

E++ (ENSINANDO C++): APLICATIVO PARA AUXILIAR INICIANTES NA PROGRAMAÇÃO

Rickson Reis Amoroso¹, Melissa Nascimento¹, Rodrigo de Oliveira Plotze¹, Anna P. Z. China¹

¹Faculdade de Tecnologia de Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

rickson.amoroso@fatec.sp.gov.br,
melissa.nascimento3@fatec.sp.gov.br,
rodrigo.plotze@fatec.sp.gov.br,
anna.china@fatec.sp.gov.br

Resumo. *Ingressar no estudo da programação pode ser intimidador ainda mais quando não se tem nenhuma vivência anterior com isso. Este artigo apresenta as etapas do desenvolvimento de um aplicativo para dispositivos Android desenvolvido com Flutter. O objetivo deste trabalho é facilitar a compreensão de conceitos básicos de programação para estudantes iniciantes, por meio de atividades gamificadas e conteúdo descomplicado para tornar mais dinâmica a compreensão, utilizando a linguagem C++ na didática. Tendo como resultado uma aplicação intuitiva que oferece uma solução gratuita e em português para o aprendizado de programação.*

Abstract. *Embarking on the study of programming can be intimidating, especially for those with no prior experience. This article presents the stages of developing an Android app using Flutter. It aims at facilitating the understanding of basic programming concepts for beginner students. The app includes gamified activities and simplified content to make comprehension more dynamic, utilizing the C++ language in its teaching methodology. The result is an intuitive application that provides a free, Portuguese-language solution for learning programming.*

1. Introdução

Quando se ingressa no ensino superior em curso que envolva programação, é comum que ocorram dúvidas e dificuldades na organização dos estudos, principalmente para os alunos iniciantes na área. Muitas vezes as disciplinas de programação do semestre de abertura podem não acolher estes alunos que nunca tiveram nenhum contato, o que pode ocasionar em uma defasagem no aprendizado e um acúmulo de dificuldades que é carregado até os outros semestres, uma vez que o aluno não teve a base de estudo para programação.

Tendo em vista essa situação, o objetivo do projeto aqui apresentado é o desenvolvimento de um aplicativo que proporcione uma introdução e prática de programação, além de oferecer a oportunidade de competir com outros estudantes visando a conquista de posições em *rankings*, que serão atribuídas com base no progresso do aluno no estudo dos conteúdos, na conclusão de atividades.

O projeto desse aplicativo desenvolvido em *Flutter*, apresenta um significativo

valor no incentivo e na aprendizagem dos alunos iniciantes que possuem conhecimento limitado ou nulo na área de programação, afinal, é importante existirem estratégias inovadoras para se aprender a programar pois isso pode evitar que o aluno se desinteresse ou se sinta perdido no curso por não conseguir aprender esse alicerce de muitos cursos de Tecnologia da Informação (MAURÍCIO; OLIVEIRA; RENHE, 2020).

Este artigo está organizado da seguinte forma: na seção 2 são apresentados estudos e pesquisas que contextualizam a teoria na qual esse artigo é baseado e uma comparação do aplicativo proposto com outras aplicações da mesma área. A seção 3 descreve e contextualiza as tecnologias e métodos utilizados no desenvolvimento do aplicativo. Na seção 4 são apresentados detalhes sobre o desenvolvimento do aplicativo, na seção 5 é apresentado um descritivo das funcionalidades do aplicativo e por fim, na seção 6 são apresentadas as conclusões e considerações finais do artigo.

2. Referencial Teórico

Nesta seção são apresentados estudos e pesquisas utilizados como base teórica para o desenvolvimento deste trabalho. Na seção 2.1 é apresentada a visão de alguns autores sobre a dificuldade que muitos estudantes enfrentam em aprender programação; na seção 2.2 é mostrado o processo para a escolha da linguagem de programação utilizada na didática do aplicativo proposto; e na seção 2.3 é feita uma comparação entre algumas soluções de ensino de programação em aplicativo apresentadas e a solução proposta com o E++.

2.1. Dificuldades na Aprendizagem de Programação

Foi observado que muitos alunos que acabam de entrar na faculdade apresentam dificuldades em começar a compreender a programação. Muitos alunos acabam por desistir do curso por conta de disciplinas que envolvem essa área (BARBOSA; FERREIRA; COSTA, 2014), por apresentarem dificuldades em entender os conceitos básicos e não conseguirem tirar dúvidas, seja por conta de falta de tempo do professor durante a aula, ou seja, por não saberem por onde começar ou até mesmo por não conseguirem formular uma pergunta.

Rocha *et al.* (2010) apontam que uma grande dificuldade é a tarefa de conseguir identificar as dúvidas que cada aluno apresenta durante o ensino da programação, devido a vários fatores como tempo do curso e da aula e também a quantidade de alunos. Isso pode ser muito ruim para o aluno porque não sanar as dúvidas, principalmente em um nível de dificuldade mais baixo, acarreta numa maior dificuldade de se entender conceitos mais avançados da disciplina.

Segundo Conde (2022), outra dificuldade em se aprender programação é o “mar de conhecimento” que esta área apresenta. São muitas tecnologias e muitas possibilidades para começar e isso pode ser assustador ou desmotivador para o aluno no início. Uma metodologia mais dinâmica como a da gamificação com uma trilha de aprendizagem guiada utilizando somente uma linguagem de programação para aprender, seria uma alternativa interessante para tentar mudar esse cenário, pois, como apontam Figueiredo *et al.* (2015, p.1156), “Emergem, na contemporaneidade, diversas abordagens e possibilidades de se ampliar as ações pedagógicas em sala de aula, assim como seu potencial, dado o contexto social e tecnológico e os novos hábitos e práticas sociais”.

2.2. Escolha da Linguagem de Programação Utilizada

No trabalho produzido por Barbosa, Ferreira e Costa (2014), foi feita uma pesquisa nas disciplinas de programação de um curso de computação quanto aos dados analisados dos índices de reprovação e aproveitamento de notas delas com o objetivo de identificar se esses dados iriam mudar de acordo com a linguagem de programação utilizada pelos estudantes. Foram utilizadas as linguagens C e *Python*. De um modo geral, a mudança de uma linguagem para outra não resultou em uma melhora significativa no rendimento dos alunos e também não indicou uma melhor compreensão de determinado conceito de algoritmo em uma linguagem de programação em relação a outra.

Isso foi levado em consideração ao escolher a linguagem de programação que seria utilizada na didática do aplicativo. Como os primeiros usuários a testarem o *app* seriam os estudantes do curso de Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Ribeirão Preto (Fatec Ribeirão Preto), sobretudo aqueles dos semestres iniciais, considerando a situação, foi utilizada a linguagem C++ nos exercícios e no conteúdo dentro da aplicação por ser a linguagem de programação adotada nas disciplinas do primeiro e segundo semestres, utilizando uma estratégia parecida com a utilizada no desenvolvimento do jogo GameProgJF, que também tem o objetivo de ajudar no estudo de programação (MAURÍCIO; OLIVEIRA; RENHE, 2020).

2.3. Comparando com soluções já existentes

Uma pesquisa foi realizada com o objetivo de identificar aplicações que compartilham da mesma premissa do aplicativo E++, proposto neste trabalho e também verificar os artigos produzidos em relação aos métodos de ensino de programação aplicados e como eles apoiam o desenvolvimento de um aplicativo que o faz de forma interativa. Cada solução encontrada foi identificada com uma letra escrita entre parênteses logo ao lado do nome da plataforma, para uma melhor compreensão da comparação e conclusões apresentadas. Como resultado, foram feitas as observações abaixo.

O *Sololearn* (a) é uma plataforma gratuita que oferece ensino de linguagens de programação por meio de lições curtas exercícios práticos e possui ambiente para competição entre os usuários. Está disponível para dispositivos *Android*, *IOS* e *Web* (SOLOLEARN, 2023).

De acordo com a loja de aplicativos *Google Play*, o *Programming Hub* (b) é um aplicativo disponível para *Android*, *IOS*, *Web* e *Desktop* que permite se aprender programação por meio de atividades curtas e exemplos de programas para a prática (GOOGLE PLAY, 2023).

O Mimo (c) é um aplicativo disponível para *Android* e *IOS* que ensina programação de uma forma mais direta por meio de exercícios curtos e criação de projetos reais e permite o usuário se conectar com outros usuários (GOOGLE PLAY, 2023).

O *Encode* (d) é um aplicativo móvel que permite ao usuário aprender programação por meio de exemplos de programas com códigos reais e desafios práticos de programação. Está disponível para *Android* e *IOS* (GOOGLE PLAY, 2023).

Analisando as plataformas listadas acima, foram feitas observações a respeito de suas soluções. Os aplicativos (a), (b), (c) e (d) requerem planos pagos para que os usuários utilizem todo o seu conteúdo. As soluções (a), (b) e (d) não estão disponíveis em português do Brasil. Os *apps* (b) e (d) não oferecem uma opção de se conectar com outros

usuários e competir.

O E++ seria uma alternativa interessante para o seu público-alvo, por se tratar de uma solução apresentada na língua portuguesa, não exigindo uma compreensão prévia da língua inglesa, por permitir ao usuário competir em *rankings* e por apresentar seu conteúdo de forma gamificada e totalmente gratuita, focando somente em uma linguagem de programação para facilitar o aprendizado.

3. Materiais e métodos

O aplicativo foi desenvolvido com o uso do *Flutter*, um *framework* (estrutura de ferramentas) de código aberto do Google que contém um SDK (*Software Development Kit* - kit de desenvolvimento de *software*), que apresenta a possibilidade de compilar um só código-fonte para várias plataformas como *Android*, *IOS* e *Web*. O *framework* utiliza componentes chamados *widgets* que são a sua hierarquia central de classes, e responsáveis por criar os elementos, que compõem a renderização do aplicativo (FLUTTER, 2023).

O *Flutter* é impulsionado pelo *Dart*, que é uma linguagem de programação do Google otimizada para cliente e orientada a objetos que foi projetada para oferecer um desenvolvimento multiplataforma mais flexível por conta de seu ambiente de execução adaptado para várias estruturas de aplicativos (DART, 2023).

Para a parte de *back-end* foi utilizado o *Firebase* do Google, que oferece o *Firebase Authentication*, um serviço de autenticação de usuários e o *Firestore*, um banco de dados não relacional (*NoSQL*) que armazena os dados em estruturas denominadas documentos e coleções. Os documentos são unidades de armazenamento, ou seja, cada registro que contém campos ou que pode ser mapeado para valores. As coleções, por sua vez, são o local onde os documentos ficam armazenados (FIREBASE, 2023).

Os protótipos do aplicativo foram feitos utilizando a plataforma Figma e os diagramas foram elaborados com o Astah UML. O Figma é uma plataforma *web* que possui várias ferramentas de *design* de interface e prototipação, oferecendo vários recursos para representar as telas de um aplicativo, site ou sistema a ser desenvolvido (FIGMA, 2023).

O Astah UML é uma ferramenta de modelagem UML que permite fazer diagramas de caso de uso, classes, sequências, entre outros, de forma simples e descomplicada (CHANGEVISISON, 2023). UML (*Unified Modeling Language*) é a Linguagem de Modelagem Unificada, criada para ser estabelecida “uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de *software*” (LUCIDCHART, 2023).

Os casos de teste foram desenvolvidos utilizando a técnica de BDD (*Behaviour Driven Development*), desenvolvimento orientado por comportamento, uma metodologia ágil que utiliza uma linguagem simples e objetiva para descrever o comportamento do sistema em diferentes situações que lhe são apresentadas. É utilizado como técnica de teste pois é considerado uma evolução do chamado TDD (*Test Driven Development*), desenvolvimento orientado a testes (DEVMEDIA, 2023). O BDD foca no motivo do código ser criado e não nas técnicas utilizadas para isso, sendo dividido em três partes, com as suas respectivas palavras-chaves (RIBEIRO, 2019): História de usuário, que descreve brevemente a funcionalidade por meio da perspectiva de uma pessoa que quer realizar uma nova ação no sistema – palavras-chave: “Eu como <x>”, x se referindo a

quem é o usuário; “Quero <y>”, y sendo o que o usuário deseja fazer; “Para <z>”, com z sendo o motivo da ação a ser realizada; Funcionalidade, que se refere a ação do sistema relacionada ao que o usuário deseja realizar; Cenário, que é a descrição do comportamento baseado em uma entrada realizado para se alcançar o resultado esperado de uma funcionalidade – palavras-chave: “Dado <a>”, onde a é ponto inicial da ação do usuário; “Quando ”, b sendo a ação; “E <c>”, com c referindo-se a uma possível nova ação; “Então <d>”, d sendo o resultado esperado das ações.

4. Desenvolvimento do aplicativo

Nesta seção são apresentadas informações sobre o desenvolvimento do aplicativo proposto, tais como o seu público-alvo e a lista de requisitos. Também são apresentados: a forma como aplicativo funciona, alguns casos de testes e a experiência de interface do usuário.

4.1. Público-alvo

O público-alvo do aplicativo é formado tanto por indivíduos que estejam iniciando a aprendizagem de programação, como também por indivíduos que desejam aprimorar seus conhecimentos.

4.2. Diagramas e requisitos funcionais e não funcionais do aplicativo

Os diagramas de classes, atividades e de caso de uso e o modelo de banco de dados estão disponíveis no *GitHub*, com acesso via *link*: <https://github.com/rickreisme/Epp/tree/main/Projeto/Diagramas>. A seguir são apresentadas as listas de requisitos funcionais e requisitos não funcionais do aplicativo.

4.3. Requisitos Funcionais

RF001 - O aplicativo deve permitir o cadastro dos alunos utilizando os dados: nome, sobrenome, *e-mail*, nome de usuário, faculdade ou escola e senha. Sendo os dados: nome, sobrenome, nome de usuário, *e-mail* e senha, obrigatórios.

RF002 - O aplicativo deve permitir o *login* dos alunos utilizando *e-mail* e senha previamente cadastrados.

RF003 - O aplicativo deve permitir ao aluno, a troca de sua senha de login caso tenha a esquecido. O método de recuperação de senha disponibilizado pelo *Firebase* deve estar disponível na tela de *login* identificado pelo botão “Esqueceu a senha?”, que levará o usuário até a tela de redefinição de senha onde será solicitado o endereço de *e-mail* informado durante o cadastro para o qual será enviado o *link* de alteração de senha.

RF004 - O aplicativo deve permitir a realização de tarefas diárias pelo aluno. Essas tarefas serão elaboradas e publicadas automaticamente pelo sistema, com base no conteúdo armazenado previamente no banco de dados.

RF005 - O aplicativo deve permitir que os alunos se conectem entre si por meio de solicitações de amizade.

RF006 - O aplicativo deve permitir que alunos que se conectaram compitam entre si por meio de *ranking* de execução de atividades por meio de *ranking* de execução de atividades que levará em consideração a pontuação e o progresso dos usuários.

RF007 - O aplicativo deve permitir o *login* do usuário com o Google e o Facebook.

RF008 - O aplicativo deve permitir o compartilhamento dos resultados dos alunos nas redes sociais.

RF009 - O aplicativo deve permitir que o aluno altere o nome e sobrenome informados no momento do cadastro, há qualquer momento.

RF0010 - As disciplinas seguintes àquela que o aluno está estudando no aplicativo deverão permanecer bloqueadas até que ele atinja a pontuação necessária para avançar para a próxima.

RF0011 - O aplicativo deve permitir que o aluno crie anotações a seu critério, sobre algum exercício ou disciplina estudados. Podendo ser excluídas e editadas.

RF0012 - O aplicativo deve permitir que o aluno crie anotações do tipo lembrete sobre algum exercício ou disciplina que esteja sendo estudado. Podendo ser excluídas e editadas.

RF0013 - O aplicativo deve permitir que o usuário visualize os termos e condições da plataforma quando quiser.

4.4. Requisitos Não Funcionais

RNF001 - Os termos e condições do aplicativo devem estar disponíveis para exibição ao aluno na tela de *login* e na opção ‘Sobre’ na tela ‘Configurações’, após clicar no botão identificado como ‘Termos e Condições’.

RNF002 - O aplicativo deverá apresentar uma tela de abertura por 3 segundos contendo somente o logo estático em um fundo na cor preta.

RNF003 - O aplicativo deve ser responsivo em diferentes tamanhos de tela e orientações, garantindo uma experiência de usuário consistente em dispositivos móveis.

RNF004 - O tempo de resposta do aplicativo a ações do usuário, como cliques em botões e interações com elementos da interface, deve ser rápido, geralmente abaixo de 0,5 segundos, para manter a sensação de fluidez e agilidade.

RNF005 - O aplicativo deve fornecer *feedback* visual claro para aprimorar a experiência do usuário durante a interação.

RNF006 - O aplicativo precisará de um dispositivo móvel que esteja conectado à internet para funcionar.

RNF007 - O aplicativo deve ser capaz de enviar notificações aos usuários para alertá-los sobre novos conteúdos, desafios, conquistas desbloqueadas e outras informações relevantes para manter o engajamento.

RNF008 - O usuário deve ser apresentado somente no tema de modo escuro, não tendo a opção de modo claro.

RNF009 - Os usuários devem ter a opção de compartilhar suas conquistas, pontuações e progresso nas redes sociais, incentivando o engajamento e a competição saudável entre os colegas.

4.5. Casos de teste

Nessa seção são apresentados alguns casos de teste do aplicativo que foram feitos utilizando a representação narrativa da metodologia BDD (*Behaviour Driven Development*). Os casos estão divididos em subseções.

4.5.1. Funcionalidade 1: Logar no Aplicativo E++

Como: um usuário; Quero: fazer *login* na minha conta do E++; Para: ter acesso ao *app*.

Cenário 1: *Login* com sucesso: Dado: que estou na tela de login do E++; Quando: preencho o campo de *e-mail* com um endereço válido e previamente cadastrado; E: preencho o campo de senha com uma senha válida e previamente cadastrada; E: toco em entrar; Então: sou direcionado para a tela ‘Matérias (*Home*)’ do E++.

Cenário 2: *Login* sem sucesso (*e-mail* incorreto): Dado: que estou na tela de *login* do E++; Quando: preencho o campo *e-mail* com um endereço inválido e/ou não cadastrado previamente; E: preencho o campo de senha com uma senha válida e previamente cadastrada; E: toco em entrar; Então: é exibido um aviso na parte inferior da tela informando que o usuário não está cadastrado.

4.5.2. Funcionalidade 2: Se cadastrar no Aplicativo E++

Como: um usuário; Quero: criar a minha conta no E++; Para: poder fazer *login* no aplicativo.

Cenário 1: Cadastro sem sucesso; Dado: que estou na tela de cadastro do E++; Quando: não preencho nenhum campo de cadastro; E: toco em ‘Finalizar Cadastro’; Então: sou direcionado para a tela de *login* do E++.

Cenário 2: Cadastro com sucesso; Dado: que estou na tela de cadastro do E++; Quando: preencho todos os campos obrigatórios com informações válidas; E: toco em ‘Finalizar Cadastro’; Então: sou direcionado para a tela ‘Matérias (*Home*)’ do E++.

4.5.3. Funcionalidade 3: Avançar nas matérias disponíveis no aplicativo

Como: um usuário; Quero: iniciar a primeira matéria; Para: iniciar o aprendizado.

Cenário 1: Iniciar a primeira matéria; Dado: que estou na tela ‘Matérias (*Home*)’ do E++; Quando: clico no módulo ‘Lógica e Algoritmo’; E: seleciono a opção para começar a matéria; Então: sou direcionado para a tela da primeira aula dessa matéria.

Cenário 2: Desbloquear próxima matéria ao concluir a anterior com sucesso; Dado: que estou na tela da última aula do módulo ‘Lógica e Algoritmo’; Quando: concluo todas as aulas da matéria; E: retorno à tela ‘Matérias (*Home*)’ do E++; Então: a próxima matéria disponível é desbloqueada para acesso.

4.5.4. Funcionalidade 4: Gerenciar anotações

Como: um usuário; Quero: criar uma nova anotação; Para: guardar alguma informação sobre algum conteúdo do aplicativo.

Cenário 1: Escrever uma nova anotação; Dado: que estou na tela ‘Caderno’ do E++; Quando: toco em ‘Minhas Anotações’; E: toco no botão ‘Anotar’; E: escrevo a anotação na caixa de diálogo aberta; E: toco em ‘Salvar’; Então: vejo a anotação salva,

exibida na lista.

Cenário 2: Editar uma anotação existente; Dado: que estou na tela ‘Caderno’ do E++; Quando: toco em ‘Minhas Anotações’; E: toco em alguma anotação da lista; E: altero as informações preenchidas anteriormente; Então: vejo a anotação com as informações atualizadas, exibida na lista.

5. Funcionamento do Aplicativo

Nesta seção são apresentadas as funcionalidades básicas do aplicativo e algumas capturas de telas para ilustrá-las. Todas as capturas de telas feitas estão disponíveis em melhor resolução no repositório no *GitHub* e podem ser acessadas através do *link*: <https://github.com/rickreisme/Epp/tree/main/Projeto/Screenshots>.

