

Desenvolvimento de um sistema de controle de estoque autônomo

Oliveira, Leonardo Ghedin
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
leonardo.go2000@aluno.ifsc.edu.br

Santos, Fernando Rodrigues
Docente da Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
fernando.rodrigues@ifsc.edu.br

Resumo - O avanço da manufatura e a produção em larga escala estão trazendo cada vez mais a necessidade dos estoques para as empresas, tanto para as pequenas quanto para as de grande porte, visto que uma boa estrutura de organização dos insumos permite executar os processos de produção com maior agilidade e eficiência, além da otimização dos recursos financeiros. As empresas que buscam a automação de seus estoques procuram métodos de gerenciamento e equipamentos que consigam lidar com a alta demanda de pedidos, assim como também a fácil organização de seu espaço físico. Neste sentido, percebeu-se a oportunidade de desenvolver um sistema de controle de estoque utilizando a Internet das Coisas, com o objetivo de melhorar a eficiência, qualidade e velocidade do processo de gestão de estoques. Assim, a proposta desse trabalho tem por finalidade desenvolver uma aplicação web integrada com uma plataforma robótica, utilizando para isso um servidor de dados como ponte entre elas, sendo possível atender às ordens de pedidos do usuário. O funcionamento do sistema se baseia na estrutura de gerenciamento de estoque dos métodos Duas Gavetas e Kanban, utilizando-os para criar uma interface onde o usuário possa fazer as requisições dos itens do estoque e o robô móvel, faça a coleta dos itens e os leve até a área de despacho.

Palavras chave – Controle de Estoque, Internet das Coisas, Aplicação Web, Plataforma Robótica.

I. INTRODUÇÃO

Desde o começo de sua história, a humanidade tem usado estoques de diferentes recursos, como alimentos e ferramentas, para apoiar sua sobrevivência e desenvolvimento, estando presente em praticamente todo o tipo de organização [1].

As grandes empresas já perceberam como a gestão de estoque pode trazer vantagens competitivas e estão olhando para os seus estoques ao longo de toda a cadeia de suprimentos. Já nas pequenas empresas, a gestão de estoque é vista como uma parte não estratégica [1].

Um dos requisitos para uma empresa entregar os seus pedidos no prazo, com eficiência e eficácia, é possuir os processos que influenciam diretamente no sistema em sincronia. Um dos processos mais importantes que podem influenciar diretamente na entrega ao cliente é o de armazenagem. É de interesse dessas organizações otimizar este processo, pois os custos de manutenibilidade são muito altos [4].

Segundo Dias (2011), uma das maiores metas dentro das empresas é a redução de seus estoques, pois com um estoque minimizado o capital investido se torna menor, aumentando a eficiência dos recursos financeiros.

Os estoques necessitam de gerenciamento, pois correspondem a 60% dos custos do produto final. As

empresas mais desenvolvidas utilizam sistemas digitais ou métodos autônomos, como robôs, para realizarem o gerenciamento dos seus estoques. De acordo com Aires (2019), esses sistemas podem reconhecer padrões e disponibilizá-los para tomada de decisão no que se refere a definição dos níveis de estoque, tempo de reposição, requisição de materiais e compra de insumos.

Alguns dos métodos mais utilizados nas empresas para o gerenciamento de seus estoques são: Duas Gavetas e Kanban. A função do método Duas Gavetas no estoque é trabalhar com uma margem de segurança, tendo sempre um estoque para o consumo principal, além de outro de suprimentos reserva [3]. Já o método Kanban atua na organização dos pedidos, trazendo o conceito de trabalhar com um estoque baixo e utilizá-lo através da demanda da fábrica, ou seja, a própria produção faz o pedido das peças, através de cartões Kanban ou via sistemas de gerenciamento mais modernos, levando então à empresa a não produzir sem antes ter os seus pedidos definidos e organizados dentro do estoque [6].

Uma gestão de estoque ineficiente pode ser um grande problema no mercado, uma vez que pode acarretar em níveis de suprimentos inadequados e o não atendimento de demandas essenciais pela falta de matéria-prima, ou a perda de produtos por vencimento [2]. O cenário do gerenciamento de estoque físico é rodeado por vários desafios, desde o excesso e a falta de produtos, passando pela propensão a erros humanos, até os elevados custos de armazenamento e processos ineficientes. A dependência exclusiva da mão de obra humana muitas vezes resulta em erros operacionais, gerando aumento da complexidade na cadeia de suprimentos com mais pessoas envolvidas, além da falta de visibilidade do processo em tempo real.

A automatização de processos industriais surge como uma estratégia que envolve o uso de sistemas que controlam as informações recebidas e transmitidas pelos usuários, proporcionando uma maior eficiência e otimização na gestão das operações. Complementar à automação, a Internet das Coisas (IoT) tem por finalidade realizar a conectividade entre diversos dispositivos, por meio de sistemas que se comunicam através da internet e geram a tomada de decisão nos processos industriais. Essa abordagem não apenas simplifica as tarefas, mas também melhora a precisão e a confiabilidade dos processos, impulsionando, assim, a produtividade e a qualidade dos produtos. [7]

Neste cenário de constante evolução tecnológica, juntamente com um mercado cada vez mais competitivo, surgem soluções de gerenciamento de estoques autônomos, porém a maioria destas são oferecidas como serviços em sistemas proprietários. Dessa forma, a sua utilização

envolve o emprego de recursos e gera dependência destes serviços, não sendo atrativos para empresas de pequeno porte. Por exemplo, em 2012 a Amazon adquiriu uma empresa de robôs de *picking* para não depender mais de softwares e equipamentos de terceiros. Logo, a automação e gerenciamento de estoque se revelam essenciais para manter a competitividade das empresas em um mercado cada vez mais global.

Diante disso, o presente trabalho visa o desenvolvimento de um sistema de gerenciamento e controle de estoque automatizado através de um robô móvel, com conectividade IoT para disponibilização de informações para os usuários em uma aplicação Web. Esse sistema terá funcionalidades baseadas nos métodos Duas Gavetas e Kanban para realizar as solicitações de produtos, bem como o monitoramento das variáveis de abastecimento e status do sistema. Sendo assim, busca-se projetar um sistema independente para gerenciamento de estoque, utilizando de ferramentas de software de livre acesso (*open source*).

II. METODOLOGIA

O estabelecimento de uma metodologia padroniza os procedimentos a serem seguidos durante o desenvolvimento do projeto.

A metodologia adotada na construção deste sistema é baseada em cinco níveis de abstração, sendo eles: Requisitos, Especificações, Arquitetura, Componentes e Integração [8].

- a) **Requisitos:** Os requisitos podem ser enxergados como funcionalidades do sistema, o que a parte de hardware e a de software deve executar e quais as circunstâncias sob quais elas devem operar.
- b) **Especificações:** As especificações de projeto podem ser caracterizadas como a descrição mais precisa do sistema, como ele se comporta, deve garantir que os requisitos sejam atendidos para guiar o desenvolvimento do sistema.
- c) **Arquitetura:** A arquitetura de projeto define como o sistema será construído, levando em consideração estruturas de hardware e software, utilizando ferramentas como diagrama de blocos com a interação entre eles.
- d) **Componentes:** Os componentes do projeto são selecionados e definidos em conformidade com as especificações e a arquitetura, sendo compatíveis para a integração do sistema como um todo.
- e) **Integração:** A integração se dá pela união dos componentes do sistema, buscando testes e a integração entre os módulos, com um plano de integração capaz de solucionar erros rapidamente.

Com base nessa abordagem, definiu-se cada nível para o desenvolvimento do sistema.

A. Requisitos

Os requisitos do projeto são distribuídos em dois grupos principais, sendo eles: os requisitos funcionais, que definem o que o sistema como um todo deve realizar; e os requisitos não funcionais, cuja definição se dá pelas restrições impostas ao projeto.

Requisitos Funcionais: A aplicação web que o usuário terá acesso deve ter uma interface intuitiva, que permita acessar as informações do estoque e enviar comandos ao robô no estoque. O robô deve ser capaz de se movimentar em todas as direções, coletar os produtos do estoque em diferentes alturas e executar suas ações de forma automática, além de ser capaz de se comunicar via internet e ter sua fonte de energia através de baterias.

Requisitos não funcionais: O sistema deve se comunicar de forma síncrona, enquanto o robô não deve usar fios de carregamento ou ser conectado diretamente a fontes de alimentação.

B. Especificações

A especificação de hardware do sistema se baseia no sensoriamento do robô e no processamento de dados e parâmetros do ambiente. São necessários sensores para medir a distância entre o robô e os produtos do estoque, bem como outros obstáculos, para evitar colisões. Esses sensores podem ser do tipo ultrassônicos e infravermelhos. Os sensores ultrassônicos precisam estar posicionados nas partes dianteira e traseira para que o robô consiga guiar-se baseado na distância em relação aos obstáculos, e assim, nenhum objeto interfira na sua locomoção. Já os sensores infravermelhos são posicionados nas laterais para garantir uma distância segura entre o robô e os outros obstáculos. A partir dessas informações, um microcontrolador principal pode realizar o controle dos movimentos do robô por meio de motores DC e servomotores. O microcontrolador que controla o robô deve se comunicar com uma aplicação Web, via Wi-Fi, e enviar dados e receber requisições do usuário para retirar do estoque os produtos solicitados.

A aplicação web é baseada em dois métodos de gerenciamento. A estrutura de software trabalhará com uma junção entre os métodos Duas Gavetas e Kanban, utilizando-os para gerar uma interface intuitiva ao usuário. O método Duas Gavetas será implementado para quando o sistema atingir uma quantidade baixa de produtos, o usuário seja alertado para agendar o abastecimento do estoque. Já o método Kanban atuará na requisição de produtos, onde o usuário poderá selecionar vários itens do estoque para posteriormente serem coletados e entregues na área de despacho, além de possuir também as listas de pedidos feitos, em processo e finalizados.

A aplicação apresentará para o usuário as informações do estoque, dos procedimentos e os status dos processos que estão sendo realizados. Para isso, o sistema precisa realizar o armazenamento de dados e hospedar da aplicação Web em um servidor. A comunicação entre o robô e o servidor será através do protocolo MQTT (*Message Queuing Telemetry Transport*), baseado em publicação e assinatura (*publish/subscribe*), onde um dispositivo é responsável por publicar (*Publisher*) a mensagem no servidor, que opera como um intermediário (*Broker*) e retransmite as mensagens recebidas aos clientes assinantes (*Subscriber*). Já a comunicação entre a aplicação web e o servidor será via HTTP, baseado em pedidos (*requests*) e respostas (*responses*). Dessa forma, será possível realizar a coleta e transferência de dados tanto para a célula robótica quanto para a aplicação Web.

C. Arquitetura

A arquitetura do sistema se baseia na estrutura em diagrama de blocos com a interação entre os módulos, separada em duas, sendo elas: Hardware e Software.

A arquitetura de hardware do sistema é dividida em três sistemas computacionais, sendo eles: Microcontrolador ATmega328, Microcontrolador ESP32 e o Servidor, conforme Figura 1 abaixo.

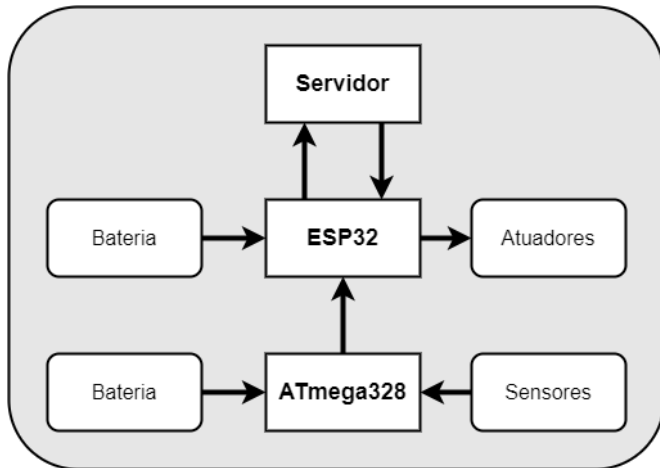


Figura 1: Arquitetura de Hardware.

