Relatório de Projeto
Residência Tecnológica em Sistemas Embarcados - Embarcatech

Projeto:

Robô Seguidor de Cor

Robô Seguidor de Cor com módulo de Visão Computacional inteligente

Residentes:

Luan Felipe Azzi

Vagner Sanches Vasconcelos

Vinícius Esperança Mantovani

Outubro de 2025

Histórico de Revisões

Data	Versão	Descrição
12/10/2025	1.0	Semana 1 - Conhecendo o cliente e o problema
19/10/2025	2.0	Semana 2 - Arquitetura e especificações
26/10/2025	3.0	Semana 3 - Escolha de Hardware e Planejamento para mudanças

Conteúdo

1 Objetivo do documento

Este relatório documenta de forma sistemática todas as atividades desenvolvidas no âmbito do projeto até a data atual. Informações adicionais e novos resultados serão incorporados em versões subsequentes deste documento, conforme o progresso do trabalho.

2 Conhecer o cliente e definir o problema.

Esta seção apresenta o contexto de negócio no qual se insere o projeto 'Robô seguidor de cor com módulo de visão computacional inteligente', abrangendo seu objetivo geral, a caracterização do problema, as principais tarefas envolvidas na solução proposta, bem como os desafios e oportunidades previstos ao longo do desenvolvimento.

2.1 Canvas da proposta de valor

Para compreender quem é o cliente e qual problema o projeto deverá resolver, bem como identificar o valor agregado pela solução e o ambiente em que ela será utilizada, foi utilizado a ferramenta Canvas da Proposta de Valor proposta por [1] e sintetizada na Fig. 1.

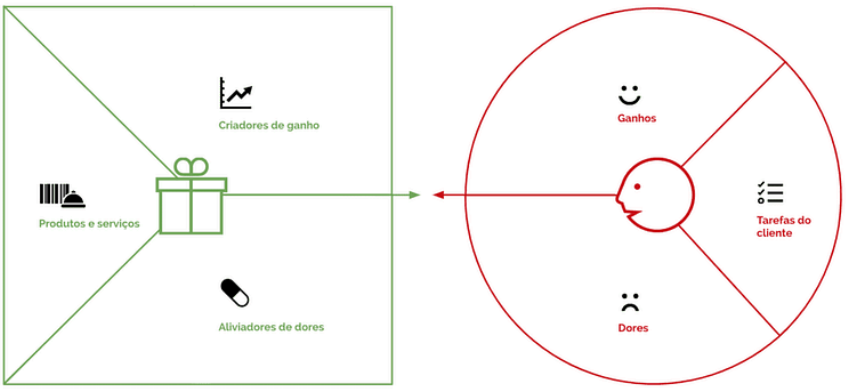

Figura 1 – Canvas da Proposta de Valor

Fonte: [1]

O **Mapa do Cliente** - lado direito do Canvas -, é utilizado para compreender profundamente o cliente. Seu propósito é identificar quem é o cliente, quais problemas o projeto deve resolver, o valor agregado pela solução e

o ambiente em que será utilizado. Já na parte da esquerda, tem-se o **Mapa de Valor**, utilizado para identificar o valor agregado pela solução e o ambiente em que ela será utilizada.

Quando ambos os mapas convergem, isso é chamado de **Encaixe da Proposta de Valor** (*Value Proposition Fit*), e isso ocorre quando a proposta de valor de uma empresa atende de forma eficaz às necessidades, dores e ganhos do cliente.

2.1.1 Mapa do Cliente

O cenário de uso ocorre em contextos educacionais.

2.1.1.1 Tarefas do Cliente

1. Tarefas Funcionais: Capacitação de estudantes; pesquisa e desenvolvimento de estratégias de controle do robô; reputação e prestígio da instituição de ensino nas competições de robótica.
2. Tarefas Sociais: Socialização; mobilidade social.
3. Tarefas Emocionais: Aprendizagem socioemocional (*soft-skills*).

2.1.1.2 Dores

Investimento inicial (CAPEX) nos robôs.

1. Custos de manutenção e reposição dos robôs.
2. Falta de força de trabalho especializada em sistemas embarcados na instituição.
3. Dificuldades com o diagnóstico de falhas nos kits.
4. Falta de padronização das provas, exigindo adaptações constantes dos robôs.
5. Falta de integração curricular.
6. *Trade-Off* entre a gestão de resultados e o processo de aprendizagem.

2.1.1.3 Ganhos

1. Pedagógicos e de aprendizagem, com a aplicação prática de teorias abstratas como: matemática; física; programação e lógica; e o estímulo ao pensamento computacional.
2. Aprendizagem baseada em projetos (PBL).
3. Desenvolvimento da cultura maker.

4. Desenvolvimento de atitudes socioemocionais (soft skills): trabalho em equipe; colaboração; resiliência; gestão da frustração; criatividade; resolução de problemas; gestão de projetos; comunicação técnica; entre outras.
5. Ganhos de marketing e visibilidade institucional.
6. Motivação e engajamento dos alunos; capacitação e valorização dos professores; fortalecimento do sentimento de pertencimento; preparação para o mercado de trabalho.

2.1.2 Mapa de Valor

O protótipo do robô seguidor de cor com módulo de visão computacional inteligente será aplicável a atividades didáticas, feiras de ciência e oficinas de robótica educacional.

2.1.2.1 Aliviadores de dores

1. Robô básico com custo similar a média de mercado.
2. Robô com elevado tempo médio entre falhas (MTBF) e peças de reposição de baixo custo.
3. Sistema de diagnóstico de falhas embarcado.
4. Módulo de visão computacional (opcional), que permite adaptações automáticas para todos os tipos de provas, e ainda a implementação de novas modalidades de controle para o robô, incluindo modelos de inteligência artificial.

2.1.2.2 Criadores de ganhos

1. Robô programável, em linguagens de alto nível e Low/No-Code, permitindo a aplicação prática de: matemática; física; programação e lógica, bem como, a aplicação de IA, com o módulo opcional de visão computacional.
2. Aplicação plena da abordagem pedagógica do PBL e da cultura maker.
3. Desenvolvimento amplo de capacidades multidisciplinares de hard e soft-skill.
4. Aumento das chances de ganhos de marketing e visibilidade institucional, principalmente com o robô embarcado com visão computacional.

5. Preparação dos alunos para o mercado de trabalho e para a vida (soft-skill).

2.1.2.3 Produtos e serviços

1. Robô móvel, sem fio e com 4 rodas, capaz de seguir faixas de cores e desviar de obstáculos de forma autônoma. Em sua versão básica, as habilidades de seguir a faixa e desviar de obstáculos ocorrem por meio de sensores de cor e distância (TOF/Ultrassônico).
2. Módulo opcional de visão computacional, que amplia as habilidades do robô, habilitando-o para competições em trilhas mais desafiadoras. Este módulo é acoplado facilmente a versão básica do robô, e também permite a incorporação de novas estratégias de controle, inclusive com técnicas de Inteligência Artificial.

2.2 Contexto de negócio

O mercado de robótica educacional no Brasil vive um momento de expansão acelerada, impulsionado principalmente pela implementação da Base Nacional Comum Curricular (BNCC), que incorporou o pensamento computacional como competência obrigatória, e ainda pela crescente conscientização sobre a importância das habilidades STEM (Ciência, Tecnologia, Engenharia e Matemática). Este marco regulatório abriu um mercado potencial de milhões de estudantes nas redes pública e privada.

Conforme [2], esse mercado terá um crescimento exponencial, chegando a US$135 milhões até 2035, o que representa um crescimento de 300% em relação a 2023.

