



Definições para o Projeto Final

1. Identificação do Problema

O crescimento acelerado de e-commerces impôs novos desafios logísticos aos centros de distribuição como a necessidade latente por agilidade, precisão e escalabilidade. Nessa lógica, empresas como Amazon, Ocado e Walmart já demonstraram que a automação logística – com uso de robôs móveis, braços robóticos, esteiras e algoritmos inteligentes – é a chave para suprir essa demanda crescente e complexa.

Porém, sistemas automatizados ainda são inacessíveis para pequenas operações ou projetos experimentais. Além disso, há lacunas práticas, como a alocação de caixas em depósitos dinâmicos, reorganização frequente por proximidade de destino, e compatibilidade entre inteligência de planejamento (ex: PDDL) e robôs físicos.

Então, visando uma POC (Proof of Concept) iremos tratar da automação completa e otimizada de um centro de distribuição, considerando os problemas principais de alocar caixas com destinos semelhantes de forma próxima e realizar isso de maneira automatizada por robô.

2. Requisitos do Projeto

- RF: Requisitos funcional
- RNF: Requisito não funcional

Código	Descrição
RF01	O braço robótico deve ser capaz de pegar uma caixa em uma posição de origem e depositá-la em uma posição de destino.
RF02	O sistema deve receber, via comunicação com backend, o local onde a caixa deve ser posicionada.
RF03	O sistema deve identificar a cor da caixa usando um sensor e associá-la a uma rota de armazenagem.
RF04	O firmware deve interpretar comandos de movimentação (ângulo dos servos) vindos do plano de execução.
RF05	O braço deve ser capaz de posicionar a caixa com precisão em prateleiras de uma maquete de estantes.
RF06	O sistema deve atualizar visualmente a operação executada.
RNF7	O tempo de resposta do sistema, do input de plano de organização até a execução, deve ser inferior a 10 segundos.
RNF8	O sistema embarcado deve operar com consumo compatível com fonte USB de bancada ou bateria 5V.

3. Lista Inicial de Materiais

Item	Descrição
01 Kit Braço Robótico MDF	https://www.huinfinito.com.br/chassis-plataformas/1637-kit-braco-robotico-mdf.html
02 BitDogLab	Plataforma do curso
01 Sensor de cor	Sensor presente no kit
02 Protoboard	Protoboard para o protótipo e ligação dos servos motores
02 cabo USB de alimentação	Para alimentar as bitdoglabs
01 Estrutura de estante	Impressa em 3d ou manualmente construída para representar os locais de depósito
04 servo motores	Presentes no braço robótico
06 caixas de fósforo	Representando as caixas, pintaremos com cores diferentes.

4. Arquitetura e Modelagem

4.1 Sensor/Atuador

Entrada:

- Sensor de Cor RGB: Identificação das caixas;
- Sensor de proximidade (opcional): Evitar colisões.

Atuadores:

- 3 Servo Motores: Movimentação do braço robótico em 3 eixos;
- 1 Servo Motor: Controle da garra.



4.2 Conectividade

Nesta camada, talvez usaremos duas placas BitDogLab ou somente uma, ainda precisamos testar para ver o que funciona melhor. No caso de usarmos 2 placas:

- Entre as duas placas: comunicação via **MQTT** com o broker Mosquitto. Isso para a escalabilidade, imaginando que em um galpão existirão vários MCUs e vários braços;
- Com o backend: comunicação **HTTP** com a API RESTful.

4.3 Dispositivo de Borda

O dispositivo de borda, como mencionado, será(ão) a(s) BitDogLab(s). Ela será responsável por realizar a leitura do sensor de cor e classificá-los em uma determinada cor/categoria. Então será realizado o envio dos dados juntamente com um plano, que faça a especificação dos pedidos, coordenadas dos pontos e número de caminhões disponíveis para entrega. A BitDogLab então receberá um plano de ação definido pelo planejador PDDL, de um domínio previamente definido. Em sequência, ela deve posicionar as caixas na estante conforme o plano de ação.

4.4 Camada de armazenamento

O sistema não prevê a persistência dos dados, a não pelo plano organizacional na cache da BitDogLab enquanto ela realiza a execução do plano. Contudo, uma integração a um banco de dados relacional como PostgreSQL ou MySQL poderia ser facilmente integrada através da API. Isso beneficiaria o sistema pela análise e mineração desses dados armazenados, o que poderia, inclusive, ajudar em ajustes futuros do sistema e do domínio PDDL.