O conjunto de sensores, que se encontram conectados ao ATmega328, tem como objetivo fornecer ao microcontrolador dados sobre o ambiente onde o robô se encontra, e essas informações são transmitidas através de cabos para o ESP32 via comunicação I2C (*Inter-Integrated Circuit*). O microcontrolador ESP32 se comunica com o servidor através da comunicação Wi-Fi, utilizando o protocolo MQTT para trocar dados, atualizações e recebimento dos pedidos do usuário. Já o servidor tem como função prover os serviços através de uma aplicação Web. A alimentação de ambos os microcontroladores se dá por meio de baterias.

A arquitetura de software (Figura 2) apresenta detalhadamente o funcionamento de cada plataforma computacional da arquitetura anterior, como os componentes de software e as comunicações entre eles, para definição do comportamento do sistema.

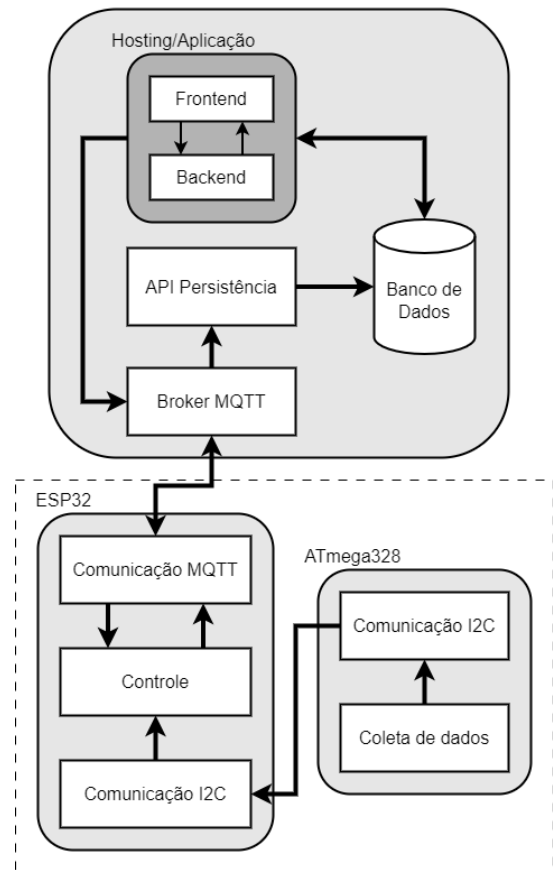


Figura 2: Arquitetura de Software

Na arquitetura proposta, somente o microcontrolador ATmega328 lê os dados dos sensores, fornecendo as informações do ambiente ao sistema. O microcontrolador ESP32 tem como função receber os dados do ATmega328 e do servidor, utilizando tais informações para controlar a plataforma robótica através dos atuadores e enviando *feedbacks* ao servidor. O servidor tem três principais funções dentro do sistema: fornecer um broker para comunicação entre os dispositivos, armazenar os dados utilizando um banco de dados, utilizando de uma API (*Application Programming Interface*) para coletar os dados do Broker e armazená-los no banco de dados, além de prover uma aplicação Web para o usuário. A aplicação Web de gerenciamento se hospeda dentro do servidor, possuindo um frontend para exibir e interagir com o usuário, bem como um backend para realizar a conexão com os outros componentes de software.

D. Componentes

Os componentes de hardware que compõe a célula robótica são: Plataforma Robótica Rocket Tank, Kit RoboARM, Kit de expansão, Plataforma Vespa ESP32, Microcontrolador ATmega328, dois sensores ultrassônicos HC-SR04, dois sensores infravermelho IR, dois motores DC de 6V e quatro servo motores DS-S006, dois conjuntos de baterias, uma de 9V e um módulo de duas baterias de 3,7V.

A integração entre os componentes Rocket Tank, RoboARM e Kit de expansão resulta na estrutura física do robô, que pode ser visualizado na Figura 3.

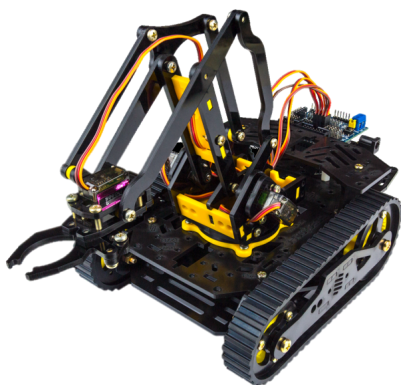


Figura 3: Célula robótica.

Os componentes de software do sistema são:

- **Broker MQTT:** para implementação do protocolo MQTT será utilizado o Broker Eclipse Mosquitto.
- **Banco de dados:** o banco de dados escolhido para o sistema é o MongoDB, que é do tipo NoSQL (*Not Only Structured Query Language*), tipo não relacional e trabalha com documentos no formato JSON (*JavaScript Object Notation*).
- **API de Persistência:** aplicação construída em linguagem Python, que tem como função ser um assinante para receber as mensagens publicadas no Broker MQTT e armazená-las no banco de dados.
- **Aplicação web:** consiste em uma *dashboard* para a integração com o usuário, desenvolvida utilizando o framework React, com linguagens de programação JavaScript e HTML (*HyperText Markup Language*), sendo a primeira para a configuração do *Back-End* e a segunda para configuração de parâmetros visuais do *Front-End*. E para as comunicações com outros componentes de software do sistema, as bibliotecas *Mongoose*, *Express* e *Cors* são utilizadas na construção do *Back-End*.
- **ESP32:** O software deste microcontrolador utiliza a linguagem de programação C/C++, com uma estrutura de programação que trabalha com arquivos no formato JSON através da biblioteca *ArduinoJson*. Ele possui um módulo para realizar a comunicação com o Broker utilizando as bibliotecas *WiFi* e *PubSubClient*, para conexão Wi-Fi e comunicação via MQTT. Outro módulo de comunicação I2C através da biblioteca *Wire*. Além de um módulo principal de controle, que é responsável por receber as informações dos pedidos e dos sensores, processar essas informações e controlar a movimentação do robô através de seus atuadores.
- **ATmega328:** O software deste microcontrolador utiliza a linguagem de programação C/C++ e sua funcionalidade consiste em realizar a leitura dos sensores, um deles através da biblioteca *Ultrasonic*, além do envio destes dados para o microcontrolador ESP32 via comunicação serial I2C.

E. Integração

A integração do sistema foi dividida em 5 etapas de desenvolvimento. Cada uma dessas etapas visa a junção entre as partes. As etapas de integração são:

- 1) Criação da plataforma Web
- 2) Conectividade com banco de dados
- 3) Implementação do Broker MQTT
- 4) Integração com a plataforma robótica
- 5) Operação do sistema completo

Esta segmentação estratégica de divisão entre etapas não apenas agiliza o processo, mas também permite uma identificação mais rápida e precisa de eventuais erros, contribuindo para a eficiência do projeto.

III. DESENVOLVIMENTO

O projeto foi executado seguindo os cinco níveis de integração apresentados na metodologia de Wolf, considerando a junção e comunicação entre as partes durante o avanço das etapas.

1) Criação da plataforma Web

O desenvolvimento do sistema se deu em um computador pessoal do tipo Notebook, que será como servidor. A plataforma Web e demais softwares para consulta e armazenamento de dados serão hospedados nele.

A aplicação Web é estruturada conforme a arquitetura apresentada na Figura 4, onde o *Front-End* utiliza o framework React para disponibilizar as informações ao usuário e se comunica através do protocolo HTTP com o usuário e o *Back-End*. O *Back-End* utiliza da biblioteca *Mongoose* para realizar a conectividade com o banco de dados e da *Express* para a criação de um servidor dedicado a API de consulta. A biblioteca *Precompiled MQTT* traz recursos para a utilização das informações recebidas através do protocolo MQTT na aplicação Web.

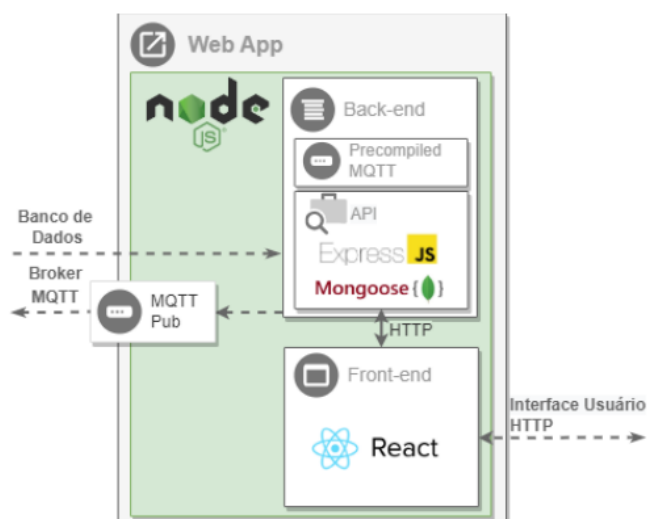


Figura 4: Estrutura de interação da aplicação Web..

A plataforma web é composta por uma aplicação contendo as seguintes abas: Home, Estoque, Kanban, Contato, Empresa e Login.

A aba Home traz uma mensagem de boas-vindas ao usuário, uma imagem do robô que irá coletar os itens no estoque, as instruções sobre o uso da plataforma e também um botão com a descrição “Realizar pedido” que quando clicado direciona o usuário para a aba Estoque, conforme pode ser visto na Figura 5. A construção da página foi realizada utilizando do *framework* React, com a utilização das linguagens de programação HTML e JavaScript para caracterização e estruturação.



Figura 5: Página Home.

A aba Estoque é onde o usuário consegue visualizar todos os itens disponíveis no estoque para requisição, trazendo o nome do produto e a sua quantidade atual. Nela, o método Duas Gavetas de gerenciamento é implementado através da coloração das quantidades. Quando o estoque de consumo principal está operando em condições normais, a quantidade de itens é apresentada na cor verde. Já quando a quantidade está abaixo de 40% de sua capacidade máxima, esta informação é apresentada ao usuário na cor vermelha, indicando que o estoque principal do produto foi consumido e que estão sendo utilizados os itens da reserva de suprimentos, que no método Duas Gavetas se trata da caixa menor. Para a construção dessa página, a biblioteca *Mongoose* foi utilizada para conexão com o banco de dados MongoDB e para disponibilizar as informações das quantidades de produtos, e assim, esta página acessa o link com os dados para apresentá-los para o usuário.

Logo abaixo de cada imagem da caixa de produto foram desenvolvidos dois campos de interação com o usuário, uma caixa de texto para inserção da quantidade do item desejado e um botão para o envio deste valor ao broker que opera no *Back-End* e posteriormente é enviado à célula robótica. A apresentação dessas informações pode ser vista na Figura 6.

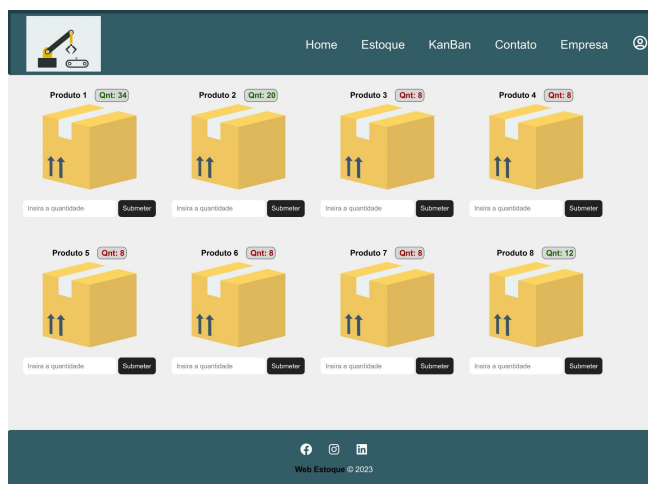


Figura 6: Página Estoque.

A aba Kanban permite ao usuário acompanhar o fluxo dos seus pedidos, informando os status das operações do robô. Esses status são divididos em três colunas, *To do*, *Doing* e *Done*. A coluna *To do* informa a fila de itens solicitados em espera para coleta, a coluna *Doing* informa os itens que estão sendo coletados atualmente, e a coluna *Done* informa os itens que já foram coletados e entregues do pedido. A API de consulta ao banco de dados disponibiliza as informações para as colunas em outros links, onde a aplicação Web utiliza de métodos HTML para realizar a consulta e trazer as informações para a página. A disposição das colunas e informações dentro da aba KanBan podem ser vistas na Figura 7.

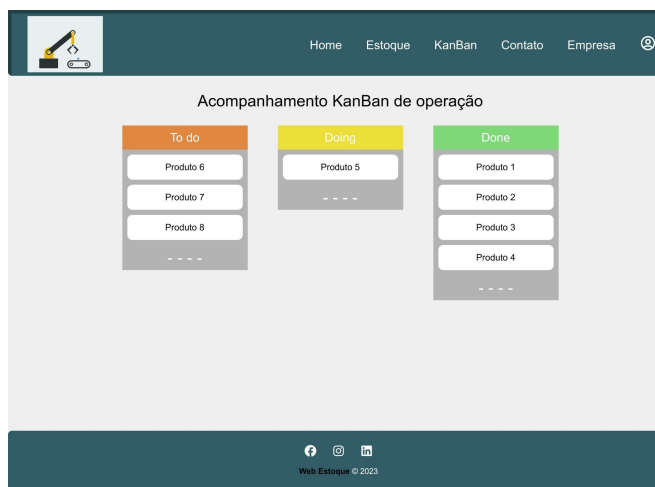


Figura 7: Página de acompanhamento KanBan.

As abas Empresa e Contato são informativas, contendo as informações sobre o projeto, empresa ou instituição de operação do sistema de gerenciamento de estoque, as pessoas envolvidas no desenvolvimento, contato dos criadores e recursos utilizados.

A aba Login foi desenvolvida para que o usuário possa acessar o sistema de forma segura, necessitando inserir as informações de usuário e senha para ser direcionado a outra aba. Após o acesso, é possível realizar alterações manuais dentro do estoque, como a exclusão de itens ou a atualização das características dos mesmos através da função de edição. O botão Deletar envia o número de identificação (ID) do

produto ao *Back-End* para uma função em JavaScript que realiza a exclusão do produto no banco de dados. Já o botão Atualizar direciona o usuário a outra aba relacionada ao produto selecionado, para que as alterações possam ser feitas e salvas, conforme Figura 8. O método de consulta de dados utiliza as mesmas bibliotecas da aba Estoque, trazendo as informações do banco de dados e apresentando-as ao usuário. O *layout* da página é similar à do estoque físico, trazendo os mesmos itens, como pode ser visto na Figura 8.

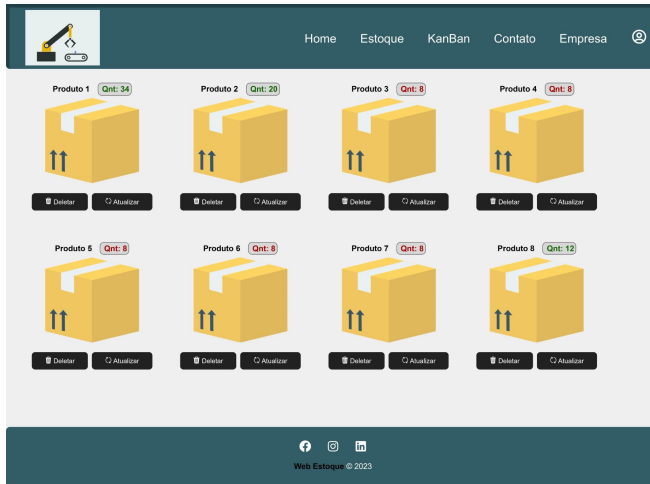


Figura 8: Página de edição e exclusão após o login.

Após o clique no botão Atualizar, o usuário é direcionado à tela com as informações do produto, para que possa realizar as alterações, conforme pode ser visto na Figura 9.

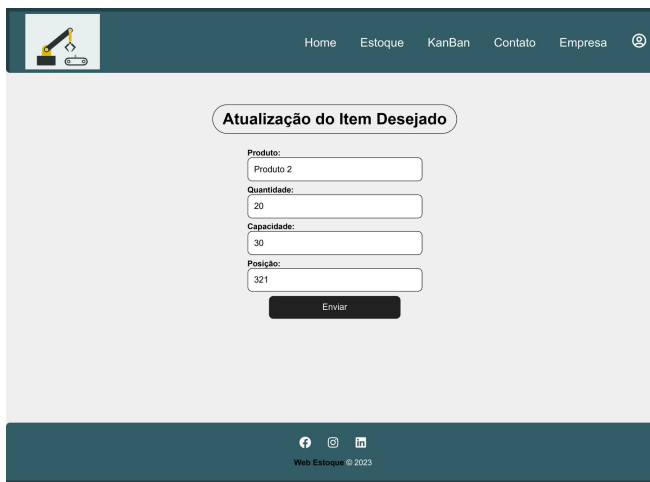


Figura 9: Aba de edição dos produtos.

Após inserir as alterações, é necessário clicar no botão Enviar para que a função de atualização que envia os novos dados do produto seja executada, e essas alterações sejam realizadas dentro do banco de dados.

2) Conectividade com banco de dados

Para que a aplicação de gerenciamento seja capaz de consultar os dados do estoque é necessário que haja uma conexão direta com um banco de dados.

O banco de dados escolhido para o projeto é o MongoDB, que possui um estilo de construção NoSQL e trabalha com documentos JSON. Os documentos possuem cinco principais campos, sendo elas: ID, Produto, Quantidade, Capacidade e Posição. A utilização de um arranjo JSON auxilia no acesso dos dados e na organização das informações, facilitando a manipulação dos dados tanto no *Back-end* da aplicação de gerenciamento quanto na utilização da mensagem no microcontrolador. A representação do arranjo utilizado no projeto pode ser visualizado na Figura 10, onde detalha os dados do produto e o ID vinculado a ele.



Figura 10: Esquema JSON.

Para a conexão entre aplicação web e banco de dados foi desenvolvido uma API, um subprograma capaz de acessar todos os dados e exibi-los em um link local no navegador, para que posteriormente a aplicação Web faça as requisições neste link e consiga acessar os dados necessários. Durante a criação da API foi necessário a utilização de bibliotecas específicas como a *Mongoose*, que realiza a conexão com o banco de dados, a *Cors*, necessária para a criação de um sub servidor para hospedar a aplicação e a *Express*, que fornece um conjunto robusto de recursos para aplicativos web e móvel.

O banco de dados utilizado possui quatro coleções: Produtos, Pedidos, Coletando e Concluídos, como mostra a Figura 11. A coleção de Produtos tem como função armazenar todos os dados dos produtos que estão disponíveis para coleta dentro do estoque. Já a de Pedidos armazena somente os produtos que o usuário solicitou para coleta na página de Estoque, criando então uma sequência de pedidos. A coleção Coletando armazena somente um produto por vez, visto que é a coleção que informa à página KanBan qual o produto que está sendo coletado atualmente pelo robô. Por fim, a coleção Concluídos armazena todos os pedidos já coletados e entregues pelo robô.

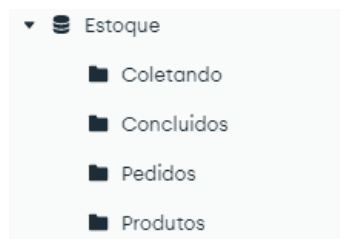


Figura 11: Coleções do banco de dados MongoDB.

As coleções Pedidos, Coletando e Concluídos são de uso exclusivo da aba Kanban, pois é a partir delas que a página consegue coletar as informações para construir o sistema Kanban de monitoramento.

3) Implementação Broker MQTT

O Broker MQTT utilizado para gerenciar a troca de informações entre a aplicação Web, banco de dados e o microcontrolador é o Eclipse Mosquitto. Esse broker foi

instalado em um notebook com o sistema operacional Windows de 64 bits e está configurado para receber mensagens internas e externas de qualquer outro dispositivo. A aplicação Web se comunica com o broker publicando as mensagens de pedidos em um tópico chamado “test” através da porta 1884, a qual foi configurada para receber mensagens através do protocolo *websockets* de conexão interna entre processos computacionais. O microcontrolador ESP32 também publica as mensagens no mesmo tópico que a aplicação Web, porém utiliza o protocolo MQTT na porta 1883, que foi configurada para receber dados de dispositivos externos ao servidor. Para isso, é necessário realizar uma liberação do firewall do sistema operacional Windows, para que as mensagens possam ser recebidas e enviadas ao microcontrolador.

Para que as mensagens publicadas pela aplicação Web e pelo microcontrolador ESP32 sejam armazenadas no banco de dados foi necessário a implementação de um programa para realizar a persistência dos dados. Este programa, construído em linguagem Python, é um assinante MQTT (*Subscriber*) que recebe todas as mensagens publicadas no Broker MQTT, no formato JSON, utilizando a biblioteca *paho.mqtt.client*. Depois disso, o programa escreve os dados da mensagem recebida no banco de dados MongoDB, através da biblioteca *MongoClient*. Esses dados são armazenados em uma coleção específica definida pelo campo “Coleção” da própria mensagem, servindo como identificação. Esse programa de persistência de dados é executado em segundo plano no servidor.

4) Integração com a plataforma robótica

Os microcontroladores foram posicionados na parte traseira da célula robótica, a fim de liberar o espaço da frente para a garra realizar as coletas dos produtos. O robô utiliza sensores para se orientar no ambiente do estoque, bem como para localizar cada produto armazenado no estoque. Os sensores ultrassônicos, posicionados na parte frontal e traseira, fornecem informações sobre a distância percorrida pelo robô, auxiliando na identificação dos pontos de coleta de cada item. Além disso, dos lados direito e esquerdo do robô, há um sensor de barreira que pode interromper o movimento de locomoção quando for acionado, indicando a proximidade de um obstáculo ou até mesmo uma pessoa durante o processo de coleta do produto.

A plataforma robótica foi programada para realizar a percepção do ambiente através desses sensores e realizar os movimentos necessários para coletar os itens do estoque e entregar na área de despacho. Para a construção da programação do microcontrolador ESP32 as bibliotecas *Wire*, *WiFi*, *PubSubClient* e *ArduinoJson* foram utilizadas. As bibliotecas *WiFi* e *PubSubClient* trabalham em conjunto para estabelecerem uma conexão com broker MQTT. A biblioteca *WiFi* foi utilizada para estabelecer uma conexão de internet com o servidor, utilizando o nome e senha da rede para que o microcontrolador pudesse realizar a conexão. A biblioteca *PubSubClient* atua na conexão com o broker, utilizando o IP da máquina onde o broker se encontra e o tópico onde as informações serão postadas para poder conectar-se. A biblioteca também inclui as funções para publicação e assinatura nos tópicos, utilizadas para

receber as mensagens e enviar ao servidor o status da célula robótica. A biblioteca *Wire* foi utilizada para realizar a conexão I2C entre os microcontroladores, designando o microcontrolador ESP32 como mestre e o microcontrolador ATmega328, com o código endereçador 0x08, como escravo do sistema. A biblioteca *ArduinoJson* foi utilizada para guardar e tratar as informações no formato JSON recebidas através do broker, disponibilizando ferramentas para a modificação, leitura e conversão das mensagens para posterior envio ao broker.

O usuário inicia o processo de solicitação de itens do estoque por meio da aplicação de gerenciamento. Ele envia informações sobre o produto, a quantidade e a posição do item no estoque, e essa mensagem é enviada para o broker, que, por sua vez, encaminha para o robô. A célula robótica recebe a solicitação e a inclui na coluna *Doing* da página Kanban na aplicação Web. A partir da informação de posição, o robô realiza a coleta do item e o transporta até a área de despacho. Quando o robô está posicionado no local correto de coleta dos produtos, são realizados os movimentos de abertura e fechamento da garra para pegar o item desejado.

O fluxo de mensagens da plataforma robótica ocorre conforme a Figura 12, onde o robô recebe as informações do pedido vindas da coleção “Pedidos”. Então, ele realiza o processo de coleta do item no estoque e envia uma mensagem para a coleção “Coletando”. Por fim, após o produto ser entregue na área de despacho, o robô envia uma mensagem para a coleção “Concluídos”.