2.3 Objetivo geral

O objetivo geral do projeto é elevar o nível de maturidade tecnológica (TRL) de um robô móvel que, além de se locomover, será capaz de seguir diferentes cores e desviar de obstáculos no trajeto. Também é escopo um módulo de Visão Computacional inteligente que, acoplado ao robô, ampliará suas habilidades e permitirá ainda a incorporação de estratégias de controle com base em IA ao robô.

2.4 Contexto do sistema

A seguir, é apresentado um diagrama de contexto do sistema. Por meio dele, é possível identificar as interações do robô com os fatores externos que atuam sobre seu funcionamento. Nesse diagrama, pode-se notar a relação do robô com obstáculos, que o fazem parar e aguardar até sua retirada, a relação com uma faixa de cor específica, a qual ele deve seguir e, a relação com o usuário, que determina o destino do robô (determinando, por consequência a cor a ser seguida) e recebe confirmações e outras informações do robô.

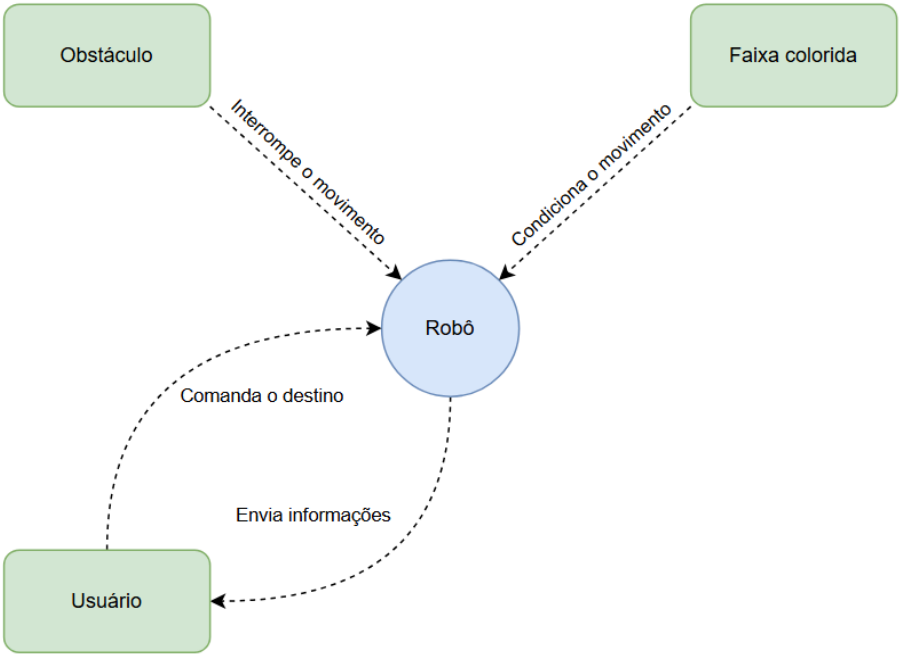

Figura 2 – Diagrama de Contexto do Sistema

Adicionalmente, tem-se um segundo diagrama, explicando as relações do Robô com fatores externos no caso de uso do módulo de visão computacional opcional. Nesse caso, o robô recebe comandos de controle do módulo, como forma de aumentar a precisão de seus movimentos majoritariamente na função de seguir cor.

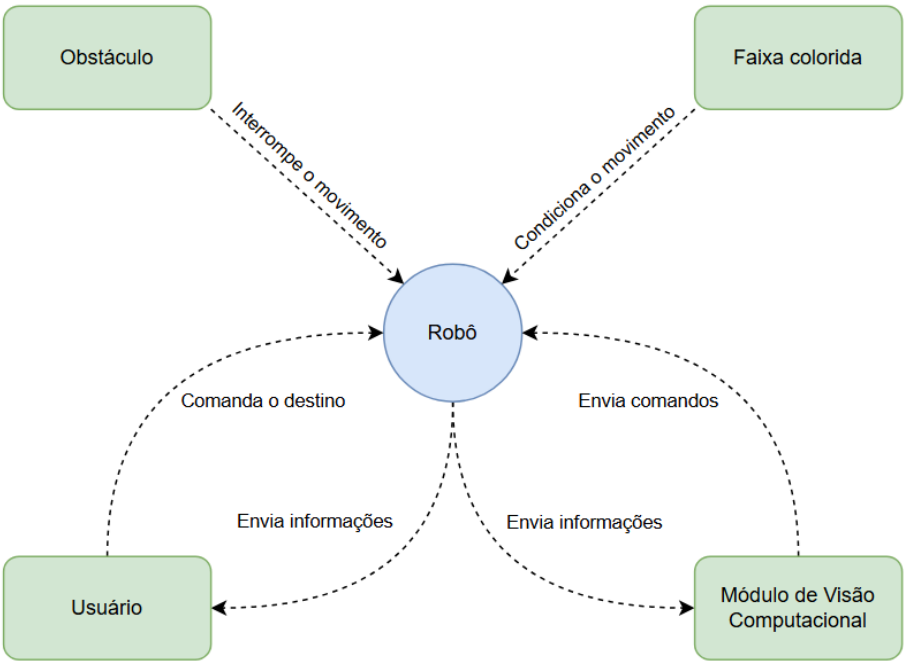

Figura 3 – Diagrama de Contexto do Sistema com Módulo de Visão Computacional

2.5 Requisitos Iniciais

A seguir, encontra-se a tabela de requisitos iniciais a ser seguida para o desenvolvimento do projeto.

ID	Tipo	Descrição breve	Prior	Critério de Aceitação
FR - 01	Func	Movimentar-se	1	O robô é capaz de se mover para frente e para trás em linha reta, fazer curvas e girar para ambos os lados
FR - 02	Func	Seguir linha	2	O robô é capaz de seguir uma linha
FR - 03	Func	Distinguir linhas de diferentes cores	3	O robô distingue a cor de diferentes linhas e segue aquela de cor correta
FR - 04	Func	Interromper o movimento na presença de obstáculo	2	O robô interrompe seu movimento caso exista um obstáculo em seu caminho, até que esse seja retirado
FR - 05	Func	Aceitar comandos de destino remotos	4	O robô recebe e interpreta comandos remotos indicando a cor a ser seguida e a segue
FR - 06	Func	Buscar a linha	4	Caso se perca da linha que

		perdida		seguia, o robô procura reencontrá-la. Caso a encontre, segue no primeiro sentido que encontrar. Caso contrário, fica parado
FR - 07	Func	Executar comandos do módulo de visão computacional	5	Caso um módulo de visão seja detectado, o robô se torna um receptor de comandos, que atua sob demandas da Micro Processing Unity (MPU) controladora do sistema de visão
FR-08	Func	Otimização de Energia	1	Otimização do consumo de energia para operação contínua do robô
FR-09	Func	Adaptação do chassi existente	5	Caso necessário, adaptar e chassi de 4 rodas existente para melhorar a performance do robô
FR-10	Func	Módulo de visão computacional	1	Sistema opcional à versão básica do robô, capaz de ampliar as suas capacidades originais de seguir faixas e desviar de obstáculos, permitindo adaptações automáticas para todos os tipos de provas, e ainda a implementação de novas modalidades de controle para o robô. Deve ser implementado em outro microcontrolador (MPU) e interagir com a BitDogLab para que esta realize o controle dos movimentos do robô. Pode utilizar visão computacional clássica e/ou moderna (redes neurais).

				Deve possuir dimensões físicas que não impactem substancialmente a performance do chassi atual do robô.
NFR - 01	Não Func	Movimentar-se	1	O robô usa seus atuadores para movimentar suas rodas de modo a garantir que ele possa seguir em linha reta, fazer curvas e girar no próprio eixo
NFR - 02	Não Func	Detectar uma linha e seguí-la	2	O robô usa seus sensores de cor para detectar a existência da linha de cor diferente da cor de fundo (branca branca por padrão) e segue por ela, verificando periódicamente para que direção deve seguir para acompanhá-la
NFR - 03	Não Func	Escolher qual cor deve seguir	3	O robô verifica a presença de linha de cor específica por meio de seus sensores de cor e, caso a encontre, segue em sua direção.
NFR - 04	Não Func	Interromper o movimento no caso de detectar uma superfície a sua frente	2	O robô detecta um obstáculo utilizando um sensor ultrassônico que o envia informações e distância periódicamente. No momento em que a distância for menor que um threshold determinado, o robô para de se mover até detectar um aumento dessa distância
NFR - 05	Não Func	Receber e atender a comandos de destino remotos (via bluetooth, wifi ou outro meio de comunicação)	4	O robô recebe um comando que o indica a cor a ser seguida, processa esse comando e busca por essa cor entre os dados atuais de seus sensores de cor. Caso a encontre, envia uma confirmação para o aparelho controlador e segue na direção da linha de cor comandada
NFR - 06	Não Func	Buscar pela linha perdida girando em seu eixo	4	Ao identificar que em nenhum de seus sensores a cor que seguia é detectada, o robô gira em seu próprio eixo até encontrá-la em um de seus sensores e, caso a

				encontre, segue no sentido em que a encontrou. Caso contrário, mantém-se parado até intervenção externa
NFR - 07	Não Func	Executar comandos do módulo de visão computacional	5	Quando em conexão com um módulo de visão computacional, o robô atua como um receptor de comandos e condiciona seu movimento ao controle da MPU controladora. Podendo ignorar os demais sensores que possui ou tratá-los em conjunto com as informações advindas da MPU
NFR - 08	Não Func	Custo	1	Robô básico com custo similar a média de mercado e peças de reposição de baixo custo
NFR - 09	Não Func	Robustez	1	Robô com elevado tempo médio entre falhas (MTBF)

Tabela 1 – Requisitos Funcionais e Não Funcionais do robô
Fonte:

2.6 Avaliação do TRL atual e avanços esperados

Atualmente, já escolhemos os componentes e conhecemos os requisitos iniciais, tendo também a plataforma base pronta (chassi de 4 rodas radiocontrolado), conforme figura 4. No entanto, a integração dos sensores de distância e cor e da câmera para realizar uma navegação autônoma existe apenas conceitualmente, sem uma prova da viabilidade prática. Por isso, avalia-se o **TRL** atual como **2 - Conceito da tecnologia formulado**.

Espera-se que, ao fim do período estipulado para realização do projeto, o **TRL** seja **4 - Validação de componente e arranjo em ambiente de laboratório**, posto que temos como principal objetivo ter um protótipo funcional que execute bem as tarefas de seguir cor e desviar de obstáculos em um circuito de testes em salas de aula e feiras de ciência. Para isso, o primeiro passo é ter uma **Prova de Conceito (TRL 3)**, integrando os sensores ao chassi e desenvolvendo um código inicial que prove que é possível ler os dados de cor e distância e, a partir disso, controlar o carrinho.

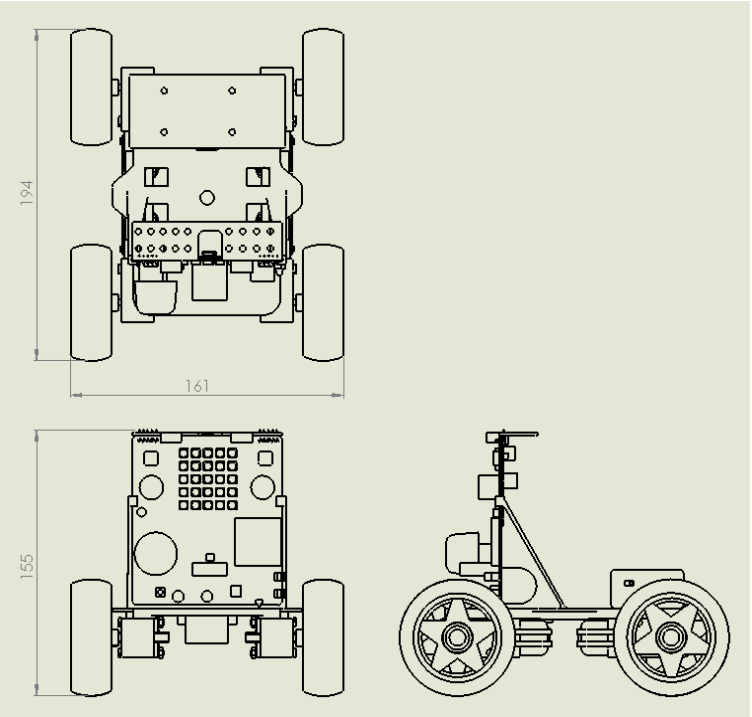

Figura 4 – Robô Móvel Skid-Steer fornecido.

Fonte: Produzido pelos autores.

Finalmente, para alcançar o **TRL 4**, deve-se refinar o protótipo construído. Para tanto, é preciso desenvolver uma estrutura final e realizar testes em diversos cenários para garantir que sua confiabilidade e desempenho estão de acordo com o que foi definido nos requisitos. Assim, teremos um produto mínimo viável com garantia de funcionamento atestada em laboratório.

Todos os desenhos para impressão 3D do chassis também foram fornecidos, por meio de um diretório compartilhado, bem como a versão atual do firmware do robô, na linguagem MicroPython, abaixo tem-se a lista de materiais utilizados no robô fornecido:

- 01 Placa BitDogLab v7;
- 04 motores tt dc 3-6V com redutor e roda de 68mm;
- 02 motores traseiros (master) com rabichos JST-SH 2P e 4 jumpers fêmea soldados;
- 02 motores dianteiros (slave) com 4 jumpers macho soldados;
- 01 chassis impresso em ABS v4;
- 02 mãos-francesas impressas em ABS v3;
- 04 parafusos M3x25;
- 04 porcas M3;
- 01 módulo ponteH tbn6612fng (módulo em PCB BitDogLab);

- 01 cabo flat IDC 02x07;
- 01 suporte para 2 baterias 18650;
- 02 baterias Li-Ion 3,7V 18650;
- 01 conector XT30 macho;
- 01 módulo bluetooth HC05;
- fita dupla-face, para colar as mãos francesas no chassis.

Para controlar o robô pelo celular, foram utilizados os aplicativos:

- Arduino Bluetooth Controller (por Giristudio);
- BT Car Controller-Arduino/ESP (por Giristudio).

3 Arquitetura e especificações técnicas (Etapa 1 - Semana 2)

Esta seção apresenta as especificações técnicas do robô em função das necessidades levantadas, bem como a estruturação da arquitetura do sistema, garantindo clareza e modularidade, e ainda relaciona os requisitos às camadas do modelo de sistema embarcado, envolvendo: sensores/atuadores; conectividade; processamento local; armazenamento; abstração; e interface.

3.1 Requisitos associados às camadas do sistema

A tabela 2 apresenta o refinamento dos requisitos levantados na semana 1 associados às camadas do sistema.

ID	Tipo	Descrição breve	Prior	Critério de Aceitação	Camada
FR - 01	Func	Movimentar-se	1	O robô é capaz de se mover para frente e para trás em linha reta, fazer curvas e girar para ambos os lados	Atuador
FR - 02	Func	Seguir linha	2	O robô é capaz de seguir uma linha	Processamento local/Sistema de Controle
FR - 03	Func	Distinguir linhas de diferentes cores	3	O robô distingue a cor de diferentes linhas e segue aquela de cor correta	Sensor de cor
FR - 04	Func	Interromper o movimento na	2	O robô interrompe seu movimento caso exista um	Sensor de Colisão

		presença de obstáculo		obstáculo em seu caminho, até que esse seja retirado	
FR - 05	Func	Aceitar comandos de destino remotos	4	O robô recebe e interpreta comandos remotos indicando a cor a ser seguida e a segue	Conectividade Processamento local/Sistema de Controle
FR - 06	Func	Buscar a linha perdida	4	Caso se perca da linha que seguia, o robô procura reencontrá-la. Caso a encontre, segue no primeiro sentido que encontrar. Caso contrário, fica parado	Processamento local/Sistema de Controle
FR - 07	Func	Executar comandos do módulo de visão computacional	5	Caso um módulo de visão seja detectado, o robô se torna um receptor de comandos, que atua sob demandas da Micro Processing Unity (MPU) controladora do sistema de visão	Conectividade Processamento local/Sistema de Controle Interface
FR-08	Func	Otimização de Energia	1	Otimização do consumo de energia para operação contínua do robô	Processamento local/Sistema de Controle
FR-09	Func	Adaptação do chassi existente	5	Caso necessário, adaptar e chassi de 4 rodas existente para melhorar a performance do robô	Interface
FR-10	Func	Módulo de visão computacional	1	Sistema opcional à versão básica do robô, capaz de ampliar as suas capacidades originais de seguir faixas e desviar de obstáculos, permitindo adaptações automáticas para todos os tipos de provas, e ainda a implementação de novas modalidades de controle para o robô. Deve ser implementado em outro microcontrolador (MPU) e interagir com a BitDogLab	Conectividade Processamento local/Sistema de Controle Interface

				para que esta realize o controle dos movimentos do robô. Pode utilizar visão computacional clássica e/ou moderna (redes neurais). Deve possuir dimensões físicas que não impactem substancialmente a performance do chassi atual do robô.	
NFR - 01	Não Func	Movimentar-se	1	O robô usa seus atuadores para movimentar suas rodas de modo a garantir que ele possa seguir em linha reta, fazer curvas e girar no próprio eixo	Atuador
NFR - 02	Não Func	Detectar uma linha e seguí-la	2	O robô usa seus sensores de cor para detectar a existência da linha de cor diferente da cor de fundo (branca branca por padrão) e segue por ela, verificando periódicamente para que direção deve seguir para acompanhá-la	Sensores
NFR - 03	Não Func	Escolher qual cor deve seguir	3	O robô verifica a presença de linha de cor específica por meio de seus sensores de cor e, caso a encontre, segue em sua direção.	Sensores Processamento local/Sistema de Controle
NFR - 04	Não Func	Interromper o movimento no caso de detectar uma superfície a sua frente	2	O robô detecta um obstáculo utilizando um sensor ultrassônico que o envia informações e distância periódicamente. No momento em que a distância for menor que um threshold determinado, o robô para de se mover até detectar um aumento dessa distância	Sensores Processamento local/Sistema de Controle
NFR - 05	Não Func	Receber e atender a comandos de destino remotos (via bluetooth, wifi ou outro	4	O robô recebe um comando que o indica a cor a ser seguida, processa esse comando e busca por essa cor entre os dados atuais de seus sensores de cor. Caso a	Sensores Conectividade Processamento local/Sistema de Controle

		meio de comunicação)		encontre, envia uma confirmação para o aparelho controlador e segue na direção da linha de cor comandada	
NFR - 06	Não Func	Buscar pela linha perdida girando em seu eixo	4	Ao identificar que em nenhum de seus sensores a cor que seguia é detectada, o robô gira em seu próprio eixo até encontrá-la em um de seus sensores e, caso a encontre, segue no sentido em que a encontrou. Caso contrário, mantém-se parado até intervenção externa	Sensores Processamento local/Sistema de Controle
NFR - 07	Não Func	Executar comandos do módulo de visão computacional	5	Quando em conexão com um módulo de visão computacional, o robô atua como um receptor de comandos e condiciona seu movimento ao controle da MPU controladora. Podendo ignorar os demais sensores que possui ou tratá-los em conjunto com as informações advindas da MPU	Conectividade Processamento local/Sistema de Controle Interface
NFR - 08	Não Func	Custo	1	Robô básico com custo similar a média de mercado e peças de reposição de baixo custo	N/A
NFR - 09	Não Func	Robustez	1	Robô com elevado tempo médio entre falhas (MTBF)	N/A
NFR - 10	Não Func	Trilhas lúdicas	1	Poster com trilhas lúdicas para aplicação de atividades com o robô móvel	N/A

Tabela 2 – Requisitos associados às camadas

3.2 Diagrama de blocos

Em sequência, são apresentados os diagramas de blocos dos principais componentes do robô, os quais foram elaborados mediante a aplicação de princípios de modularidade e de padrões de projeto consagrados, visando à clareza e à eficiência da arquitetura proposta.

3.2.1 Hardware

Seguem abaixo os diagramas de blocos dos módulos do robô móvel, que funciona independente de qualquer acessório, e do sistema de visão computacional, que será um opcional, mas estenderá sobremaneira as capacidades do robô.

3.2.1.1 Robô Móvel

Na Figura 5 abaixo, tem-se um diagrama de blocos do hardware do robô a ser desenvolvido. Nesse diagrama, pode-se notar a diferença de cores entre conjuntos de blocos:

- Amarelo: Bateria, componente essencial para alimentação da MCU, da Ponte H e dos motores;
- Verde: Ponte H e Motores, blocos compostos, o primeiro pela ponte h tbn6612fng do robô e o segundo pelos motores traseiros e dianteiros. Este é o conjunto atuador, responsável por induzir a movimentação do robô;
- Roxo: Módulo Bluetooth, periférico responsável pela comunicação bluetooth com o controlador remoto (smartphone);
- Azul: Raspberry Pi Pico W, cérebro do robô, MCU que recebe dados dos sensores e envia dados aos atuadores, responsável pelo controle do robô e aquisição de informações relevantes;
- Vermelho: Sensores de Distância e Sensores de Cor, componente responsável pelo sensoriamento do robô. O primeiro bloco se refere aos sensores de distância do robô, utilizados para a detecção de obstáculos e, o segundo, refere-se aos sensores de cor do robô, utilizados para guiar o robô na direção da linha de cor desejada.

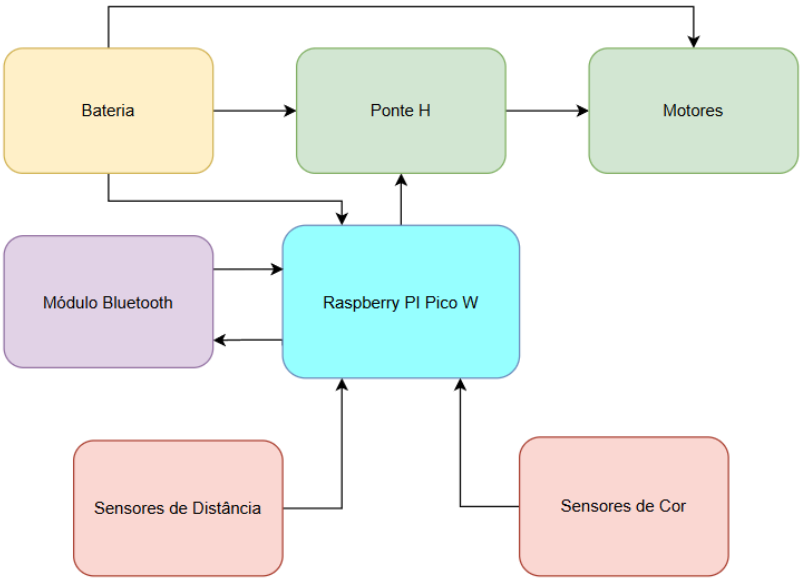

Figura 5 - Diagrama de Blocos de Hardware do Robô

Fonte: Produzido pelos Autores.