4.5 Camada de abstração

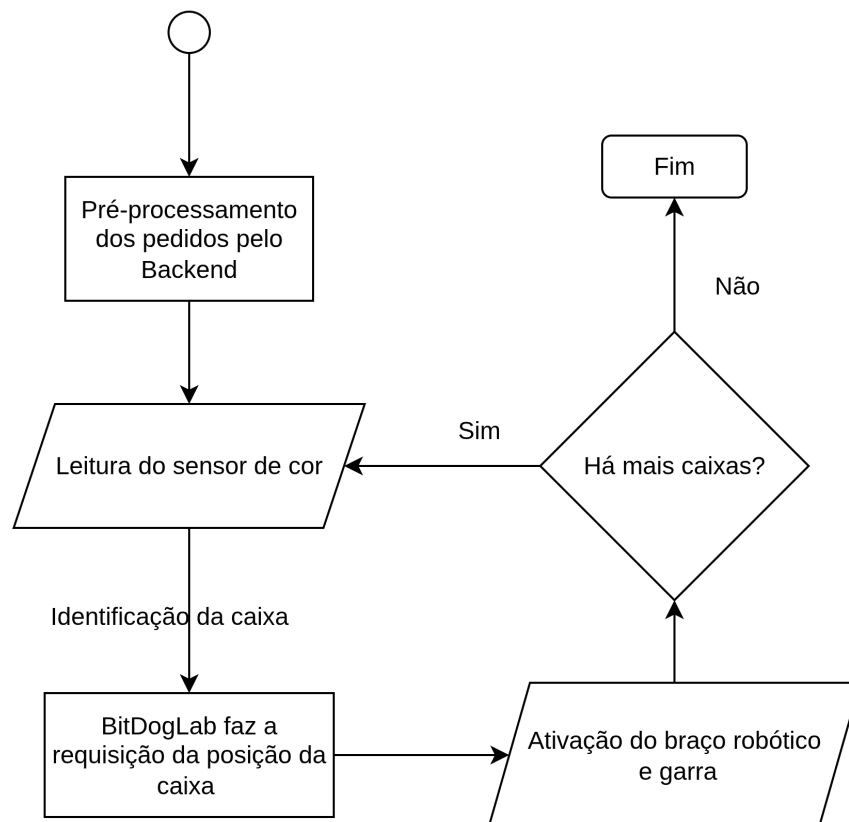
O plano organizacional, definido pelo planejador, é, por si só, uma abstração da complexidade do problema na totalidade. Além disso, será desenvolvida uma API interna

uniforme para o código da camada de apresentação e lógica de controle, escondendo detalhes de hardware e rede.

