



UNIVERSIDADE FEDERAL RURAL DO SEMI ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO ENGENHARIA DE COMPUTAÇÃO

Hércules Janoca Santana Izidério

Proposta Pedagógica com Interface de Software para o Robô TAN

PAU DOS FERROS/RN

2024

Hércules Janoca Santana Izidério

Proposta Pedagógica com Interface de software para o Robô TAN

Monografia apresentada à Universidade Federal Rural do Semi-Árido como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Pedro Thiago Valério de Souza

Coorientador: Prof. Dr. Francisco Carlos Gurgel da Silva

PAU DOS FERROS/RN

2024

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

I98p Izidério, Hércules Janoca Santana.
Proposta pedagógica com interface de software
para o robô TAN / Hércules Janoca Santana
Izidério. - 2024.
36 f. : il.

Orientador: Pedro Thiago Valério de Souza.
Coorientador: Francisco Carlos Gurgel da
Silva Segundo.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Engenharia de
Computação, 2024.

1. Bluetooth. 2. Deficiência. 3.
Interatividade. 4. Tecnologia. 5. Software. I.
Souza, Pedro Thiago Valério de, orient. II. Silva
Segundo, Francisco Carlos Gurgel da , co-orient.
III. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).
Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

Hércules Janoca Santana Izidério

Proposta Pedagógica com Interface de Software para o Robô TAN

Monografia apresentada à Universidade Federal Rural do Semi-Árido como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Defendido em: 27 / 03/ 2025

BANCA EXAMINADORA

Pedro Thiago Valério de Souza, Prof. Dr.(UFERSA)
Presidente

Francisco Carlos Gurgel da Silva Segundo, Prof. Dr.(UFERSA)
Membro Examinador

Cecílio Martins de Sousa Neto, Prof. Dr.(UFERSA)
Membro Examinador

Adller de Oliveira Guimarães, Prof. Dr.(UFERSA)
Membro Examinador

AGRADECIMENTOS

Primeiramente, agradeço a Deus, fonte de força e sabedoria, por me conceder saúde, perseverança e a oportunidade de trilhar este caminho. Sem sua graça e proteção, essa jornada teria sido ainda mais desafiadora. Cada conquista, cada aprendizado e cada obstáculo superado foram possíveis porque Ele sempre esteve presente, iluminando meus passos e fortalecendo minha fé.

Expresso minha profunda gratidão à minha tia, Maria das Dores Janoca de Santana, e à minha mãe, Maria do Socorro Janoca de Santana, por todo o amor, apoio e incentivo incondicional ao longo dessa caminhada. O carinho e a dedicação de ambas foram fundamentais para que eu pudesse seguir em frente, enfrentando desafios com coragem e determinação.

Agradeço também à minha madrinha, Ana Paula Tavares, e à minha namorada, Maria Wenia de Sá, por estarem ao meu lado durante toda essa trajetória. Seu apoio, compreensão e incentivo foram essenciais para que eu mantivesse a motivação e seguisse firme em meus objetivos. Saber que pude contar com vocês em cada momento de dificuldade e celebração tornou essa jornada ainda mais significativa e especial.

Sou imensamente grato aos meus professores, que desempenharam um papel fundamental em minha formação acadêmica. Em especial, agradeço ao Professor Francisco Carlos Gurgel da Silva Segundo e ao Professor Pedro Thiago Valério de Souza, por sua dedicação, paciência e orientação ao longo do meu percurso. Suas contribuições foram essenciais para o desenvolvimento deste trabalho e para minha evolução como estudante e profissional.

Agradeço ainda à banca examinadora pela disponibilidade e pelas contribuições valiosas que enriqueceram este projeto. Seu olhar crítico e suas considerações foram fundamentais para o aprimoramento deste trabalho. Também deixo meu sincero agradecimento aos meus amigos Davi Martins e Rita Sampaio, por toda a amizade, companheirismo e parceria ao longo dessa caminhada. Compartilhar essa jornada com vocês tornou tudo mais leve e gratificante.

Por fim, meu agradecimento a todos que, de alguma forma, contribuíram para a realização deste projeto. Sejam com palavras de incentivo, ensinamentos, apoio emocional ou qualquer outra forma de colaboração, cada gesto fez diferença nessa trajetória. A todos, meu muito obrigado!

Ama a Deus durante toda sua vida, e invoca-o
para a tua salvação.

Eclesiástico 13:18

RESUMO

A tecnologia tem desempenhado um papel essencial no desenvolvimento cognitivo e educacional de pessoas com deficiências, contribuindo para a estimulação do raciocínio lógico, da coordenação motora e da resolução de problemas. No entanto, ainda há uma carência de projetos que integrem a robótica educacional de forma acessível e interativa para esse público. Diante desse cenário, este projeto tem como objetivo o desenvolvimento de um software para o controle remoto do Robô TAN, permitindo sua utilização em atividades pedagógicas voltadas para o estímulo do pensamento lógico e da tomada de decisões. O software foi projetado para estabelecer comunicação com o Robô TAN via tecnologia *Bluetooth*, possibilitando o envio de comandos para movimentação em quatro direções: frente, trás, esquerda e direita. A plataforma utilizada no desenvolvimento desse projeto foi o *App Inventor*, uma plataforma onde é possível fazer a parte de *design* e programar o aplicativo em forma de blocos. Dessa maneira, o usuário poderá liderar o robô por trajetórias desafiadoras, exigindo planejamento e estratégia para alcançar o destino final. Essa experiência proporciona um aprendizado mais interativo, incentivando a autonomia. O objetivo é tornar o ensino mais acessível e envolvente, promovendo maior integração entre os participantes e a tecnologia. Durante a criação do software, foram realizadas pesquisas sobre o impacto da robótica educacional no aprendizado de pessoas com deficiência intelectual, além da análise de plataformas semelhantes em diferentes microcontroladores, como o Arduino, para identificar funcionalidades que pudessem enriquecer a aplicação. A validação do sistema ocorreu por meio de testes práticos, que mostraram a eficácia do controle remoto, a interação dos usuários e os benefícios do uso da ferramenta no desenvolvimento de habilidades cognitivas. Espera-se que esse projeto contribua para ampliar o uso da tecnologia assistiva na educação, tornando o aprendizado mais inclusivo e acessível.

Palavras-chave: bluetooth; deficiência; interatividade; tecnologia; software.

ABSTRACT

Technology has played a crucial role in the cognitive and educational development of people with disabilities, contributing to the stimulation of logical reasoning, motor coordination, and problem-solving skills. However, there is still a lack of projects that integrate educational robotics in an accessible and interactive way for this audience. Given this scenario, this project aims to develop software for remotely controlling the TAN Robot, allowing its use in pedagogical activities focused on stimulating logical thinking and decision-making. The software was designed to establish communication with the TAN Robot via Bluetooth technology, enabling the transmission of movement commands in four directions: forward, backward, left, and right. The platform used for this project was App Inventor, which allows both the design and block-based programming of the application. This way, users can guide the robot through challenging trajectories, requiring planning and strategy to reach the final destination. This experience promotes a more interactive learning process, encouraging autonomy. The goal is to make education more accessible and engaging, fostering greater interaction between participants and technology. During the software development process, research was conducted on the impact of educational robotics on the learning process of individuals with intellectual disabilities, along with an analysis of similar platforms using different microcontrollers, such as Arduino, to identify functionalities that could enhance the application. The system was validated through practical tests, which demonstrated the effectiveness of remote control, user interaction, and the benefits of the tool in cognitive skill development. It is expected that this project will contribute to expanding the use of assistive technology in education, making learning more inclusive and accessible.

Keywords: bluetooth; deficiency; interactivity; technology; software.

