

Desenvolvimento de um Robô Educativo com IA Embarcada para Reconhecimento Visual

João Pavão^{1,2}

¹ Universidade dos Açores – Pós-Graduação PRIA

² Escola Básica e Integrada de Ginetes, Ponta Delgada
joao.pavao@email.com

Resumo. O presente artigo apresenta o desenvolvimento do AzorCAR, um robô educativo open source concebido no âmbito da pós-graduação PRIA – Universidade dos Açores. O sistema integra um microcontrolador Arduino, controlo de motores com driver L298P e uma câmara HuskyLens (DFRobot). Esta é uma câmara de visão computacional com diversos modos integrados, incluindo reconhecimento de cor, TAGs visuais, objetos e seguimento de linha. Foram exploradas funcionalidades como o seguimento de alvos visuais, aproximação por distância com base no tamanho do objeto detetado, e paragem condicionada por critérios espaciais e temporais, com testes práticos realizados em arenas físicas construídas especificamente para o AzoresBot 2025. O projeto foi concebido com o objetivo de promover aprendizagens ativas em contexto competitivo e educativo, aliando robótica, visão computacional e pensamento computacional. A plataforma física foi desenhada de forma modular, permitindo futuras expansões com novos sensores e integração com ROS 2. Embora esta integração ainda não tenha sido realizada, reconhece-se o seu elevado potencial, sendo assumida como uma próxima etapa evolutiva. O artigo descreve a estrutura física do robô, a lógica de controlo implementada em Arduino, os testes realizados e os resultados observados, discutindo ainda perspetivas futuras, como o uso de ambientes simulados e um comparativo entre diferentes câmaras aplicáveis à robótica educativa.

Palavras-chave: Robótica Educativa, Visão Computacional, Arduino, HuskyLens, AzoresBot, Robô Modular

1 Introdução

A robótica educativa tem vindo a afirmar-se como uma ferramenta pedagógica essencial no ensino das disciplinas STEM (Ciências, Tecnologias, Engenharia e Matemática), promovendo o pensamento computacional [1], a criatividade e a aprendizagem ativa [2]. A sua integração em contextos formais e informais tem demonstrado um impacto positivo na motivação dos alunos e no desenvolvimento de competências técnicas e transversais, alinhando-se com as abordagens construcionistas defendidas por Papert [3].

Nos Açores, o ensino da robótica enfrenta desafios específicos decorrentes da insularidade, como o acesso limitado a recursos, à formação especializada e a contextos

laboratoriais adequados. Paradoxalmente, estas limitações representam também uma oportunidade para o desenvolvimento de soluções locais, modulares e acessíveis, ajustadas à realidade das escolas e clubes educativos da região.

Entre os vetores de inovação mais relevantes na robótica educativa destaca-se a visão computacional, que permite dotar os robôs de capacidades percetivas próximas das interações humanas. No entanto, a adoção de soluções baseadas em câmara tem sido tradicionalmente condicionada por fatores como a complexidade de implementação (por exemplo, com OpenCV), a exigência computacional e as dificuldades de integração com microcontroladores simples.

Neste contexto, a câmara HuskyLens da DFRobot surge como uma alternativa pedagógica altamente eficaz: incorpora múltiplos modos de reconhecimento visual com inteligência artificial embarcada, incluindo deteção de cor, reconhecimento de objetos, TAGs visuais e seguimento de linha, permitindo a sua utilização de forma autónoma e com comunicação serial simplificada. Estas características tornam-na particularmente adequada para projetos educativos com base em Arduino ou micro:bit, mesmo em contextos com recursos limitados [4].

A realização do AzoresBot, competição regional de robótica educativa, constitui uma oportunidade privilegiada para o desenvolvimento e validação de soluções robóticas em ambiente prático e motivador. Este evento, em crescimento contínuo, promove a experimentação, a aprendizagem ativa e a partilha de boas práticas entre professores, alunos e entusiastas da robótica na região.

É neste enquadramento que se insere o projeto AzorCAR, um robô educativo modular e open source, concebido com o objetivo de explorar, em ambiente real, funcionalidades como o seguimento de alvos visuais, a aproximação com base na distância e a paragem condicionada por critérios espaciais e temporais, utilizando apenas componentes acessíveis e linguagem de programação Arduino. A estrutura física foi desenhada com foco na modularidade, permitindo futuras expansões, como a integração de sensores adicionais e a transição para uma arquitetura baseada em ROS 2. Embora essa integração ainda não tenha sido realizada, o seu desenvolvimento inicial já foi iniciado, sendo assumida como uma perspectiva estratégica de evolução para futuras edições do AzoresBot e projetos mais avançados.

2 Metodologia e Arquitetura do Sistema

2.1 Conceção e Protótipos Físicos

O desenvolvimento físico do projeto AzorCAR passou por duas versões distintas de robôs móveis concebidas pelo autor, cada uma com objetivos e características complementares no contexto educativo e experimental.

Protótipo 1. Foi construído com base num modelo 3D de autor externo, adaptado a partir do projeto publicado por Michael Klements [5]. A versão original, disponível online, inclui os ficheiros STL, códigos de controlo e tutoriais explicativos para a integração da câmara HuskyLens com Arduino. Este chassi foi originalmente desenvolvido num projeto anterior do mesmo autor, centrado num robô de desvio de obstáculos com Arduino [6]. O modelo foi ajustado para se adaptar aos materiais disponíveis

localmente e às necessidades do AzorCAR. O resultado revelou-se funcional e estável, sendo o primeiro sistema utilizado em testes práticos com a HuskyLens [4] e controlo de motores com driver L298P. A estrutura rígida e o design fechado permitiram validar a integração dos principais componentes e simular as primeiras interações com a arena física.

Protótipo 2. Concebido de raiz, introduz uma abordagem modular e escalável, inspirada nos robôs utilizados em competições como a RoboCup. A sua estrutura base consiste numa plataforma octogonal perfurada, com cantos chanfrados e um padrão de furos regulares, permitindo a fixação flexível de componentes eletrónicos, sensores e elementos mecânicos. Esta geometria oferece versatilidade e expansibilidade, facilitando a montagem de níveis adicionais e a adaptação a diferentes contextos. O robô foi projetado com a integração futura de ROS 2 em mente, prevendo espaço interno e pontos de ancoragem para sistemas mais avançados. Assume-se como a versão principal do projeto e será utilizada na apresentação prática no AzoresBot 2025.

Fig. 1. Modelação do AzorCAR no Tinkercad (protótipo 2 com os vários níveis)