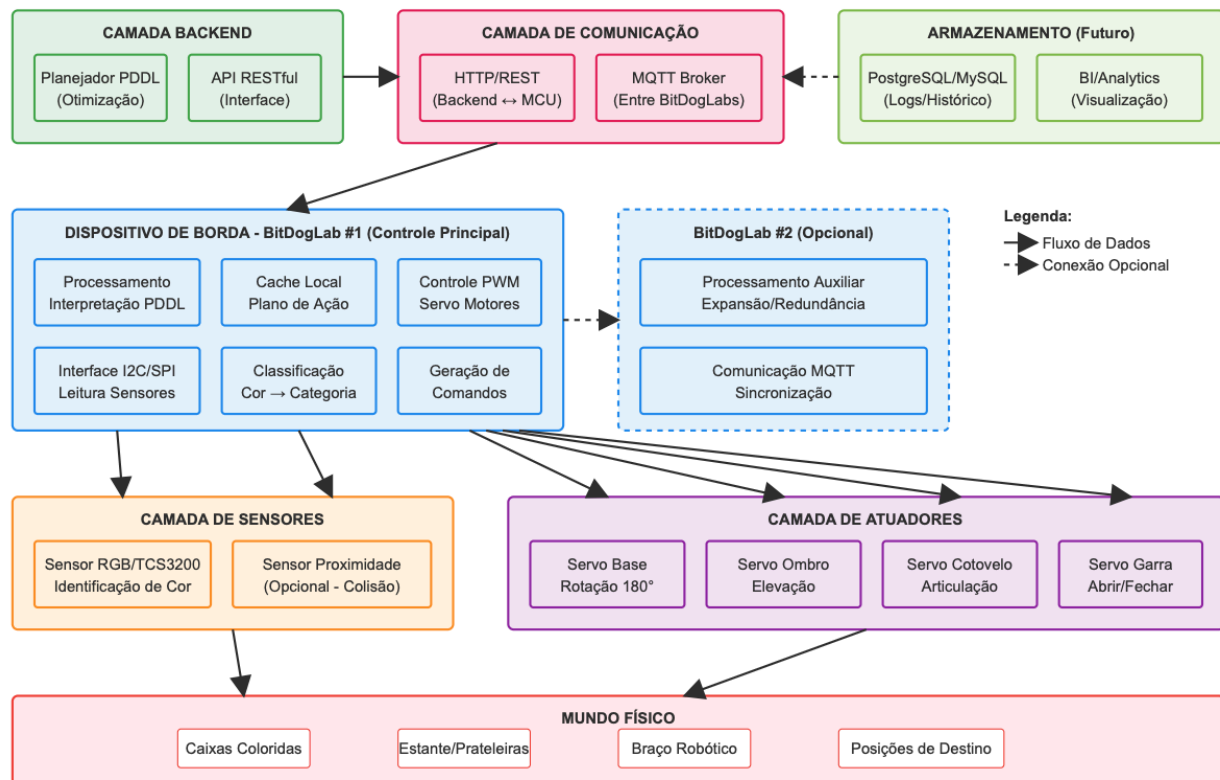
4.6 Camada de Apresentação

Com uma futura integração com um banco de dados e a análise e mineração desses dados, pode ser desenvolvido um BI que mostre a eficiência dos planos e possíveis ganhos da implementação do sistema. Dessa forma, pode se analisar de uma forma mais visual o impacto do sistema no sistema logístico de organização de estoque da empresa.

5. Fluxograma do Software



6. Diagrama de Blocos



7. Explicação funcional

O processo inicia-se na camada de backend, onde o planejador PDDL (Planning Domain Definition Language) analisa o conjunto de pedidos, destinos e restrições logísticas para gerar um plano otimizado de organização. Nesse sentido, este plano define precisamente onde cada caixa deve ser posicionada na estante, agrupando itens com destinos similares para maximizar a eficiência na expedição.



Uma vez gerado, o plano de organização é disponibilizado através de uma API RESTful que estabelece a ponte entre a inteligência de planejamento e o sistema embarcado. Dessa forma, a BitDogLab, funcionando como dispositivo de borda, requisita este plano via protocolo HTTP e armazena as instruções em sua memória local. Assim, o sistema pode operar de forma autônoma mesmo em caso de intermitência na comunicação com o backend.

Nessa lógica, a BitDogLab então entra em modo de operação, iniciando um ciclo contínuo de identificação e movimentação. O sensor RGB TCS3200, conectado via interface I2C, realiza varreduras constantes da área de entrada onde as caixas são depositadas e quando uma caixa é detectada, o sensor captura sua assinatura cromática através da análise das componentes vermelho, verde e azul refletidas pela superfície da embalagem.

Logo após, os dados brutos do sensor são processados pelo firmware embarcado, que implementa um algoritmo de classificação por limiares adaptativos. Então, cada uma das cores identificadas é mapeada para uma categoria específica que, por sua vez, está associada a uma rota de entrega no plano previamente carregado. Por exemplo, caixas vermelhas podem representar entregas expressas para a região norte, enquanto caixas azuis indicam envios regulares para a região sul.