SUMÁRIO

1 INTRODUÇÃO	09
1.1. Objetivo Geral	10
1.2. Objetivos Específicos	10
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 Importância da pedagogia na inclusão de pessoas com deficiência	11
2.2 Uso de robôs em atividades educacionais	12
2.3 MIT App Inventor	14
3 DESENVOLVIMENTO DO CONTROLE DO ROBÔ TAN	18
3.1 Designer e interface do software	18
3.1.1 Botões de comandos	18
3.1.2 Botão de conexão bluetooth	19
3.1.3 Botão seguir	19
3.1.4 Interface completa do software	20
3.2 Codificação em bloco do controle	21
3.2.1 Blocos de programação para conexão bluetooth	21
3.2.2 Bloco de programação para armazenar lista de comandos	22
3.2.3 Comandos de movimentação	23
3.2.4 Indicação da posição inicial das movimentações	24
3.2.5 Bloco de programação para o botão seguir	25
3.2.6 Bloco de ativação do clock	26
4 VALIDAÇÃO DO SISTEMA	28
4.1 Planejamento	28
4.2 Teste do aplicativo	30
4.3 Resultados	32
5 CONSIDERAÇÕES FINAIS	34
5.1 Trabalhos Futuros	35
REFERÊNCIAS	36

1 INTRODUÇÃO

A tecnologia tem desempenhado um papel fundamental na inclusão e no desenvolvimento de habilidades em crianças e pessoas com deficiência. Recursos interativos, como *softwares* educacionais, robótica e dispositivos conectados, proporcionam novas formas de aprendizado, estimulando habilidades cognitivas, motoras e sociais. Quando desenvolve-se projetos voltados a esse público, é notável que por meio da tecnologia podemos abrir um horizonte de oportunidades para essas pessoas. A aplicação de tecnologias assistivas na educação especial tem sido amplamente estudada e evidenciada como uma estratégia eficaz para promover maior engajamento e independência dos participantes (Gomes et al., 2019).

Neste contexto, o projeto teve início com a ideia de desenvolver um aplicativo para o controle do robô TAN, inspirado no conceito do *Blockly Maze*, que é um jogo onde podemos criar trajetórias utilizando blocos de programação, e por conta da similaridade com a programação em blocos, a plataforma escolhida para o desenvolvimento foi o *App Inventor*. O objetivo é criar uma experiência lúdica, desafiadora e divertida, em que o usuário deve conduzir o robô para que ele consiga desviar de obstáculos. O ambiente do aplicativo foi desenvolvido de forma que chame a atenção dos seus usuários, trazendo cores vivas nos botões. Essa abordagem explora o potencial das tecnologias assistivas no estímulo ao raciocínio lógico e à resolução de problemas, elementos essenciais para o desenvolvimento cognitivo e motor de pessoas com deficiência (LOPES, BARROSO, SERPA, 2015).

A plataforma App Inventor, desempenha um papel crucial no projeto, permitindo a criação de aplicativos sem a necessidade de conhecimentos avançados em programação. Através dessa ferramenta, o *software* desenvolvido viabiliza a comunicação com o robô por meio de comandos enviados via *bluetooth*. Além disso, durante essa etapa, foi essencial considerar a acessibilidade e o engajamento do público-alvo, composto por pessoas com deficiências. Dessa forma, o aplicativo foi desenvolvido de maneira intuitiva, com uma interface amigável e visualmente atraente. Ao integrar tecnologia, inclusão e aprendizado, este projeto se apresenta como uma iniciativa inovadora para apoiar o desenvolvimento cognitivo e motor de pessoas com deficiência. O uso de um robô em um ambiente desafiador, aliado ao controle interativo via *software*, transforma o aprendizado em uma experiência motivadora e acessível, reforçando habilidades essenciais que podem ser aplicadas em diversas situações da vida cotidiana.

1.1. Objetivo Geral

Desenvolver e validar uma solução tecnológica educacional que combine robótica e *software* de controle via *bluetooth*, com o objetivo de proporcionar uma experiência inclusiva e interativa capaz de estimular o desenvolvimento cognitivo e senso de direcionamento de pessoas com deficiência, por meio da resolução de problemas lógicos.

1.2. Objetivos Específicos

- Realizar uma revisão bibliográfica para identificar fundamentos teóricos e boas práticas relacionadas ao uso de tecnologia no desenvolvimento educacional de pessoas com deficiência.
- Pesquisar *softwares* e ferramentas existentes que utilizem robótica e controle remoto via *bluetooth*, analisando suas funcionalidades e limitações.
- Criar um projeto funcional composto por um robô controlado pelo aplicativo desenvolvido, conectado via *bluetooth* que atenda ao público-alvo.
- Realizar testes práticos em ambientes controlados para avaliar a funcionalidade do *software* e a interação do público-alvo com o sistema, considerando o engajamento e o desenvolvimento de habilidades cognitivas e motoras.
- Analisar qualitativamente os resultados obtidos nos testes, identificando a eficácia da solução proposta e possíveis melhorias para ampliar sua aplicabilidade.

2 FUNDAMENTAÇÃO TEÓRICA

O trabalho proposto tem um papel importante na inclusão de pessoas com deficiência, visando trazer novas possibilidades para o aprendizado e o desenvolvimento de habilidades cognitivas e motoras, fazendo com que essas pessoas tenham oportunidade de se desenvolver e ter destaque na nossa sociedade. Por meio da interação com dispositivos programáveis, os participantes são estimulados a resolver problemas, ações planejadas e descobrir com muita alegria a resolução de algumas situações, tornando o processo de ensino mais dinâmico e acessível. O uso de robôs em atividades educacionais permite a adaptação de desafios conforme o nível de cada indivíduo, promovendo uma aprendizagem personalizada e incentivando a autonomia. No projeto utiliza-se o robô TAN, esse robô é composto por sensores de distância, seguidor de linha e um módulo bluetooth. Essa conexão *bluetooth* ajuda a se conectar com os controles convencionais e com o controle proposto neste projeto. A utilização de uma interface chamativa, ou seja, com todos os botões coloridos, obtém uma leveza na utilização do aplicativo e faz com que os usuários gostem de usar o *software*.

O Robô TAN foi desenvolvido em um trabalho anterior pelo engenheiro Felipe Cardoso, ex-discente da Universidade Federal Rural do Semi-Árido (UFERSA). No aplicativo inicial, o robô era controlado por um sistema convencional, onde os comandos eram enviados em tempo real, permitindo a movimentação imediata conforme a entrada do usuário. No entanto, esse controle difere da abordagem proposta neste trabalho, que busca uma interação mais planejada e estratégica, baseada na criação de uma sequência de comandos antes da execução.

No contexto do desenvolvimento do controle, o *App Inventor* facilita a construção de uma interface amigável para controlar os movimentos do robô via *bluetooth*, tornando a experiência mais acessível e motivadora. Além disso, a resposta do robô aos comandos enviados pelo usuário contribui para a compreensão da relação entre ação e consequência, reforçando conceitos de direcionamento e de planejamento.

2.1 Importância da pedagogia na inclusão de pessoas com deficiência.

A educação molda o caráter e nos ajuda a ter certa facilidade no processo de socialização. A estimulação e a aprendizagem precoce ajudam essas pessoas a se encaixar com mais facilidade nos ambientes que estão inseridos. Sendo assim, a educação deve acompanhar o ritmo individual de cada aluno, com isso, pessoas com deficiência devem ser

ensinadas de acordo com seu nível de aprendizagem, para que eles não percam o encanto e a maravilha do ensino. Dessa forma, os professores devem estimular a autonomia e favorecer a construção do conhecimento de forma significativa. Nesse contexto, a tecnologia tem sido fornecida uma aliada indispensável para potencializar o processo educacional, permitindo a criação de recursos interativos que estimulam a participação ativa dos alunos. Segundo o artigo "Impacto das tecnologias digitais no desenvolvimento cognitivo infantil", baseado nas teorias de Piaget e Vygotsky, as interações digitais podem influenciar a construção do conhecimento e o desenvolvimento social das crianças. Os *Softwares* educativos, aplicativos interativos e robótica pedagógica são ferramentas que favorecem a aprendizagem por meio da experimentação e do jogo, aspectos essenciais para o desenvolvimento cognitivo e motor de crianças e pessoas com deficiência.

A tecnologia é um caminho encantador para usuários com deficiência, ela apresenta mecanismos interessantes e divertidos para essas pessoas. Nesse caso, quando falamos de jogos de celulares, robôs ou qualquer outra coisa tecnológica, não estamos falando apenas em divertimento, mas sim, em potencialização do desenvolvimento da evolução motora e raciocínio lógico. Em relação ao *software* que está sendo criado, ele vai ajudar na compreensão da relação entre ação e consequência, reforçando conceitos espaciais e de planejamento.

Dessa forma, a pedagogia, aliada à tecnologia, desempenha um papel essencial na promoção da inclusão educacional e no desenvolvimento das potencialidades das pessoas com deficiência. A personalização do ensino, o uso de metodologias lúdicas e o acesso a ferramentas tecnológicas inovadoras possibilitam uma experiência de aprendizado mais rica, estimulante e significativa. Como destaca Mantoan (2006), a inclusão não deve ser adaptada apenas ao acesso à escola, mas uma garantia de uma educação de qualidade, às necessidades de cada aluno, respeitando sua individualidade e promovendo sua participação ativa na sociedade.