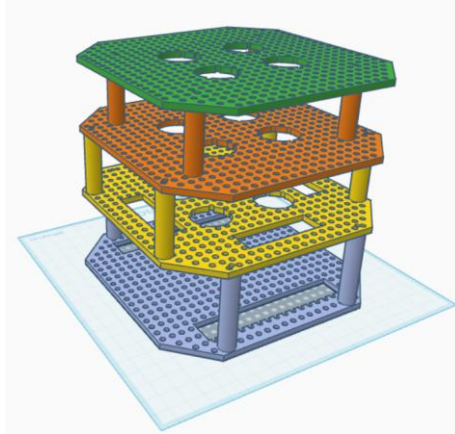
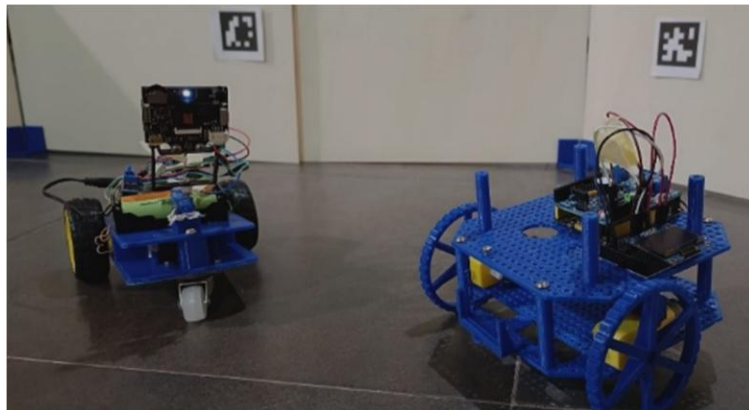
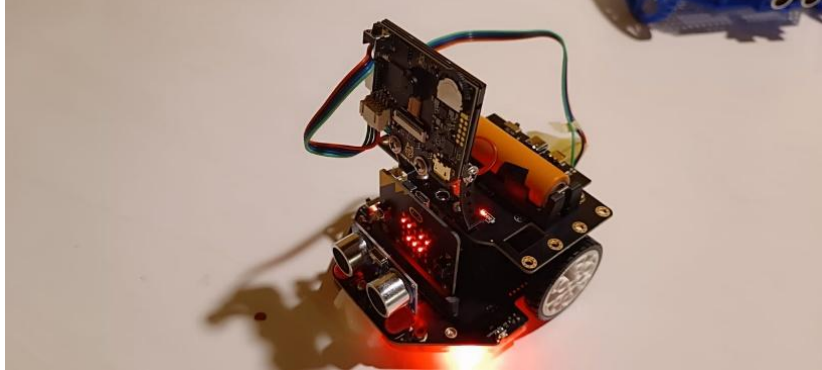


Fig. 2. Protótipos AzorCAR: à esquerda, modelo 1 com chassi 3D adaptado; à direita, modelo 2 com estrutura modular desenvolvida de raiz.



Maqueen Plus V2. Paralelamente, é relevante mencionar esta plataforma comercial da DFRobot [7], amplamente adotada no contexto da robótica educativa. É compatível com o micro:bit e com suporte nativo para a câmara HuskyLens [4], constituindo uma alternativa acessível e pronta a usar em ambientes escolares e clubes de programação.

Fig. 3. Maqueen Plus V2 com a camera Huskylens acoplada



2.2 Componentes de Hardware e Sensores

O sistema AzorCAR integra um conjunto de componentes eletrônicos amplamente utilizados em projetos de robótica educativa, selecionados com base na sua disponibilidade local no Clube de Robótica e na compatibilidade com o ambiente de desenvolvimento Arduino. A unidade de controlo principal é um microcontrolador Arduino UNO R3 ou R4 WiFi, que comunica com os motores através de um driver L298P, integrado num shield standard. Embora existam alternativas como o DRV8833 ou o tradicional L298N, optou-se pelo L298P pela sua robustez, facilidade de uso e integração simplificada com motores DC de baixa potência.

Fig. 4. O driver L298P + Arduino Uno 3, montados no AzorCAR (protótipo 2)



O sensor central do sistema é a câmara HuskyLens [4], fornecida pela DFRobot, que incorpora algoritmos de visão computacional com inteligência artificial embarcada.

Esta câmara dispõe de sete modos principais de operação, destacando-se, no contexto do projeto AzorCAR, os seguintes:

- Reconhecimento de Cores, com aprendizagem simples ou múltipla;
- Reconhecimento de TAGs Visuais, com identificação automática de etiquetas com IDs únicos (1 a 8);
- Seguidor de Linha, com capacidade de aprendizagem de trajetos e curvas.

Estes modos permitiram implementar comportamentos como o seguimento de alvos visuais, a deteção de comandos através de etiquetas visuais e a navegação assistida por linha. A comunicação com o Arduino é realizada via UART, garantindo resposta em tempo real. Outras funcionalidades da HuskyLens, como o reconhecimento de rostos, objetos ou classificação supervisionada com IA, não foram exploradas nesta fase, mas são referenciadas em anexo como potenciais extensões futuras.

Fig. 5. Detalhe da câmara HuskyLens integrada no robô AzorCAR.



Além destes elementos principais, o sistema integra vários componentes de suporte: motores DC com rodas e suportes adaptados, rodas impressas em 3D como alternativa às comerciais, servo motor opcional para varrimento horizontal da câmara, e um ecrã OLED de 1.3" com driver SH1106 e interface I2C, usado para visualização local de dados. A alimentação é assegurada por powerbanks USB e/ou packs de pilhas recarregáveis, garantindo a mobilidade do robô em ambientes de prova.

Fig. 6. Ecrã OLED presente no AzorCAR, permitindo visualização local de dados.



2.3 Construção da Arena e Materiais

A arena para o AzorCAR Challenge foi desenhada de forma quase circular e de modo a ser facilmente implementada em qualquer local. Foram projetadas duas arenas destinadas ao Desafio I e ao Desafio II.

Fig. 7. Esboço da arena para o Desafio I, com os marcadores visuais e as TAGs

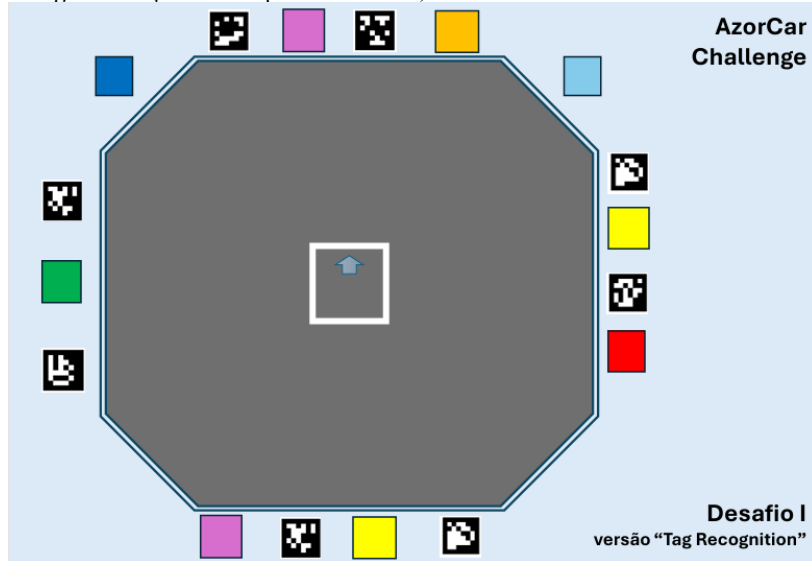
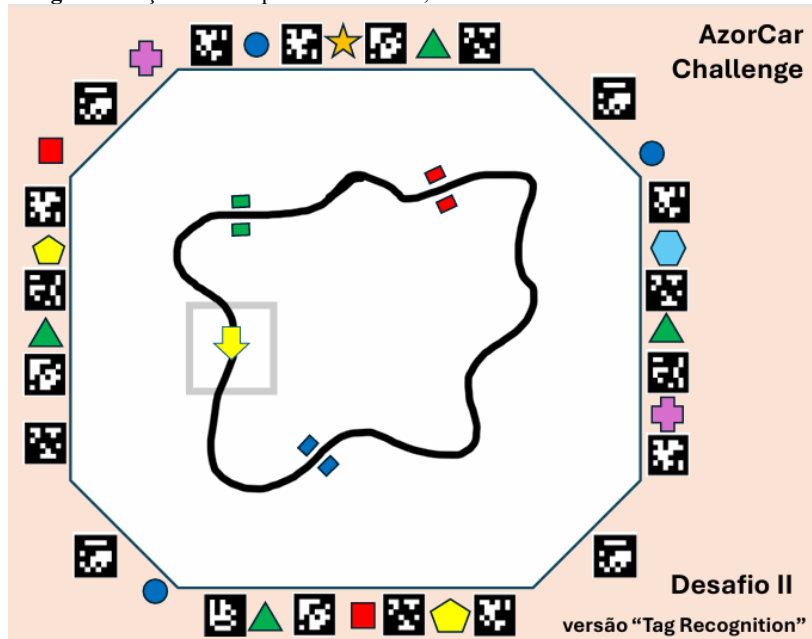


Fig. 8. Esboço da arena para o Desafio II, com os marcadores visuais e as TAGs



A arena utilizada nos testes e demonstrações do projeto AzorCAR foi construída com materiais acessíveis e adaptados à realidade local, permitindo a sua reprodução por escolas ou clubes de robótica com recursos limitados. A estrutura base é composta por placas padronizadas de contraplacado ($30 \times 42 \times 0,4 \sim 0,5$ cm), facilmente encontradas em carpintarias locais. A ligação entre placas foi realizada através de suportes impressos em 3D, desenhados com ângulos variados (90° , 135° , em T e em cruz), possibilitando a construção de formas complexas como octógonos, labirintos ou corredores.

Fig. 9. Vista da arena modular construída com placas de contraplacado e conectores 3D.



Esta abordagem modular inspira-se em metodologias aplicadas em contextos de competição como a RoboCupJunior Rescue League [10], promovendo flexibilidade na montagem e testagem de cenários variados. O sistema permite ainda a criação de pisos elevados, rampas, zonas de entrada e obstáculos móveis, compatibilizando-se com desafios de navegação autónoma e testes de algoritmos visuais.

Na sinalização visual da arena foram utilizados marcadores coloridos, com padrões aprendidos pela câmara HuskyLens, tal como TAGs visuais com IDs únicos, impressas em papel e vinil plastificado. Futuramente, serão utilizados tapetes claros com trilhos escuros para o modo de segue-linha.

A conceção teve em conta os requisitos práticos do AzoresBot e desafios semelhantes descritos em outras competições educativas, como o Vulcano Challenge [11], onde a adaptabilidade do terreno e a perceção visual desempenham papéis críticos no desempenho do robô.

Fig. 10. Protótipo 1 do AzorCAR em teste numa arena com marcadores de cor e TAGs visuais.



2.4 Organização e Desenvolvimento de Código

O desenvolvimento do sistema AzorCAR seguiu uma abordagem modular, iterativa e com forte orientação pedagógica. Todos os programas foram desenvolvidos na Arduino IDE, organizados por diretórios temáticos (motores, sensores, testes, integração), com reutilização de funções e documentação clara, facilitando a replicação e o uso educativo.

A fase inicial focou-se na validação individual dos componentes, com testes dedicados aos motores DC, à câmara HuskyLens e ao ecrã OLED SH1106. Estes ensaios permitiram confirmar a comunicação entre dispositivos, calibrar tempos e velocidades, e verificar a resposta aos estímulos visuais em tempo real.

Com base nos requisitos dos desafios do AzoresBot 2025, foram desenvolvidas rotinas de controlo autónomo, que associam a deteção visual à decisão de movimento. As estratégias programadas incluíram:

- Correção direcional com base na posição horizontal do alvo (coordenada X);
- Aproximação guiada pela dimensão da moldura detetada (largura/altura);
- Paragem automática quando dentro da zona ideal;
- Execução de varrimentos iniciais (rotação 360°) para reconhecimento global do ambiente.

Estas soluções foram implementadas de forma incremental, permitindo testes progressivos e controlo rigoroso sobre os parâmetros de entrada e saída. Esta lógica favoreceu o desenvolvimento técnico e a aprendizagem ativa em ambientes de ensino.

Adicionalmente, foram testadas funcionalidades visuais avançadas, como seguimento de linha e leitura de TAGs com ID, evidenciando a versatilidade da câmara HuskyLens e a sua integração via UART com o Arduino UNO [8].

A interface gráfica local foi assegurada por um ecrã OLED 1.3", que permitiu a visualização de dados como coordenadas, tamanho da moldura e estado do sistema. A biblioteca U8g2 foi utilizada pela sua simplicidade e compatibilidade [9].

O desenvolvimento beneficiou ainda do uso de ferramentas baseadas em inteligência artificial para apoio à programação, como assistentes de geração e depuração de código, que aceleraram a escrita de funções e permitiram otimizações estruturais relevantes. Esta abordagem reflete uma realidade crescente na prática pedagógica e de engenharia: o uso assistido de ferramentas de IA como meio de potenciar o raciocínio algorítmico, a experimentação segura e o aperfeiçoamento contínuo do código. Durante o desenvolvimento do projeto AzorCAR, foram utilizadas plataformas como ChatGPT, Google Gemini, Grok e Copilot para geração de trechos de código, depuração e sugestões estruturais, sempre com foco na compreensão, adaptação e validação prática em contexto educativo.

3 Implementação do Desafio com Comportamento Autónomo Baseado em Visão Computacional

Esta secção descreve a lógica de funcionamento, o algoritmo de controlo e a estratégia prática de implementação do comportamento autónomo desenvolvido para a participação no AzoresBot 2025. A prova em questão (Desafio I) consiste na deteção sequencial de marcadores visuais — cores aprendidas ou TAGs — colocados numa arena construída em contraplacado, exigindo que o robô execute uma aproximação controlada, paragem temporária e reinício do processo para o marcador seguinte.

Fig. 11. Marcadores Visuais a serem utilizados no AzoresBot 2025

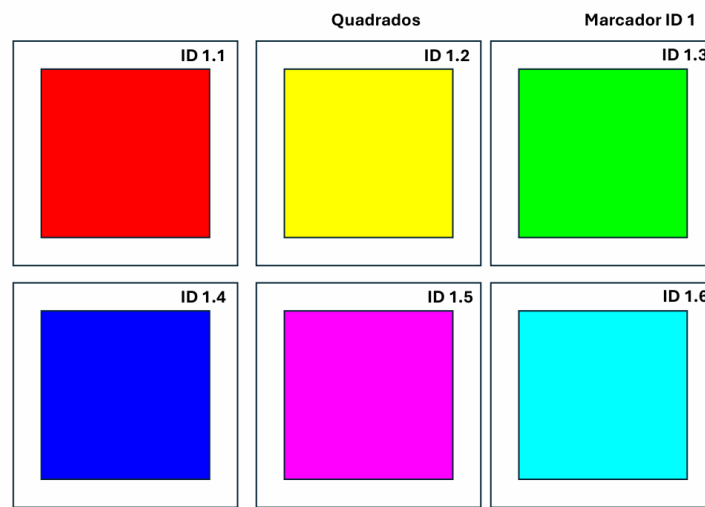


Fig. 12. Marcadores TAG a serem utilizados nos AzoresBot 2025

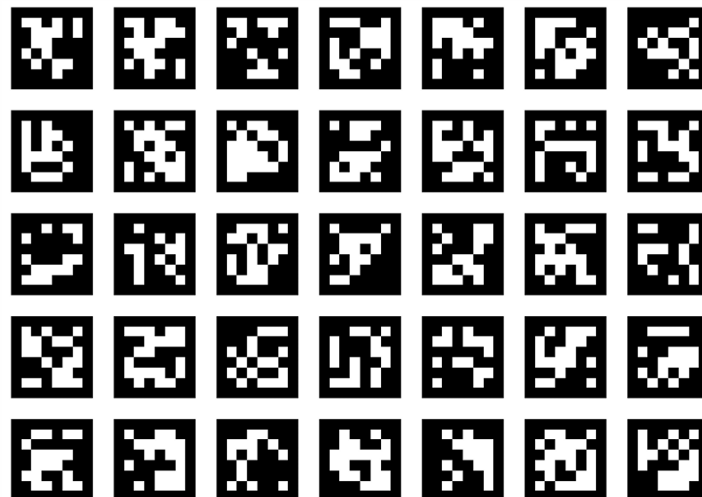


Fig. 2. Camera Huskylens a aprender uma determinada forma ou cor

3.1 Regras e Lógica da Prova do Desafio I

A lógica comportamental do sistema AzorCAR para o Desafio I segue uma sequência de ações bem definida:

- O robô inicia a partir do centro da arena e realiza uma rotação completa de 360°, simulando uma varredura inicial do ambiente;
- Após detetar o primeiro marcador visual (ex. ID1), previamente aprendido pela câmara HuskyLens, o robô aproxima-se até atingir uma distância segura entre 15 e 25 cm — estimada com base na largura (W) e altura (H) da moldura detetada;
- Quando atinge a distância ideal, o robô executa uma paragem estável de 10 segundos;
- Após este tempo, reinicia a procura do próximo marcador definido na sequência (ID2, ID3, ...).

O desempenho do sistema é avaliado com base na capacidade de:

- Detetar corretamente os IDs programados;
- Manter paragem estável durante o tempo definido;
- Evitar aproximações incorretas ou deteções erradas devido a ruído visual ou iluminação instável.

Algoritmo de Controlo

O algoritmo de controlo foi implementado na plataforma Arduino UNO R4 WiFi, e baseia-se em três componentes principais:

Análise da moldura (W/H)

A aproximação ao marcador é regulada com base nas dimensões da moldura detetada pela HuskyLens. Os valores de largura (W) e altura (H) foram previamente calibrados, associando intervalos a distâncias reais, com base em testes com régua. Se a moldura

for demasiado pequena (objeto longe), o robô avança; se for grande demais (objeto muito próximo), recua. Dentro da zona ideal (15–25 cm), o robô para automaticamente.

Correção de alinhamento horizontal (X)

O valor da coordenada X do centro da moldura é utilizado para ajustar a orientação lateral:

- Se o X estiver abaixo do limiar esquerdo, o robô roda para a direita;
- Se estiver acima do limiar direito, roda para a esquerda;
- Quando o alvo está centrado, avança ou para, conforme o caso.

Máquina de estados

O sistema opera segundo uma lógica sequencial de cinco estados:

1. Varrimento inicial (rotação de 360°);
2. Procura ativa do marcador (ID1, ID2, ...);
3. Aproximação e alinhamento com base em W/H e X;
4. Paragem de 10 segundos frente ao marcador;
5. Reinício do ciclo para o ID seguinte.

Validação Prática com TAGs Visuais

Os testes mais recentes permitiram validar com sucesso a lógica de deteção sequencial com TAGs visuais. Numa arena com cinco marcadores distintos, o robô completou uma rotação de 360°, detetou corretamente o ID1, aproximou-se de forma suave e parou durante 10 segundos. Após esse tempo, iniciou nova rotação, detetou o ID2 e repetiu o comportamento com sucesso. Estes resultados demonstram a viabilidade do controlo autónomo baseado em visão computacional, embora se reconheçam ainda oportunidades de melhoria na calibração dos motores, gestão da iluminação e integração de feedback visual via ecrã OLED.

3.2 Regras e Lógica da Prova do Desafio II

A lógica comportamental do sistema AzoCAR para o Desafio II, denominado Missão Combinada Inteligente, integra navegação autónoma, detecção visual e tomada de decisão com memória. O robô deve executar uma sequência de ações bem definida:

- Seguir uma linha preta no chão utilizando sensores de linha, com a câmara HuskyLens A configurada no modo Line Tracking.
- Detectar zonas coloridas no chão (vermelho, verde ou azul) utilizando um sensor dedicado ou a câmara HuskyLens B configurada no modo Color Recognition.
- Ao identificar uma zona colorida, parar, realizar uma rotação completa de 360° e contar os marcadores visuais verticais correspondentes à cor detectada, utilizando a HuskyLens B no modo Object Recognition ou Color Recognition.
- Exibir a contagem dos marcadores no display OLED de 1.3”.

O desempenho do sistema é avaliado com base na capacidade de:

- Seguir a linha preta com precisão, sem desvios significativos.
- Identificar corretamente as zonas coloridas no chão.
- Contar com exatidão os marcadores visuais correspondentes à cor detectada.
- Exibir a contagem no display OLED de forma clara e em tempo real.
- Gerenciar a sincronização entre as duas câmeras HuskyLens para evitar interferências.

Algoritmo de Controle

O algoritmo de controle, implementado na plataforma Arduino UNO, baseia-se em quatro componentes principais:

Seguimento de Linha

A HuskyLens A, configurada no modo Line Tracking, detecta a linha preta no chão e fornece coordenadas para manter o robô alinhado. O algoritmo ajusta a velocidade dos motores DC com base na posição da linha: se a linha estiver à esquerda, o robô gira à direita; se estiver à direita, gira à esquerda; se centrada, avança reto.

Deteção de Zonas Coloridas

A HuskyLens B, configurada no modo Color Recognition, ou um sensor de cor dedicado, identifica zonas coloridas (vermelho, verde ou azul) no chão. Ao detectar uma zona, o robô interrompe o seguimento de linha e inicia a parada. A identificação da cor é calibrada previamente para minimizar erros devido a variações de iluminação.

Contagem de Marcadores Visuais

Após a parada, o robô realiza uma rotação completa de 360°, durante a qual a HuskyLens B, configurada no modo Object Recognition ou Color Recognition, conta os marcadores visuais verticais correspondentes à cor da zona detectada. A contagem é baseada em IDs pré-aprendidos ou na correspondência de cores.

Máquina de Estados

O sistema opera segundo uma lógica sequencial de seis estados:

1. Seguimento da linha preta.
2. Deteção de uma zona colorida.
3. Parada do robô na zona detectada.
4. Rotação de 360° para escaneamento dos marcadores visuais.
5. Contagem dos marcadores correspondentes à cor detectada.
6. Exibição da contagem no display OLED e retomada do seguimento da linha.

4 Testes e Resultados

Esta secção descreve os testes realizados com o robô AzorCAR em ambiente real, evidenciando os principais comportamentos observados durante a execução da prova prática, bem como algumas limitações técnicas detetadas. Os ensaios foram efetuados numa arena construída com placas de contraplacado de dimensões padronizadas, sobre as quais foram dispostos marcadores visuais coloridos, seguindo as orientações propostas no regulamento do evento AzoresBot 2025.

Durante os testes com uma cor alvo (ID1), previamente aprendida na câmara HuskyLens, o robô demonstrou comportamentos consistentes e funcionais. A deteção foi fiável mesmo com pequenas variações de iluminação, e a aproximação ao marcador foi realizada com base no tamanho da moldura detetada (largura e altura – W/H), previamente calibrada com régua para representar distâncias reais entre 15 e 25 cm. Esta lógica é comum em sistemas embarcados de visão computacional, permitindo estimativas de profundidade simples e eficazes [5]. Ao atingir a distância desejada, o robô executava corretamente uma paragem estável de 10 segundos, mantendo a posição sem oscilações significativas. A correção horizontal com base na coordenada X do objeto revelou-se eficaz, permitindo alinhar o alvo com o centro do campo de visão. A rotação inicial de 360 graus, implementada como etapa de varrimento antes da deteção, foi igualmente executada com sucesso, sem desvios relevantes.

Adicionalmente, foram testados comportamentos complementares, como a deteção sequencial de outros marcadores (ID2 e ID3), ainda em fase experimental, bem como a ativação de elementos de feedback como LEDs e mensagens no ecrã OLED durante a execução das tarefas.

Do ponto de vista técnico, o sistema AzorCAR demonstrou robustez dentro dos limites definidos para o desafio. A câmara HuskyLens revelou bom desempenho nos modos utilizados — reconhecimento de cor, TAGs visuais e seguimento de linha —, com resposta em tempo real e comunicação estável via UART [6]. Os motores DC comportaram-se de forma previsível, sendo possível controlar os movimentos com precisão, embora fosse necessário ajustar manualmente a velocidade consoante o tipo de superfície.

Apesar dos resultados positivos, foram identificadas algumas limitações práticas. A luz ambiente interferiu pontualmente na deteção de algumas cores mais sensíveis (como o amarelo), exigindo cuidados adicionais na iluminação da arena. O ecrã OLED apresentou instabilidade visual (flickering), possivelmente devido à gestão energética ou à frequência de atualização gráfica. Além disso, a calibração dos limites de W/H revelou-se crítica, exigindo rigor na aprendizagem inicial dos objetos, dada a sua dependência das condições de luminosidade e da distância real ao marcador.

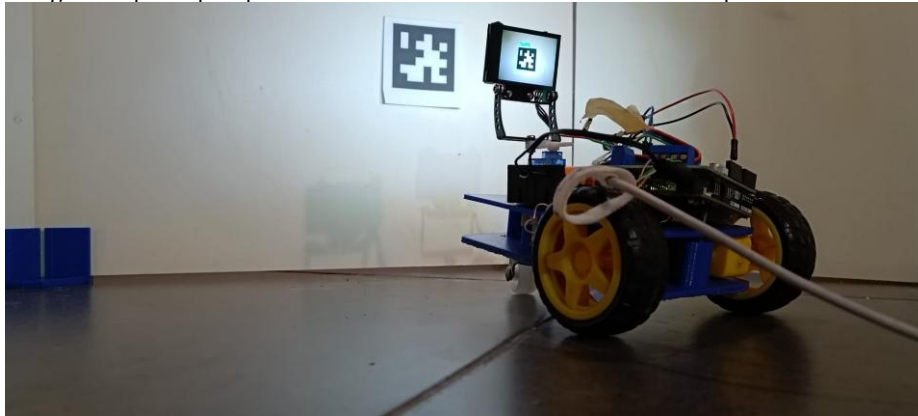
Estes testes permitiram validar o comportamento autónomo do robô e forneceram dados importantes para o ajuste fino de parâmetros, servindo de base para futuras iterações e expansão das funcionalidades previstas no projeto.

Testes com TAGs Visuais

Tendo em conta as limitações identificadas na deteção baseada em cor — nomeadamente a elevada sensibilidade à iluminação ambiente e a necessidade de calibração rigorosa dos limiares HSV (Hue, Saturation, Value) — foram realizados testes adicionais utilizando TAGs visuais (ID1, ID2) no modo Tag Recognition da câmara HuskyLens. Esta abordagem revelou-se consideravelmente mais fiável, com menor necessidade de afinação dos parâmetros de movimento, permitindo uma identificação clara e imediata dos marcadores, mesmo sob luz artificial noturna ou em ambientes não controlados [12].

Durante os ensaios, o robô executou uma rotação inicial completa de 360°, seguida de uma varrida em passos, com deteção ativa em cada pausa. Ao identificar a TAG1, foi ativada uma rotina de aproximação suave, com correções laterais baseadas na posição horizontal (coordenada X) e movimentos de avanço ou recuo em função do tamanho da moldura detetada (largura e altura). Quando a posição ideal foi atingida, o sistema permaneceu estável durante 10 segundos, conforme previsto no regulamento. Após essa pausa, iniciou-se automaticamente uma nova busca ativa até encontrar a TAG2, completando com sucesso a transição entre marcadores.

Fig. 14. O protótipo 1 parou à frente da TAG identificada como ID1 e a parar durante 10 s.



De forma geral, a utilização de TAGs visuais revelou-se claramente mais eficaz do que a deteção baseada em cor. A deteção binária e sem ambiguidades, a menor dependência de condições ambientais e a associação de cada TAG a um comportamento específico tornam esta abordagem indicada para ambientes educativos. Além de garantir um desempenho mais fiável em situações de iluminação instável ou em cenários dinâmicos, reduz significativamente o tempo de calibração e a frustração durante os testes, facilitando a aprendizagem e a compreensão dos conceitos de visão computacional e robótica por parte dos alunos.

Testes com Arduino UNO R4 WiFi e Comunicação Web

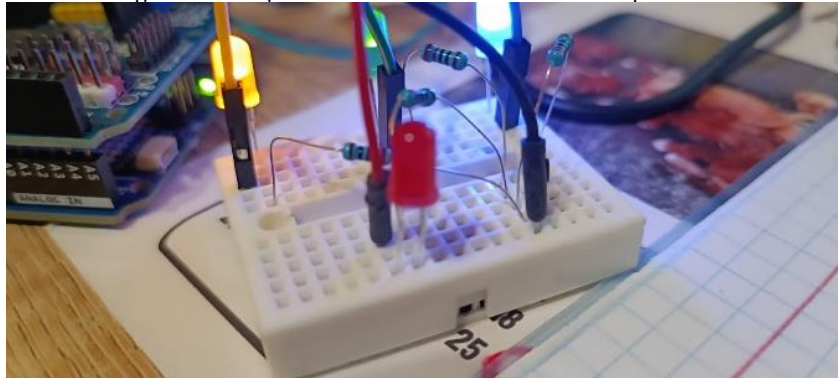
Durante o processo de desenvolvimento do AzorCAR, foram realizados testes exploratórios com o microcontrolador Arduino UNO R4 WiFi [8], de modo a avaliar o seu potencial para controlo remoto via rede local, bem como a sua integração com sensores e simulação de ações robóticas. Estes testes decorreram com sucesso e evidenciaram a versatilidade desta nova placa da Arduino, sobretudo no contexto educativo.

O primeiro código implementado consistiu num servidor Web embutido no Arduino, acessível via browser em qualquer dispositivo ligado à mesma rede Wi-Fi. A interface web permitia enviar comandos simples como start, stop e status, os quais ativavam quatro LEDs ligados aos pinos digitais do Arduino, simulando os estados do robô (ligado, parado, motor esquerdo, motor direito). Esta abordagem revelou-se eficaz para simular comportamentos e testar respostas do sistema sem depender de motores reais, o que facilitou a depuração e o ensino do controlo lógico.

Fig. 15. Interface via browser que permitiu controlar os LEDs via Wi-Fi



Fig. 16. Simulação de estados do robô com LEDs em protoboard



Num segundo momento, este sistema foi expandido para incluir a câmara Husky-Lens, configurada no modo Color Recognition. Com base nos IDs atribuídos a diferentes cores, o Arduino tomava decisões locais, como acionar LEDs distintos conforme o ID detetado (ex: virar à esquerda, à direita, parar). A interface web manteve-se funcional, permitindo alternar entre os estados do sistema, mesmo sem ligação à Internet — o que destaca a capacidade offline do UNO R4 WiFi como ponto de acesso local autónomo.

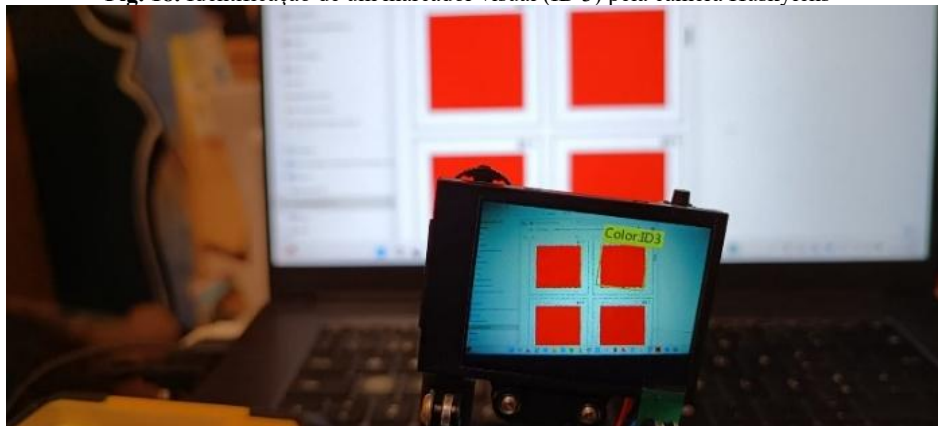
Fig. 17. Interface via browser que permitiu visualizar a identificação de um marcador, via Wi-

```

  ID Detetado: 3
  Parar
  ID Detetado: 3
  Parar
  ID Detetado: 3
  Parar
  ID Detetado: 3
  Parar
  Pedido recebido: GET /stop HTTP/1.1
  Pedido recebido:

```

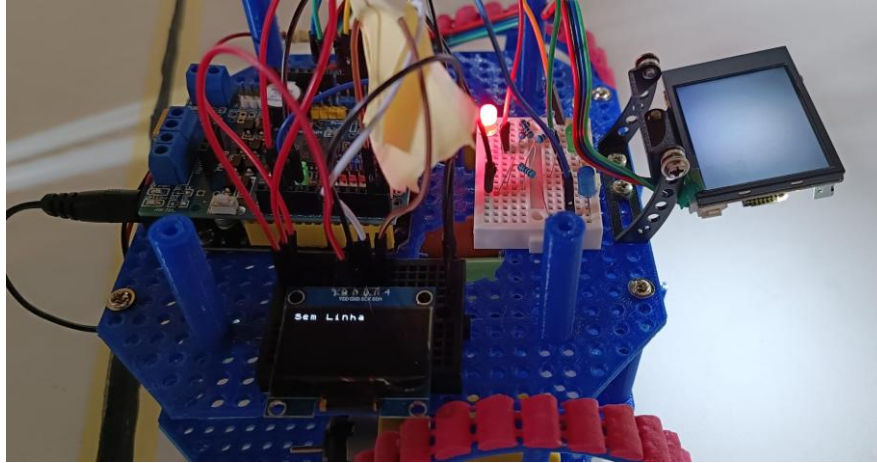
Fi

Fig. 18. Identificação de um marcador visual (ID 3) pela camera Huskylens

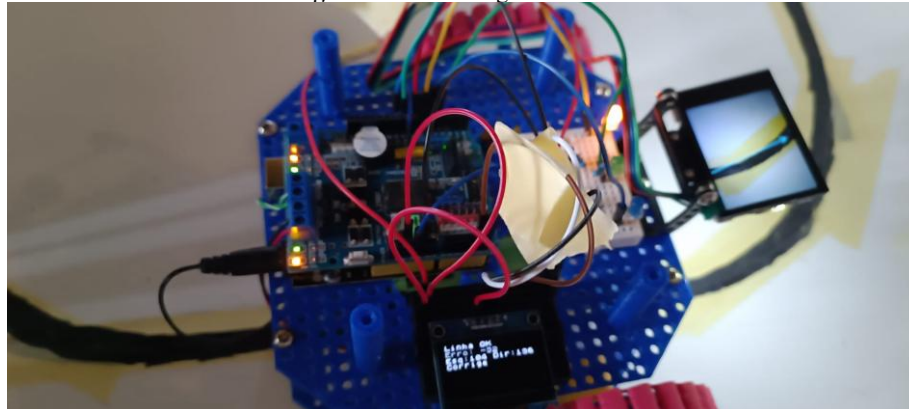
Estes testes de bancada demonstram que o Arduino UNO R4 WiFi é uma excelente opção para projetos com interatividade via rede, nomeadamente para fins pedagógicos. A possibilidade de comandar e visualizar o estado do sistema remotamente, bem como de integrar sensores e lógica embarcada, posiciona este microcontrolador como uma ferramenta estratégica para futuras versões do AzorCAR, especialmente em contextos onde o acesso à Internet não está garantido, mas se pretende um ambiente de programação moderno e acessível.

Testes com a função de Segue Linha

Durante os testes de seguimento de linha, o AzoCAR demonstrou desempenho robusto ao navegar por um tapete com uma linha preta, utilizando a câmara HuskyLens configurada no modo Line Tracking e apontada para o chão.

Fig. 3. AzorCAR com a câmera HuskyLens configurada no modo Line Tracking

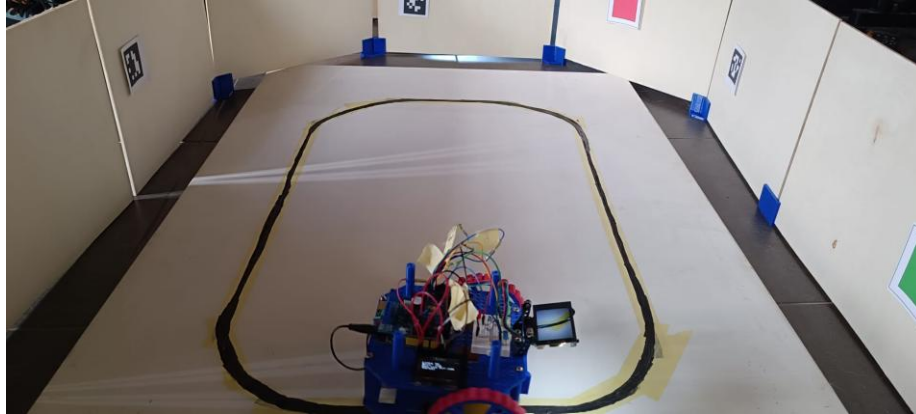
O algoritmo de controle proporcional, implementado no Arduino UNO com ganho ($K_p = 0.5$), calculou o erro com base na posição da linha em relação ao centro da imagem (160 pixels, resolução 320x240), ajustando as velocidades dos motores DC para manter desvios dentro de uma tolerância de 20 pixels. Com calibrações precisas, o robô seguiu a linha com alta estabilidade, avançando a uma velocidade base de 120 (escala PWM 0-255) e corrigindo desvios laterais de forma suave, com ciclos de avanço de 200 ms seguidos por paradas de 150 ms. Essas paradas foram cruciais para permitir o processamento em tempo real dos dados da HuskyLens, garantindo que o robô não saísse da linha durante a navegação.

Fig. 20. AzorCAR seguindo a linha

O feedback visual foi proporcionado por quatro LEDs indicadores: verde (pino 4) para movimento reto, azul (pino 5) para correção à esquerda, amarelo (pino 6) para correção à direita e vermelho (pino 7) piscando a cada 500 ms durante paradas ou falhas na detecção da linha. O display OLED SH1106, utilizando a biblioteca U8x8lib, exibia mensagens de status (ex.: "Linha OK", "Reto", "Corrige", "Parado") e informações

quantitativas, como o erro de alinhamento e as velocidades dos motores esquerdo e direito, facilitando o diagnóstico em tempo real.

Fig. 21. Simulação da função Segue Linha, na arena do AzorCAR no Desafio II



5 Discussão e Perspetivas Futuras

5.1 Limitações Identificadas

O desenvolvimento do AzorCAR demonstrou a eficácia de uma abordagem acessível e incremental à robótica educativa, mas também expôs limitações técnicas e desafios relevantes. A deteção múltipla de marcadores visuais, embora prevista na lógica do sistema, encontra-se ainda em fase inicial. A transição entre diferentes IDs ou cores requer maior robustez na gestão de estados e maior controlo sobre os tempos de resposta da câmara e do sistema de decisão.

Além disso, o comportamento do robô continua fortemente dependente de uma calibração manual e precisa dos parâmetros visuais, nomeadamente das dimensões da moldura (largura e altura) que definem a zona de paragem ideal. Esta calibração é sensível a fatores como a iluminação ambiente, o tipo de superfície e a altura da câmara, exigindo ajustes frequentes para garantir fiabilidade nos testes.

5.2 Transição Tecnológica: de Arduino a ROS 2 com Passagem Intermédia por Raspberry Pi e OpenCV

A arquitetura do projeto AzorCAR foi desenhada de forma modular e progressiva, permitindo não apenas um início acessível com o microcontrolador Arduino UNO R3/R4 WiFi, mas também uma transição tecnológica gradual para plataformas mais poderosas. Embora o objetivo final seja a integração com o Robot Operating System 2 (ROS 2), considerou-se pedagógica e tecnicamente mais eficaz a adoção de uma fase intermédia, baseada no uso de um Raspberry Pi 5 com câmara Pi e biblioteca OpenCV.

Essa abordagem oferece múltiplas vantagens. Em primeiro lugar, permite que os alunos desenvolvam competências em programação em Python e em visão

computacional com OpenCV, ferramentas amplamente usadas em robótica moderna. A câmara Pi, aliada à capacidade computacional do Raspberry Pi, possibilita um processamento local mais flexível das imagens captadas, sem depender de hardware especializado como a HuskyLens. Técnicas como detecção de cor em HSV, rastreamento de objetos, identificação de contornos e até reconhecimento facial tornam-se assim exequíveis com uma curva de aprendizagem controlada. Esta fase também prepara o terreno para a adoção posterior de ferramentas como MediaPipe e algoritmos de detecção baseados em deep learning.

Fig. 22. Logótipo oficial da versão Jazzy Jalisco do sistema ROS 2, lançado em 2024.



Só após este domínio intermédio é que se justifica a introdução do ROS 2. De facto, o ROS 2, embora poderoso, possui uma complexidade significativa, exigindo conhecimentos prévios sobre tópicos, nós, serviços, ações, e sistemas de build como o colcon. Foi neste contexto que o autor configurou um sistema com Ubuntu 24.04 no Raspberry Pi 5 e procedeu à instalação do ROS 2 Jazzy Jalisco, criando workspaces, pacotes em Python e C++, e testando ferramentas como o `rqt_graph` e o `Turtlesim`, o que permitiu validar o funcionamento de publicadores e subscritores num ambiente real [12].

Fig. 23. Execução do nó talker usando ROS 2 (`demo_nodes_cpp`) no terminal, publicando mensagens no tópico.

```
joaopavao@joaopavao:~$ ros2 run demo_nodes_cpp talker
[INFO] [1751248862.735613570] [talker]: Publishing: 'Hello World: 1'
[INFO] [1751248863.735584340] [talker]: Publishing: 'Hello World: 2'
[INFO] [1751248864.735579073] [talker]: Publishing: 'Hello World: 3'
[INFO] [1751248865.735579250] [talker]: Publishing: 'Hello World: 4'
[INFO] [1751248866.735579316] [talker]: Publishing: 'Hello World: 5'
[INFO] [1751248867.735581326] [talker]: Publishing: 'Hello World: 6'
[INFO] [1751248868.735577040] [talker]: Publishing: 'Hello World: 7'
[INFO] [1751248869.735579477] [talker]: Publishing: 'Hello World: 8'
[INFO] [1751248870.735580209] [talker]: Publishing: 'Hello World: 9'
[INFO] [1751248871.735589053] [talker]: Publishing: 'Hello World: 10'
```

Fig. 24. Execução do nó listener usando ROS 2 (`demo_nodes_cpp`) no terminal, recebendo mensagens de um tópico.

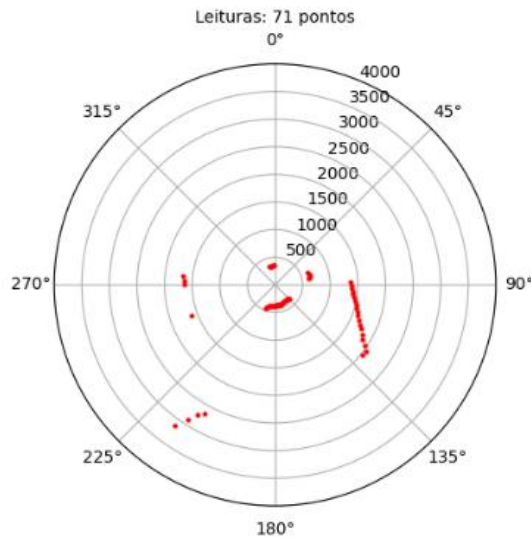
```
joaopavao@joaopavao:~$ ros2 run demo_nodes_cpp listener
[INFO] [1751248939.735773753] [listener]: I heard: [Hello World: 78]
[INFO] [1751248940.735890076] [listener]: I heard: [Hello World: 79]
[INFO] [1751248941.735871268] [listener]: I heard: [Hello World: 80]
[INFO] [1751248942.735900386] [listener]: I heard: [Hello World: 81]
[INFO] [1751248943.735883152] [listener]: I heard: [Hello World: 82]
[INFO] [1751248944.735881308] [listener]: I heard: [Hello World: 83]
[INFO] [1751248945.735881611] [listener]: I heard: [Hello World: 84]
[INFO] [1751248946.735882469] [listener]: I heard: [Hello World: 85]
[INFO] [1751248947.735882995] [listener]: I heard: [Hello World: 86]
[INFO] [1751248948.735885761] [listener]: I heard: [Hello World: 87]
```

A lógica do AzorCAR, centrada em blocos independentes de hardware (motores, sensores, câmara), é compatível com este tipo de evolução tecnológica. A comunicação entre placas poderá, futuramente, ser estabelecida por ROS Serial, integrando dados provenientes de Arduinos em nós ROS 2 de nível superior. Espera-se também utilizar simuladores como Gazebo ou RViz para desenvolvimento e teste de algoritmos de navegação, reconhecimento e mapeamento (SLAM), aproveitando os benefícios de ambientes virtuais para prototipagem rápida e segura [13].

Este percurso pedagógico — Arduino → Raspberry Pi + OpenCV → ROS 2 — permite equilibrar acessibilidade inicial com complexidade gradual, reforçando a aprendizagem por etapas e promovendo a compreensão progressiva de conceitos de engenharia robótica e visão computacional.

Outro elemento explorado de forma preliminar no projeto AzorCAR foi o uso de sensor LiDAR (Light Detection and Ranging), nomeadamente o modelo RPLIDAR A2 da Slamtec [14], que permite realizar mapeamento em tempo real de ambientes através da emissão e receção de feixes laser rotativos. Durante os testes iniciais feitos, foi possível obter leituras estáveis de distâncias em 360° e visualizar os dados com scripts em Python e ferramentas gráficas básicas. O LiDAR assume um papel central em sistemas robóticos autónomos avançados, sendo amplamente utilizado em aplicações como navegação indoor, deteção de obstáculos, SLAM (Simultaneous Localization and Mapping) e veículos autónomos. A sua integração futura com ROS 2 será estratégica, permitindo enriquecer o conjunto de dados sensoriais disponíveis para tomada de decisão em tempo real, além de potenciar o uso de algoritmos robustos de planeamento e localização espacial. Esta experiência prática reforça o potencial do AzorCAR como plataforma escalável, apta a evoluir de um projeto educativo para uma base experimental avançada de robótica móvel.

Fig. 25. Representação gráfica polar das leituras obtidas com o sensor RPLIDAR A2 (Slamtec).



5.3 Comparativo de Câmaras

O projeto AzorCAR iniciou-se com a integração da câmara HuskyLens, da DFRobot, mas prevê-se a inclusão futura de outras soluções para visão computacional. Por esse motivo, foi delineado um pequeno comparativo técnico-pedagógico entre câmaras frequentemente utilizadas em contextos de robótica educativa: HuskyLens, Pi Camera, ESP32-CAM e webcams USB genéricas.

A HuskyLens apresenta-se como uma opção robusta para ambientes educativos devido à sua IA nativa embutida, que permite reconhecimento de cores, tags visuais, objetos e seguimento de linha sem necessidade de programação de baixo nível. Comunica facilmente com microcontroladores via UART ou I2C e não requer integração com sistemas operativos complexos. O seu principal ponto fraco é a ausência de interface com o PC, o que limita a visualização e análise avançada de dados. Ainda assim, pela sua simplicidade e eficácia em atividades com Arduino ou micro:bit, continua a ser recomendada em contexto pedagógico inicial [12].

A Pi Camera, associada ao Raspberry Pi, oferece grande flexibilidade quando usada com bibliotecas como OpenCV. Embora exija mais conhecimentos técnicos e configuração inicial, permite implementar algoritmos de deteção em tempo real, reconhecimento facial, contornos ou leitura de QR codes. O seu custo é moderado e o apoio comunitário é vasto.

Fig. 26. Placa ESP32-CAM com camera integrada e suporte para comunicação Wi-Fi.



A ESP32-CAM constitui a opção mais económica, combinando microcontrolador e câmara num único módulo com Wi-Fi integrado. É uma solução interessante para projetos compactos e remotos, embora a sua capacidade de processamento e qualidade de imagem sejam limitadas. Finalmente, as webcams USB, quando ligadas a um computador ou Raspberry Pi, proporcionam alta qualidade de imagem e integração direta com OpenCV, sendo ideais para testes rápidos ou projetos com maiores recursos computacionais.

A tabela seguinte sintetiza este comparativo:

Tabela 1. Comparativo de câmaras para robótica educativa

Câmara	Funcionalidades	Compatibilidade	Interface PC	Custo (€)	Facilidade de Uso	Apoio Comunitário
HuskyLens	IA nativa (cor, TAGs, linha)	Arduino, micro:bit	nenhuma	45–60	Alta	Alto
Pi Camera	Via OpenCV	Raspberry Pi, ROS 2	USB/CSI	25–70	Média	Alto
ESP32-CAM	Via OpenCV, Wi-Fi	ESP32, Arduino	Wi-Fi/USB	5–10	Baixa	Moderado
Webcam USB	Via OpenCV	PC, Raspberry Pi	USB	10–40	Média	Variável

Este comparativo reforça a ideia de que a HuskyLens constitui uma solução equilibrada para o arranque de projetos de robótica educativa com visão computacional. No entanto, o Raspberry Pi com OpenCV oferece uma evolução natural em direção a ambientes de maior complexidade e autonomia. O AzorCAR está assim preparado para integrar ambas as soluções em diferentes fases de ensino ou competição, respeitando as competências e os recursos disponíveis em cada contexto.

6 Disponibilização e Reprodutibilidade

Com o intuito de promover a ciência aberta e incentivar a reutilização pedagógica e técnica deste trabalho, o projeto AzorCAR encontra-se documentado e publicado em acesso livre através do GitHub. O repositório inclui código-fonte, esquemas eletrônicos, desenhos 3D, apresentações, artigos e outros materiais complementares:

<https://github.com/joaopavao1979/AzorCAR>

7 Conclusão

O projeto AzorCAR demonstrou que é possível desenvolver um robô educativo funcional, modular e acessível, utilizando componentes simples e uma câmara de visão computacional como a HuskyLens. Através de uma abordagem incremental e orientada para testes práticos, foi possível implementar comportamentos autónomos relevantes para desafios educativos como os propostos no AzoresBot 2025.

O sistema provou ser uma base sólida para aprendizagem ativa em programação, eletrónica e visão computacional, podendo ser facilmente replicado e adaptado para outros fins. Apesar das limitações identificadas, nomeadamente a necessidade de calibração frequente e a fase inicial da deteção múltipla, o projeto encontra-se numa trajetória evolutiva clara, com potencial de expansão para ambientes mais avançados com OpenCV, simulação e ROS 2.

O AzorCAR afirma-se, assim, como uma ferramenta versátil para o ensino da robótica nos Açores e um contributo real para a dinamização de clubes e práticas educativas STEM na região.

Referências

1. Wing, J.M.: Computational thinking. *Communications of the ACM* **49**(3), 33–35 (2006)
2. Bers, M.U.: *Coding as a Playground: Programming for Personal Expression*. Routledge (2020)
3. Papert, S.: *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York (1980)
4. DFRobot: *HuskyLens AI Camera Wiki*. Disponível em: https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_DFR0428. Último acesso: 21 jul. 2025
5. Klements, M.: *Adding AI Vision to a Robot Car Using a HuskyLens*. The DIY Life. Disponível em: <https://www.the-diy-life.com/adding-ai-vision-to-a-robot-car-using-a-huskylens/>. Último acesso: 21 jul. 2025
6. Klements, M.: *Arduino-Based Obstacle Avoiding Robot Car*. The DIY Life. Disponível em: <https://www.the-diy-life.com/arduino-based-obstacle-avoiding-robot-car/>. Último acesso: 21 jul. 2025
7. DFRobot: *Maqueen Plus V2 for micro:bit – Wiki*. Disponível em: https://wiki.dfrobot.com/Maqueen_Plus_V2_for_micro_bit_SKU_ROB0148. Último acesso: 21 jul. 2025
8. Arduino: *Arduino UNO R4 WiFi Documentation*. Disponível em: <https://docs.arduino.cc/hardware/uno-r4-wifi>. Último acesso: 22 jul. 2025
9. Olikraus: *U8g2: A Monochrome Graphics Library for Embedded Systems*. GitHub Repository. Disponível em: <https://github.com/olikraus/u8g2>. Último acesso: 21 jul. 2025
10. ROBOCUPJUNIOR Rescue. *RoboCupJunior Rescue — página oficial*. Disponível em: <https://junior.robocup.org/rescue/>. Acesso em: 22 jul. 2025.
11. Domingos, M., Pedro, F., Ramos, A., Funk, M.G., Mendes, A., & Cascalho, J. (2023). *Vulcano: A new robotic challenge for legged robots*. *Frontiers in Robotics and AI*, 9, Article 1057832. <https://doi.org/10.3389/frobt.2022.1057832>
12. DFRobot: *Course 663 – HuskyLens Tag Recognition*. Learn DFRobot. Disponível em: <https://learn.dfrobot.com/course-663.html>. Último acesso: 21 jul. 2025
13. ROS 2 Documentation: *Jazzy Jalisco (2025)*. Disponível em: <https://docs.ros.org/en/jazzy/index.html>. Último acesso: 21 jul. 2025
14. Slamtec: *RPLIDAR A2 – 360° Laser Scanner Development Kit*. Disponível em: <https://www.slamtec.com/en/Lidar/A2>. Último acesso: 21 jul. 2025