

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA ELETRÔNICA

ANTONIO THOMAZ BITTENCOURT KOZAK
HENRIQUE LOPES SENGER

ALMOXARIFADO INTELIGENTE

OFICINA DE INTEGRAÇÃO – RELATÓRIO FINAL

CURITIBA

2022

ANTONIO THOMAZ BITTENCOURT KOZAK
HENRIQUE LOPES SENGER

ALMOXARIFADO INTELIGENTE

Relatório Final da disciplina Oficina De Integração, do curso de Engenharia Eletrônica, apresentado aos professores que ministram a mesma na Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina.

Orientador: Prof. Sergio Moribe

CURITIBA

2022

RESUMO

. ALMOXARIFADO INTELIGENTE. 31 f. Oficina de Integração – Relatório Final – Curso de Engenharia Eletrônica, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2022.

O projeto visa construir um sistema capaz de gerenciar os estoques de um almoxarifado de maneira automática. Esse sistema permite que múltiplos usuários possam adicionar e remover itens de compartimentos predefinidos de um almoxarifado sem se preocupar em registrar essas mudanças manualmente. Neste trabalho demonstramos uma prova de conceito do sistema de uma possível implementação deste sistema.

Palavras-chave: almoxarifado, automação, projeto

ABSTRACT

. AUTOMATED WAREHOUSING SYSTEM. 31 f. Oficina de Integração – Relatório Final – Curso de Engenharia Eletrônica, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2022.

This project objective is developing an automated warehousing system capable of keeping track automatically of the amount of items in assigned compartments. This system allows that multiple users can add or remove items, and it automatically keeps an updated history of every transaction without the need for human interaction. In this work we built a proof of concept to show a possible implementation of this system.

Keywords: warehouse, automation, project.

LISTA DE FIGURAS

FIGURA 1	– Extensômetro	11
FIGURA 2	– Ponte de Wheatstone de extensômetros (TRENT, 2019)	11
FIGURA 3	– Esquemático do controlador	14
FIGURA 4	– Esquemático do modulo	16
FIGURA 5	– Diagrama de componentes do software	17
FIGURA 6	– Estágio inicial da construção	21
FIGURA 7	– Modelo da estrutura, vista frontal	22
FIGURA 8	– Modelo da estrutura, vista traseira	22
FIGURA 9	– Vista frontal da estrutura	23
FIGURA 10	– Vista traseira da estrutura	23
FIGURA 11	– Editando compartimento	25
FIGURA 12	– Visualização do estoque através do site	25
FIGURA 13	– Cronograma do projeto	27

SUMÁRIO

1	INTRODUÇÃO	8
1.1	MOTIVAÇÃO	8
1.2	OBJETIVOS	8
1.2.1	Objetivo geral	8
1.2.2	Objetivos específicos	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	CÉLULAS DE CARGA	10
3	METODOLOGIA	12
3.1	PROJETO DE HARDWARE	12
3.1.1	Implementação do controlador	12
3.1.2	Implementação do módulo	15
3.2	PROJETO DE SOFTWARE	17
3.2.1	Framework e bibliotecas	17
3.2.2	Protocolo de Comunicação Controlador-Modulo	18
3.2.3	API JSON	19
3.2.4	Interface WEB	19
3.2.5	Implementação do Banco de Dados	19
3.2.6	Implementação do Firmware do modulo	20
3.3	PROJETO MECÂNICO	21
4	EXPERIMENTOS E RESULTADOS	24
5	CRONOGRAMA E CUSTOS DO PROJETO	27
5.1	CRONOGRAMA	27
5.2	CUSTOS	28
6	CONCLUSÕES	29
6.1	CONCLUSÕES	29
6.2	TRABALHOS FUTUROS	29
	REFERÊNCIAS	31

1 INTRODUÇÃO

Neste trabalho é apresentada a realização da prova de conceito de um sistema de almoxarifado automatizado. Neste capítulo explicamos a motivação por trás da elaboração deste projeto, e os objetivos gerais da elaboração deste trabalho.

1.1 MOTIVAÇÃO

O gerenciamento de estoques é um problema comum a diversos setores da sociedade. Seja na indústria onde é necessário ter um controle de ferramentas e insumos utilizados. No comércio onde existe a necessidade de manter os estoques dos produtos a serem vendidos, até escritórios onde existe a necessidade de manter estoques de materiais de escritório utilizados no dia a dia, como papel, canetas, grampos e etc. Normalmente esse gerenciamento é feito por funcionários, que além de consumir tempo também sofre com o problema de erros humanos, que acabam tornando esses sistemas pouco confiáveis.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo deste projeto é através do uso de sistemas eletrônicos, propor uma solução de um almoxarifado completamente automatizado. Capaz de manter a quantidade de produtos sempre atualizados e mantendo um histórico completo do que, quanto e por quem foi retirado. O sistema conta com duas interfaces, sendo a principal através de um módulo local que permite que os usuários entrem suas credenciais e retirem ou adicionem os itens desejados e uma interface WEB, que a partir de um portal WEB ou uma API JSON permite os administradores do sistema alterarem as configurações dos itens armazenados, adicionar usuários e integrar

com outros sistemas da instituição aonde o sistema está sendo implementado.

1.2.2 OBJETIVOS ESPECÍFICOS

Para esta implementação de prova de conceito, os requisitos funcionais do sistema que consideramos essenciais e implementamos durante esse projeto são:

- O sistema permite que sejam cadastrados os itens contidos em cada um dos compartimentos, com seus respectivos pesos.
- O sistema deve manter um histórico de tudo que foi adicionado ou removido por cada usuário.
- O sistema é capaz de realizar cadastros de usuários, cada um com suas credenciais únicas de identificação.
- O sistema deve ser capaz de se conectar a uma rede, onde pode ser configurado e ter seus dados acessíveis via um portal web e uma API JSON.
- O sistema é flexível, de modo que possa ser futuramente expandido com a adição de múltiplos módulos conectados a um controlador, e que os módulos possam ser modificados adicionando sensores e atuadores específicos à aplicação.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são explicados os seguintes conceitos que embasam o funcionamento teórico do projeto

2.1 CÉLULAS DE CARGA

Células de carga são os sensores tipicamente usados na medição de massa. Entre seus principais pontos positivos está o fato de se tratar de um sensor bastante versátil, podendo medir precisamente desde miligramas até toneladas dependendo do modelo da célula.

Uma célula de carga é normalmente composta de quatro extensômetros (*strain gauges*), que são elementos resistivos elásticos, como mostrado na Figura 1. Ao sofrer uma deformação, o comprimento total da resistência aumenta e sua área transversal diminui, e como a resistência do elemento é dada pela equação (1), onde R é a resistência total (Ω), ρ a resistividade do material, L o comprimento do fio (m) e A a área transversal do fio (m^2) temos então que a resistência total é proporcional a essas mudanças.

$$R = \frac{\rho L}{A} \quad (1)$$

Os extensômetros são fixados em uma barra metálica e conectados, formando uma ponte de Wheatstone como ilustrado na Figura 2. A característica que torna a ponte de wheatstone interessante para essa aplicação é que quando o circuito esta perfeitamente balanceado, isso é, a resistência de todos extensômetros é a mesma, a tensão medida V_O é igual a 0V. E analisando o circuito, temos que V_O de maneira genérica é dada pela equação (2). Temos então que V_O é altamente sensível a variação da resistência dos extensômetros. E como o valor de V_O é centrado em 0, esse sinal pode ser alimentado a um amplificador de alto

ganho, e esse valor amplificado pode ser então digitalizado e enviado a um microcontrolador. E utilizando um objeto de massa conhecido, podemos associar essa variação de resistência devido a deformação dos extensômetros a uma massa conhecida, assim pode ser utilizar essa associação para medir massas genéricas.

$$V_0 = \left[\frac{R_3}{R_3 + R_2} - \frac{R_2}{R_1 + R_2} \right] \cdot V_{ex} \quad (2)$$

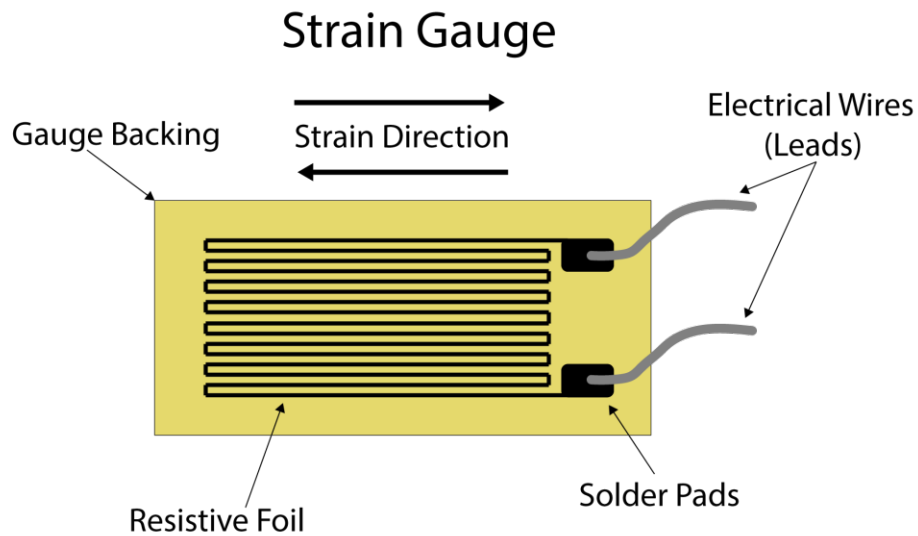


Figura 1: Extensômetro

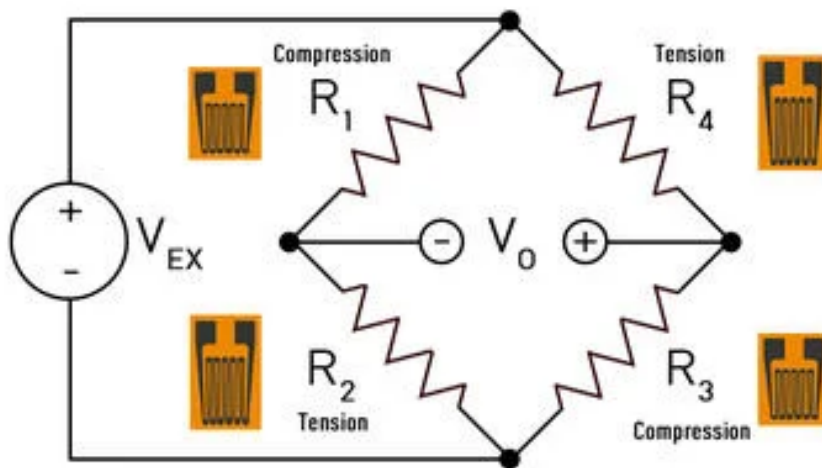


Figura 2: Ponte de Wheatstone de extensômetros (TRENT, 2019)

3 METODOLOGIA

3.1 PROJETO DE HARDWARE

A implementação do hardware do projeto consiste de dois dispositivos. Sendo eles o controlador, que é responsável por todo controle de lógica da aplicação, e o módulo, que é um dispositivo responsável pelo controle dos sensores e atuadores que realizam a contagem dos itens armazenados. A escolha dessa topologia com dois dispositivos distintos foi feita para aumentar a flexibilidade do sistema, de modo que em um mesmo barramento, múltiplos módulos possam estar conectados em um único controlador, permitindo que de acordo com a necessidade do cliente múltiplos módulos com diferentes configurações de sensores e atuadores possam ser integrados.