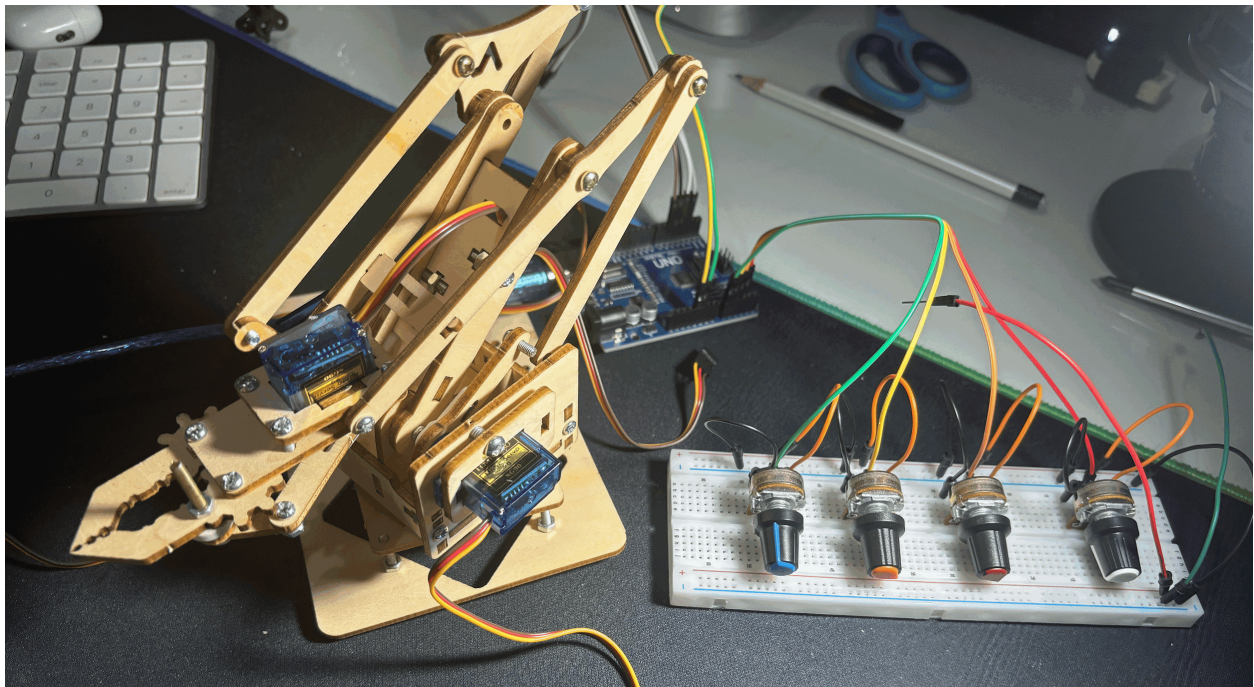
Com a categoria determinada, o sistema consulta o plano em cache para identificar a posição exata na estante onde a caixa deve ser armazenada. Dessa maneira, a decisão considera não apenas o destino final, mas também a sequência ótima de carregamento dos caminhões, minimizando o tempo de expedição.

A fase de execução física inicia com a geração de comandos de controle para os quatro servo motores do braço robótico. O firmware estaria configurado para calcular a cinemática inversa necessária para posicionar a garra na localização da caixa detectada. De forma que, primeiro, o servo da base rotacionaria para alinhar o braço com a posição da caixa e, simultaneamente, os servos do ombro e cotovelo ajustam seus ângulos para posicionar a garra

na altura correta., utilizando sinais PWM (Pulse Width Modulation) com frequência de 50Hz para controlar precisamente a posição angular de cada servo. Nessa lógica, ao alcançar a posição alvo, o servo da garra é acionado para fechar e segurar firmemente a caixa. Então, um sensor de proximidade pode ser utilizado para confirmar a captura bem-sucedida e evitar colisões durante o movimento. E logo depois, executa-se a trajetória calculada até a posição designada na estante, onde a sequência é invertida: a garra se abre, depositando a caixa, e o braço retorna à posição inicial.

Em suma, após completar uma operação de armazenamento na estante, o sistema imediatamente retorna ao modo de varredura, buscando a próxima caixa a ser processada e então o ciclo se repete. É importante dizer que cada operação completa é registrada internamente, atualizando o estado do plano de organização.

8. Protótipo inicial





Na fase inicial, o braço robótico em MDF foi montado e conectado a um Arduino para testes preliminares, utilizando potenciômetros como controle manual para mapear os limites seguros de cada servo motor para que não tenham movimentos bruscos que possam danificar as peças.

Com os parâmetros mecânicos validados, a próxima etapa consiste na substituição do Arduino pela plataforma BitDogLab, conforme especificado na arquitetura do projeto. Além disso, como próximo passo para validar a funcionalidade básica do sistema, será realizado um teste de manipulação utilizando uma caixa de fósforo como objeto de referência (simulando uma caixa de uma empresa de logística). Nessa lógica, esse teste verificará a capacidade do braço robótico de executar o ciclo completo de pick-and-place, onde objetivo será confirmar que o sistema consegue realizar a tarefa de pegar um objeto em uma posição e depositá-lo em outra, estabelecendo a base para operações mais complexas com o sensor de cor e o planejador PDDL.

Coordenação:



Financiadores:

