RoboLearn: Um protótipo de aplicativo móvel voltado ao ensino de robótica

Bernardo Gomes Nocêra[1], **Luciane T. W. Agner**[1]

[1]Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro Oeste (UNICENTRO)
Alameda Élio Antonio Dalla Vecchia, 838 – Vila Carli – CEP 85040-167
Guarapuava – PR – Brasil

bernardo.nocera1@gmail.com, lagner@unicentro.br

Abstract. *This paper presents the development of a mobile application prototype called RoboLearn, aimed at teaching robotics to beginners. The application integrates theoretical content, interactive simulations, and gamified activities through an intuitive interface developed with Flutter. The prototype includes features such as progress tracking, component dictionary and interactive quizzes. The main objective is to make access to robotics education more accessible, overcoming barriers such as lack of physical resources and accessible educational materials. The results demonstrate the feasibility of using mobile technologies to support robotics learning in an engaging and structured way.*

Resumo. *Este artigo apresenta o desenvolvimento de um protótipo de aplicativo móvel denominado RoboLearn, voltado ao ensino de robótica para iniciantes. O aplicativo integra conteúdos teóricos, simulações interativas e atividades gamificadas através de uma interface intuitiva desenvolvida com Flutter. O protótipo inclui funcionalidades como acompanhamento de progresso, dicionário de componentes, quizzes interativos e videoaulas. O objetivo principal é tornar o acesso ao ensino de robótica mais acessível, superando barreiras como a falta de recursos físicos e materiais didáticos. O resultado oferece a possibilidade de uso de tecnologias móveis para apoiar o aprendizado de robótica de forma envolvente e atrativa.*

1. Introdução

Este Trabalho de Conclusão de Curso (TCC) desenvolveu um aplicativo cujo nome é RoboLearn. Seguiu uma abordagem iterativa e centrada no usuário, combinando princípios de *design* de interação humano-computador. Durante o desenvolvimento, utilizou-se a ferramenta Trello[1] para organizar e acompanhar as tarefas de forma visual, baseando-se no método Kanban, que consiste na organização de atividades em colunas que representam diferentes estágios do fluxo de trabalho (como "A fazer", "Em andamento" e "Concluído"), permitindo melhor controle e visualização do progresso do projeto.

1.1. Contextualização

A sociedade contemporânea está cada vez mais permeada por tecnologias digitais, e o domínio de conhecimentos nas áreas de ciência, tecnologia, engenharia e matemática (do

[1]https://trello.com/

inglês *Science, Technology, Engineering and Mathematics* - STEM) tornou-se essencial para a formação de cidadãos críticos e preparados para os desafios do século XXI. Nesse contexto, a robótica educacional pode ser utilizada como uma abordagem pedagógica inovadora, capaz de promover o aprendizado ativo, interdisciplinar e significativo.

A robótica, ao integrar conceitos de eletrônica, programação e lógica matemática, permite que os estudantes experimentem a teoria na prática, desenvolvendo habilidades técnicas e cognitivas de maneira concreta. Além disso, ao lidar com montagem, testes e correção de projetos, os alunos são estimulados a exercitar criatividade, resiliência, resolução de problemas e trabalho em equipe [Santos, 2024].

No entanto, apesar de seus benefícios, o ensino de robótica ainda enfrenta barreiras, especialmente no contexto educacional brasileiro. Muitos jovens estudantes, particularmente aqueles em idade escolar (12-18 anos), têm contato limitado com essa área devido à falta de recursos físicos, como *kits* de robótica, e à ausência de materiais didáticos acessíveis, estruturados e adaptados para iniciantes. Este público-alvo específico foi escolhido por representar a faixa etária mais receptiva a tecnologias móveis e com maior potencial de desenvolvimento de competências STEM fundamentais para sua formação acadêmica e profissional futura [Zem and Borges, 2017].

Paralelamente, observa-se o crescente uso de aplicativos móveis como ferramenta de apoio à aprendizagem. Sua popularidade entre os jovens, aliada à onipresença dos *smartphones*, torna os dispositivos móveis plataformas promissoras para a disseminação de conteúdos educacionais. O uso de interfaces amigáveis, elementos interativos e recursos de gamificação pode potencializar o interesse dos estudantes por temas como robótica e tecnologia [Gumennykova et al., 2019].

Nesse cenário, o desenvolvimento de um aplicativo móvel voltado ao ensino introdutório de robótica se apresenta como uma alternativa viável para incentivar o acesso a esse conhecimento. Ao unir teoria, simulações práticas e atividades interativas em uma única plataforma, busca-se facilitar a aprendizagem de conteúdos técnicos de maneira acessível, dinâmica e lúdica.

1.2. Justificativa

A carência de soluções educacionais que combinem acessibilidade, interatividade e qualidade no ensino de robótica motivou a criação do presente trabalho. A maioria das ferramentas similares disponíveis está voltada a públicos que já possuem familiaridade com programação ou eletrônica, o que representa uma barreira significativa para estudantes iniciantes [Lima and Borges, 2020].

Além disso, embora existam ferramentas para auxiliar no aprendizado de programação como simuladores tal qual o Tinkercad[2] e ambientes de programação visual como o Scratch[3] e o mBlock[4], essas não são projetadas especificamente para dispositivos móveis. Frequentemente, carecem de integração com uma proposta pedagógica auxiliar, o que limita sua aplicabilidade em contextos escolares e na aprendizagem autônoma.

O desenvolvimento de um aplicativo como o RoboLearn, com foco na simplici-

[2]https://www.tinkercad.com/

[3]https://scratch.mit.edu/

[4]https://mblock.cc/

dade de uso, *design* acessível e metodologia baseada em gamificação, pode preencher uma lacuna no ensino de robótica para iniciantes. A proposta visa não apenas apoiar a compreensão de conceitos básicos, mas também fomentar o interesse pela área tecnológica, podendo contribuir para a formação de estudantes mais preparados para o futuro.

1.3. Objetivos

1.3.1. Objetivo Geral

O principal objetivo deste TCC foi desenvolver um protótipo de aplicativo móvel voltado ao ensino de robótica, com foco na apresentação de conteúdos teóricos, simulações práticas e atividades interativas, de forma acessível e atrativa para iniciantes.

1.3.2. Objetivos Específicos

- Uma ferramenta educacional para auxiliar no ensino de conceitos fundamentais de robótica foi proposta.
- Conteúdos didáticos sobre componentes como sensores, atuadores e controladores foram criados.
- Uma interface interativa baseada em princípios de IHC e gamificação foi projetada.
- *Quizzes* e desafios para reforçar o aprendizado foram criados.
- Novos conhecimentos em ferramentas modernas de desenvolvimento multiplataforma, como Flutter e Firebase, foram adquiridos.

2. Fundamentação Teórica

2.1. Robótica Educacional e sua Importância

A robótica educacional tem se consolidado como uma ferramenta eficaz no processo de ensino-aprendizagem, especialmente por seu caráter interdisciplinar e seu potencial de engajamento dos estudantes. Principalmente através de plataformas digitais, os alunos são incentivados a aplicar conhecimentos de áreas como matemática, física e programação de forma prática, descontraída e concreta [Silva and Costa, 2023].

O principal diferencial da robótica no contexto educacional está na capacidade de transformar o aluno em protagonista do seu aprendizado. Ao montar, programar e testar seus próprios projetos, o estudante desenvolve não apenas competências técnicas, mas também habilidades como trabalho em equipe, criatividade, resolução de problemas e persistência diante de falhas [Santos and Lima, 2022].

Além disso, a robótica estimula uma aprendizagem ativa e significativa. Ver um protótipo funcionando a partir de um código escrito pelo próprio aluno torna o conteúdo mais tangível e desperta o interesse pelo estudo. Isso contribui para tornar o processo educacional mais dinâmico e conectado à realidade tecnológica em que vivemos [Oliveira and Ferreira, 2023].

Outro ponto importante é o seu papel na inclusão digital. Ao ser adotada em escolas públicas e privadas, a robótica educacional pode democratizar o acesso à tecnologia

e preparar os estudantes para os desafios do século XXI, inclusive no mercado de trabalho, que cada vez mais valoriza competências relacionadas à STEM [Gumennykova et al., 2019].

Desde os anos 1980, o potencial transformador da tecnologia no ambiente escolar, ao defender a aprendizagem por meio da experimentação e da construção ativa do conhecimento [Papert, 1980]. Valente (2003) complementa essa visão ao argumentar que o computador pode ser um instrumento essencial na construção do conhecimento.

2.2. Tecnologias Utilizadas no Desenvolvimento

O desenvolvimento do protótipo proposto envolveu o uso de diversas ferramentas e tecnologias que contribuíram para garantir a qualidade, usabilidade e portabilidade do produto final. Essas ferramentas serão descritas a seguir.

2.2.1. Flutter

O Flutter foi utilizado como *framework* de desenvolvimento multiplataforma, permitindo a criação de uma única base de código para aplicações Android e iOS. Sua estrutura baseada em *widgets* facilita a construção de interfaces responsivas e personalizadas Payne (2019).

2.2.2. Dart

A linguagem de programação Dart, nativamente suportada pelo Flutter, foi empregada na implementação da lógica do aplicativo. Com uma sintaxe moderna e recursos orientados a objetos, Dart proporciona produtividade e desempenho durante o desenvolvimento Payne (2019).

2.2.3. Figma

O Figma[5] foi utilizado nas etapas de *design* de interface e prototipagem. Por ser uma ferramenta colaborativa baseada na nuvem, permitiu a criação de *wireframes* e simulações que seguem os princípios de Interação Humano-Computador (IHC), com foco em usabilidade e acessibilidade.

2.2.4. Firebase

O Firebase[6], que é uma plataforma de serviços em nuvem do Google, foi considerado para o gerenciamento de dados e autenticação de usuários. Sua integração com *Flutter* permite a sincronização eficiente de dados e funcionalidades em tempo real.

[5]https://www.figma.com/
[6]https://g.co/kgs/q3Tf4pH

2.2.5. Visual Studio Code

O Visual Studio Code[7] foi o ambiente de desenvolvimento adotado. Suas extensões específicas para Flutter e Dart, aliadas a recursos de depuração e controle de versão, tornaram o processo de codificação mais ágil e organizado.

2.2.6. ChatGPT

O ChatGPT[8], modelo de linguagem desenvolvido pela *OpenAI*, foi utilizado como ferramenta de apoio para correções ortográficas e sugestões de aprimoramento gramatical durante a redação deste trabalho. Ressalta-se que a produção de conteúdo original e a análise crítica permaneceram de responsabilidade total do autor.

2.2.7. GitHub

O GitHub[9] foi utilizado para o controle de versionamento e colaboração no desenvolvimento do projeto. Através de repositórios e *branches*, foi possível manter o histórico de alterações e realizar testes de forma controlada.

2.2.8. Tinkercad

O Tinkercad foi usado como simulador de circuitos, auxiliando na criação de exemplos práticos relacionados à robótica. Sua interface intuitiva permite simular sensores, atuadores e controladores, facilitando a compreensão dos conceitos pelos usuários do aplicativo Diyarova et al. (2024).

2.3. Interação Humano-Computador e Design de Interfaces

O desenvolvimento de aplicações educacionais requer especial atenção aos princípios de Interação Humano-Computador (IHC), especialmente quando o público-alvo são iniciantes em tecnologia. Rogers et al. (2011) destacam que o *design* centrado no usuário é fundamental para criar interfaces eficazes e intuitivas.

No contexto de aplicações móveis educacionais, Nielsen (1994) estabelece heurísticas de usabilidade que foram consideradas no desenvolvimento do RoboLearn:

- **Visibilidade do status do sistema**: Manter os usuários informados sobre o progresso nas atividades.
- **Correspondência entre sistema e mundo real**: Usar linguagem familiar ao público-alvo.
- **Controle e liberdade do usuário**: Permitir navegação livre entre os conteúdos.
- **Consistência e padrões**: Manter padrões visuais em todas as telas.
- **Prevenção de erros**: *Design* que minimiza possibilidades de erro.
- **Flexibilidade e eficiência de uso**: Interface adaptável a diferentes níveis de experiência.

[7] https://code.visualstudio.com/
[8] https://chatgpt.com/
[9] https://github.com/

2.4. Gamificação no Contexto Educacional

A gamificação, definida por Deterding et al. (2011) como ”o uso de elementos de *design* de jogos em contextos não-lúdicos”, tem se mostrado uma estratégia eficaz para aumentar o engajamento em aplicações educacionais. Kapp (2012) identifica elementos-chave da gamificação aplicáveis ao contexto educacional:

- **Pontuação**: Sistema de recompensas que motiva a participação.
- **Níveis e progressão**: Estrutura hierárquica que orienta o aprendizado.
- **Desafios**: Tarefas que estimulam o desenvolvimento de habilidades.
- **Feedback imediato**: Respostas rápidas às ações do usuário.
- **Narrativa**: Contexto que dá significado às atividades.

No ensino de robótica, a gamificação pode ser particularmente eficaz por transformar conceitos abstratos em desafios concretos e mensuráveis, aumentando a motivação dos estudantes para explorar tópicos técnicos complexos.

2.5. Design Responsivo e Desenvolvimento Mobile

O desenvolvimento para dispositivos móveis apresenta desafios únicos relacionados à diversidade de tamanhos de tela, capacidades de processamento e formas de interação. Prensky (2001) destaca que os ”nativos digitais” esperam interfaces intuitivas e responsivas, adaptadas aos padrões de uso de dispositivos móveis. O *design* responsivo em aplicações educacionais considerados no desenvolvimento desse projeto foram:

- **Adaptabilidade**: *Layouts* que se ajustam a diferentes resoluções.
- **Acessibilidade**: Interfaces usáveis por pessoas com diferentes habilidades.
- **Performance**: Otimização para dispositivos com recursos limitados.

2.6. Pensamento Computacional na Educação

Segundo Wing (2006) o pensamento computacional pode ser definido como ”o processo de resolução de problemas que inclui decomposição, reconhecimento de padrões, abstração e algoritmos”. No contexto da robótica educacional, esses conceitos são fundamentais para o desenvolvimento de habilidades cognitivas dos estudantes. O RoboLearn buscou incorporar alguns dos pilares deste pensamento através da decomposição de problemas complexos em etapas menores, do reconhecimento de padrões nos componentes e circuitos eletrônicos, da abstração dos conceitos essenciais buscando facilitar a compreensão. O processo de depuração é integrado pelo sistema de *feedback* imediato, permitindo que os usuários identifiquem e corrijam as atividades logo ao submeter sua resposta.

3. Desenvolvimento

3.1. Casos de Uso do Sistema

O Diagrama de Casos de Uso é uma representação gráfica que define como os usuários (ou atores) interagem com um sistema, como um aplicativo de *software* e constitui uma ferramenta fundamental da UML (do inglês *Unified Modeling Language*) para modelagem de sistemas orientados a objetos. Segundo Sommerville (2011), os casos de uso descrevem as interações entre os usuários e o sistema, especificando as funcionalidades que devem ser implementadas do ponto de vista do usuário final.

A definição dos casos de uso foi essencial para identificar as funcionalidades que o RoboLearn deve oferecer ao usuário final. A Figura 1 apresenta o Diagrama de Casos de Uso, onde o ator ”Usuário” interage com os principais recursos do sistema, como cadastro, *login*, edição de perfil, acesso às aulas, realização de atividades interativas, consulta ao dicionário de componentes e acesso às configurações, que possibilita ao usuário se desconectar da conta atual, entre outras funcionalidades.

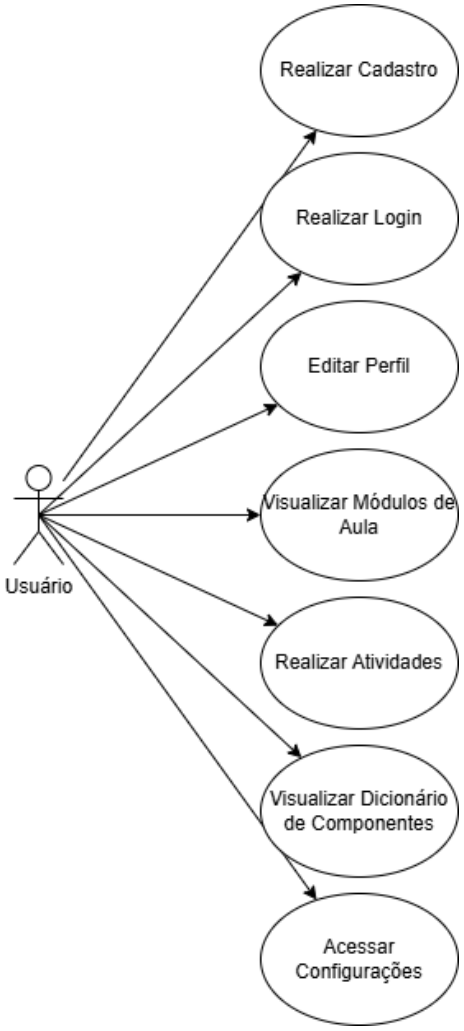


Figura 1. Diagrama de casos de uso do aplicativo RoboLearn.

3.2. Fluxograma

A Figura 2 apresenta o fluxograma do protótipo proposto, mostrando a arquitetura de navegação e o fluxo de interação do usuário com o sistema. Esse diagrama ilustra o processo iniciado pela verificação de autenticação do usuário, seguindo por duas rotas principais: o cadastro de novos usuários ou o *login* para usuários já existentes.

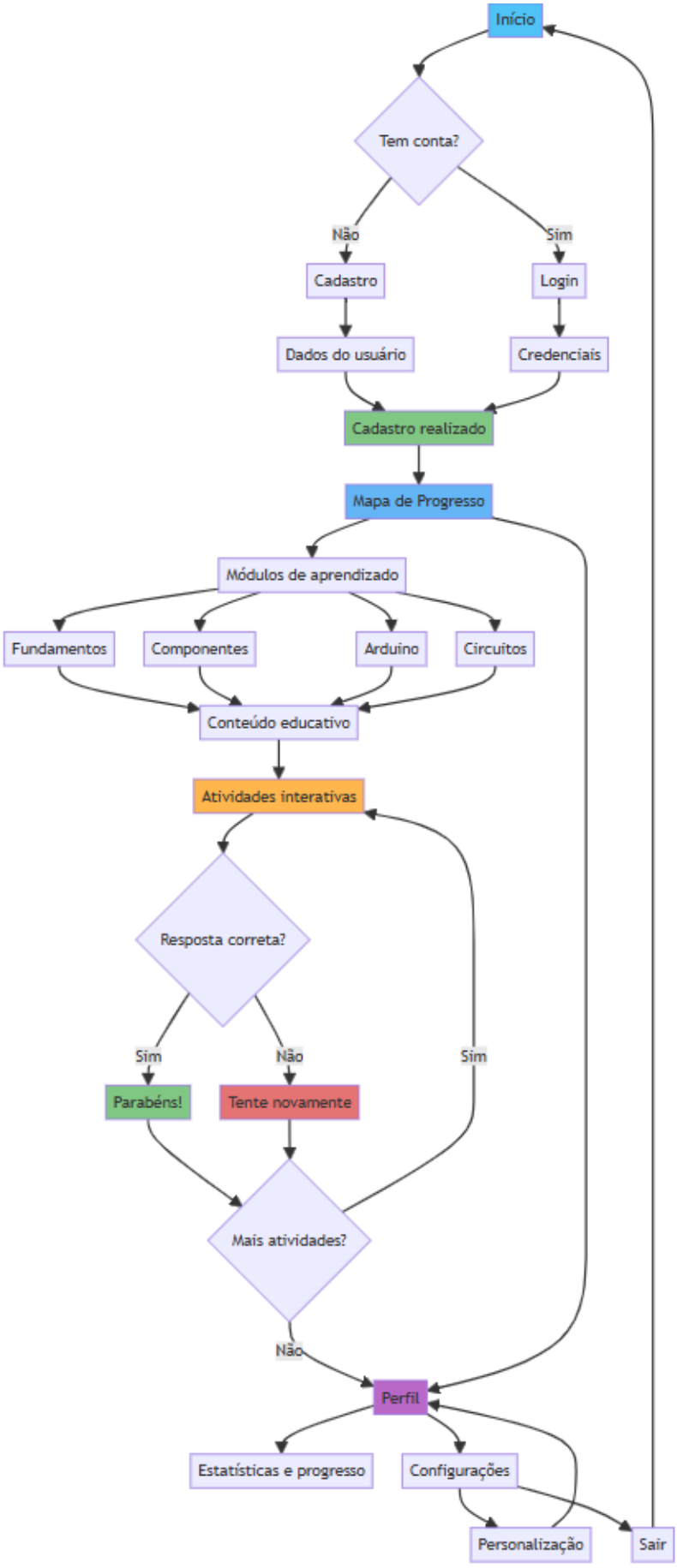


Figura 2. Fluxograma do aplicativo RoboLearn.

Conforme apresentado na Figura 2, após a autenticação bem-sucedida, o sistema direciona o usuário para o Mapa de Progresso, que constitui a interface principal da aplicação. Esta tela centraliza o acesso aos quatro módulos educacionais fundamentais: Fundamentos da Eletrônica, Componentes Eletrônicos, Simulação com Arduino e Circuitos. Cada módulo oferece conteúdo educativo estruturado seguindo alguns princípios de gamificação, como pontuação em forma de experiência (XP - *Experience Points*) que é obtida após realizar uma atividade, os desafios através de atividades de múltipla escolha e *feedback* imediato com uma mensagem caso acerte ou erre a questão, similares aos implementados em plataformas como Duolingo[10].

O fluxo de aprendizagem é caracterizado por atividades interativas que implementam um sistema de *feedback* imediato. Quando o usuário responde corretamente às questões, recebe reforço positivo através da mensagem "Parabéns!", enquanto respostas incorretas geram o *feedback* "Tente novamente", promovendo assim a aprendizagem por tentativa e erro. Este mecanismo de avaliação contínua permite que o usuário repita atividades até alcançar o domínio do conteúdo.

O sistema incorpora ainda funcionalidades de acompanhamento de progresso através do módulo Perfil, onde são apresentadas estatísticas detalhadas do desempenho do usuário. Por sua vez, as configurações permitem a personalização da experiência de uso, incluindo ajustes de interface e preferências individuais. O fluxograma demonstra a natureza cíclica do processo de aprendizagem, onde o usuário pode navegar entre diferentes módulos e repetir atividades conforme necessário.

A arquitetura apresentada evidencia a implementação de princípios da gamificação, fundamentais para manter o engajamento do usuário em ambientes de aprendizagem móvel voltados ao ensino de eletrônica e programação com Arduino.

3.3. Banco de Dados

O sistema de gerenciamento de dados do aplicativo RoboLearn foi implementado utilizando o Firebase, como citado anteriormente. A sua escolha é justificada devido à facilidade de integração com o Flutter através de poucas linhas de comando apresentadas em sua própria documentação detalhadamente, escalabilidade automática e suporte nativo para aplicações móveis multiplataforma. A estrutura do banco de dados foi projetada seguindo um modelo NoSQL baseado em documentos, que permite maior flexibilidade no armazenamento de dados comparado aos bancos relacionais tradicionais. Esta escolha é particularmente adequada para aplicações educacionais móveis voltadas ao público jovem, pois permite a rápida adaptação da estrutura de dados conforme novas funcionalidades são implementadas, além de oferecer melhor performance em consultas frequentes de pequeno volume, que são características essenciais para manter o engajamento dos usuários jovens que esperam respostas instantâneas. A natureza não-relacional também facilita o armazenamento de conteúdo educacional como texto, imagens e dados de progresso, sem a necessidade de esquemas rígidos. As principais coleções implementadas incluem:

Usuários: Armazena informações pessoais dos estudantes, incluindo nome, *email*, senha criptografada, data de cadastro e configurações de perfil. Cada usuário possui

[10]https://www.duolingo.com/

ainda subcoleções para controle de progresso individual e histórico de atividades realizadas.

Componentes: Funciona como um dicionário digital com informações detalhadas sobre componentes eletrônicos utilizados em robótica, incluindo descrições técnicas, aplicações práticas, especificações e imagens de referência.

Progresso: Registra o avanço individual de cada usuário através dos módulos por meio da pontuação obtida.

A segurança dos dados apresentados é garantida através das *Firebase Security Rules*, regras de acesso configuráveis da plataforma que asseguram que usuários só possam acessar e modificar suas próprias informações com base em sua autenticação. Além disso, todas as comunicações entre o aplicativo e o banco de dados são automaticamente criptografadas pelo Firebase através de conexões HTTPS, atendendo aos padrões de segurança exigidos.

3.4. Implementação e Telas

Nesta seção serão apresentadas e descritas as principais telas do projeto, descrevendo suas variações e interações, de forma a exemplificar a experiência do usuário. Também serão abordadas as funcionalidades mais relevantes implementadas, explicando como cada uma contribui para o funcionamento do aplicativo e sua integração com o sistema.

3.4.1. Configuração e Integração com Firebase

Para integrar o banco de dados ao aplicativo, foi necessário instalar alguns pacotes e dependências como o *FlutterFire*. Este conjunto de *plugins* conecta os serviços do Firebase aos aplicativos Flutter, oferecendo funcionalidades como autenticação de usuários, banco de dados em tempo real e armazenamento de arquivos. A principal integração do Firebase no projeto é realizada conforme é mostrado no Código 1 que é adicionado ao arquivo *main* (principal) do projeto.

Listing 1. Inicialização do Firebase e execução do aplicativo RoboLearn

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const RoboLearnApp());
}
```

Esta configuração realiza três etapas essenciais: primeiro, "*WidgetsFlutterBinding.ensureInitialized()*" garante que o sistema de *widgets* do Flutter esteja pronto antes de executar operações assíncronas; em seguida, "*Firebase.initializeApp()*" conecta o aplicativo aos serviços do Firebase usando as configurações específicas da plataforma; por fim, "*runApp()*" inicializa a interface do usuário. O termo "*async/await*" assegura que o Firebase seja completamente configurado antes do carregamento da aplicação para evitar possíveis erros.

3.4.2. Sistema de Roteamento e Navegação

A navegação entre as diferentes telas do aplicativo foi implementada utilizando o sistema de rotas nomeadas do Flutter, que oferece uma maior organização, além de facilitar a manutenção do código. O mapeamento das rotas foi definido conforme apresentado no Código 2.

Listing 2. Definição das rotas nomeadas utilizadas no aplicativo Flutter

```
  initialRoute: '/',
  routes: {
    '/': (context) => LoginPage(),
    '/home': (context) => HomePage(),
    '/register': (context) => RegisterPage(),
    '/dictionary': (context) => DictionaryPage(),
    '/component': (context) => ComponentPage(),
    '/class': (context) => ClassPage(),
    '/exercise': (context) => ExercisePage(),
    '/profile': (context) => ProfilePage(),
    '/config': (context) => ConfigPage()
  },
```

Cada rota corresponde a uma tela específica da aplicação: a rota inicial ('/') direciona para a tela de *login*, '*/home*' para o painel principal com os módulos educacionais, '*/dictionary*' para o dicionário de componentes, '*/exercise*' para as atividades interativas, '*/profile*' para o perfil do usuário e '*/config*' para as configurações.

3.4.3. Sistema de Registro de Usuário

O sistema de registro de novos usuários foi implementado seguindo boas práticas de validação e de experiência do usuário. A funcionalidade utiliza controladores de texto para capturar e gerenciar os dados de entrada do formulário, incluindo nome, nome de usuário, *e-mail*, senha e uma confirmação de senha como precaução para garantir que a senha foi digitada corretamente. O sistema incorpora validações no lado do cliente (*client-side*) para garantir a integridade dos dados antes de enviá-los ao serviço Firebase Authentication, conforme exemplificado no Código 3.

Listing 3. Trecho da implementação do sistema de registro de usuário com validação de senhas

```
final TextEditingController nameController =
    TextEditingController();
final TextEditingController userController =
    TextEditingController();
final TextEditingController emailController =
    TextEditingController();
final TextEditingController passwordController =
    TextEditingController();
final TextEditingController confirmPasswordController =
    TextEditingController();

bool _isLoading = false;
```

```
void registerUser() async {
  if (passwordController.text != confirmPasswordController.text) {
    _showErrorDialog('As senhas nao coincidem.');
    return;
  }
}
```

É utilizado de "*TextEditingController*" para cada campo do formulário, com o intuito de permitir acesso aos valores inseridos pelo usuário. A variável "*isLoading*" controla o estado da interface durante o processo de registro, desabilitando botões e exibindo indicadores visuais de carregamento para melhorar a experiência do usuário. A validação inicial verifica se as senhas coincidem antes de prosseguir com a criação da conta, sendo a camada de segurança que previne erros comuns de digitação. Em caso de falha na validação, o método "*showErrorDialog*" exibe uma mensagem de erro clara e contextual ao usuário, seguindo princípios de usabilidade que facilitam a correção de problemas no formulário.

3.4.4. Tela de Login e Cadastro

A tela apresentada na Figura 5 é a primeira que o usuário verá ao acessar o aplicativo. Nela, existem dois campos de texto a serem preenchidos: um para o nome do usuário e outro para a senha. Além disso, há dois botões: o primeiro tem a função de realizar o *login*, enquanto o segundo direciona o usuário para a tela de Cadastro. Esta última tela é semelhante à de *login*, mas com a adição de dois campos extras: um para o *e-mail* e outro para a confirmação da senha. Após o preenchimento dos campos, o usuário pode ser redirecionado de volta para a tela de *login* por meio dos botões localizados na parte inferior.



Figura 3. *
(a) Protótipo da tela de Login (Figma)

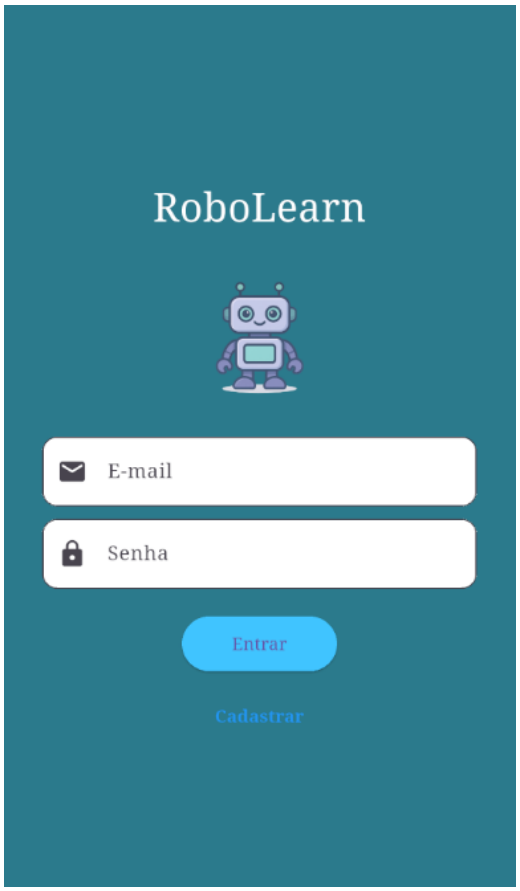


Figura 4. *
(b) Tela final de Login no app

Figura 5. Comparação entre o protótipo e a tela final de Login.

3.4.5. Tela Inicial

A tela inicial, ilustrada na Figura 8, é exibida após a autenticação do usuário e funciona como um painel principal de navegação dentro do aplicativo. Nela, o usuário se depara com uma grade de botões modulares que direcionam para diferentes áreas do conteúdo, como aulas, dicionário de componentes e demais funcionalidades. Cada módulo é representado por um ícone distinto, seguindo uma estética visual lúdica, acessível e relacionada ao conteúdo abordado. Na parte inferior, há uma barra de navegação fixa, presente também nas demais telas, com atalhos rápidos para as principais seções do aplicativo: a própria tela inicial, a tela de atividades, a tela de perfil e, por fim, a de configurações, o que favorece a usabilidade e o acesso fluido aos conteúdos.



Figura 6. *
(a) Protótipo da tela inicial (Figma)



Figura 7. *
(b) Tela final inicial do app

Figura 8. Comparação entre o protótipo e a tela final inicial.

3.4.6. Tela de Dicionário de Componentes

A tela de Dicionário de Componentes, apresentada na Figura 11, permite ao usuário explorar uma lista organizada de componentes eletrônicos utilizados em projetos de robótica. Os componentes são exibidos como cartões contendo ícone, nome e, ao toque, levam a uma nova tela com uma descrição completa, imagem ampliada, aplicações e curiosidades. Um campo de busca na parte superior facilita a localização dos itens desejados. A categorização visual, com ícones representativos, visa auxiliar na identificação rápida dos componentes por estudantes iniciantes.

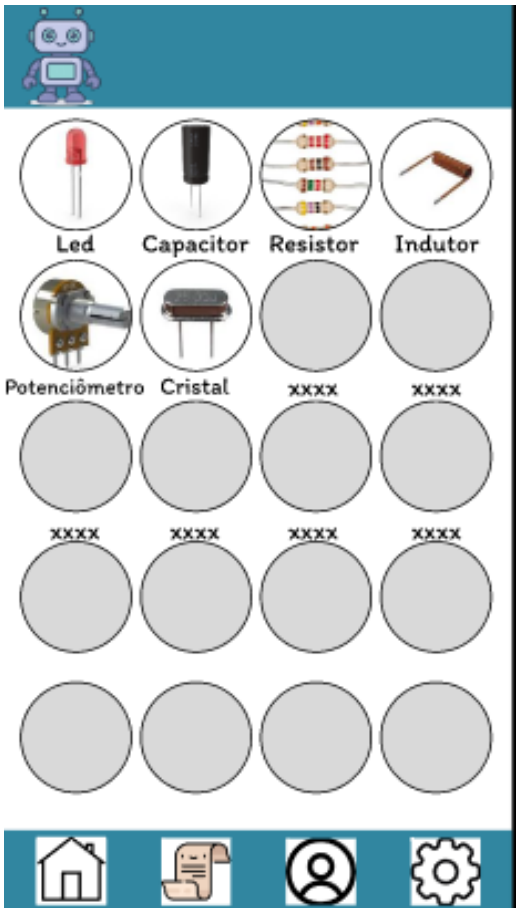


Figura 9. *
(a) Protótipo da tela de Dicionário (Figma)



Figura 10. *
(b) Tela final de Dicionário no app

Figura 11. Comparação entre o protótipo e a tela final de Dicionário.

3.4.7. Tela de Atividade

A tela de Atividade, ilustrada na Figura 14, é responsável por apresentar os desafios interativos relacionados aos conteúdos vistos nas aulas. São disponibilizadas questões em formatos variados, como múltipla escolha, verdadeiro ou falso e associação de imagens. O sistema fornece *feedback* imediato após cada resposta, informando se o usuário acertou ou errou e explicando a justificativa correta. Há ainda uma barra de progresso no topo da tela indicando o avanço na atividade. Ao final da sequência de perguntas, o usuário recebe uma pontuação e uma mensagem de incentivo com base no seu desempenho.



Figura 12. *
(a) Protótipo da tela de Atividade (Figma)

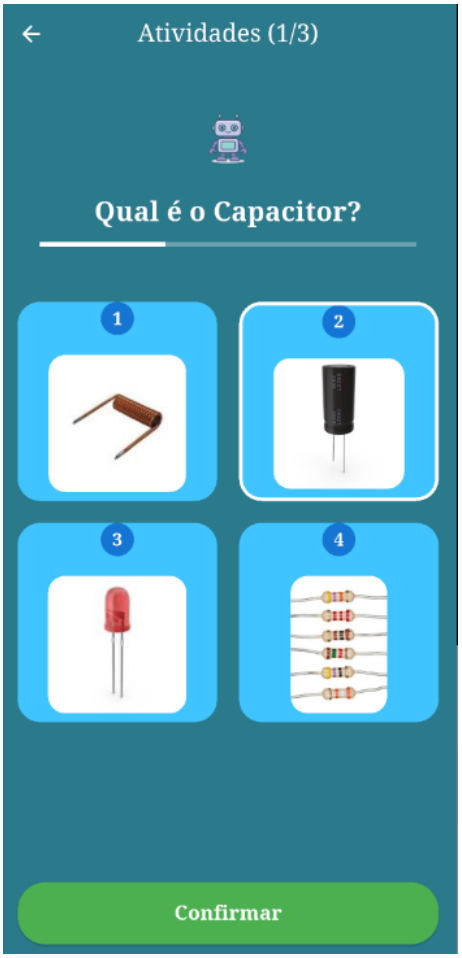


Figura 13. *
(b) Tela final de Atividade no app

Figura 14. Comparação entre o protótipo e a tela final de Atividade.

3.4.8. Tela de Perfil

A tela de Perfil, mostrada na Figura 17, apresenta ao usuário um resumo do seu progresso no aplicativo. Nela, é possível visualizar o nome de usuário, nível de XP, total de aulas concluídas, atividades realizadas e futuramente conquistas desbloqueadas. Também há opções para edição do perfil, como troca de senha ou nome, além do botão para encerrar a sessão e retornar à tela de *login*. O *layout* da tela é organizado em seções com ícones visuais que indicam estatísticas e opções de personalização, reforçando o aspecto gamificado do aplicativo.



Figura 15. *
(a) Protótipo da tela de Perfil (Figma)



Figura 16. *
(b) Tela final de Perfil no app

Figura 17. Comparação entre o protótipo e a tela final de Perfil.

4. Trabalhos Correlatos

Diversos estudos e iniciativas têm explorado o uso da tecnologia para apoiar o ensino de robótica, principalmente por meio de plataformas digitais, simulações e abordagens gamificadas. Nesta seção, são apresentados trabalhos relacionados que se aproximam do objetivo deste projeto, assim como seus diferenciais em relação à proposta do aplicativo RoboLearn.

O aplicativo ArduinoDroid, por exemplo, permite a compilação e envio de códigos para placas Arduino diretamente de dispositivos móveis, mas não apresenta suporte pedagógico, nem conteúdo teórico estruturado para iniciantes. Já o simulador Tinkercad oferece recursos robustos de montagem de circuitos e simulação virtual, sendo amplamente utilizado no ensino informal de eletrônica. No entanto, sua interface é voltada a usuários com conhecimento prévio e não possui recursos de gamificação ou curadoria pedagógica Diyarova et al. (2024).

Plataformas como o Scratch e o mBlock apresentam ambientes visuais voltados ao ensino de lógica e programação com recursos de robótica, principalmente no contexto da educação básica. Apesar de possuírem linguagem acessível e serem amplamente utilizadas, não são otimizadas para dispositivos móveis e exigem a instalação de *softwares* específicos, limitando sua acessibilidade.

Um estudo conduzido por Gumennykova et al. (2019) destaca a importância da gamificação no ensino de robótica, evidenciando que alunos expostos a experiências lúdicas e interativas demonstram maior engajamento e retenção de conteúdo. No entanto, os autores ressaltam que há uma carência de aplicativos móveis que integrem teoria, prática e desafios interativos em um único ambiente digital.

Diante disso, o aplicativo RoboLearn busca se destacar ao combinar conteúdo didático estruturado, ilustrações explicativas, atividades gamificadas e simulações com uma interface intuitiva, acessível e responsiva. Seu diferencial está na integração entre teoria e prática, com navegação simples e proposta modular, ideal para iniciantes no universo da robótica.

5. Resultados

Como principal resultado deste trabalho foi criado um protótipo denominado RoboLearn que poderá auxiliar no ensino de robótica para estudantes iniciantes, integrando conteúdo teórico estruturado, atividades práticas simuladas e elementos de gamificação em uma única plataforma móvel.

O protótipo implementado apresenta quatro módulos educacionais principais: Fundamentos da Eletrônica, Componentes Eletrônicos, Simulação com Arduino e Circuitos. Cada módulo foi estruturado seguindo princípios de *microlearning*, que consiste na divisão do conteúdo em pequenas unidades de aprendizagem que podem ser consumidas rapidamente, permitindo que os usuários avancem gradualmente através de lições curtas e objetivas, com *feedback* imediato sobre seu desempenho.

Dentre as principais vantagens do protótipo proposto é possível citar: Acessibilidade, eliminando a necessidade de equipamentos especializados ou *kits* de robótica físicos; Interface intuitiva, desenvolvida seguindo princípios de Interação Humano-Computador; Aprendizagem gamificada, incorporando elementos como sistema de pontuação, com material didático organizado de forma progressiva; Flexibilidade de uso, possibilitando acesso ao conteúdo a qualquer momento e local.

Espera-se que o RoboLearn possa trazer contribuições para o cenário educacional brasileiro, especialmente em instituições que enfrentam limitações de recursos físicos para o ensino de robótica. O aplicativo pode demonstrar um potencial para despertar o interesse de jovens estudantes pelas áreas STEM, além de facilitar o acesso a conhecimentos tecnológicos avançados.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento de um protótipo de aplicativo móvel para o ensino de robótica, denominado RoboLearn, que integra conteúdo teórico, simulações práticas e elementos de gamificação em uma plataforma acessível e intuitiva. A pesquisa

partiu da identificação de lacunas no ensino de robótica no contexto educacional brasileiro, caracterizado pela escassez de recursos físicos e materiais didáticos adequados para estudantes iniciantes.

O protótipo desenvolvido demonstra como tecnologias modernas de desenvolvimento móvel, especificamente Flutter e Firebase, podem ser empregadas para criar soluções educacionais robustas. A implementação seguiu princípios consolidados de Interação Humano-Computador e gamificação, resultando em uma interface que equilibra funcionalidade técnica com experiência de usuário atrativa.

Os resultados obtidos evidenciam a viabilidade de utilizar dispositivos móveis como plataforma para o ensino de conceitos complexos de robótica e eletrônica. O RoboLearn conseguiu traduzir conteúdos tradicionalmente dependentes de equipamentos físicos em experiências digitais interativas.

Para trabalhos futuros, sugere-se a implementação de funcionalidades *offline*, expansão do catálogo de componentes eletrônicos, expansão para *web* com o intuito de tornar o conteúdo mais acessível e possibilitar a inclusão de novos conteúdos. Recomenda-se também a condução de estudos de usabilidade com grupos diversificados de estudantes e o desenvolvimento de parcerias com instituições educacionais para validação em ambiente escolar real.

O RoboLearn pode representar um passo importante na direção de soluções tecnológicas que buscam facilitar o acesso ao ensino de robótica no Brasil, podendo contribuir para a formação de uma nova geração de profissionais capacitados em áreas críticas para o desenvolvimento tecnológico do país.

Referências

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: defining gamification. pages 9–15.

Diyarova, L. et al. (2024). The use of arduino for educational purposes. *Trudy Universiteta*, 3.

Gumennykova, T. P. et al. (2019). Gamification features of stream-education components with education robotics. *Applied Aspects of Information Technology*, 2(1):45–65.

Kapp, K. M. (2012). The gamification of learning and instruction: game-based methods and strategies for training and education. *John Wiley & Sons*.

Lima, A. P. d. and Borges, M. A. F. (2020). Desafios da robótica educacional acessível e inclusiva no brasil. In *Anais do Workshop de Desafios da Computação aplicada à Educação (DesafIE!)*, pages 1–10, Porto Alegre, RS, Brasil. SBC. Acesso em: 25 jun. 2025.

Nielsen, J. (1994). Usability engineering. *Morgan Kaufmann*.

Oliveira, J. S. and Ferreira, L. M. (2023). Aprendizagem ativa e tecnologia: Impactos da robótica no engajamento estudantil. *Revista de Educação e Tecnologia*, 18(3):78–92. Acesso em: 26 jun. 2025.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.

Payne, R. (2019). *Beginning App Development with Flutter*. Apress.

Prensky, M. (2001). Digital natives, digital immigrants. *On the horizon*, 9(5):1–6.

Rogers, Y., Sharp, H., and Preece, J. (2011). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.

Santos, M. F. and Lima, P. R. (2022). Desenvolvimento de competências através da robótica: O aluno como protagonista. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*, volume 33, pages 123–135, Porto Alegre, RS, Brasil. SBC. Acesso em: 26 jun. 2025.

Santos, R. A. (2024). Robótica educacional: aprendizado através da criação. Escolhendo Bem. Acesso em: 25 jun. 2025.

Silva, A. B. and Costa, C. D. (2023). Robótica educacional: Uma abordagem interdisciplinar para o ensino de stem. *Revista Brasileira de Informática na Educação*, 31(2):45–62. Acesso em: 26 jun. 2025.

Sommerville, I. (2011). *Engenharia de software*. Pearson Prentice Hall, São Paulo, 9 edition.

Valente, J. A. (2003). O papel do computador no processo ensino-aprendizagem. *Boletim do Salto para o Futuro*.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

Zem, R. and Borges, M. A. F. (2017). Robótica educacional no brasil: questões em aberto, desafios e perspectivas futuras. *Revista Ibero-Americana de Estudos em Educação*, 12(4):2108–2121. Acesso em: 25 jun. 2025.