3.1.1 IMPLEMENTAÇÃO DO CONTROLADOR

O controlador foi implementado utilizando uma placa de desenvolvimento com o microcontrolador esp32-wroom- 32, ele atua como o cérebro do sistema, sendo responsável por toda parte lógica do sistema. A Figura 3 mostra o esquemático do controlador, e as funções de cada componente são:

- Regulador de tensão LM7805: Regulador linear que regula a tensão de entrada de uma fonte externa de 12V a 5V, que é o valor utilizado pela placa do microcontrolador e a maioria dos periféricos.
- ESP32: Microcontrolador que além de controlar todos outros periféricos do controlador, também é responsável por se comunicar com os módulos.
- Teclado 4x4: Teclado matricial utilizado pelos usuários locais para interagir com o sistema.

- RTC DS1307: Real Time Clock com bateria externa, necessário para permitir que o sistema tenha sempre uma base de tempo correta, mesmo se desligado.
- SDCard: Leitor de cartão SD com um cartão microSD de 32 GB. É utilizado para armazenagem não volátil do sistema em um banco de dados.
- Display 16x2: Display LCD de 16x2 linhas utilizadas para fornecer informações sobre o sistema ao usuário local.
- Buzzer: Auto falante buzzer que fornece feedback auditivo ao usuário ao interagir com o sistema.

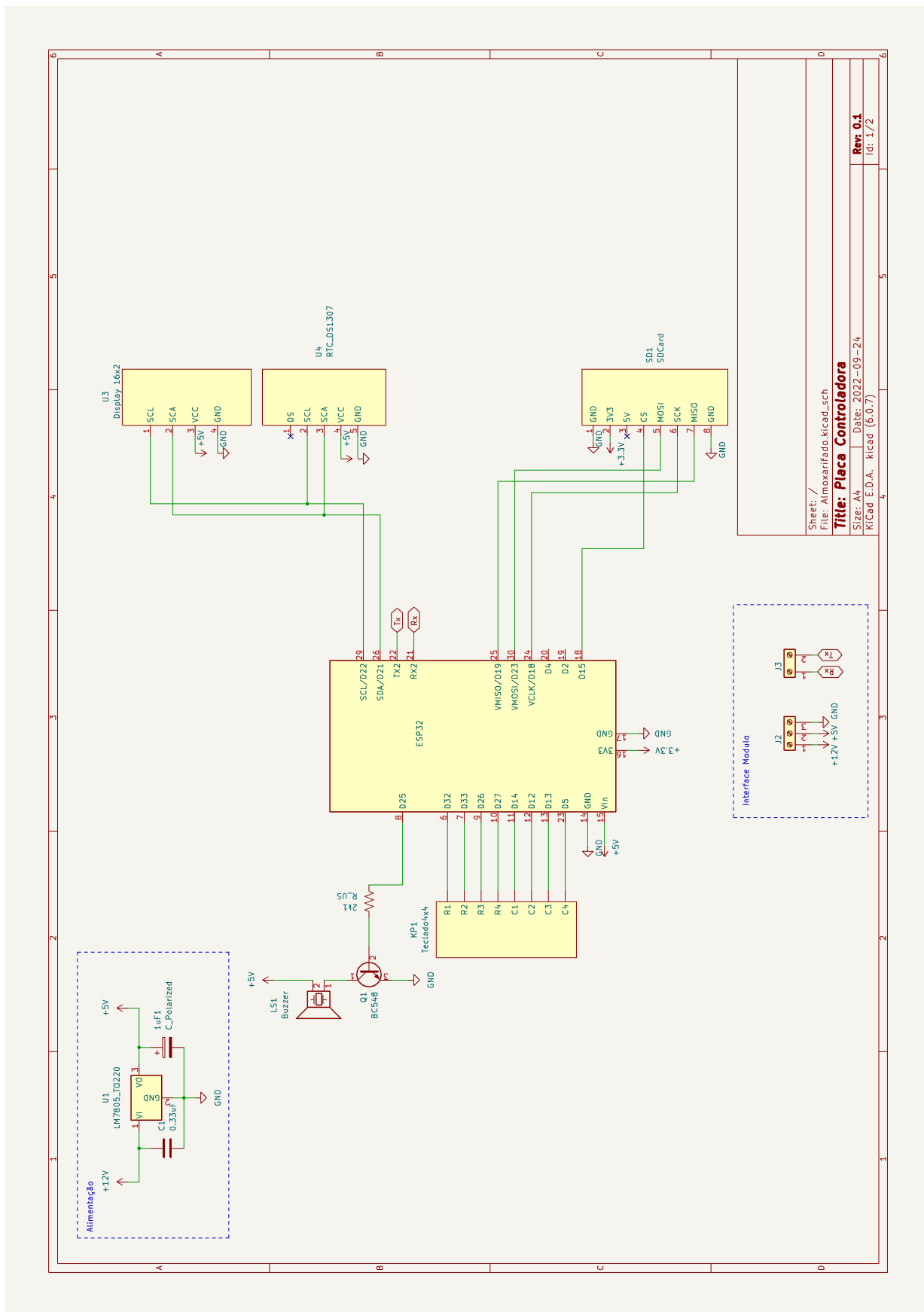


Figura 3: Esquemático do controlador

3.1.2 IMPLEMENTAÇÃO DO MÓDULO

Para esta implementação de prova de conceito, o módulo foi implementado utilizando uma placa de desenvolvimento com o microcontrolador Cortex M4 STM32411F. O esquemático do módulo é mostrado na Figura 4. Na implementação desta prova de conceito ele conta com os componentes:

- Microcontrolador STM32F411: Responsável pela leitura e controle dos sensores e atuadores e pela comunicação com o controlador via uma interface serial UART.
- Módulo HX711: Módulo com circuito integrado especializado na leitura de células de carga, responsável pela amplificação do sinal obtido da célula de carga e conversão analógica/digital do sinal amplificado.
- Célula de carga de 1kg: Célula de carga capaz de medir cargas de até 1kg.
- Relé 12V: Relé controlado pelo microcontrolador responsável pelo acionamento da fechadura.
- Fechadura Solenoide 12V: Fechadura solenoide que abre ao serem aplicados 12V.

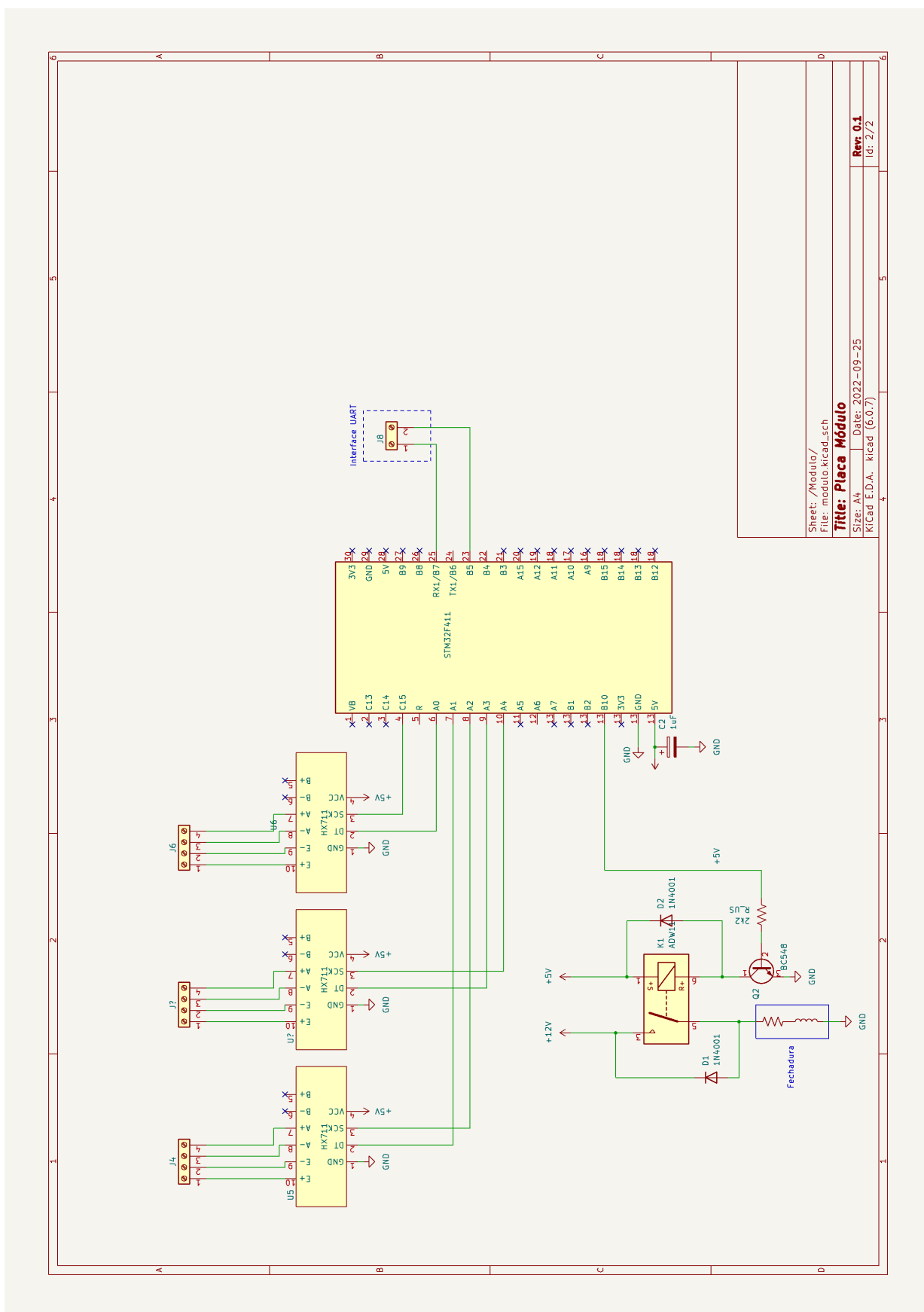


Figura 4: Esquemático do módulo

3.2 PROJETO DE SOFTWARE

A implementação do software é composta de diversos componentes como ilustrado na Figura ???. Nesta seção são descritos brevemente como cada um desses componente foi implementados.

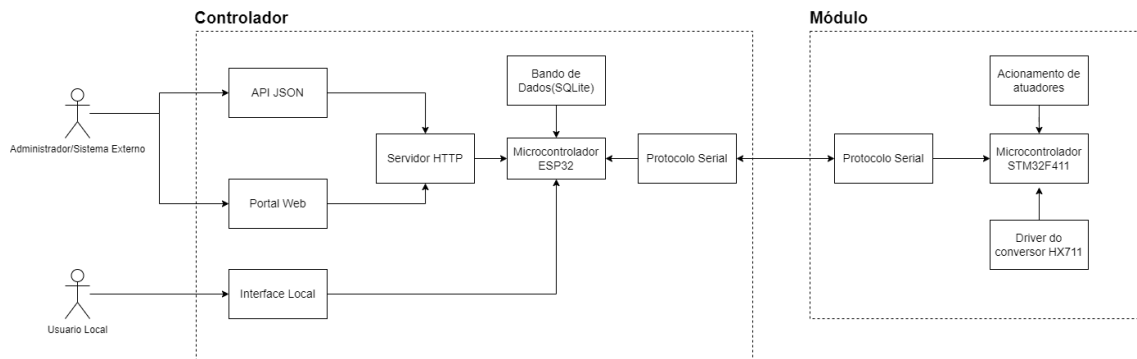


Figura 5: Diagrama de componentes do software

3.2.1 FRAMEWORK E BIBLIOTECAS

A implementação do software do controlador foi feita utilizando a camada de abstração da Arduino Framework. Essa framework nos permite utilizar diretamente bibliotecas feitas originalmente para o Arduino no ESP32 sem maiores modificações. As bibliotecas utilizadas nesse projeto foram:

- **ESP Async Server (DEV, 2022):** Versão melhorada do servidor Web incluído no ESP-IDF. Além de permitir conexões assíncronas ela também fornece funções de configuração avançadas usadas no projeto.
- **Arduino Json (BLANCHON, 2022):** Biblioteca de serialização JSON otimizada para sistemas embarcados. Utilizada no processo de geração de respostas da API JSON.
- **Liquid Crystal I2C (SCHWARTZ, 2018):** Biblioteca de displays LCD que utilizam a interface I2c.
- **RTC Lib (JEELAB, 2019):** Biblioteca que implementa o driver para diversos circuitos integrados de RTC, como o DS1307 utilizado no projeto.

- Arduino Keypad (STANLEY, 2012): Biblioteca de interface para teclados matriciais.

3.2.2 PROTOCOLO DE COMUNICAÇÃO CONTROLADOR-MÓDULO

Para a comunicação entre o controlador e o módulo, foi necessário implementar um protocolo binário semi-duplex entre eles. A escolha do protocolo semi-duplex se da ao fato de que o sistema foi projetado para casos onde múltiplos módulos podem estar conectados ao mesmo barramento serial, assim, toda comunicação é iniciada pelo controlador e apenas respondida pelo modulo referenciado na mensagem. Outra vantagem de uma conexão semi-duplex é que o sistema pode ser futuramente adaptado para usar padrões de transmissão de longa distancia como o RS-485 sem a necessidade de grandes modificações.

Os campos do pacote implementado são descritos na Tabela 1 e na Tabela 2 estão todos os comandos que foram implementados na prova de conceito.

Nome	Tipo	Descrição
moduleID	uint8	ID único do modulo a receber essa mensagem
dataSize	uint32	Quantidade de dados na mensagem(em bytes)
commandID	uint8	ID do comando a ser executado pelo modulo
data	uint8[]	Data especifica do comando, com o tamanho total de dataSize bytes
crc8	uint8[]	CRC utilizado para verificar a integridade da mensagem

Tabela 1: Campos do pacote serial

Comando	Descrição
CMD_PING	Checa conectividade com o modulo
CMD_PING_RESP	Resposta ao comando ping
CMD_GET_LC	Controlador requisita a ultima leitura da célula de carga id
CMD_GET_LC_RESP	Resposta do modulo ao comando CMD_GET_LC com
CMD_RESET_TARE	Reseta a tara da célula de carga id
CMD_RESET_TARE_RESP	Indica que o comando CMD_RESET_TARE foi bem sucedido
CMD_OPEN_LOCK	Indica ao modulo para ativar o solenoide da fechadura id
CMD_RESET_LOCK_RESP	Indica que o comando CMD_RESET_TARE foi bem sucedido
CMD_SET_TARE	Seta o valor de tara de um modo especifico id
CMD_SET_TARE_RESP	Resposta com o valor de tara

Tabela 2: Mensagens implementadas

3.2.3 API JSON

A fim de facilitar a integração com outros sistemas externos. Ao ser conectado a uma rede todas as funções do sistema como adicionar/editar usuários, modificar módulos e listar o histórico de um compartimento podem ser feitos através de uma API Restful utilizando JSON. Além disso, também implementamos uma interface web servida diretamente pelo ESP32. Essa interface faz uso da API JSON, permitindo que qualquer administrador faça modificações ao sistema diretamente em um navegador.

3.2.4 INTERFACE WEB

A interface web foi feita utilizando HTML e Javascript, sendo que todas suas requisições para o controlador são feitas através da API JSON implementada. A funcionalidade foi implementada utilizando Javascript puro, sem depender de bibliotecas externas. Já o design foi feito com o auxílio do Bootstrap(BOOTSTRAP, 2022).

3.2.5 IMPLEMENTAÇÃO DO BANCO DE DADOS

Levando em conta todas as limitações de armazenamento, memória e poder computacional de sistemas embarcados como o ESP32, escolhemos utilizar o banco de dados SQLite (SQLITE, 2022). Ele se mostrou a alternativa ideal para nossa aplicação, pois oferece um banco de dados relacional SQL utilizando apenas cerca de 160KB de FLASH e 4KB de RAM.

Outra grande vantagem da utilização do SQLite sobre uma implementação própria do banco de dados é a facilidade de migração de dados de outros bancos de dados, facilitando a integração com sistemas externos.

Para esta prova de conceito foram implementadas três tabelas, sendo uma para os usuários, uma para os compartimentos e uma para o histórico das mudanças dos compartimentos, os campos são mostrados em mais detalhes nas Tabelas 3, 4 e 5.

Nome	Tipo	Descrição
ID	INTEGER	Chave primária do usuário
nome	TEXT	Nome do usuário
login	TEXT	Identificador único usado no login do usuário (numérico)
ultimo_acesso	TEXT	Data do ultimo acesso deste usuário
privilegio	INTEGER	Nível de acesso do usuário
api_key	TEXT	Chave única de autenticação utilizada na API JSON

Tabela 3: Tabela User do banco de dados

Nome	Tipo	Descrição
ID	INTEGER	Chave primária do compartimento
object_name	TEXT	Nome do objeto sendo armazenado no compartimento
object_amount	INTEGER	Quantidade de objetos no compartimento
object_weight	REAL	Peso do objeto (em gramas)
tare	REAL	Tara do compartimento(em gramas)
mode	INTEGER	Modo de operação (retirada ou empréstimo)
last_change	TEXT	Data da última modificação do compartimento

Tabela 4: Tabela Comparment do banco de dados

Nome	Tipo	Descrição
ID	INTEGER	Chave primária do item no histórico
user_id	INTEGER	Chave do usuário que gerou essa transação
compartment_id	INTEGER	Chave do compartimento envolvido na transação
change	INTEGER	Quantidade de objetos adicionados ou removidos
date	TEXT	Data da transação

Tabela 5: Tabela Comparment_history do banco de dados

3.2.6 IMPLEMENTAÇÃO DO FIMRWARE DO MODULO

O firmware do módulo foi feito utilizando a IDE própria fornecida pelo fabricante, a STM32CubeIDE. O código escrito para o módulo foi escrito pela equipe, apenas utilizando o HAL do microcontrolador fornecido pelo fabricante. Foi necessário a implementação própria do driver do conversor de célula de carga HX711 devido ao fato dele usar um protocolo serial não convencional. O resto do código que consiste na interface serial foi adaptado do código do controlador.

3.3 PROJETO MECÂNICO

A equipe optou por utilizar chapas de papelão rígido para a construção da estrutura física do almoxarifado. A escolha por esse material se deve ao fato de que ele pode ser facilmente modelado em diversos formatos, facilitando o processo de construção. Como contraponto, não é um material muito resistente mecanicamente, por isso utilizou-se varetas de madeira para dar uma maior rigidez à estrutura. A Figura 6 mostra o início da construção da maquete, com as varetas acopladas às paredes internas.



Figura 6: Estágio inicial da construção

Outro desafio foi a construção da porta que permite o acesso aos componentes do almoxarifado. Para essa parte, foi utilizado uma chapa de madeira mdf já que o papelão é facilmente distorcido após vários movimentos. Além do espaço para depositar os objetos uma vez que a porta é aberta, planejou-se um espaço para a fixação dos componentes eletrônicos e para o controlador. A Figura 7 mostra o modelo para a construção do almoxarifado, levando-se em conta um espaço para depositar o controlador (à direita nesta Figura).

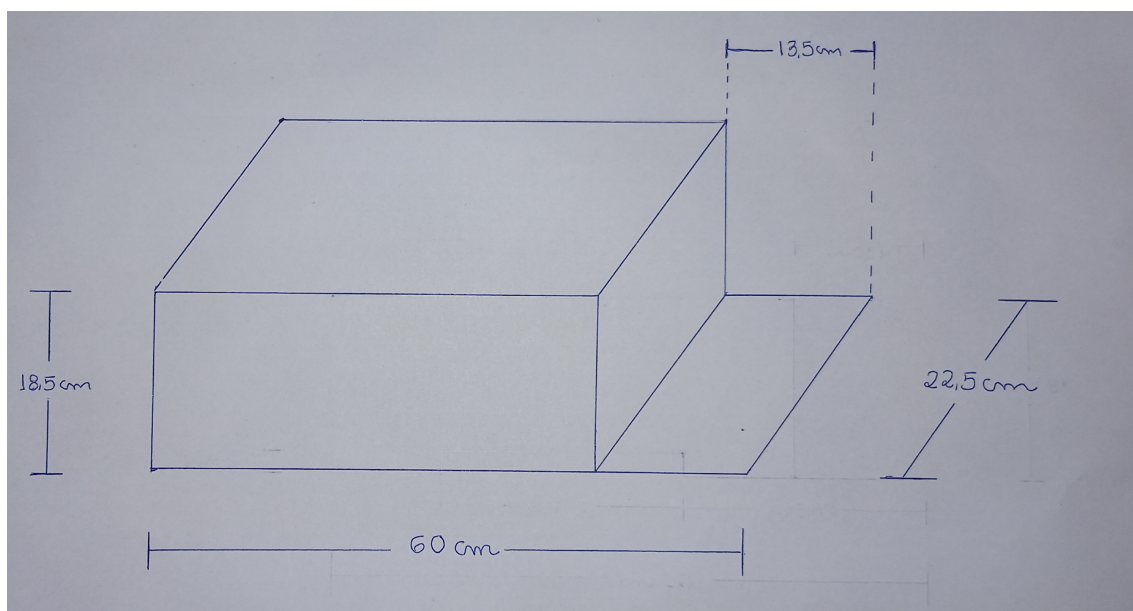


Figura 7: Modelo da estrutura, vista frontal

A Figura 8 mostra a estrutura física vista de traz. Foi deixada uma abertura de (24,5 x 2,3) cm para a fiação do controlador chegar aos módulos, ao solenoide e ao fim de curso.

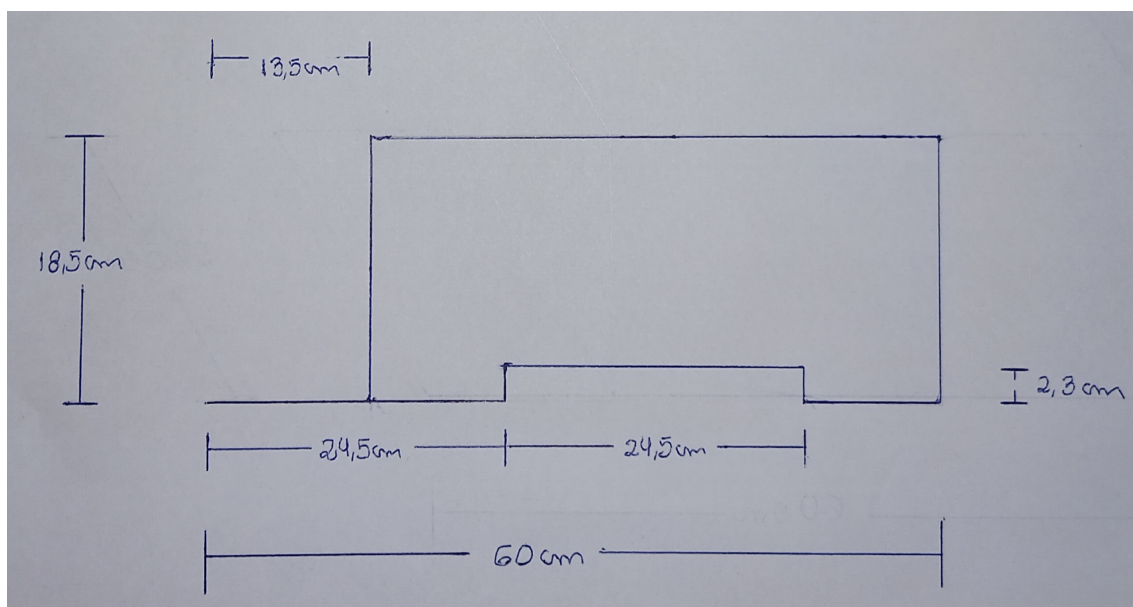


Figura 8: Modelo da estrutura, vista traseira

Após a construção, decidiu-se personalizar o almoxarifado pintando-o de azul marinho e preto (esta última a cor para a porta). Também fixou-se o controlador na maquete

de modo que o projeto fosse mais facilmente portátil (além da maquete, só é necessário a fonte de alimentação de 12V). As Figuras 9 e 10 mostram o resultado final.



Figura 9: Vista frontal da estrutura

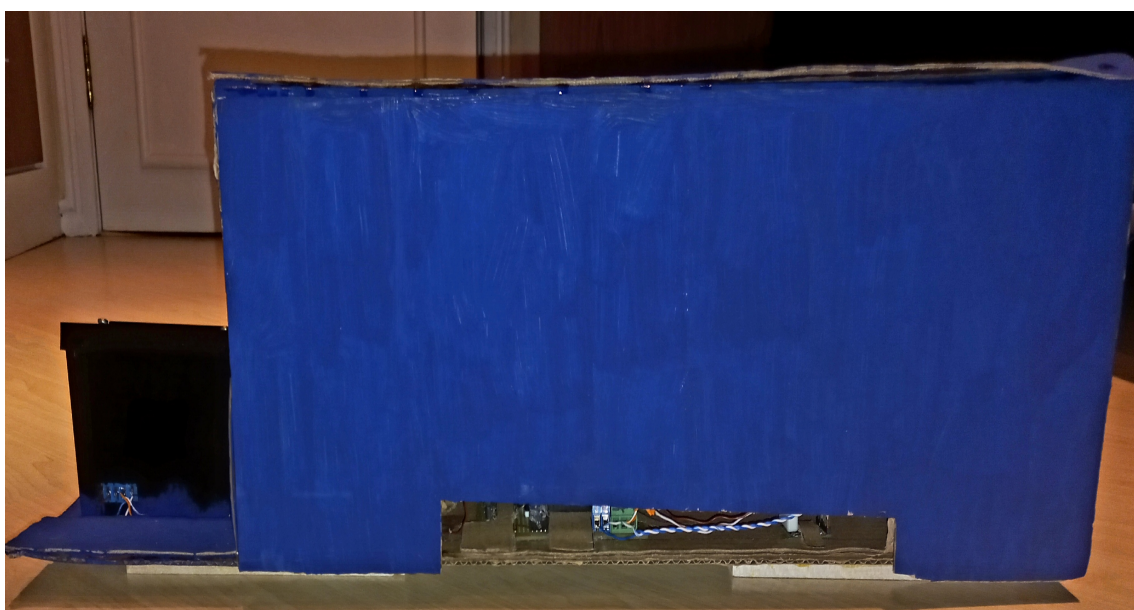


Figura 10: Vista traseira da estrutura

Aproveitou-se o resto das varetas de madeira para compor os "pés" da maquete, de modo que a base da maquete ficasse suspensa e estável.

4 EXPERIMENTOS E RESULTADOS

Os resultados obtidos nesse projeto foram demonstrados cada etapa em nosso blog, que pode ser encontrado em "https://atbk2000.github.io/Almoxarifado_Automatico". E o video com a demonstração do projeto em funcionamento pode ser visto em "<https://www.youtube.com/embed/w8k9mWNdKU>"

Na seção seguinte são descritos em mais detalhes os detalhes da implementação final.

Na Tabela 6 temos as principais características do prova de conceito finalizada:

Parâmetro	Valor
Número de compartimentos	3
Peso máximo por compartimento	1 kg
Peso mínimo por item	5g

Tabela 6: Características do almoxarifado construído

Tendo em vista as características da Tabela 6 e o fato de que os objetos depositados em um mesmo módulo devem possuir o mesmo peso, escolheu-se pilhas como carga. Tomou-se o cuidado para que as pilhas fossem da mesma marca para que não houvesse uma variação de massa significativa. A Tabela 7 mostra os itens utilizados e suas respectivas massas.

Item	Massa
Pilha AAA	11.4g
Pilha AA	23.4g
Pilha C	72.6g

Tabela 7: Itens guardados no almoxarifado

Uma vez escolhidos os itens, é necessário cadastrá-los no banco de dados. Para isso foi utilizado o site hospedado pelo ESP32, que permite essa configuração. A Figura 11 mostra esse processo. Importante lembrar que antes de cadastrar o objeto, é bom utilizar a opção de tara para que massa do compartimento em si não seja contabilizada, apenas do objeto. Além do mais, é uma boa prática de tempos em tempos calibrar os módulos.

Figura 11: Editando compartimento

Depois dos objetos serem cadastrados, é possível visualizar a quantidade disponível em estoque. Na Figura 12 é possível ver as pilhas disponíveis no almoxarifado.

ID	ITEM	QUANTIDADE	MODO	ULTIMO ACESSO	OPÇÕES
0	Pilha AA	0	0	2022-12-04 18:30:45	Editar Historico
1	Pilha C	0	0	2022-12-04 18:32:07	Editar Historico
2	Pilha AAA	0	0	2022-12-04 18:31:43	Editar Historico

Figura 12: Visualização do estoque através do site

Também, através do site, é possível adicionar mais usuários, sendo eles usuários comuns ou administradores (possui todos os privilégios). Desse modo, alguém que desejar ter acesso ao almoxarifado deve digitar seu ID e sua senha no teclado disponível ao lado do

almoxarifado. Se suas credenciais forem válidas, a porta é liberada e o usuário pode retirar os itens guardados ou devolvê-los.

5.2 CUSTOS

A lista de componentes utilizados e seus respectivos custos são listados na Tabela 8. O gasto total do projeto foi de R\$ 320,50. O bruto do custo do projeto foi justamente nas células de carga e seus respectivos módulos e nos microcontroladores.

Componente	Preço	Quantidade	Adquirido
ESP32 WROOM32	R\$40,00	1	Não
STM32F411	R\$30,00	1	Não
Módulo HX711	R\$18,00	4	Sim
Célula de carga 1Kg	R\$15,00	4	Sim
Placa Padrão 10x10cm	R\$15,00	1	Sim
Placa Padrão 10x5cm	R\$7,00	1	Sim
Módulo Rtc DS1307	R\$20,00	1	Sim
Módulo leitor cartão SD	R\$12,00	1	Sim
Teclado matricial 4x4	R\$10,00	1	Sim
Fechadura solenoide 12v	R\$35,00	1	Sim
Regulador de tensão LM7085	R\$0,50	1	Sim
Placa de papelão rígido 30x100 cm	R\$5,00	6	Sim
Chapa de mdf 30x42 cm	R\$3,50	1	Não
Regulador de tensão LM7085	R\$0,50	1	Sim
Componentes eletrônicos passivos	R\$5,00	1	Sim
Parafusos	R\$5,00	1	Sim

Tabela 8: Lista de materiais

6 CONCLUSÕES

6.1 CONCLUSÕES

Neste trabalho a equipe conseguiu realizar com sucesso a implementação da prova de conceito de um sistema que foi proposta. Mesmo com as limitações orçamentarias e de tempo, o sistema de prova de conceito é bom o suficiente para demonstrar a solução proposta para o problema de automatização de um almoxarifado.

O maior desafio na implementação do projeto foi a parte de software, por envolver muitas tecnologias desconhecidas pela equipe. Esse problema foi contornado alocando uma quantidade de tempo consideravelmente maior do que havia sido planejado. Mas o resultado obtido embora não ideal, é bom o suficiente para demonstrar as funcionalidades do sistema.

Outro grande desafio foi a falta de expertise e ferramentas da equipe para a construção da parte mecânica do projeto. O que acabou resultando numa base mecânica simples que embora cumpra seu proposito em demonstrar o sistema, deixou a desejar de maneira geral.

6.2 TRABALHOS FUTUROS

O sistema foi projetado com a expansibilidade em mente, logo existem muitas ideias que foram descartadas por não serem praticas na implementação da prova de conceito, mas que seriam interessantes em um sistemas real.

Entre elas, seria interessante a implementação de um sistema com múltiplos módulos, sendo capaz de gerenciar áreas muito maiores, como almoxarifados reais ou armazéns.

Outra adição interessante seria a adição de sensores que permitissem que itens distintos pudessem ocupar o mesmo compartimento, através de uso de RFID de longa

distancia.

Também foi pensando em futuramente utilizar um microprocessador no lugar de um microcontrolador, devido a complexidade do software sendo executado no Esp32, provavelmente faria mais sentido utilizar um computador de placa única como o Raspberry pi. Pois além de fornecer muito mais recursos como memória e armazenamento não volátil, esses sistemas contam com software que facilitariam muito na hora da implementação de um servidor web ou banco de dados. Além de fornecer mais segurança e mais funcionalidades em relação a nossa implementação.

REFERÊNCIAS

BLANCHON, B. **ArduinoJson: Efficient JSON serialization for embedded C++**. 2022. Disponível em: <<https://arduinojson.org/>>. Acesso em: 1 de Dezembro de 2022.

BOOTSTRAP. **Build fast, responsive sites with Bootstrap**. 2022. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 3 de Dezembro de 2022.

DEV me-no. **ESPAsyncWebServer**. 2022. Disponível em: <<https://github.com/me-no-dev/ESPAsyncWebServer>>. Acesso em: 3 de Dezembro de 2022.

JEELAB. **RTCLib**. 2019. Disponível em: <<https://github.com/adafruit/RTCLib>>. Acesso em: 1 de Dezembro de 2022.

SCHWARTZ, M. **LiquidCrystal Arduino library for I2C LCD displays**. 2018. Disponível em: <https://github.com/johnrickman/LiquidCrystal_I2C>. Acesso em: 1 de Dezembro de 2022.

SQLITE. **SQLite**. 2022. Disponível em: <<https://www.sqlite.org/index.html>>. Acesso em: 3 de Dezembro de 2022.

STANLEY, A. B. M. **Keypad Library for Arduino**. 2012. Disponível em: <<https://playground.arduino.cc/Code/Keypad/>>. Acesso em: 1 de Dezembro de 2022.

TRENT, D. **Strain Gauge Load Cell Basics**. 2019. Disponível em: <<https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html>>. Acesso em: 1 de Dezembro de 2022.