3.2.1.2 Sistema de Visão

A Figura 6 apresenta as entradas/saídas do módulo de visão computacional.

Figura 6– Módulo de Visão Computacional

Fonte: Produzido pelos Autores.

Conforme apresentado na Figura 7, existem vários *hardwares* no mercado que, em tese, poderiam atender a aplicação escopo deste projeto.

Figura 7– Tiny Image Classification Benchmark.

Fonte: [3]

Considerando parâmetros como: custo; capacidade computacional; consumo de energia; e facilidade de aquisição no mercado local, nesta 1ª etapa de projeto estão sendo considerados para este projeto o hardwares: ESP-CAM e Raspberry Pi Zero 2 W.

3.2.2 Software

Da mesma forma que para o hardware, são apresentados agora os diagramas de software do robô móvel e seu acessório, o sistema de visão computacional.

3.2.2.1 Robô Móvel

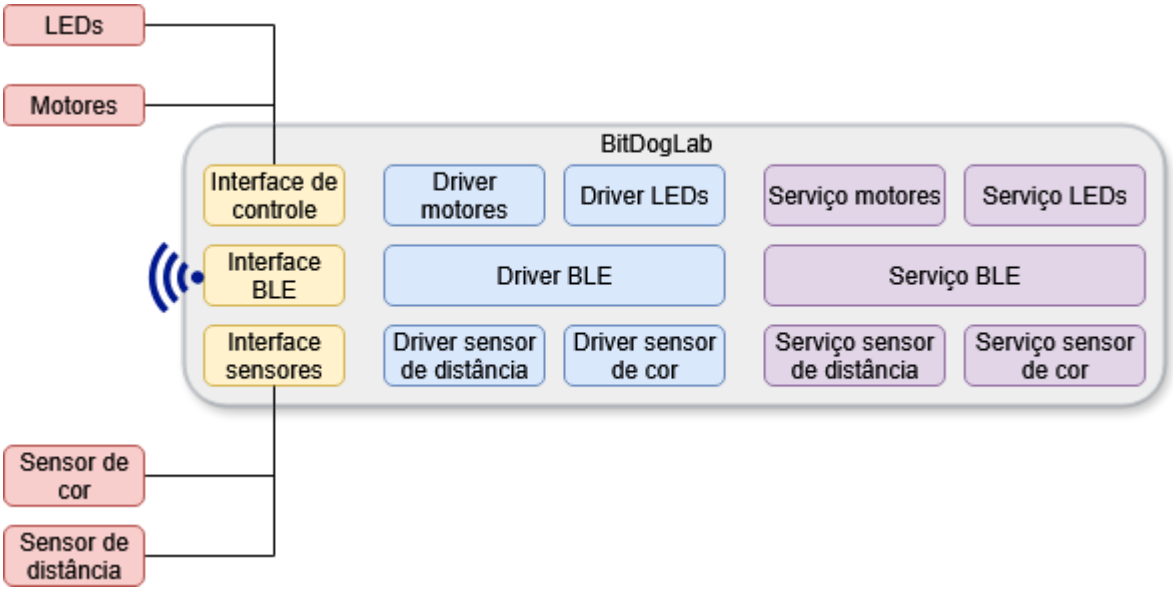

Figura 8– Diagrama de software do robô móvel.

Fonte: Produzido pelos Autores.

3.2.2.2 Sistema de Visão

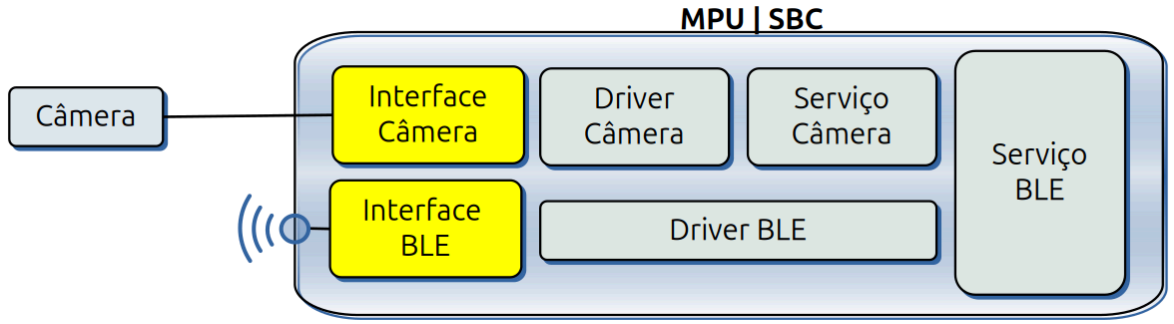

Figura 9 – Tiny Diagrama de Software do Sistema de Visão.

Fonte: Produzido pelos Autores.

3.3 Diagramas de camadas e interfaces entre módulos

3.3.0.1 Robô Móvel

A Seguir, apresenta-se o diagrama de camadas e interfaces entre os módulos do robô:

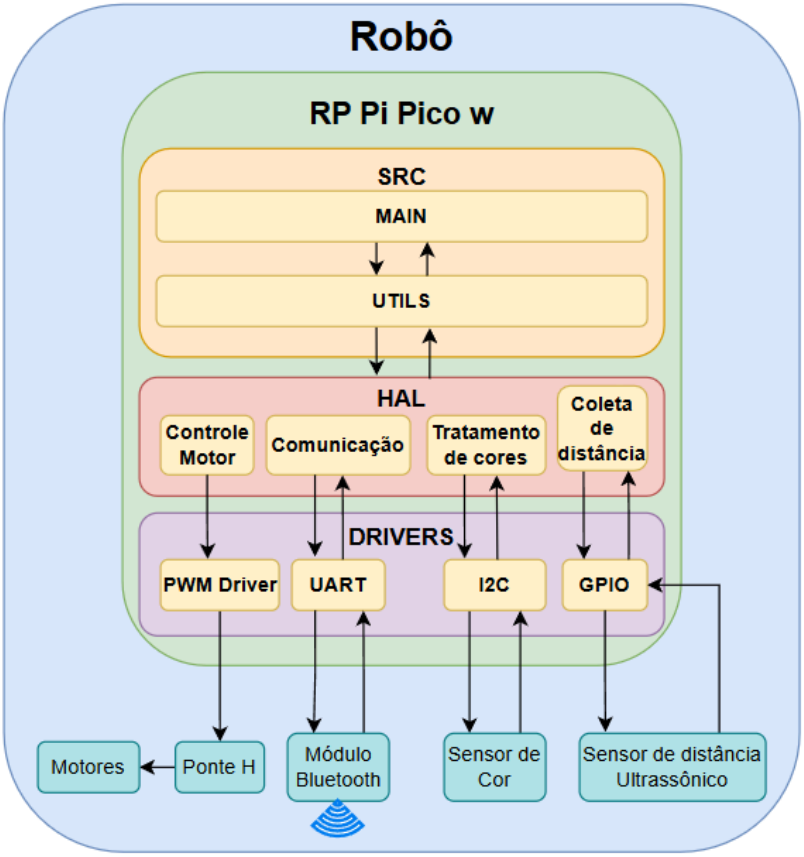

Figura 10 – Tiny Diagrama de Software do robô móvel.

