até a próxima coleta.

$$t = \frac{max - m_n}{crescimento} \quad (4.3)$$

Também foi implementado um filtro por intervalo de tempo, que consiste em iterar sobre a lista de medições e efetuar o cálculo da média dos valores dentro de um mesmo intervalo de tempo, criando uma nova lista com as médias dos valores. As opções disponibilizadas no filtro são de 15 minutos, 30 minutos, 1 hora, 2 horas, 4 horas e 8 horas. Caso nenhuma das opções seja selecionada, todas as medições serão exibidas com o horário exato em que ocorreram, ou seja, com um intervalo de 5 minutos entre elas. Todos os dados estatísticos citados acima são diretamente afetados pelo filtro, visto que o cálculo é feito sobre os valores após a filtragem.

4.5.2 Dados obtidos

Nas Figuras de 4.5 a 4.7 é possível observar as medições e estatísticas para alguns dos diferentes intervalos de tempo disponíveis no filtro da dashboard, em que o eixo x representa as datas e horários das medições e o eixo y os valores das medições em kg.

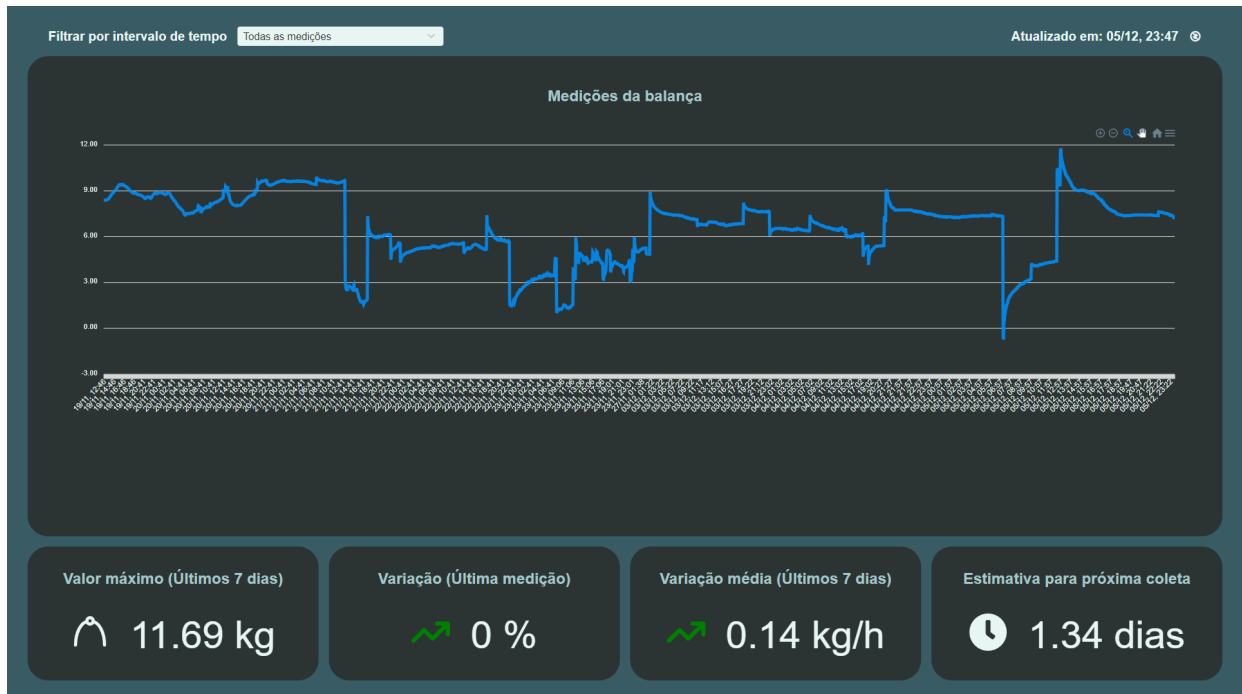


Figura 4.5: Dashboard sem filtragem

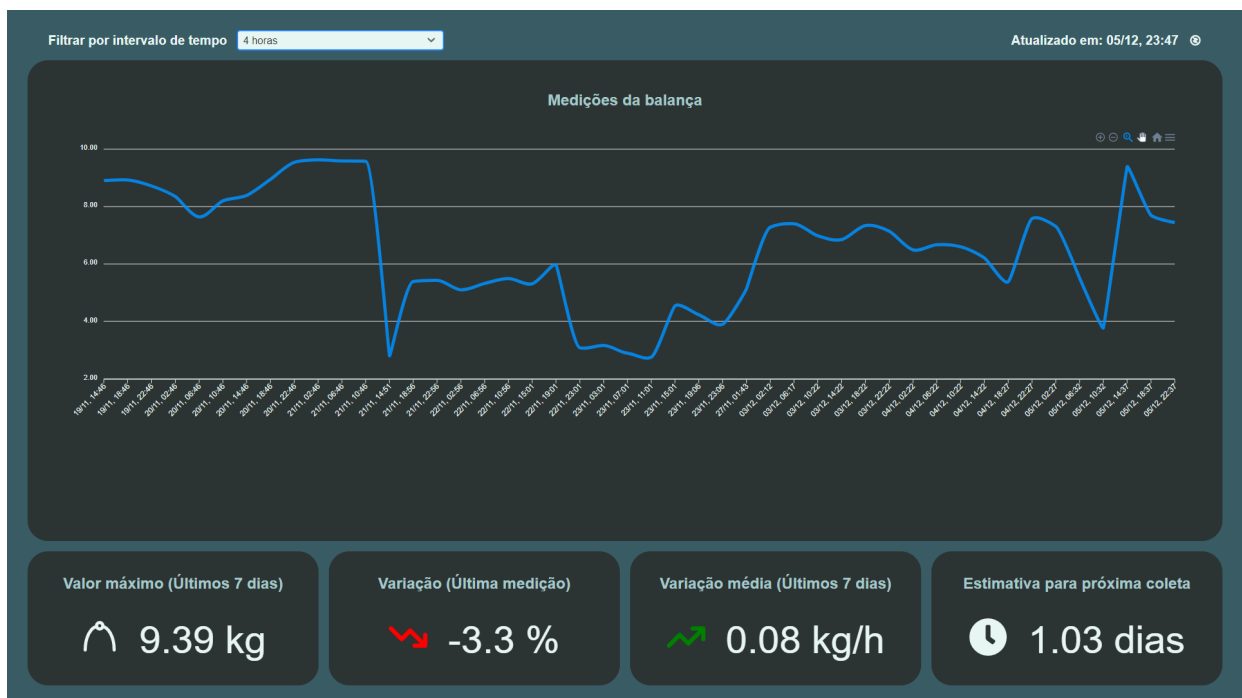


Figura 4.6: Dashboard com configuração do filtro de 4 horas entre os dados

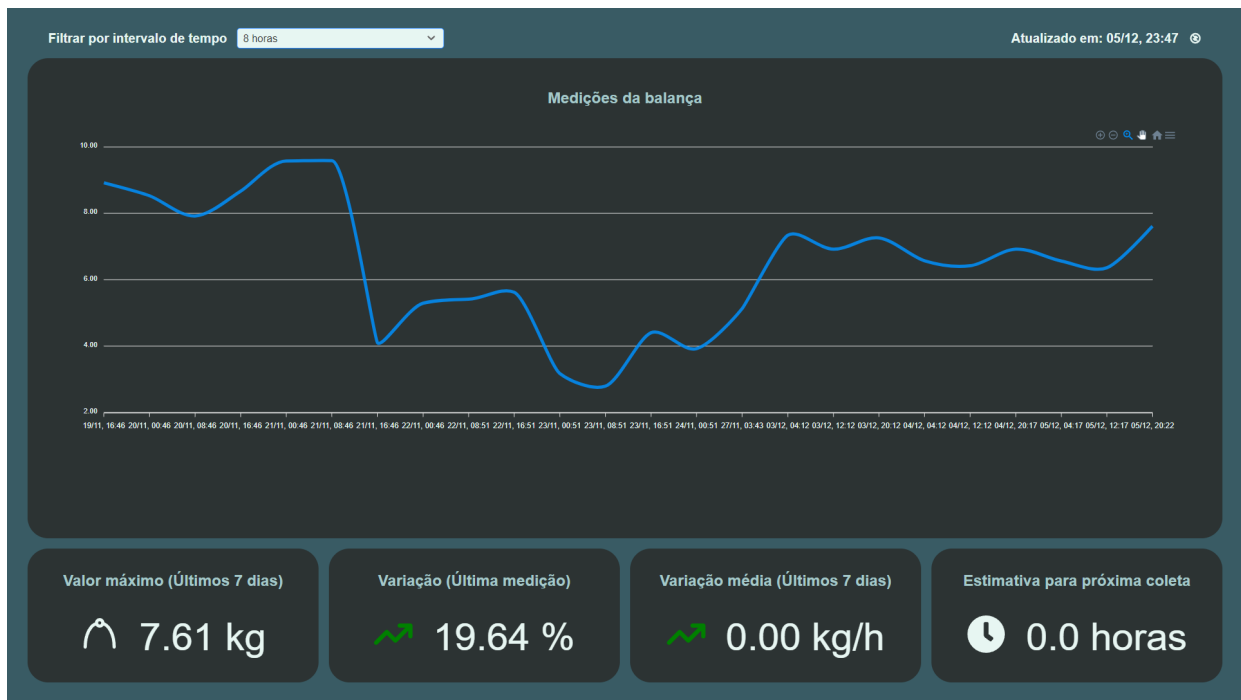


Figura 4.7: Dashboard com configuração do filtro de 8 horas entre os dados

Ao analisar os resultados obtidos, é possível notar que o gráfico filtrado por menores intervalos de tempo entre as medições apresenta uma maior quantidade de ruídos, resultando em picos e vales muito acentuados. Por conta disso, existe uma alta dificuldade de interpretar o gráfico aprofundadamente nesses casos, não sendo fácil de identificar corretamente o peso em um momento específico do tempo (sem dar zoom no gráfico). Em contrapartida, ao filtrar por maiores intervalos de tempo, em que é calculado a média dos valores medidos no período, é possível observar uma redução nos ruídos e, conseqüentemente, uma suavização nas curvas, facilitando a observação da variação do peso ao longo do tempo.

Também é importante considerar a taxa de precisão desejada para o sistema. Quando aplicado no contexto do armazenamento de lixo, é desejado obter grandes variações de massa sobre os resíduos armazenados, na ordem dos quilogramas. Sendo assim, pode não haver a necessidade de utilizar um conversor AD de tamanha precisão quanto o HX711. Considerando que este módulo é um conversor de 24 bits, com um ganho de 128, existe uma alta sensibilidade do sistema a quaisquer ruídos que ocorram. Portanto, a utilização de conversores de menores resoluções ou a realização do descarte dos bits menos significativos nas leituras do HX711 poderia contribuir para a minimização dos ruídos obtidos.

Outra questão importante de comentar é que em intervalos menores, devido ao volume de dados, é facilmente perceptível uma redução de performance na plataforma, com uma maior demora para carregar a página e maior tempo de resposta ao mexer no gráfico. Foi efetuada uma tentativa de otimizar a performance com a implementação um cache dentro do próprio rails que não mais acessaria esses dados no banco de dados a cada acesso (apesar de ainda realizar um processamento em cima do cache para aplicar os filtros), acessando-o apenas quando o cache

expirasse (a cada 12 horas) ou quando fosse explicitamente solicitado para limpar o cache e atualizar os dados. Apesar de mitigar bastante o problema, o volume de dados, mesmo que cacheado, continua sendo bem alto, sendo necessário carregar um gráfico com mais de 2 mil pontos por semana (no intervalo de 5 minutos). De modo geral, apesar de não ser uma lentidão tão elevada a ponto de impedir o uso da plataforma, com intervalos de menos de 1 hora a performance afeta diretamente a experiência do usuário, enquanto que para intervalos acima de 1 hora, o problema de performance é praticamente imperceptível. Sendo assim, com o tempo, pode ser necessário a implementação de um limite para a quantidade de dados a serem exibidos na interface ou uma diminuição na taxa de amostragem para evitar que o volume de dados impeça a plataforma de ser utilizada.

Ainda analisando os gráficos, é possível observar que existe uma tendência de ocorrer um decaimento nas medições das células de carga ao longo do tempo. Esse fenômeno é comum em células de carga de extensômetro, uma vez que a deformação causada por variações de peso no suporte interno das células de carga pode demorar um pouco para se estabilizar. No entanto, não foi possível validar se essa é a única causa para a ocorrência desse decaimento. Alguns outros fatores podem contribuir para que isso ocorra, como flutuações na temperatura, qualidade dos módulos de hardware e até a montagem da estrutura da balança em si, cuja base que fica sobre as células de carga não foi propriamente fixada, podendo resultar em medições imprecisas quando não devidamente posicionada.