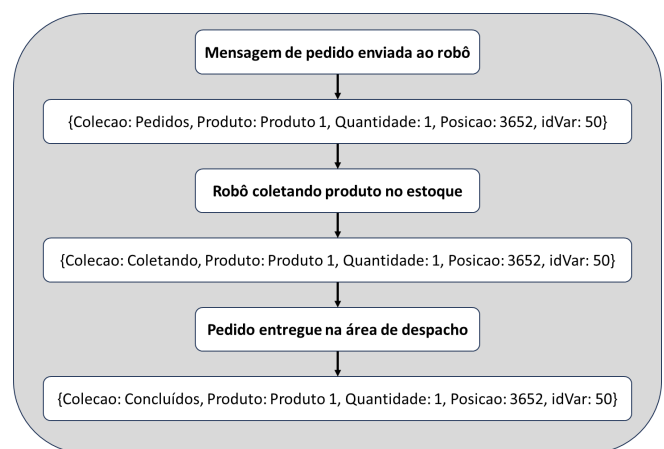


Figura 12: Fluxo de troca de mensagens durante o processo de coleta.

Assim que o banco de dados recebe a mensagem da coleção “Concluídos”, o sistema atualiza o item para da coluna *Doing* para a coluna *Done* na página Kanban.

5) Funcionamento do sistema completo

As aplicações de software e o sistema do robô se comunicam, enviando e recebendo mensagens do Broker MQTT através de inscrições ou publicações.

Na Figura 13 pode ser observado o modelo de troca de mensagens entre essas partes, tendo como componente central o broker Mosquitto. A Aplicação Web publica as mensagens com os dados em um tópico definido no sistema, a Aplicação de Persistência se inscreve para receber as

mensagens do broker nesse mesmo tópico e armazenar as informações no banco de dados. E o microcontrolador da Célula Robótica atua como *Publisher* e *Subscriber*, enviando as atualizações e recebendo as informações com os pedidos do broker.

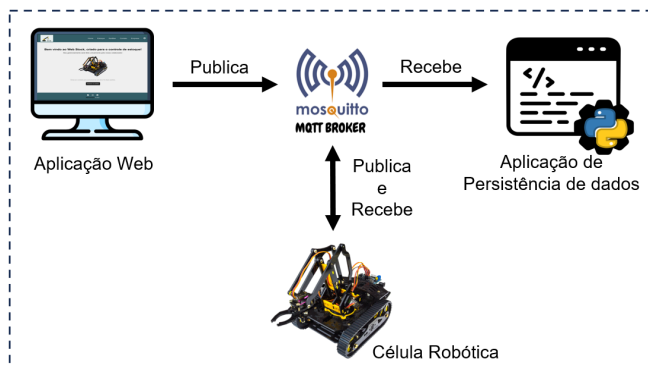


Figura 13: Modelo de troca de mensagens.

Com os produtos adicionados manualmente no estoque, o usuário tem duas possibilidades de operação dentro do sistema de gerenciamento: realizar a alteração ou exclusão destes itens, ou realizar a solicitação de pedido. Para realizar as alterações o usuário deve fazer o login no sistema, na aba de Login, e clicar no campo de atualização, onde a aba com as informações do item aparece para a alteração dessas características.

Ao acessar a aba de Estoque, o usuário pode efetuar a sua solicitação de pedido, inserindo a quantidade desejada no campo “Quantidade” e, em seguida, clicando no botão “Submeter”. Assim que a solicitação for feita, o pedido é imediatamente encaminhado para a coluna *To do* na página Kanban e também é publicado no Broker MQTT. O microcontrolador ESP32 do robô recebe a mensagem com os dados e inicia o processamento do pedido assim que a informação chega.

Quando o microcontrolador ESP32 recebe a mensagem de pedido, o item é automaticamente movido da coluna *To do* para a *Doing* na página Kanban. Enquanto o robô realiza a coleta do pedido, o usuário tem a possibilidade de efetuar outras solicitações, criando uma fila de pedidos para o robô coletar. Após a coleta do produto, o robô retorna à posição inicial e envia uma mensagem de “Concluído” para o broker. O programa de persistência recebe essa mensagem e insere na coleção do banco de dados denominada “Concluídos”, efetuando a transição do pedido da coluna *Doing* para *Done*.

Durante o processo de coleta dos itens dentro do estoque, o microcontrolador ATmega328 realiza as leituras dos sensores e envia os dados via I2C para o microcontrolador ESP32, que utiliza essa informação para se locomover até o local onde o produto está localizado. A posição do produto é fixa e pré determinada no estoque, garantindo que um item esteja sempre no mesmo local.

IV. RESULTADOS E DISCUSSÕES

Para a validação do sistema desenvolvido foram realizados testes de coleta de dois produtos, sendo solicitado ao robô a coleta do Produto 1 e Produto 2 em sequência. Os testes foram realizados em uma superfície plana e lisa, utilizando dois blocos de madeira com dimensões de 60x35x15mm e peso aproximado de 0,017kg (17g) representando os produtos no estoque. Os itens foram posicionados em diferentes alturas, 6cm e 14cm em relação ao solo. Os produtos foram posicionados a uma distância de 40cm do posicionamento inicial do robô na base, conforme Figura 14.



Figura 14: Testes de coleta.

Com os testes realizados pode-se observar os resultados referentes às seguintes partes do projeto: Aplicação Web, Banco de Dados e Plataforma Robótica.

A aplicação Web trouxe para o usuário ferramentas importantes para realizar um gerenciamento e controle de estoque, com a implementação de funcionalidades com os métodos Duas Gavetas e KanBan de forma discreta e intuitiva. A visualização das colunas da aba KanBan atualizam automaticamente conforme o fluxo do processo, trazendo uma informação precisa sobre o status atual da célula robótica no estoque. O método Duas Gavetas trouxe uma boa visualização dos níveis de estoque ao usuário, dando destaque nas quantidades dos produtos com a mudança das colorações. A divisão da aplicação Web em abas possibilitou uma navegação fluida dos itens no site.