3.3.0.2 Sistema de Visão

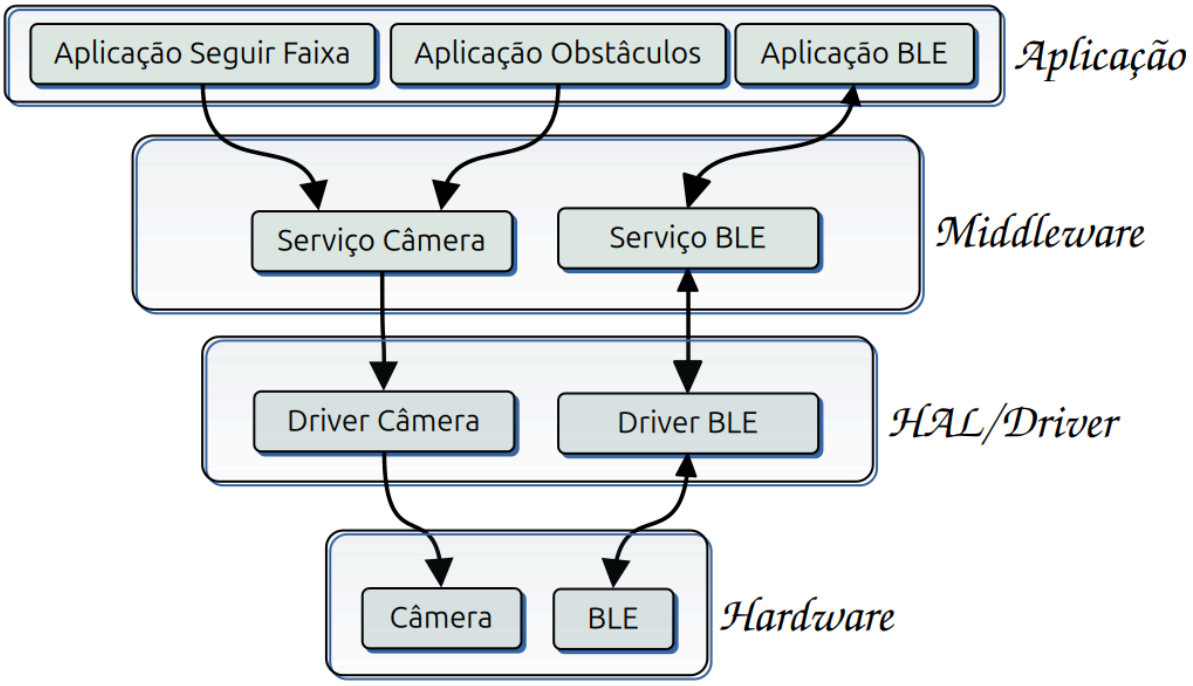

Figura 11 – Diagrama de camadas do Sistema de Visão.
Fonte: Produzido pelos Autores.

4 Escolha de Hardware e Planejamento para Mudanças (Etapa 1 - Semana 3)

Os objetivos desta etapa são:

- Selecionar o hardware e tecnologias coerentes com as especificações do sistema;
- Planejar a resiliência a mudanças de componentes ou tecnologias; e
- Documentar as justificativas técnicas das escolhas realizadas.

4.1 Matriz comparativa de hardware

Nesta seção serão comparados: microcontroladores; SBC[1]; sensores; módulos de rede; fontes de alimentação; entre outros.

[1] SBC: Single Board Computer (Computador de Placa Única).

4.1.1 Robô móvel

A tabela 4 apresenta a matriz comparativa de hardwares candidatos a atender o módulo do robô móvel, escopo deste projeto.

	Opção A	Opção B	Opção C	Critério
MCU / Placa	BitDogLab V7 (RP2040 / Pico W)	BitDogLab V6 (RP2040 / Pico W)	-	Integração com kit, disponibilidade de GPIO, custo
Sensor de cor	TCS34725	TCS3200	TCRT5000	Precisão na leitura de cor e facilidade I²C
Sensor de distância	VL53L1X (TOF)	HC-SR04	-	Precisão e imunidade a superfície/tipo de solo
Comunicação Externa	HC-05	Wi-Fi	BLE	Facilidade de uso para controle manual e integração
Ponte H	TBN6612FNG	L298N	-	Corrente, eficiência, montagem
Motores	4× TT 3–6V	-	-	Eficiência, velocidade
Alimentação	Bateria 18650 3.7V	-	-	Tensão compatível, autonomia, segurança
Facilidade de Aquisição	Fácil	Médio	Fácil	Facilidade em comprar os itens
Custo aproximado (R$)	300 a 500			-

Tabela 4 – Matriz comparativa de hardware para robô móvel.

Fonte: Produzido pelos Autores.

4.1.2 Sistema de Visão

A tabela 5 apresenta a matriz comparativa de hardwares candidatos a atender o módulo de visão computacional, escopo deste projeto.

	ESP32-CAM	XIAO ESP32S3	Nicla-Vision	Raspberry Pi Zero 2 W	Grove Vision AI V2
Processador	Dual-Core Tensilica LX6 240 MHz (ESP32-S)	Dual-Core LX7 240 MHz (ESP32-S3)	ARM Cortex-M7 480 MHz (STM32H747AII6)	SBC ARM Quad-Core 64-bit Cortex-A53 1 GHz (Pi RP3A0)	ARM Cortex-M55 (HX6538 ASIC[2] de **IA Ethus-U55**)
Bluetooth	4.2 (BLE)	5.0 (BLE)	4.2 (BLE)	4.2 (BLE)	Não Possui
Tempo Classificação [3]	687 ms	142 ms	86 ms	11 ms	6 ms
FPS[4]	1.5	7.0	11.6	91.0	167
Potência [W]	300 mW	525 mW	600 mW	1500 mW	420 mW
Tensão de Alimentação	5 V e 3.3 V	5 V e 3.7 V	5 V e 3.7 V	5 V	5 V e 3.3 V
Tensão dos GPIO (Operação)	3.3 V	3.3 V	3.3 V	3.3 V	3.3 V
Custo [R$]	90,00	240,00	1.250,00	400,00	350,00
Facilidade de Aquisição	Muito Fácil	Difícil	Fácil	Muito Fácil	Difícil

Tabela 5 – Matriz comparativa de hardware para módulo de visão.

Fonte: Produzido pelos Autores.

Considerando as características, os dois hardwares candidatos a atender o módulo de visão computacional são: **ESP32-CAM** e **Raspberry Pi Zero 2 W**.

4.2 Justificativas das escolhas

Nesta seção, todas as escolhas serão justificadas de forma técnica e econômica.

[2] ASIC: Circuito integrado de aplicação específica, neste caso, aplicações de IA.
[3] Tempo de Classificação: Tempo que o hardware demora para fazer a inferência em um modelo de classificação de imagens.
[4] FPS: Quadros por Segundo, número de imagens que uma câmera captura e exibe a cada segundo.

4.2.1 Robô móvel

Microcontrolador: A escolha é justificada pela necessidade do uso da BitDogLab no projeto. As características, como o processador dual-core, e principalmente, o suporte oficial a MicroPython, alinham-se perfeitamente com os objetivos do projeto.

Sensor de Distância: Para a prova de conceito e validação em laboratório, o HC-SR04 é a melhor escolha, dado que seu custo extremamente baixo cumpre o requisito de manter o robô básico com um custo similar à média de mercado. Tecnicamente, seu alcance e precisão são suficientes para a detecção de obstáculos em ambientes controlados. Já o VL53L1X é robusto para detecção de obstáculos em superfícies diversas, podendo ser usado como fallback.

Sensor de Cor (TCS34725): Apesar de um custo superior ao TCS3200, o TCS34725 é a escolha técnica superior, dado que sua interface I2C economiza pinos GPIO no Pico W e simplifica o hardware. A leitura direta de valores RGB, juntamente com o filtro de IR, oferece maior precisão na distinção de cores sob diferentes condições de iluminação, o que é fundamental para o requisito FR-03. Em último caso, há o TCRT5000 que é apenas um sensor infravermelho para linhas pretas e brancas, sendo muito confiável e utilizado em seguidores de linha.

4.2.2 Sistema de Visão

De todas as alternativas apresentadas na tabela 4, os dois hardwares candidatos a atender o módulo de visão computacional são: ESP32-CAM e Raspberry Pi Zero 2 W.

Exceto pela dificuldade de aquisição no mercado local, o **Grove Vision AI V2** seria a melhor opção para este caso de uso, uma vez que, por um bom preço, consegue entregar uma excepcional capacidade computacional, consumindo relativamente pouca potência elétrica; contudo, esse hardware não possui bluetooth, que é um requisito para o módulo de visão. Importante destacar aqui que esse hardware não é um microcontrolador ou uma SBC, ele é um módulo acelerador de IA, também conhecido como co-processador de visão. Sua principal função é ser um "periférico inteligente" que deve se conectar a outra placa (host), como a BitDogLab por exemplo, dando a ela "superpoderes" de visão computacional, com consumo de energia baixíssimo.

A **Nicla-Vision** tem um preço alto pelo desempenho que entrega.

O **XIAO ESP32S3**, assim como o **Grove Vision AI V2,** apresenta dificuldades de aquisição no mercado local.

Desta forma, as melhores alternativas de hardware são: **ESP32-CAM** e **Raspberry Pi Zero 2 W.**

A **ESP32-CAM** é a opção mais barata, contudo, possui baixíssima poder computacional, comparado às outras alternativas, assim, um ponto de atenção para utilizá-la, em projetos de visão computacional, será a escolha de bibliotecas/frameworks que requerem menos capacidade computacional.

A **Raspberry Pi Zero 2 W** apresenta boa capacidade computacional, permitindo a utilização de bibliotecas/frameworks mais sofisticados de visão computacional, contudo, um ponto de atenção é sua tensão de alimentação de 5V. Assim, para utilizá-lo, será necessário elevar a tensão das baterias e regulá-la, o que pode ser realizado com módulos do tipo: Módulo Carregador Duplo USB 5V 2A para Bateria 18650 - Modelo: JX-887Y.

4.3 Plano de abstração de hardware (HAL)

Os módulos robô móvel (Unidade de Controle do Robô) e visão computacional (Unidade de Processamento de Visão) formam um sistema distribuído, composto por duas unidades de processamento distintas:

- UCR (Robô Móvel): Na qual a BitDogLab (Pi Pico W), controlará o "baixo nível" do robô: controle dos motores; sensores (cores, obstáculos); gestão de energia; interface com usuário; entre outras. Tudo isso independentemente do módulo opcional de visão computacional, conforme Figura 10.
- UPV (Visão Computacional): Na qual um sistema microcontrolado (ESP32-CAM) ou um computador de placa única (Pi Zero 2 W) expandirá as capacidades da UCR como um processador de aplicação de alta performance, responsável pela visão computacional, capaz de identificar várias cores de faixas e obstáculos, enviando essas informações para o UCR executar o controle do sistema, conforme Figura 11.

Ambas as unidades (UCR e UPV) se comunicam por meio de BLE[5]. Além disso, o robô móvel pode ainda ser controlado por um aplicativo (app), também por meio de BLE.

[5] BLE: Bluetooth Low Energy, tecnologia de comunicação sem fio projetada para transferir dados de forma eficiente com baixo consumo de energia.

Sendo assim, o plano HAL define interfaces para ambas as unidades de processamento e o protocolo que as une, bem como para o app. Contudo, **o desenvolvimento do app não é escopo deste projeto.**

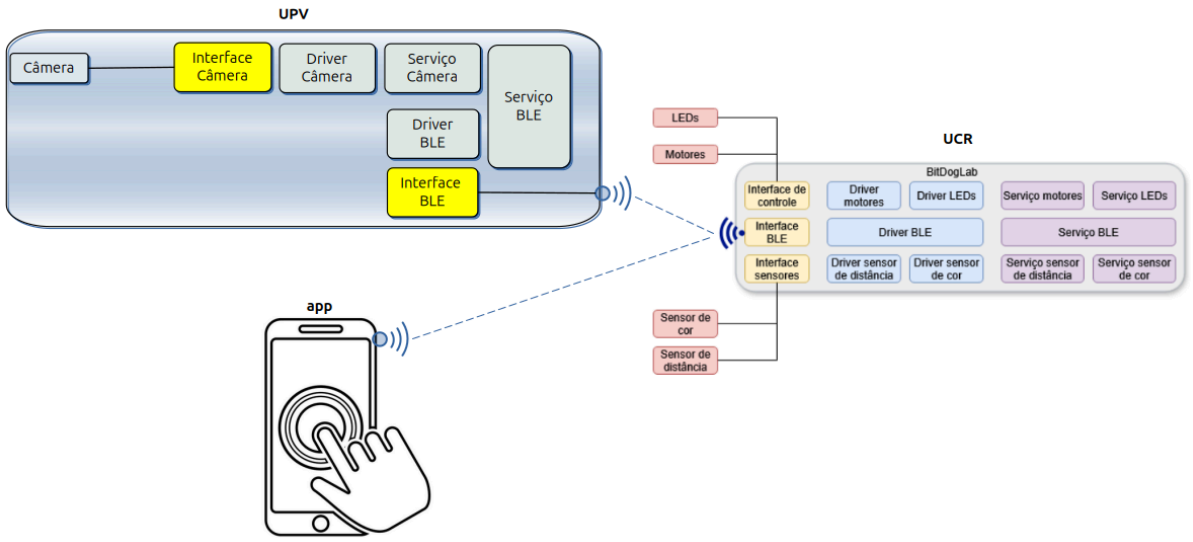

Figura 12 – Diagrama de camadas do Sistema de Visão.

Fonte: Produzido pelos Autores.

4.3.1 Visão geral do sistema

O sistema será desacoplado, sendo:

- UCR (Pi Pico W): Executa a Figura 10, controlando motores, lendo sensores de proximidade/cor e obedecendo a comandos;
- UPV (Pi Zero e/ou ESP32-CAM): Executa o Figura 11, capturando imagens, executa a lógica de IA ("Seguir Faixa", "Obstáculos") e enviando comandos para a UCR executá-los;
- APP: Comando manual do robô por meio de aplicativo.

4.3.2 Definição do Protocolo de Comunicação (IPC[6])

A comunicação entre a UCR e a UPV é a espinha dorsal deste design, e será implementada por meio do uso de Bluetooth/UART[7].

O módulo comunicação, conforme a Figura 10, na UCR será o hal_ipc.

Já o protocolo será baseado em pacotes (CMD/Response)[8]:
[START_BYTE (0xFE)] [LENGTH] [COMMAND_ID] [PAYLOAD...] [CHECKSUM].

4.3.2.1 Plano de HAL - Parte A: Robô Móvel (UCR)

Este HAL será implementado em C/C++ (Pico SDK):

- **hal_motor.h**: Abstrai a "Ponte H" e o "PWM Driver".
- **hal_tof.h**: Abstrai o "Sensor Ultrassônico" e a "Coleta de Distância" (GPIO).
- **hal_surface.h:** Abstrai o "Sensor de Cor" (I2C).
- **hal_ipc.h**: Abstrai o driver "UART" e implementa o lado escravo do protocolo IPC.