O aspecto lúdico no ensino também desempenha um papel crucial na aprendizagem de crianças com deficiência. Segundo Kishimoto (2010), os jogos educativos avançados para o desenvolvimento cognitivo, permitem que a criança aprenda de forma prazerosa e significativa. Além disso, atividades lúdicas promovem o engajamento e facilitam a construção do conhecimento de maneira mais natural.

2.2 Uso de robôs em atividades educacionais

A robótica educacional vem se tornando uma maneira incrível de deixar o aprendizado mais prático e encantador. Quando os alunos interagem com robôs, eles não apenas aprendem conteúdos teóricos, mas também desenvolvem habilidades importantes, como lógicas, detalhadas, motoras e criativas. Para Seymour Papert (1980), aprender fazendo é muito mais eficaz do que apenas absorver informações de forma passiva. E é exatamente isso que a robótica permite: unir teoria e prática de uma forma dinâmica, fazendo com que os estudantes realmente entendam e se interessem pelo que estão aprendendo. Quando vemos e sentimos que a interação está sendo feita de forma proveitosa, a vontade de produzir novos meios de conectar pessoas a tecnologia só aumenta, e junto disso, vem a determinação de fazer algo que eleve a educação e a sociedade em geral.

Uma das maiores vantagens de usar robôs na escola é que elas estimulam a curiosidade e a criatividade. Os alunos testam ideias, são livres para errar, concordar e tentar de novo, o que pode fazer parte do processo de aprendizagem. Além disso, aprender dessa forma, como por exemplo, tentando tirar um robô de um labirinto, pode ser muito mais divertido do que apenas decorar fórmulas e textos. Quando os alunos conseguem ver o resultado do que programaram, isso traz uma sensação de conquista e incentivo para continuar aprendendo.

Figura 2 - Crianças ajudando no desenvolvimento de robôs.



Fonte: JGalion (2018)

Na Figura 2, temos crianças desenvolvendo robôs, mostrando a importância de aprender tecnologia desde cedo, e assim prepará-las para um mundo cada vez mais digital e inovador.

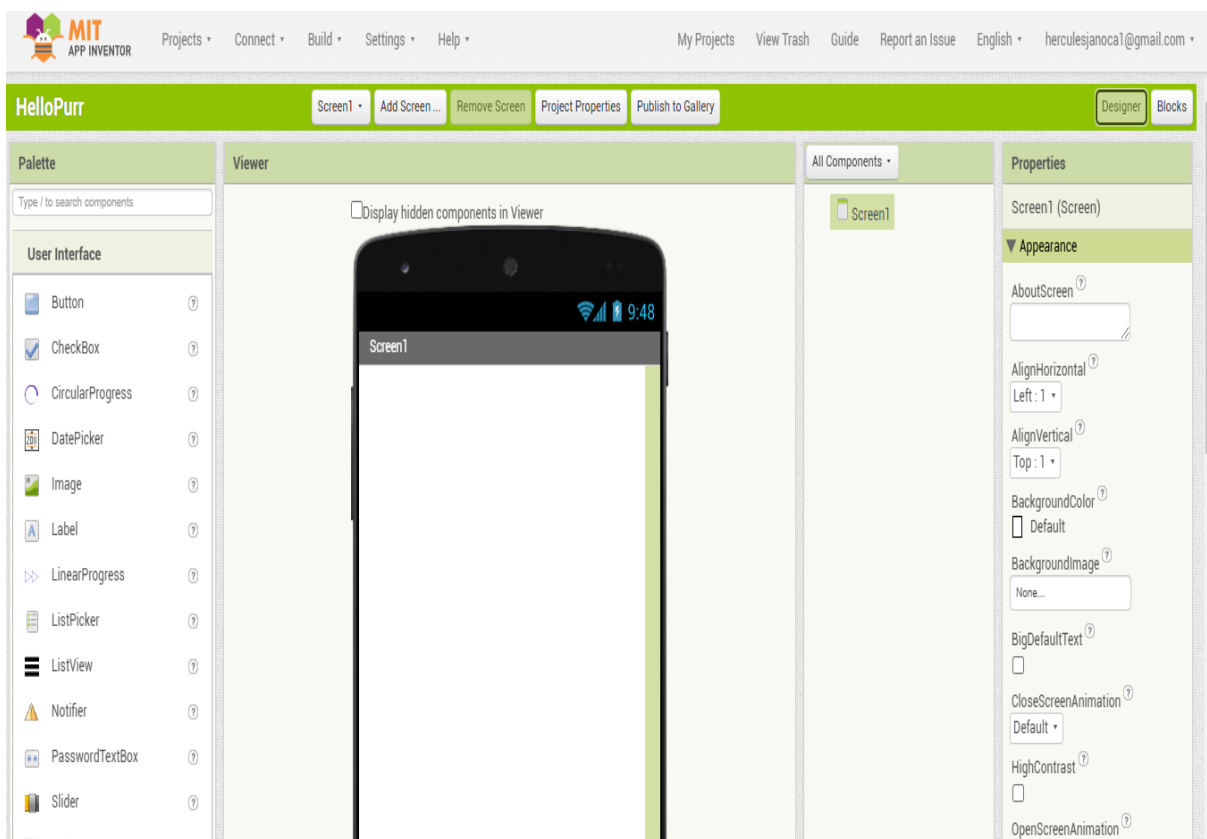
2.3 MIT App Inventor

O MIT App Inventor é uma plataforma que ajuda a facilitar o desenvolvimento de aplicativos, ela permite que qualquer pessoa, mesmo sem experiência em programação, consiga criar jogos e aplicativos funcionais para dispositivos Android. O projeto foi desenvolvido originalmente pelo Google, mas, em 2011, foi transferido para o *Massachusetts Institute of Technology* (MIT), onde continua sendo aprimorado. A ideia por trás da ferramenta é tornar uma programação mais acessível e interativa, ajudando estudantes, professores e entusiastas da tecnologia a explorarem o mundo do desenvolvimento de *software*.

Uma característica importante do App Inventor é sua acessibilidade. Como é uma ferramenta gratuita e baseada na *web*, qualquer pessoa com um computador e acesso à internet pode começar a criar aplicativos sem precisar instalar *softwares* pesados. Ou seja, não é necessário que o usuário baixe nenhuma IDE para construir seus aplicativos. Isso possibilita que mais pessoas, independentemente de sua formação ou recursos financeiros, possam explorar o mundo da programação.

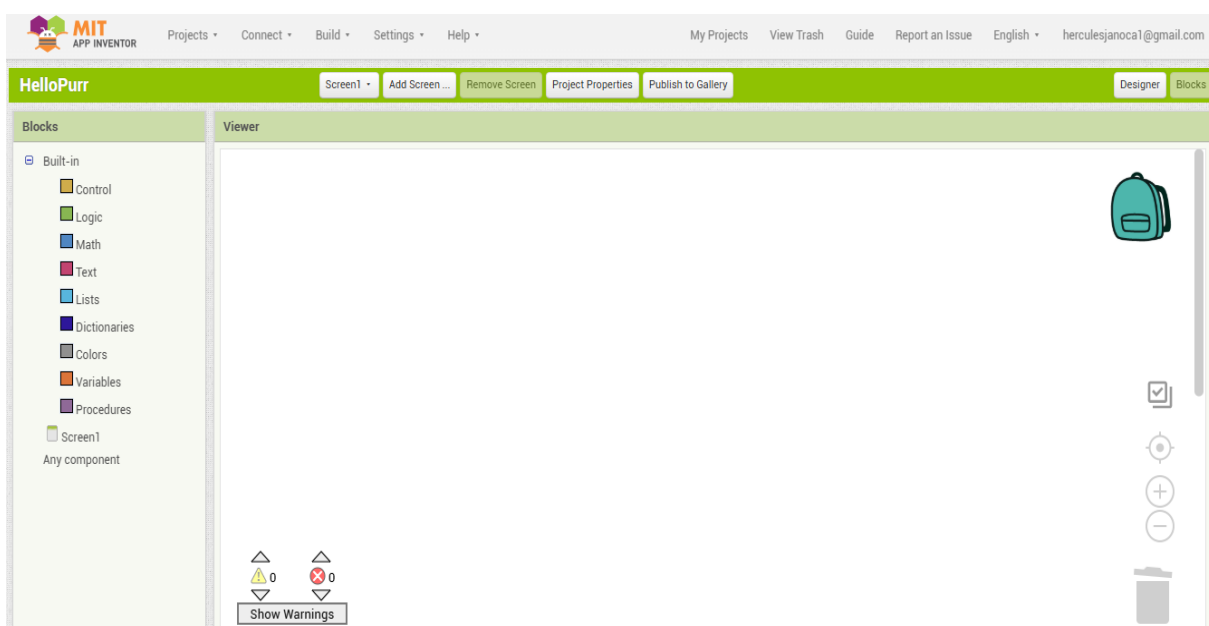
O MIT App Inventor é dividido em duas partes principais: o *Design* e a Programação em Blocos como ilustrado nas Figuras 3 e 4. Essa separação facilita o desenvolvimento dos aplicativos, fazendo com que o usuário primeiro monte sua interface da maneira que achar melhor, depois programe todo o funcionamento do seu aplicativo. Esse modelo torna a criação de aplicativos mais acessíveis, especialmente para iniciantes, pois não exige conhecimento avançado em linguagens de programação tradicionais.

Figura 3 - Local de desenvolvimento do designer MIT App Inventor.



Fonte: MIT App Inventor

Figura 4 - Local da programação em blocos no MIT App Inventor.



Fonte: MIT App Inventor.

Referente a parte de designer do software, ela vai ser o local onde o usuário irá construir a interface do seu jogo ou aplicativo. No *designer* será possível adicionar botões, imagens, listas e colocar cores em qualquer componente que desejar. Todos os componentes são organizados de forma simples e intuitiva, permitindo que o usuário arraste para a tela de desenvolvimento e posicione como quiser.

Entre os principais componentes disponíveis no designer, destacam-se:

- **Botões:** Servem para executar as ações quando pressionados.
- **Caixas de texto:** Permite ao usuário digitar informações.
- **Rótulos:** Textos fixos na tela.
- **Imagens:** Permite ao usuário inserir imagens.
- **Sensores:** O usuário pode incluir acelerômetro, GPS, bússola, entre outros.
- **Mídia:** Permite adicionar vídeos e câmaras.
- **Conectividade:** Inclui *bluetooth* e conexão com a internet.

Em relação a parte de programa em blocos, o MIT App Inventor utiliza um sistema de blocos coloridos que se encaixam, semelhante às peças de um quebra-cabeça. Esse método visual substitui a escrita de códigos tradicionais e facilita o aprendizado da lógica de programação.

Os blocos são divididos em categorias, como:

- **Blocos de Controle:** Permite criar condições e repetições, como “se-então” e “enquanto”.
- **Blocos de Matemática:** Executam cálculos e operações numéricas.

- Blocos de texto: Manipulam todo tipo de texto, como juntar palavras ou convertem números para texto.
- Bloco de variáveis: Armazenam informações para serem usadas no aplicativo.
- Bloco de procedimentos: Criam funções personalizadas para organizar o código.
- Bloco de componentes: Controlam os elementos adicionados no *designer*, como botões, sensores e bluetooth.

3 DESENVOLVIMENTO DO CONTROLE DO ROBÔ TAN

Esta seção descreve o *designer* e o blocos de programas do controle do robô, dessa maneira, mostrando seus detalhes e apresentando seu desenvolvimento.

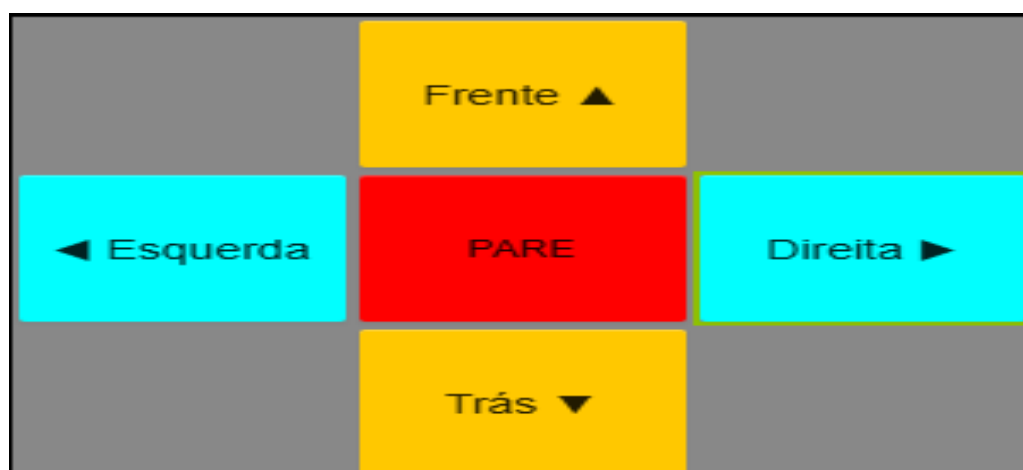
3.1 Designer e interface do software

O desenvolvimento do designer no MIT App Inventor foi pensado para tornar o controle do Robô TAN intuitivo e acessível, especialmente para crianças e jovens com Síndrome de Down . A interface foi projetada com botões grandes e coloridos, permitindo que os usuários possam navegar de forma simples e eficiente. A ideia principal é proporcionar uma experiência educativa e interativa, ajudando os participantes a desenvolver habilidades motoras, cognitivas e lógicas enquanto conduzem o robô. Vale salientar que o controle pode ser utilizado em projetos semelhantes que utilizem outro tipo de microcontrolador, como o arduino.

3.1.1 Botões de comandos

A figura 5 ilustra o design dos botões de movimento. As cores vibrantes no aplicativo servem para capturar a atenção e tornar a experiência mais envolvente. Estudos realizados pelo *site* Psicologia.Design e outros pesquisadores de usabilidade demonstram que núcleos bem definidos podem facilitar a identificação dos botões e melhorar a interação com o aplicativo, principalmente para pessoas com dificuldades cognitivas.

Figuras 5 - Botões de controle de direção.



Fonte: Autoria própria (2025)

O principal controle do robô é composto por botões direcionados que enviam comandos para que ele vá para frente, atrás, esquerda, direita e pare. Dessa forma, os usuários podem testar diferentes estratégias para conduzir o robô pelo labirinto, estimulando o pensamento lógico e a criatividade.

3.1.2 Botão de conexão *bluetooth*

Outro elemento essencial do *design* é o botão de conexão *bluetooth*, que possibilita a comunicação entre o aplicativo e o robô. Esse botão garante que, antes de qualquer comando ser enviado, o dispositivo esteja devidamente pareado com o robô, evitando falhas na execução dos movimentos.

A Figura 6, ilustra o botão para conexão *bluetooth*, essa funcionalidade é fundamental para garantir que os comandos sejam transmitidos de forma correta, tornando a experiência mais fluida e sem interrupções.

Figura 6 - Botão para conexão via Bluetooth.



Fonte: Autoria própria (2025).

3.1.3 Botão seguir

A figura 7 apresenta outra funcionalidade do aplicativo, o botão “Seguir”, que permite que o robô execute uma sequência de comandos de uma só vez. Isso significa que o usuário pode planejar uma trajetória, instruir os botões direcionados na ordem desejada, e apenas depois apertar o botão "Seguir" para que o robô execute toda a trajetória programada.

Figura 7 - Botão seguir do aplicativo.



Fonte: Autoria própria (2025)

Esse conceito é inspirado em jogos educativos, como o [Blockly Maze](#), onde os alunos precisam montar uma sequência de movimentos antes de vê-los em ação, no blockly é utilizado o botão executar ao invés de “seguir”. Essa abordagem reforça o raciocínio lógico e a tomada de decisões.

3.1.4 Interface completa do software

A experiência de guiar o robô, não apenas ensina conceitos básicos de programação e robótica, mas também proporciona um momento de diversão e conquista para os participantes. Dessa maneira, utilizamos uma interface simples e chamativa para que os jovens e as crianças gostem de utilizar o aplicativo.

Figura 8 - Interface do controle do robô TAN.



Fonte: Autoria própria (2025)

Para usuários com deficiência, essa interação pode ser extremamente gratificante, pois permite que eles desenvolvam autonomia e confiança ao verem o robô respondendo aos seus comandos. O aprendizado se torna lúdico e acessível, incentivando a criatividade e promovendo a inclusão digital de forma interativa e dinâmica.

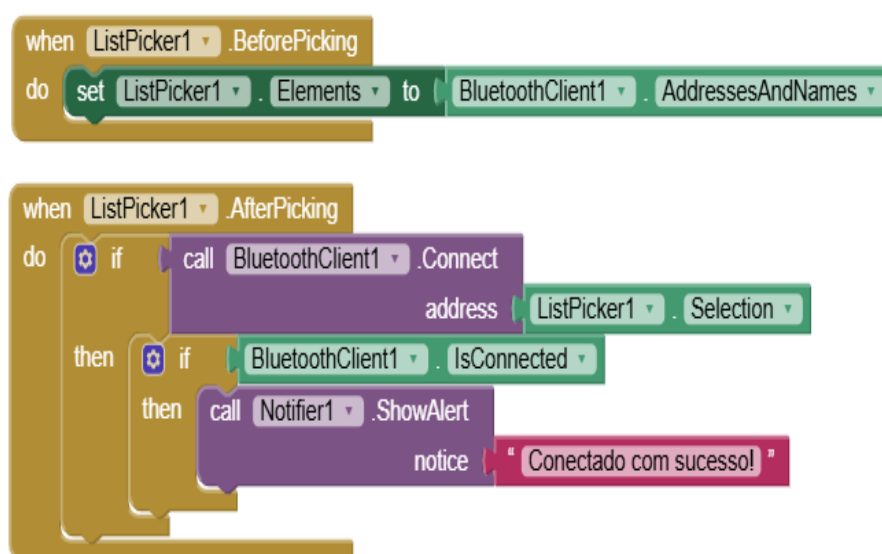
3.2 Codificação em blocos do Controle

A programação em blocos do *MIT App Inventor* serve para criar aplicativos móveis de forma intuitiva e acessível, sem a necessidade de escrever código tradicional. Ele permite que usuários, especialmente iniciantes, desenvolvam aplicativos para Android usando blocos visuais que representam comandos e lógica de programação.

3.2.1 Blocos de programação para conexão bluetooth

A Figura 9, demonstra a codificação da conexão bluetooth. Esse código em blocos do *MIT App Inventor* é responsável por estabelecer a conexão Bluetooth entre o aplicativo e um dispositivo externo, como o Robô TAN.

Figura 9 - Código em blocos para conexão bluetooth



Fonte: Autoria própria (2025)

Ele utiliza um *ListPicker* para exibir os dispositivos disponíveis e um *BluetoothClient* para gerenciar a conexão. O primeiro bloco, "*When ListPicker1.BeforePicking*", define que antes do usuário selecionar um dispositivo, a lista de elementos do *ListPicker1* será preenchida com os nomes e endereços dos dispositivos *Bluetooth* disponíveis. Isso garante que o usuário possa escolher um dispositivo válido para se conectar.

O segundo bloco, "*When ListPicker1.AfterPicking*", é ativado logo após o usuário selecionar um dispositivo na lista. O código tenta conectar ao dispositivo escolhido usando o *BluetoothClient1.Connect*, passando pelo endereço do dispositivo selecionado. Em seguida, há uma estrutura condicional "*if*" que verifica se a conexão foi bem-sucedida por meio do bloco *BluetoothClient1.IsConnected*. Caso a conexão seja estabelecida corretamente, o aplicativo exibe uma mensagem de alerta com o texto "Conectado com sucesso!", utilizando o *Notifier1.ShowAlert*.

Esse código garante que o processo de conexão do dispositivo seja intuitivo e automatizado, permitindo que o usuário escolha facilmente um *Bluetooth* e receba um aviso caso a conexão seja bem-sucedida. Caso a conexão falhe, o código não fornece uma mensagem de erro, mas isso poderia ser adicionado para melhorar a usabilidade do aplicativo. Esse mecanismo é essencial para a comunicação entre o aplicativo e o robô, permitindo o envio de comandos de movimentação após a conexão ser estabelecida.

3.2.2 Bloco de programação para armazenar lista de comandos

A Figura 10 apresenta a codificação de armazenamento da lista. Esse bloco de código apresentado trata da inicialização da variável chamada global *ListaComandos*. Essa variável é definida como uma lista vazia e terá a função de armazenar todos os comandos que o usuário selecionar para controlar o robô. Esse passo é essencial, pois permite que o aplicativo registre a sequência e guarde os movimentos antes de enviá-los para execução. Sem essa lista, o robô não poderia seguir um conjunto de instruções predefinido, comprometendo a proposta do projeto.

Figura 10 - Bloco de variável para armazenar a lista de comandos.



Fonte: Autoria própria (2025)

A importância desse bloco reside no fato de que a *ListaComandos* funciona como uma memória temporária dos comandos que serão executados. Sempre que houver um botão de movimento para pressionar, o comando correspondente será armazenado nessa lista, garantindo que o usuário possa planejar seu percurso antes de ativá-lo. Esse conceito é

semelhante à programação sequencial, permitindo que o usuário tenha um controle preciso sobre as ações do robô.

3.2.3 Comandos de movimentação

Nessa parte da programação em bloco, observamos a implementação dos comandos de movimento. Cada botão representa uma direção específica: *F_botao* para frente ("F"), *B_botao* para trás ("B"), *L_botao* para esquerda ("L"), *R_botao* para direita ("R") e *S_botao* para parar ("S"). Sempre que um desses botões for pressionado, seu comando correspondente será adicionado à *ListaComandos*, garantindo que o aplicativo registre cada ação desejada antes da execução.

Figura 11 - Comandos para movimentação.



Fonte: Autoria própria (2025)

A Figura 11 ilustra os comandos de movimentação. A estrutura desses blocos é simples e eficiente: cada vez que um botão é pressionado, o comando correspondente é inserido na lista de comandos já existente. Essa abordagem permite que o usuário construa uma sequência de movimentos antes de executar o percurso do robô. Isso torna o aplicativo mais interativo e planejado, em vez de apenas enviar comandos individuais e imediatos.

Outro ponto relevante é que essa funcionalidade permite que o robô seja controlado de forma mais intuitiva e pedagógica. Especialmente para o público-alvo do projeto, como usuários com deficiência, essa abordagem facilita a compreensão da lógica por trás dos comandos, permitindo que eles desenvolvam habilidades cognitivas e de planejamento ao organizar a sequência antes da execução.

Por fim, essa estrutura de código garante que o botão "Seguir", responsável por iniciar a execução dos comandos, tenha uma lista completa de instruções para processar. Assim, quando acionado, o robô poderá seguir todo o trajeto registrado, reforçando a importância de *ListaComandos* como um elemento-chave no funcionamento do aplicativo.

3.2.4 Indicação da posição inicial das movimentações.

A figura 12 apresenta a variável global *IndiceAtual* com o valor 0. Essa variável desempenha um papel fundamental no controle da execução da sequência de comandos armazenados na *ListaComandos*. Sua função é indicar qual posição da lista está sendo processada no momento, permitindo que o aplicativo envie os comandos para o robô de maneira ordenada e estruturada. Sempre que um comando for executado, essa variável é incrementada para avançar ao próximo item da lista, garantindo que o robô siga corretamente a trajetória programada.

Figura 12 - Índice atual da lista de comandos.



Fonte: Autoria própria (2025)

A importância desse bloco está na organização e no controle do fluxo de comandos. Sem essa variável, o aplicativo não teria um ponto de referência para saber qual comando deve ser enviado em seguida, podendo comprometer a execução da sequência. Além disso, ao

definir *IndiceAtual* como 0 no início do processo, o código garante que a sequência sempre comece do primeiro comando armazenado, evitando erros na navegação do robô. Essa abordagem é essencial para garantir que o robô execute os movimentos na ordem desejada pelo usuário, proporcionando uma experiência mais interativa e personalizada.

3.2.5 Bloco de programação do botão seguir.

A Figura 13 demonstra o bloco “seguir”, que é responsável por programar o comportamento do botão que irá liberar as ações, ou seja, inicia a execução da sequência de comandos armazenados na *ListaComandos*. Quando o usuário clica nesse botão, o código primeiro define a variável *IndiceAtual* como 0, garantindo que a execução comece sempre do primeiro item da lista. Esse passo é essencial para evitar que o robô execute comandos da metade da lista caso o botão seja pressionado mais de uma vez sem inicializar corretamente.

Figura 13 - Blocos de programação do botão seguir.



Fonte: Autoria própria (2025)

Em seguida, há uma verificação para garantir que a lista de comandos não esteja vazia. Isso é feito pelo bloco *if not is list empty? get global ListaComandos*. Essa verificação é importante porque impede que o sistema tente acessar elementos inexistentes, o que pode causar erros na execução do código. Se a lista estiver vazia, o botão "Seguir" não fará nada, evitando um funcionamento inesperado do robô.

Caso a lista contenha comandos, o primeiro item da sequência é enviado via Bluetooth para o robô. Isso é realizado pelo bloco call *BluetoothClient1.SendText*, que utiliza select list item get global ListaComandos *index* 1 para selecionar o primeiro item da lista e transmiti-lo ao robô. Esse passo é fundamental para garantir que o primeiro movimento programado seja enviado corretamente antes que o temporizador assuma o controle da execução dos próximos comandos.

Após o envio do primeiro comando, o bloco *set Clock1.TimerEnabled to true* é ativado. Isso significa que o temporizador do aplicativo (*Clock1*) fluirá para funcionar, permitindo que os próximos comandos da lista sejam enviados automaticamente em intervalos regulares. Esse mecanismo garante que o robô execute os comandos na ordem correta e com o tempo adequado entre cada um, simulando uma movimentação fluida e controlada.

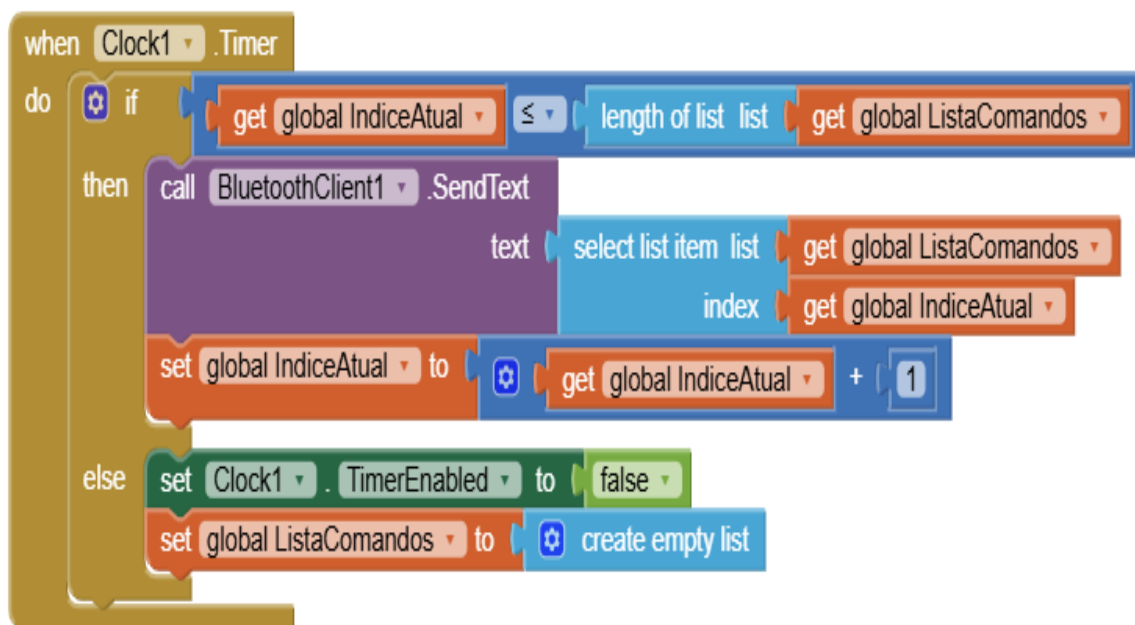
No geral, essa estrutura de programação garante que o usuário possa planejar um trajeto para o robô antes de fazê-lo, tornando o controle mais eficiente e interativo. Além disso, ao verificar a lista antes de iniciar e ao ativar o temporizador, o código evita erros e melhora a experiência do usuário, garantindo que o robô siga exatamente os comandos que foram selecionados anteriormente.

3.2.6 Bloco de ativação do clock

A Figura 14 apresenta os blocos referentes ao *clock*. O clock vai funcionar como um temporizador que vai permitir que os comandos sejam enviados um por um, garantindo que o robô execute a sequência de instruções desejadas com um intervalo de tempo adequado. Isso é essencial para permitir que o robô processe cada comando antes de receber o próximo, evitando sobreposição de instruções e garantindo que ele execute os movimentos corretamente.

A primeira parte do bloco verifica se o *ÍndiceAtual* (variável global) ainda está dentro do tamanho da lista de comandos. Se essa condição for verdadeira, ele seleciona o comando correspondente ao índice atual e o envia via *Bluetooth* para o robô. Em seguida, o *ÍndiceAtual* é incrementado em +1, garantindo que, na próxima execução do temporizador, o próximo comando da lista será enviado. Esse mecanismo é importante porque evita que todos os comandos sejam enviados de uma só vez, permitindo que o robô execute cada movimento de forma ordenada.

Figura 14 - Bloco referente ao clock.



Fonte: Autoria própria (2025)

Caso a condição inicial seja falsa (ou seja, quando todos os comandos já tiverem sido enviados), o bloco entra na parte *else*. Aqui, ele desativa o *Clock1* (*TimerEnabled* = *false*), impedindo que o temporizador continue rodando sem necessidade. Além disso, a *ListaComandos* é redefinida para uma lista vazia, garantindo que uma nova sequência de comandos possa ser programada sem interferência de comandos antigos. Essa abordagem evita erros e repetições indesejadas na execução do robô.

Este bloco é essencial para o funcionamento do aplicativo, pois organiza o envio dos comandos em sequência com um intervalo controlado. Sem esse controle, o robô poderia receber múltiplos comandos ao mesmo tempo, causando movimentos descoordenados. Além disso, a estrutura garante que o aplicativo possa ser reutilizado várias vezes sem precisar ser reiniciado, já que a lista de comandos é apagada e pode ser preenchida novamente para uma nova execução.

4 VALIDAÇÃO DO APLICATIVO

A validação da proposta foi realizada para observar o funcionamento do controle e as respostas do *software*. Os principais pontos observados foram as respostas dos comandos, como frente, direita, esquerda, trás e o botão parar. Também foi analisada a resposta do botão seguir, onde ele esperava todos os comandos serem feitos, para logo após acionar o robô. E englobado nos testes para validação, também observamos a conectividade do robô com o aparelho celular, para que não houvesse nenhum problema na hora de mandar os comandos. Essa etapa de validação foi dividida em três partes: Planejamento, teste do aplicativo e resultados.

4.1 Planejamento

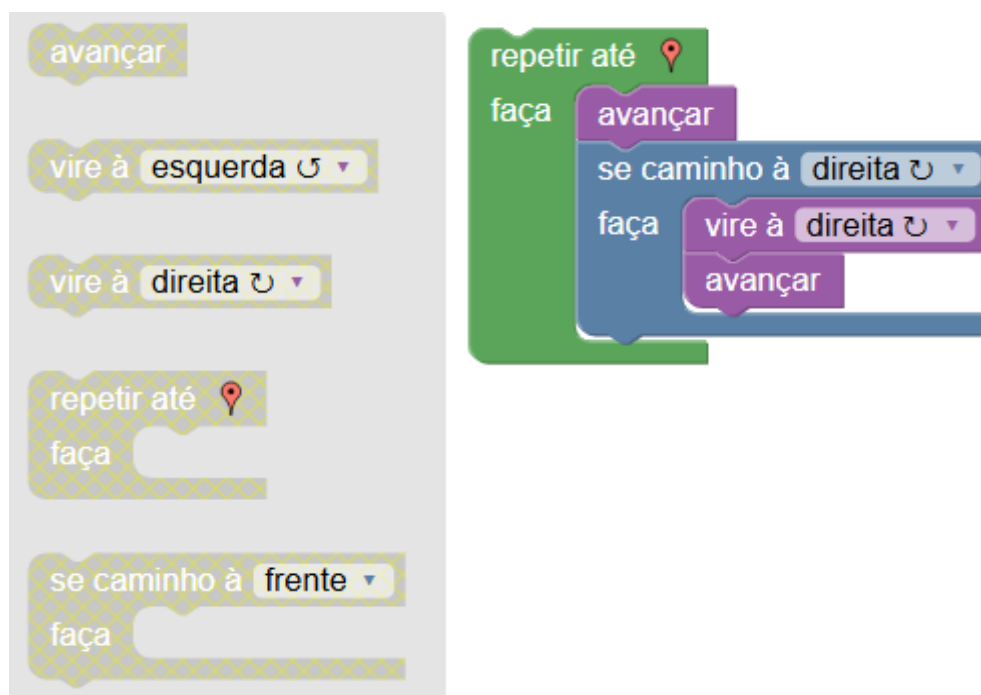
O planejamento do projeto teve início com a ideia de desenvolver um aplicativo para o controle do robô TAN , inspirado no conceito do Blockly Maze. O objetivo principal era criar um ambiente onde o usuário precisasse programar toda a trajetória do robô antes de iniciar seu movimento, promovendo o pensamento lógico e o planejamento antecipado das ações. Essa abordagem foi escolhida para o cultivo da organização e a compreensão da sequência de comandos, tornando o uso do robô uma experiência educativa e interativa.

Figura 15 - Tela de trajetória do Blockly Maze.



Fonte: Block Maze

Figura 16 - Códigos em bloco do Blockly Maze.



Fonte: Blockly Maze

Nas Figuras 15 e 16, temos a representação da interface do *Blockly Maze*, onde foi notória a similaridade com a parte de programação em bloco do App Inventor. Diante desse ponto, foi necessário buscar uma ferramenta que permitisse um desenvolvimento simplificado e eficiente, levando à escolha do próprio *MIT App Inventor*.

Inicialmente, para validar a conectividade entre o aplicativo e o robô, foi desenvolvido um controle básico, esse primeiro protótipo serviu para testar a comunicação via *Bluetooth* e garantir que o robô pudesse responder corretamente às instruções recebidas. Após essa etapa, iniciou-se o desenvolvimento do controle definitivo, que incorporou a funcionalidade da lista de comandos, possibilitando que os usuários registrassem uma sequência completa de movimentos antes de iniciar a execução.

É necessário pontuar que o planejamento também envolveu um período de adaptação ao *MIT App Inventor*, visto que essa foi a primeira vez que a ferramenta foi utilizada na elaboração de um projeto. Foi necessário um tempo para familiarização com sua interface, seus blocos de programação e seus recursos específicos.

4.2 Teste do aplicativo

A Figura 17 demonstra os primeiros testes de direcionamento utilizando um controle convencional. Os testes do aplicativo foram todos feitos na APAE(Associação dos Pais e Amigos dos Excepcionais), uma associação não governamental que atende pessoas com Deficiência Intelectual, Síndrome de Down e Transtorno de Espectro do Autismo. A APAE de Pau dos Ferros atende em média 17 alunos, entre 13 e 37 anos.

Primeiramente, foram feitos análises de direcionamento, onde essas crianças e jovens tiveram a oportunidade de controlar o robô passando por cores pré-definidas, como vermelho(guiar o robô para frente), amarelo(guiar para direita), azul(guiar para esquerda) e o verde(guiar para trás). Dessa forma, eles podem melhorar o senso de direcionamento e noções de espaço.

Figura 17 - Jovem com Síndrome de Down treinando com controle convencional.



Fonte: Autoria própria (2025)

Essa noção de espaço e direção, ajuda de forma expressiva na utilização do nosso aplicativo, pois já tendo esse conhecimento mínimo de posicionamento, os usuários poderão montar a trajetória do robô com uma dificuldade reduzida. Ao estimular essas habilidades, os

usuários podem visualizar mentalmente o percurso desejado e planejar melhor a trajetória antes de fazê-lo, ou que está alinhado diretamente com a proposta do aplicativo.

Com um mínimo de noção espacial, essas crianças poderão entender conceitos básicos como frente, trás, esquerda e direita, tornando o uso da interface mais intuitivo. Por exemplo, ao planejar uma rota no aplicativo, elas precisarão prever como o robô se deslocará ao receber cada comando, o que reforçará a importância do raciocínio lógico e da orientação espacial. Esse processo não apenas melhora a interação com o aplicativo, mas também fortalece habilidades cognitivas que podem ser aplicadas em outras áreas do aprendizado e da vida cotidiana.

Após o treinamento inicial com o controle convencional, apresentamos aos alunos a proposta principal do projeto: o controle baseado em lista de comandos. De início, foi claro que eles sentiram certa diferença na interação, uma vez que, no controle convencional, o robô respondia instantaneamente aos comandos conforme os botões eram estreitos. Já na nova abordagem, era necessário planejar toda a trajetória antes de iniciar a execução, o que representava um desafio adicional.

Para tornar o aprendizado mais didático, realizamos um estudo sobre a posição do robô e verificamos que, ao acionar o comando "frente", ele percorria 10 centímetros. Com essa informação, posicionamos um obstáculo a 50 centímetros de distância e questionamos cada aluno sobre quantas vezes o botão "frente" deveria ser acionado para que o robô se aproximasse do obstáculo sem colidir. As respostas variaram: alguns alunos indicaram sete ou nove repetições, enquanto outros pressionaram o botão diversas vezes, acreditando que o robô responderia imediatamente. No entanto, três alunos calcularam corretamente e indicaram cinco repetições, acertando a quantidade exata.

Após esse primeiro desafio, aumentamos o nível de dificuldade ao incluir um novo obstáculo à direita do robô. Dessa forma, após percorrer os 50cm, seria necessário que o robô se deslocasse para a esquerda, já que o caminho à direita estava bloqueado. Nesse momento, a maioria dos alunos demonstrou um desempenho superior, pois ao observarem a barreira, direcionaram corretamente o robô para o lado oposto. Um erro recorrente identificado durante a atividade foi a falta do comando "pare" ao final da sequência, o que fazia com que o robô continuasse se movendo após concluir a trajetória planejada. A sequência correta de comandos para esse cenário deveria ser: Frente, Frente, Frente, Frente, Frente, Pare, Esquerda, Frente, Pare.

4.3 Resultados

Os testes iniciais de direcionamento, realizados em 13 de março de 2025, tiveram como objetivo principal identificar as dificuldades dos usuários em relação à orientação espacial e noção de deslocamento. A análise desses desafios permitiu um desenvolvimento mais preciso do aplicativo, garantindo que ele atendesse de maneira eficaz aos objetivos propostos. Essa fase preliminar, na qual foi utilizado o controle convencional, foi essencial para compreender as principais dificuldades enfrentadas pelos usuários. A partir dessas observações, foi possível ajustar e aprimorar a experiência do aplicativo, tornando-o mais acessível e didático.

No dia 21 de março de 2025, foram realizados os testes finais utilizando o controle desenvolvido para o aplicativo. As avaliações tiveram início às 9h30 da manhã e foram concluídas às 11h30. O procedimento teve início com a apresentação da interface do aplicativo aos usuários, destacando o posicionamento e a funcionalidade de cada botão. Além da explicação individual de cada comando, foi enfatizado o funcionamento da lista de comandos, demonstrando que os usuários deveriam selecionar as instruções desejadas para construir a trajetória que o robô deveria seguir. O botão "Pare" recebeu uma explicação mais detalhada, uma vez que seu uso era fundamental para o correto funcionamento da sequência de ações. Os usuários foram orientados de que, sempre que uma sequência de movimentos fosse programada, o botão "Pare" deveria ser acionado para encerrar o processo antes da inclusão de novos comandos. Por exemplo, caso um usuário adicionasse quatro movimentos para frente e, em seguida, desejasse virar para a esquerda, seria necessário pressionar o botão "Pare" antes de selecionar a nova direção. Esse procedimento foi reforçado para garantir que a execução das ações ocorra de forma ordenada e previsível.

O botão "Seguir" também precisou de uma explicação detalhada, pois representa o elemento central para a execução dos movimentos programados. Como os usuários estavam habituados ao controle convencional, nenhuma ação era respondida de maneira imediata, foi necessário reforçar que, no novo modelo de controle, o robô apenas iniciaria suas posições após a ativação do botão "Seguir". Durante as instruções, foi enfatizado que o objetivo do aplicativo era permitir uma programação antecipada da trajetória, diferentemente do controle convencional. Dessa forma, os usuários compreenderam que, após inserir os comandos na lista, o robô permaneceria parado até que o botão "Seguir" fosse obrigatório, garantindo assim que a sequência planejada fosse realizada de maneira estruturada e previsível.

Figura 18 - Utilização do aplicativo proposto.



Fonte: Autoria própria (2025)

A Figura 18 ilustra os testes na APAE, onde foram utilizadas caixas devidamente posicionadas para servirem de obstáculos. Depois de todos os estudos, constatamos o ótimo funcionamento do aplicativo, precisando apenas de algumas calibrações e reparos no software. Essas atualizações serão mencionadas para trabalhos futuros.