Com o filtro de 8 horas de intervalo, a Figura 4.7 é o que mais se assemelha ao gráfico do cenário ideal hipotético ilustrado na Figura 4.4. Uma das principais diferenças que é possível citar é o decaimento observado em alguns períodos, apesar de serem bem menos perceptíveis se comparados aos outros filtros. Outra diferença que fica bem evidente, mas que já era esperada, é que no cenário real, as variações não seguem um padrão bem definido como no hipotético. Em contrapartida, é possível notar que o comportamento de ambos os gráficos se assemelha ao apresentar intervalos que consistem de um crescimento lento seguida de uma brusca queda nos resultados das medições, que é o comportamento esperado levando em conta o contexto da aplicação.

Em relação às estatísticas, percebe-se que a tendência é que o valor máximo diminua na medida que o intervalo de tempo dos filtros aumenta, uma vez que é bem comum que ocorram picos devido à ruídos ou aumentos bruscos de carga sobre a balança, que demora um tempo até se estabilizar. Como os filtros funcionam a partir do cálculo do valor médio no período de tempo, quanto maior esse período, menos esses picos irão afetar o resultado final desse dado.

A variação média, por sua vez, é um dado interessante de se observar, pois como o seu cálculo não tende a ser muito afetado por ruídos (apesar de ser afetado pelo problema da deriva), é difícil de afirmar qual dos filtros apresenta um crescimento médio o mais próximo do real. No entanto, ao tirar a média para obter um ponto intermediário entre os valores dessa informação nos diferentes gráficos, obtêm-se um crescimento médio de cerca de 0.09 kg/h, o que significa que seriam necessários pouco mais de 4 dias e meio para a realização de uma nova coleta caso o sistema fosse esvaziado por completo (considerando uma capacidade máxima de 10kg).

A estimativa para a próxima coleta é um dado que também apresenta uma variação inconstante para os diferentes intervalos de tempo, principalmente por usar como parâmetros para seu cálculo o valor de pico e a própria variação média, da qual não se tem muita certeza quanto à sua confiabilidade em relação ao sistema real. Apesar disso, com exceção do gráfico com a configuração de filtro de 8 horas, que apresentou uma estimativa de 0 horas para a próxima coleta (já que se encontrava em seu ponto máximo), a variação da estimativa de tempo não é tão relevante a ponto de afetar a implementação de políticas de coleta. Com uma variação de 12 horas entre o menor e o maior valor, a coleta provavelmente seria programada para o mesmo dia.

5 CONCLUSÃO

5.1 ANÁLISE DOS RESULTADOS

Os resultados apresentados no capítulo anterior mostram que as funcionalidades implementadas pelo sistema poderiam ser eficientemente utilizadas para o monitoramento do volume de resíduos sólidos armazenados em algum local. Também são fornecidas estatísticas que facilitam o acompanhamento da variação dos resíduos armazenados, o que permite estimar e programar a data ideal para a sua coleta.

Com esses resultados, verifica-se o cumprimento dos principais objetivos traçados para o sistema: mapeamento das ferramentas necessárias para o desenvolvimento, construção da estrutura e calibragem da balança eletrônica, elaboração de software responsável pelo seu funcionamento (desde a parte de medição até o envio dos dados pela rede), desenvolvimento de um processo para captação e armazenamento dessas informações e criação de uma interface para exibição dos dados captados. Todos esses objetivos, quando integrados, cumprem o propósito geral do projeto, que consiste no desenvolvimento de um dispositivo de aferição de peso, cuja função seria medir o volume de resíduos sólidos dispostos sobre ele e disponibilizar essa informação de forma acessível, visando possibilitar a otimização do processo de coleta seletiva de lixo, reduzindo, conseqüentemente, seus custos de implementação.

Apesar de válida a sua funcionalidade, foi observada uma quantidade elevada de ruídos e altas taxas de decaimento nas medições ao longo do tempo. Estes fatores afetam diretamente o funcionamento e a confiabilidade do sistema, podendo causar incoerências e, conseqüentemente, previsões errôneas quanto ao momento ideal para o recolhimento dos resíduos. Também é importante a realização de testes dentro do contexto em que será aplicado, que é a pesagem de resíduos sólidos, fornecendo uma noção melhor do como o sistema se comportaria no contexto da coleta de lixo.

5.2 PRÓXIMOS PASSOS

Nessa seção, serão apresentadas algumas melhorias que poderiam vir a ser implementadas e integradas à solução desenvolvida.

5.2.1 Melhoria na captação dos dados

A obtenção de dados mais precisos e de alta confiabilidade resultam em um sistema que agrega muito mais valor ao seu público alvo, que são os agentes responsáveis pela coleta de resíduos. Por conta disso, a maior prioridade no aprimoramento do dispositivo desenvolvido deve ser verificar

as possíveis causas e reduzir ao máximo os ruídos observados, assim como o decaimento no valor das medições ao longo do tempo.

Uma das principais fontes de ruídos é montagem e conexão dos componentes eletrônicos e mecânicos que compõem o aparelho. Seria interessante a criação de uma Placa de Circuito Impresso com os componentes eletrônicos integrados (HX711 e ESP32) em conjunto com um conector PCI de fio para placa, para conexão apropriada com as células de carga. Ademais, para atingir uma maior estabilidade no sistema, seria de extrema importância a fixação da placa superior que atua como base da balança, ou até mesmo a construção de uma estrutura mecânica mais apropriada.

O decaimento, por sua vez, não apresentou uma causa em potencial bem definida, podendo ser causado pela qualidade das células de carga utilizadas, variações de temperatura, variações de umidade e até a montagem do sistema em si. Seria preciso investigar mais a fundo sobre essa adversidade para obter um aumento da confiabilidade e precisão do sistema como um todo.

5.2.2 Performance

Com a tendência de realizar medidas indefinidamente, o volume de dados gerados pelo o dispositivo de pesagem tendem a ser cada vez maior. Dessa forma, por mais bem otimizada que seja a aplicação, em algum momento ela deixaria de funcionar apropriadamente por não suportar tamanha quantidade de dados. A implementação de cache de dados e filtros na dashboard ajudam a mitigar esse problema. No entanto, caso seja desejado visualizar os dados de pequenos intervalos de tempo, seu processamento tenderá a ser extremamente demorado, piorando cada vez mais, tornando inviável o uso da plataforma a médio/longo prazo.

Para resolver essa adversidade, uma opção é limitar a quantidade de dados que serão exibidos e processados em cada requisição. Ao utilizar um limite de 600 dados, por exemplo, a performance seria agradável e se manteria estável entre os diferente filtros de tempo. Além do mais, mesmo com esse limite, ainda seria possível observar um amplo histórico de variações, apresentando períodos de 2 dias (5 minutos entre as medições) até 4 anos (8 horas entre as medições).

Uma opção alternativa seria e estabelecer um limite por período de tempo, o que já mitigaria o problema do aumento do volume de dados crescer indefinidamente, mas nesse caso a performance não seria muito boa para menores intervalos de tempo, a não ser que sejam definidos limites distintos para cada período.

Para reduzir custos de hospedagem devido a um aumento contínuo do banco de dados, também é importante descartar, periodicamente, as medições mais antigas e que não são mais de interesse.

5.2.3 Escalabilidade

Como o objetivo traçado para esse projeto tratava apenas de desenvolver e validar o funcionamento do um sistema de pesagem, não existiu a necessidade de lidar com múltiplos dispositivos.

Todavia, para que pudesse realmente ser aplicado no contexto da coleta seletiva, seria necessária a construção em escala dos aparelhos de pesagem.

Neste cenário, seria necessária a implementação de algum mecanismo para verificar o local no qual cada um dos dispositivos se encontra e disponibilizá-lo na interface do usuário. Essa informação seria imprescindível, uma vez que não haveria utilidade alguma os valores de medições de balanças cuja localização é desconhecida. Indo ainda mais além, poderia ser esse dado para a implementação de um algoritmo para traçar as melhores rotas de coleta em uma região com base na estimativa pra lotação de capacidade em cada uma das balanças.

Também seria necessário efetuar algumas alterações nos softwares de envio e de coleta de dados para suportar múltiplos aparelhos. Uma opção seria o uso de um UUID (Identificador Único Universal) em cada um deles que seria enviado em conjunto com os valores de medição e, posteriormente, armazenado em uma nova coluna na tabela de medições do banco de dados.

Para a conexão dos aparelhos à rede, é desejável o incremento do módulo de software de Wi-Fi para permitir a inserção das credenciais de Wi-Fi com o uso de um celular, sem a necessidade de carregar manualmente essas credenciais para no esp32.

Na interface do usuário também seriam necessárias mudanças. Em primeiro lugar, teria de ser criada uma nova página na aplicação na qual seriam listados todas os dispositivos integrados ao sistema. A partir dela, seria possível acessar as dashboards individuais de cada uma das balanças (que seria a interface desenvolvida nesse projeto). Adicionalmente, poderia ser implementado um gráfico similar ao já desenvolvido, mas contendo várias linhas, cada um representando um dispositivo em um local diferente. Neste caso, cabe um cuidado ainda maior com performance, pois o volume de dados cresceria exponencialmente com o aumento de dispositivos, sendo necessário limitar a quantidade de informações dispostas em cada página.

Outra funcionalidade de grande importância para a implementação em larga escala desses dispositivos seria a configuração e habilitação de atualizações OTA (*Over The Air*), a qual permite a atualização, de forma remota, do software que está carregado no microcontrolador esp32.

5.2.4 Segurança

Nesse projeto não foi desenvolvido nenhum mecanismo visando o aumento da segurança do sistema. A primeira melhoria nesse quesito a ser implementada seria na conexão com o agente MQTT, que atualmente é um broker público, o qual permite que qualquer usuário conectado à internet possa enviar e receber mensagens por esse canal. Também pode ser desejado limitar o acesso à aplicação web, sendo preciso implementar algum módulo de autenticação dentro dela, o que é algo bem comum para esse tipo de aplicação, não sendo tão difícil de encontrar bibliotecas com esse propósito.

Caso seja habilitada a função de atualizações OTA, também é de extrema importância tomar os devidos cuidados com a segurança, visto que abre mais um vetor para ataques de usuários

maliciosos, que podem se infiltrar e obter acesso aos dispositivos, possibilitando ataques DDoS, envio de mensagens de spam e roubar dados.

5.2.5 Monitoramento e automação

Em alguns momentos, foi identificado uma interrupção em algumas funcionalidades do sistema, resultando em um período sem a coleta de dados. Acabou sendo necessário reiniciá-lo manualmente, seja desligando e ligando de volta o esp da tomada, seja de alimentação reiniciando a thread de execução do processo responsável por efetuar a conexão mqtt.

Para impedir que isso seja necessário, é desejável otimizar o software de forma a verificar, periodicamente, se está tudo funcionando corretamente (enviando/recebendo mensagens a cada 5 minutos) e, caso não seja o caso, reinicia automaticamente o processo que apresenta falhas para buscar resolver o problema. Também é interessante configurar e conectar um LCD ao dispositivo, fornecendo um visor para exibição de logs e medições em tempo real para os usuários na presença do aparelho.

REFERÊNCIAS BIBLIOGRÁFICAS

AWS AWS. *O que é MQTT?* Disponível em: <<https://aws.amazon.com/pt/what-is/mqtt>>.

Brasil 2010 BRASIL. Lei nº 12.305, de 2 de agosto de 2010. *Diário Oficial da União*, Brasília, DF, 2010. ISSN 1677-7042. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2010/lei/112305.htm>.

Bravo et al. 2018 BRAVO, T. L.; PEÇANHA, A. L.; WERNER, E. T.; SANTOS, A. A. O. Educação ambiental e percepção da implantação de coleta seletiva de lixo urbano em de alegre, es. *Revista Gestão & Sustentabilidade Ambiental*, v. 7, n. 1, p. 375–396, 2018.

Carbonnelle 2023 CARBONNELLE, P. *PYPL Popularity of Programming Language*. 2023. Disponível em: <<https://pypl.github.io/PYPL.html>>.

Charts CHARTS, A. *React Charts*. Disponível em: <<https://apexcharts.com/docs/react-charts/>>.

Conke e Nascimento 2018 CONKE, L. S.; NASCIMENTO, E. P. d. A coleta seletiva nas pesquisas brasileiras: uma avaliação metodológica. *urbe. Revista Brasileira de Gestão Urbana*, SciELO Brasil, v. 10, p. 199–212, 2018.

contributors 2023 CONTRIBUTORS, W. *Load cell* — *Wikipedia, The Free Encyclopedia*. 2023. 2023-11-20. Disponível em: <https://en.wikipedia.org/wiki/Load_cell>.

Costa 2022 COSTA, L. A. F. D. da. *Controle de estoque através de variação de peso usando técnicas de baixo consumo e comunicação sem fio*. 87 p. Monografia — Universidade de Brasília, Brasília, DF, Brasil, 2022.

Dinculeană e Cheng 2019 DINCULEANĂ, D.; CHENG, X. Vulnerabilities and limitations of mqtt protocol used between iot devices. *Applied Sciences*, MDPI, v. 9, n. 5, p. 848, 2019.

Galante 2019 GALANTE, G. J. *Flask vs Django*. 2019. Disponível em: <<https://community.revelo.com.br/flask-vs-django/>>.

Globo 2023 GLOBO. *Dia Mundial da Reciclagem: 96% dos resíduos produzidos no Brasil não são reaproveitados*. 2023. Disponível em: <<https://g1.globo.com/jornal-nacional/noticia/2023/05/17/dia-mundial-da-reciclagem-96percent-dos-residuos-produzidos-no-brasil-nao-sao-reaproveitados.ghtml>>.

Globo 2023 GLOBO. *Mesmo proibidos, lixões a céu aberto ainda ameaçam a saúde em um terço dos municípios brasileiros*. 2023. Disponível em: <<https://g1.globo.com/jornal-nacional/noticia/2023/05/29/mesmo-proibidos-lixoes-a-ceu-aberto-ainda-ameacam-a-saude-em-um-terco-dos-municipios-brasileiros.ghtml>>.

Guimarães, Farias e Bandeira GUIMARÃES, S. P.; FARIAS, L. G. de; BANDEIRA, V. Interface jtag.

Hat 2022 HAT, R. *O que é um container Linux??* 2022. Disponível em: <<https://www.redhat.com/pt-br/topics/containers/whats-a-linux-container>>.

IBM IBM. *O que é o Docker?* Disponível em: <<https://www.ibm.com/br-pt/topics/docker>>.

Kashif 2022 KASHIF. *What Are Advantages and Disadvantages of Arduino*. 2022. Disponível em: <<https://linuxhint.com/advantages-and-disadvantages-arduino/>>.

Monteiro et al. 2021 MONTEIRO, M. S.; FILHO, F. L. de C.; SOUZA, P. d. O. Moura e; COSTA, V. S.; LUZ, G. P. C. P. da; CARVALHO, L. F. d. O.; SOUSA, R. T. de. Solid waste management and monitoring system for smart cities: development of a low-cost sustainable iot architecture using gprs/gsm. In: *2021 Workshop on Communication Networks and Power Systems (WCNPS)*. [S.l.: s.n.], 2021. p. 1–6.

Olujinmi 2022 OLUJINMI, T. *Raspberry Pi Pico vs. ESP32: Which Microcontroller Is Right for You?* 2022. Disponível em: <<https://www.makeuseof.com/raspberry-pi-pico-vs-esp32-microcontroller/>>.

Rodrigues e Santana 2012 RODRIGUES, W.; SANTANA, W. C. Análise econômica de sistemas de gestão de resíduos sólidos urbanos: o caso da coleta de lixo seletiva em palmas, to. *urbe. Revista Brasileira de Gestão Urbana*, SciELO Brasil, v. 4, p. 299–312, 2012.

Sacramento 2022 SACRAMENTO, G. *Aplicação web: o que é, diferença para website, como funciona e mais!* 2022. Disponível em: <<https://rockcontent.com/br/talent-blog/aplicacao-web>>.

Semiconductor SEMICONDUCTOR, A. *24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales*. Disponível em: <https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf>.

Souza 2022 SOUZA, F. *O que são sistemas embarcados?* 2022. Disponível em: <<https://embarcados.com.br/o-que-sao-sistemas-embarcados/>>.

Vailshery 2023 VAILSHERY, L. S. *Most used web frameworks among developers worldwide, as of 2023*. 2023. Disponível em: <<https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>>.