Lógica de Implementação: O **main.c** da UCR irá registrar um callback[9]. Quando um **CMD_SET_MOTOR** chegar, o callback será disparado, e ele chamará **hal_motor_set_speed().**

---

[6] IPC: Inter Process Communication, é o conjunto de mecanismos e técnicas que permite a programas e processos se comunicarem e trocarem dados entre si em um sistema microcontrolado.

[7] Bluetooth/UART: UART é a tradução livre de Receptor-Transmissor Assíncrono Universal, que é um circuito ou protocolo de hardware que permite a comunicação serial (bit a bit) entre dois dispositivos. Já o Bluetooth fornece a camada de conexão sem fio.

[8] Pacotes (CMD/Response): É um protocolo baseado em pacotes com o formato "comando/resposta" (CMD/Response) é um modelo de comunicação comum usado em redes e sistemas de hardware. Ele funciona com um sistema de requisição e resposta: um cliente envia um pacote contendo um comando (CMD), e o servidor responde com um pacote que contém a resposta (Response).

[9] Callback: pedaço de código executável que é passado como parâmetro para algum método, é esperado que o método execute o código do argumento em algum momento.

4.3.2.2 Plano de HAL - Parte B: Visão Computacional (UPV)

Este HAL é o mais crítico para a portabilidade. O "Serviço Câmera" e as "Aplicações", Figura 11, serão escritos em C++ ou Python e consumirão está HAL. A implementação dos arquivos .c/.py da HAL será diferente para o Pi Zero e para o ESP32-CAM, mas os cabeçalhos (.h) serão idênticos.

- **hal_camera.h:** Abstrai a captura de imagem.
  Implementação (Pi Zero): Usará libcamera (C++) ou picamera2 (Python).
  Implementação (ESP32-CAM): Usará esp_camera (C).
- **hal_ble.h:** Abstrai a comunicação externa (Figura. 11).
  Implementação (Pi Zero): Usará BlueZ (via D-Bus ou bibliotecas Python).
  Implementação (ESP32-CAM): Usará NimBLE ou Bluedroid (ESP-IDF).
- **hal_rcu_link.h**: Abstrai a comunicação com a UCR. Ele implementa o lado mestre do protocolo IPC sobre UART.
  Implementação (Pi Zero): Usará a biblioteca serial (Python) ou termios (C) para controlar o /dev/ttyS0.
  Implementação (ESP32-CAM): Usará os drivers UART (ESP-IDF).

4.4 Análise de Riscos

Na tabela 6 serão analisados os riscos envolvidos na substituição de componentes, com proposições de alternativas de mitigação.

Robô Móvel		
Componente	**Risco**	**Mitigação**
Sensor de Distância (HC-SR04)	O sensor pode apresentar leituras imprecisas com objetos que absorvem som ou em ângulos agudos, comprometendo a detecção de obstáculos.	Se o HC-SR04 se mostrar inadequado, ele pode ser substituído por um sensor ToF (VL53L0X), que é mais preciso.
Sensor de Cor	Variações extremas de	A camada HAL pode

(TCS34725)	iluminação no ambiente podem afetar a precisão da leitura de cor, fazendo o robô se perder.	incluir rotinas de calibração que a aplicação pode invocar no início da operação. Isso torna o sistema mais resiliente a diferentes ambientes. Caso nenhum dos sensores de cor apresentem boa confiabilidade, pode-se usar sensores infravermelhos que detectam apenas linhas pretas.
Motores e Ponte H	Desgaste mecânico ou queima da Ponte H	Os motores TT e a ponte H são componentes de baixo custo e facilmente encontrados no mercado.
Módulo de Visão		
Componente	**Risco**	**Mitigação**
Hardware (ESP32-CAM)	O baixo poder de processamento da ESP32-CAM pode levar a uma baixa taxa de quadros por segundo (FPS) e alto tempo de classificação. Isso pode fazer com que o robô reaja lentamente a curvas, obstáculos ou mudanças na faixa, comprometendo seu desempenho.	Utilizar bibliotecas e frameworks que demandam menor capacidade computacional. Priorizar modelos de IA otimizados para microcontroladores (TinyML) e algoritmos de visão clássica mais simples, focados no processamento de cores em vez de detecção complexa de objetos.
Hardware (Raspberry Pi Zero 2 W)	O Pi Zero 2 W consome significativamente mais	Incorporar um módulo regulador de tensão

	energia (1500 mW) e exige uma fonte de 5V, incompatível com a tensão nativa das baterias de 3.7V. Isso pode reduzir drasticamente a autonomia do robô e exigir componentes adicionais, aumentando o custo e a complexidade do projeto.	(step-up/boost) para elevar e estabilizar a tensão das baterias para 5V, como o modelo JX-887Y sugerido anteriormente. No software, implementar rotinas de economia de energia, como desativar periféricos não utilizados e ajustar a frequência do processador dinamicamente.
Câmera	Variações na iluminação do ambiente (sombras, reflexos), desfoque de movimento devido à alta velocidade ou um ângulo de montagem inadequado podem degradar a qualidade da imagem. Isso pode levar a detecções incorretas da faixa colorida ou de obstáculos.	Projetar um suporte para a câmera que a posicione em um ângulo ótimo e minimize reflexos. Se necessário, adicionar iluminação própria com LEDs para garantir condições de luz consistentes. Implementar no software da UPV rotinas de calibração de cor e normalização de imagem que se ajustem às condições do ambiente no início da operação.

Tabela 6 – Análise de riscos e mitigação

Fonte: Produzido pelos Autores.

4.5 Diagrama consolidado do sistema completo

A Figura 13 apresenta o diagrama de software e camadas detalhado dos módulos robô móvel e visão computacional.

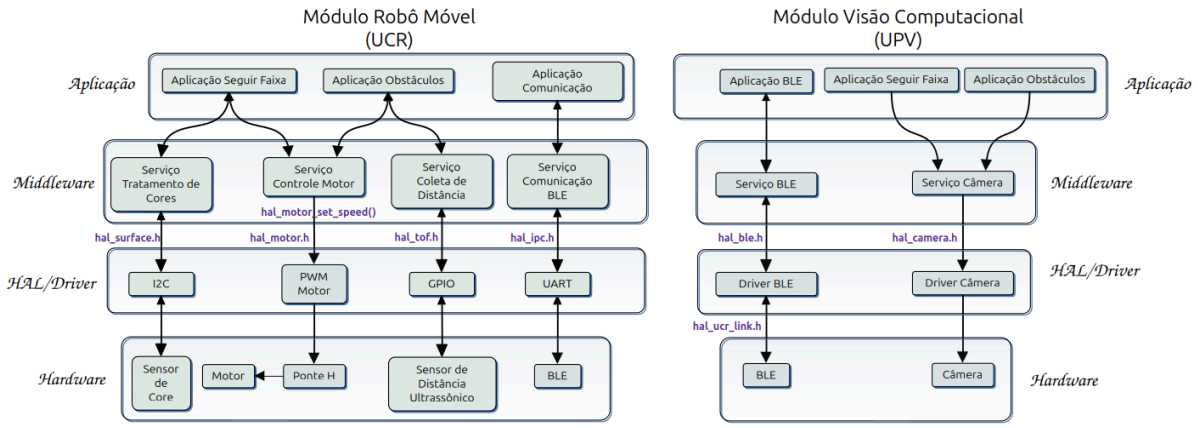

Figura 13 – Diagrama de camadas e software detalhado.

Fonte: Produzido pelos Autores.

5 Referências

[1] A. Osterwalder, G. Bernarda, Y. Pigneur "Value Proposition Design: Como construir propostas de valor inovadoras", 2019.

[2] Market Research Future (MRF), "Brazil Educational Robots Market Research Report - Forecast to 2035", 2024.

[3] M. Rovai, "Edge AI Engineering: Hands-on with the Raspberry Pi", 2025.