É notável que o uso contínuo desse aplicativo pode trazer uma série de benefícios para os usuários com deficiência, contribuindo significativamente para seu desenvolvimento cognitivo, motor e social. A proposta do aplicativo não se limita apenas ao entretenimento, mas também se destaca como uma ferramenta educativa que auxilia no raciocínio lógico, na cooperação motora e na tomada de decisões. Outro ponto relevante é que o raciocínio lógico aplicado no uso do aplicativo pode ser transferido para outras áreas da aprendizagem, como matemática, leitura e escrita. Ao compreenderem que cada ação tem uma consequência direta, os usuários começam a construir um pensamento mais analítico, o que pode ajudá-los em situações que precisam de planejamento e estratégia.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento deste aplicativo foi um processo desafiador e enriquecedor, resultando em uma ferramenta tecnológica capaz de auxiliar no aprendizado e no desenvolvimento de pessoas com deficiência. Desde a concepção da ideia até a implementação final, cada fase foi planejada cuidadosamente para garantir uma experiência intuitiva e eficaz. A inspiração no *Blockly Maze* proporcionou um modelo interativo que estimula o pensamento lógico, a organização espacial e a autonomia, permitindo que os usuários programem a trajetória do robô antes de sua movimentação.

A realização de testes foi essencial para avaliar a usabilidade e a eficácia do aplicativo. Os testes iniciais, utilizando um controle convencional, foram fundamentais para compreender as dificuldades enfrentadas pelos usuários em relação ao espaço e orientação. Já na fase de testes finais, utilizando uma lista de comandos, foi possível observar como os usuários interagem com a nova metodologia e como sua capacidade de planejamento e tomada de decisões foi aprimorada ao longo do processo. Esses experimentos permitiram ajustes e melhorias que tornaram o software mais acessível e adequado às necessidades do público-alvo.

Além de sua funcionalidade prática, o aplicativo apresenta uma grande relevância pedagógica. Ele pode ser utilizado como um recurso complementar no ensino, auxiliando no desenvolvimento da progressão motora, da noção de sequência e da resolução de problemas. O fato dos usuários precisarem prever a trajetória do robô antes de sua execução incentiva a reflexão e a organização do pensamento, habilidades essenciais para diversas atividades do cotidiano.

Do ponto de vista do desenvolvimento pessoal, o aplicativo promove ganhos significativos na autonomia e na autoconfiança dos usuários. A possibilidade de testar diferentes configurações de comandos, observar os resultados e corrigir possíveis erros proporciona um ambiente de aprendizado experimental, no qual os usuários se sentem motivados a explorar e aprimorar suas habilidades. Além disso, a interação com a tecnologia fortalece a capacidade de adaptação e a inclusão digital, aspectos cada vez mais importantes na sociedade moderna.

Dessa forma, o projeto cumpriu seu propósito de criar um software acessível, educativo e estimulante. A experiência adquirida ao longo desse desenvolvimento reforça a importância de investir em soluções inovadoras para a educação especial, promovendo aprendizado, inclusão e autonomia. Os resultados positivos obtidos indicam que o aplicativo

pode ser um recurso importante para o ensino e a reabilitação, contribuindo para um ambiente educacional mais sonoro, inclusivo e adaptado às necessidades de cada indivíduo.

5.1 Trabalhos futuros

A validação do sistema, realizada com o público-alvo, comprovou a eficiência do nosso aplicativo. Dessa maneira, esse trabalho contribuiu de forma positiva para o avanço da pedagogia e tecnologia nas escolas. Sendo um passo importante para o desenvolvimento de habilidades e uma maior autonomia de escolhas. No entanto, ainda há espaço para melhorias, como:

- Melhorar a calibragem das direções, fazendo com que o giro para esquerda e direita seja perfeitamente de 90 graus.
- Uma análise mais profunda referente a parada em distâncias diferentes, onde pode ser o desgaste do motor ou até mesmo o atrito do local testado.
- Melhorar a interface, deixando ela ainda mais atrativa.
- Adicionar outras funcionalidades que ajudem a facilitar o uso de pessoas com Síndrome de Down.
- Confeção do tabuleiro com caminhos que podem ser alterados. Dessa forma, melhorar o raciocínio dos usuários.
- Acrescentar um comando de voz. Dessa forma, o usuário pode falar as direções e o aplicativo adicionar na lista de comandos.
- Adicionar uma inteligência artificial para ajudar os usuários nas resoluções dos trajetos.

REFERÊNCIAS

ESEMREDE. A importância da robótica educacional na inclusão de jovens e crianças neurodiversos. Disponível em:

<<https://esemrede.com.br/colunista/a-importancia-da-robotica-educacional-na-inclusao-de-jovens-e-criancas-neurodiversos/>>. Acesso em: 09 fev. 2025.

FOCUS. Novas tecnologias na educação inclusiva de crianças com síndrome de Down. Disponível em: <<https://periodicos.faculdaedefocus.com.br/revista-multidisciplinar-focus/articloe/download/41/35/123>>. Acesso em: 09 fev. 2025.

ARAÚJO, Wellington Wagner Oliveira; PALHETA, Paola da Silva; NASCIMENTO, Juliane Nascimento do. Robótica educacional desplugada no ensino infantil: surge como uma alternativa inclusiva e acessível. Disponível em: <https://editorarealize.com.br/editora/anais/conedu/2024/TRABALHO_COMPLETO_EV200_MD1_ID10629_TB6959_01102024153219.pdf>. Acesso em: 11 fev. 2025.

A FUNÇÃO DAS TECNOLOGIAS ASSISTIVAS NA EDUCAÇÃO ESPECIAL: FERRAMENTAS E RECURSOS PARA A APRENDIZAGEM. Disponível em: <file:///C:/Users/hercu/Downloads/[170]-A+FUN%C3%87%C3%83O+DAS+TECNOLOGIAS+ASSISTIVAS+NA+EDUCA%C3%87%C3%83O+ESPECIAL-+FERRAMENTAS+E+RECURSOS+PARA+A+APRENDIZAGEM.pdf>. Acesso em: 14 fev. 2025.

SANTOS, Larissa Medeiros Marinho dos. A robótica educacional como ferramenta multidisciplinar: um estudo de caso para a formação e inclusão de pessoas com deficiência. Disponível em: <https://www.academia.edu/55696719/A_rob%C3%B3tica_educacional_como_ferramenta_multidisciplinar_um_estudo_de_caso_para_a_forma%C3%A7%C3%A3o_e_inclus%C3%A3o_de_pessoas_com_defici%C3%Aancia>. Acesso em: 18 fev. 2025.

USP São Carlos. Robótica na educação infantil. Projeto de robótica educacional para crianças em idade pré-escolar. Disponível em: <<https://www.youtube.com/watch?v=YXfYEhtxi0E>>. Acesso em: 18 fev. 2025.

MIT App Inventor. Tutorial App Inventor: Criando o primeiro projeto. Disponível em: <<https://www.youtube.com/watch?v=itT8fZkd5IE>>. Acesso em: 20 fev. 2025.

JOGOS DO BLOCKLY. Disponível em: <<https://blockly.games/maze?lang=pt-br&level=5&skin=0>>. Acesso em: 08 fev. 2025.

APAE. Disponível em: <https://www.apaefranca.org.br/blog/vem-ai-35-festa-di-san-gennaro?gad_source=1&gclid=Cj0KCQjwv_m-BhC4ARIsAlqNeBuikt0NwqFr9QwvYVggN74otcFVoMW6heArON-FbOGUUV6CRho-4waAgyXEALw_wcB>. Acesso em: 23 fev. 2025.

O PEDAGOGO E O DESENVOLVIMENTO DE CRIANÇAS COM SÍNDROME DE DOWN NA EDUCAÇÃO INFANTIL. Disponível em:

<<https://repositorio.ucs.br/xmlui/bitstream/handle/11338/4932/TCC%20Scheila%20Severo%20Melo%20Amaral.pdf?sequence=1&isAllowed=y>>. Acesso em: 12 mar. 2025.

Mit App Inventor: desenvolvendo aplicativos para smartphone. Disponível em:

<https://www.makerhero.com/blog/mit-app-inventor-desenvolvendo-aplicativos-para-smartphone/?srsltid=AfmBOoq4gASVD7CaOdO6h_9mJ04pGf2fC8xsVmoR7a8Tq_ZW7PbPLs9H>.

Acesso em: 22 fev. 2025.

Psicología cognitiva no design UX: como minimizar a Carga Cognitiva na Experiência do Usuário. Disponível em:

<https://psicologia.design/psicologia-cognitiva-no-design-ux-como-minimizar-a-carga-cognitiva-na-experiencia-do-usuario/?utm_source=chatgpt.com>. Acesso em: 02 de mar.2025