O banco de dados não relacional atendeu as necessidades da solução possibilitando a criação de coleções específicas para cada item e etapa do método KanBan.

A precisão de coleta do robô a nível de protótipo atendeu às necessidades de coleta dos produtos em estoque, porém, após realizar a solicitação das peças ao robô por diversas vezes percebeu-se que um acúmulo de erro de posição vai sendo gerado. O erro de posição ocorre devido a necessidade de rotação de 180° para voltar até a área de despacho, onde ele realiza essa rotação se baseando no tempo de acionamento dos motores, podendo então variar a distância conforme os níveis das baterias. A variação de posição pode ser controlada com a utilização de um encoder rotativo que mede a rotação dos motores.

Os sensores da célula robótica obtiveram respostas rápidas para a tomada de decisão do robô e leitura constante das informações. Porém, as medições desses sensores podem ter um delay de transmissão de dados de alguns milissegundos, devido à troca de dados entre os microcontroladores via I2C. Contudo, o atraso das informações não comprometeram a solução.

O braço robótico RodoARM possui uma capacidade de coleta de até 80g, quatro vezes maior do que os itens utilizados nos testes, o que possibilitou que a garra coletasse os itens de maneira precisa e sem deixá-los cair. Os blocos de madeira utilizados possuem um revestimento liso em sua área externa, o que impossibilita o aumento de peso utilizando o mesmo tipo de material. Para conseguir carregar itens de até 80g no projeto seria necessário que eles possuíssem encaixes para a garra, não deixando que os itens escorregassem.

Esse sistema é escalável em todas as partes. A plataforma Web pode ter as suas funcionalidades ampliadas com a criação de novas páginas de produtos e descrição dos itens do estoque, como também com a implementação de novos métodos de gerenciamento, baseados em teorias da Engenharia de Produção e Administração. Já o banco de dados pode ser expandido com a criação de novas coleções, inserção de novos itens, devido a sua estrutura do tipo não relacional e dados no formato JSON. Diferente de bancos de dados relacionais, do tipo SQL, que não permitem uma fácil expansão. E também o volume de coleta de produtos no estoque pode ser significativamente aumentado com o uso

de mais células robóticas dentro do estoque, conectadas ao sistema e à aplicação Web.

A escalabilidade do sistema se dá pela modificação do código, que foi desenvolvido de forma modular e com componentes independentes, podendo ser replicado para outras aplicações de gerenciamento.

V. CONSIDERAÇÕES FINAIS

O presente trabalho apresentou um sistema de gerenciamento e controle de estoque, capaz de coletar produtos dentro de um estoque e trazê-los até uma área de despacho, disponibilizando as informações ao usuário em diversas abas ao longo do site.

Foi possível realizar a coleta de itens selecionados ao longo de uma área delimitada ao robô e concluir o processo de coleta, informando ao usuário na aba Kanban que o processo requisitado foi finalizado.

O sistema mostrou-se eficaz no gerenciamento e controle dos produtos no estoque, conseguindo criar filas de pedidos para que as informações fiquem em espera para o robô. A utilização de duas baterias para a alimentação do robô se mostrou suficiente para a aplicação dos testes, visto que para um número elevado de diversos pedidos será necessário fazer recarga das mesmas.

Com isso, tem-se como sugestões de trabalhos futuros: implementação das aplicações em um servidor dedicado, local ou em nuvem; disponibilização da aplicação Web de forma remota e online; uso de outros métodos de gerenciamento de estoques; melhorias da precisão da célula robótica ou a substituição da mesma por outra mais robusta; implementação de novas funcionalidades para o robô, como armazenamento de itens para dentro do estoque; uso de baterias com maior capacidade, bem como um sensoramento do nível que gere informações para recarga programada.

VI. REFERÊNCIAS

- [1] AIRES, Clayton Silva França; ALMEIDA, Gabrielly de Jesus; SILVEIRA, Sidioney Onézio. **INTELIGÊNCIA ARTIFICIAL NA GESTÃO DE ESTOQUE**. Fatec Guarulhos, Guarulhos, p. 1-7, mar. 2022.
- [2] PISCHING, André Marcos. **Arquitetura para descoberta de equipamentos em processos de manufatura com foco na Indústria 4.0**. São Paulo 2018. Acesso em: 15 de Março de 2023.
- [3] DIAS, Marco Aurélio e. **Administração de Materiais**. 5. ed. São Paulo: Atlas S.A., 2010. 537 p.
- [4] FREITAS, Juliane Aline de; GOMES, Marco Antônio. **SISTEMA DE ARMAZENAGEM AUTOMATIZADO**. 2015. 43 f. TCC (Graduação) - Curso de Tecnólogo., Universidade Tecnológica Federal do Paraná, Curitiba, 2015.
- [5] GARCIA, Eduardo Saggioro; REIS, Leticia Mattos Tavares Valente dos; MACHADO, Leonardo Rodrigues; FERREIRA, Virgílio José Martins. **Gestão de Estoques**. Rio de Janeiro: E-Papers, 2006. 141 p. Disponível em: <https://books.google.com.br/books?id=AvfRM51NlCQC&lpq=PA3&dq=gestão%20de%20estoque&lr&hl=pt->

BR&pg=PA2#v=onepage&q&f=false. Acesso em: 05 abr. 2022.

- [6] SILVA, Karen Milena da. **THE IMPORTANCE OF THE KANBAN SYSTEM FOR THE MANAGEMENT AND STOCK CONTROL OF A COMPANY.** Disponível em: file:///C:/Users/leogh/Desktop/administrador,+55-590-Arquivo+do+artigo-2470-1-15-20190604.pdf. Acesso em: 19 out. 2023.
- [7] MAGRANI, Eduardo. **A Internet das Coisas: Direito** Rio Editora, 2018. 192 p. Disponível em: <https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/23898/A%20internet%20das%20coisas.pdf?sequence=1&isAllowed=y>. Acesso em: 16 de maio de 2022.
- [8] WOLF, W. **Computers as Components Principles of Embedded Computing System Design.** 2. ed. Burlington: Elsevier, 2008. v. 2.