5.1. Cadastro e Login

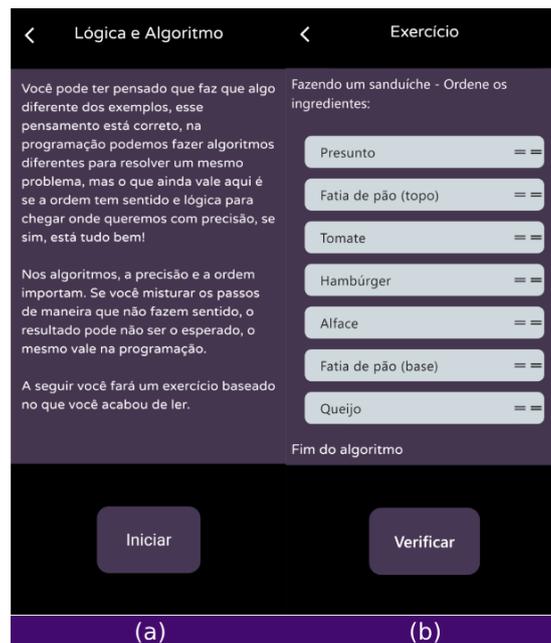
O aplicativo requer que o usuário esteja autenticado em uma conta previamente cadastrada na tela de cadastro (Figura 1b). O *login* é com *e-mail* e senha no formulário de autenticação (Figura 1a). Ao tocar no botão 'Esqueceu a senha?', inserindo o *e-mail* cadastrado, e tocando no botão 'Enviar' é enviado um *e-mail* com as instruções a serem seguidas pelo usuário (Figura 1c).



Figura 1. (a) tela de *login*; (b) tela de cadastro; (c) tela de redefinição de senha
Fonte: (Autoria própria, 2024)

5.2. Desbloqueio de Material de Estudo e Sistema de Pontuação

O material de estudo é desbloqueado à medida em que o usuário conclui o conteúdo do módulo anterior. O E++ adota um sistema de pontuação gamificada. Serão atribuídos 2 pontos para cada atividade concluída e 5 pontos para cada módulo concluído. A pontuação no E++ é independente e não tem qualquer relação com o sistema de notas oficial da faculdade em que o usuário está matriculado.



**Figura 2. (a) introdução do módulo ‘Lógica e Algoritmo’; (b) exercício de ‘Lógica e Algoritmo’
Fonte: (Autoria própria, 2024)**

Na figura acima é possível ver a tela de introdução e o exercício do módulo ‘Lógica e Algoritmo’ (Figura 2a e 2b).

5.3. Telas Principais

Após o *login*, a barra de navegação inferior oferece acesso rápido a cinco telas principais: ‘Matérias (*Home*)’: a primeira tela que o usuário logado visualiza, traz a visão de todas as disciplinas disponíveis para estudos. (Figura 3a); ‘Colegas’: tela onde é possível ver sugestões e convites de amizade (função desativada). (Figura 4a); ‘*Ranking*’: tela onde o usuário consegue conferir as melhores pontuações da semana no seu ciclo de colegas. (Figura 4b); ‘Caderno’: tela que possibilita o acesso aos conteúdos salvos (função desativada) e possibilidade de fazer anotações sobre as disciplinas. (Figura 3c); ‘Perfil’: tela são exibidas as informações do perfil do usuário, conquistas e também uma opção para adicionar lembretes (Figura 3b).

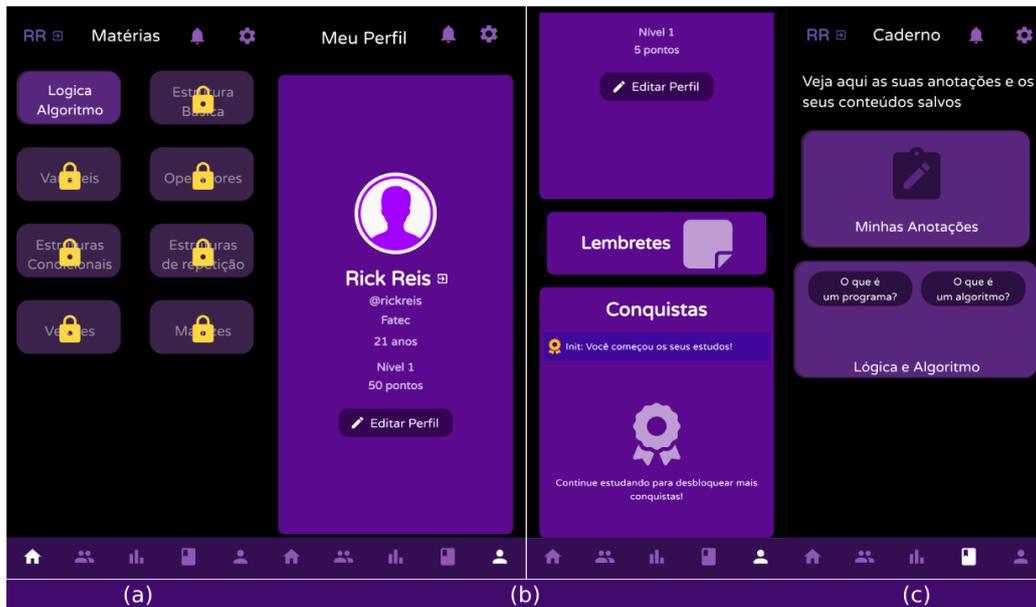


Figura 3. (a) tela ‘Matérias (Home)’; (b) tela ‘Perfil’; (c) tela ‘Caderno’
Fonte: (Autoria própria, 2024)

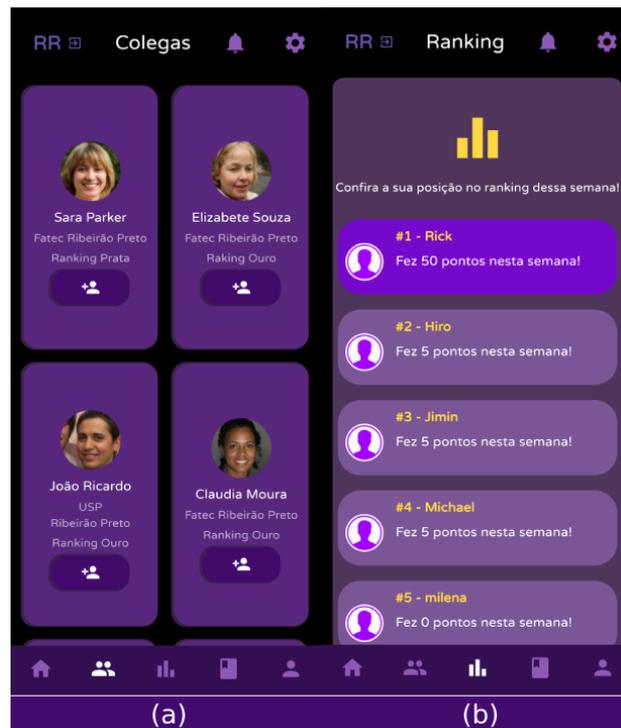


Figura 4: (a) Tela ‘Colegas’ e (b) Tela ‘Ranking’
Fonte: (Autoria própria, 2024)

5.4. Autenticação com *Firebase*

O *Firebase* é utilizado na autenticação e no armazenamento de dados. *E-mails* e senhas ficam registrados no *Firebase Authentication*, enquanto outros dados de cadastro são armazenados na coleção ‘usuários’ do *Firebase Cloud Firestore*.

6. Considerações Finais

O artigo apresentou um aplicativo desenvolvido em *Flutter*, projetado para auxiliar no estudo de programação para iniciantes. O trabalho incluiu a motivação do problema identificado, a contextualização das tecnologias e metodologias utilizadas, a apresentação dos requisitos funcionais e não funcionais, e a descrição do funcionamento do aplicativo, exemplificada com capturas das telas. Pode-se afirmar que o objetivo do projeto foi alcançado com sucesso, ao desenvolver um aplicativo de fácil utilização que proporciona uma introdução à base de programação de forma clara e intuitiva. Os usuários que testaram a atividade proposta no módulo de lógica e algoritmo afirmaram que conseguiram realizar o exercício sem dificuldades e que compreenderam bem o conteúdo e as instruções fornecidas.

Durante o desenvolvimento do aplicativo, foram encontrados alguns desafios. A primeira dificuldade foi explorar mais a fundo o *Flutter* e lidar com funções e ferramentas mais complexas, como a integração com o *Firebase* em diversas partes do aplicativo, o que, por si só, já representou uma dificuldade significativa. Implementar funcionalidades como o *ranking*, a exibição de informações na tela de perfil e a criação de anotações exigiu bastante tempo e estudo. Para o desenvolvimento do conteúdo e das atividades do módulo de lógica e algoritmo, foi necessária uma vasta pesquisa em bibliografias da linguagem C++ e a adaptação de todo esse conteúdo de forma mais intuitiva, para que um usuário sem conhecimento prévio pudesse entender sem problemas, o que foi trabalhoso e demandou tempo. A responsividade das telas do aplicativo em diferentes tamanhos de dispositivos *Android* também representou um grande desafio, sendo necessário focar em dispositivos e tamanhos mais populares. A instabilidade do *Flutter* foi uma das maiores dificuldades encontradas; sempre que o desenvolvimento era retomado após algum tempo, alguma atualização inutilizava parte do código ou gerava erros desconhecidos, exigindo tempo para investigação e resolução. Sincronizar as versões do aplicativo nos dispositivos utilizados para o desenvolvimento também foi um processo dificultado pela instabilidade do *framework*, pois sempre que era feito o *download* do repositório remoto para sincronizar com o local, o *Flutter* deixava alguma inconsistência entre os dois.

O maior foco da proposta era entregar um aplicativo que apresentasse uma boa solução para o aprendizado dos conceitos básicos de programação, a entrega inicial ficou dentro do esperado, contendo uma apresentação sobre o que é programação e a inicialização do módulo 'Lógica e Algoritmo' bem como o primeiro jogo sobre o conteúdo apresentado no mesmo. Em um momento futuro, espera-se implementar a funcionalidade de solicitações de amizade entre os usuários e também a criação do conteúdo dos outros módulos do aplicativo ('Estrutura Básica', 'Variáveis', 'Operadores', 'Estruturas Condicionais', 'Estruturas de Repetição', 'Vetores e Matrizes'), para que o E++ ofereça uma plataforma completa de introdução à programação.

7. Referências

- BARBOSA, Alexandre de A.; FERREIRA, Dyego ÍS; COSTA, Evandro B. (2014) Influência da linguagem no ensino introdutório de programação. In: III Congresso Brasileiro de Informática na Educação (CBIE 2014) – XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014), 2014, Dourados. *Anais...* Dourados: UFGD, 2014. p. 612.
- CHANGEVISISON. (2023) Powerful and Fast UML Diagramming Software – Astah.

- Astah. Disponível em <<https://astah.net/pfproducts/astah-uml/>>. Acesso em: 21 de set. de 2023.
- CONDE, Julliana. (2022) Por que programação é tão difícil? Sudo apt-get update (Linkedin). Disponível em <<https://www.linkedin.com/pulse/por-que-programação-é-tão-difícil-juliana-conde/?originalSubdomain=pt>>. Acesso em: 14 de fev. de 2023
- DART. (2023) Dart Overview. Dart. Disponível em <<https://dart.dev/overview>>. Acesso em: 21 de set. de 2023.
- DEVMEDIA. (2023) Desenvolvimento orientado por comportamento (BDD). DevMedia. Disponível em <<https://www.devmedia.com.br/desenvolvimento-orientado-por-comportamento-bdd/21127>>. Acesso em: 19 de out. de 2023.
- FIGMA. (2023) Figma: The Collaborative Interface Design Tool. Figma. Disponível em <<https://www.figma.com>>. Acesso em: 21 de set. de 2023
- FIGUEIREDO, M.; PAZ, T.; JUNQUEIRA, E. (2015) Gamificação e educação: um estado da arte das pesquisas realizadas no Brasil. In: IV Congresso Brasileiro de Informática na Educação (CBIE 2015), 2015, Alagoas. *Anais...* Alagoas: UFAL, 2015.
- FIREBASE. (2023) Documentação do Firebase. Firebase. Disponível em <<https://firebase.google.com/docs/build?hl=pt>>. Acesso em 21 de set. de 2023
- FLUTTER. (2023) Flutter – Build apps for any screen. Flutter. Disponível em <<https://flutter.dev>>. Acesso em: 21 de set. de 2023
- GOOGLE PLAY. (2023) Aprenda a programar/codar: Mimo. Google Play. Disponível em <<https://play.google.com/store/apps/details?id=com.getmimo>>. Acesso em: 19 de jun. de 2023.
- GOOGLE PLAY. (2023) Encode: Learn to Code. Google Play. Disponível em <https://play.google.com/store/apps/details?id=com.upskew.encode&hl=pt_BR&gl=US>. Acesso em: 19 de jun. de 2023.
- GOOGLE PLAY. (2023) Programming Hub: Learn to code. Google Play. Disponível em <https://play.google.com/store/apps/details?id=com.freeit.java&hl=en_US>. Acesso em: 18 de jun. de 2023.
- LUCIDCHART. (2023) O que é um diagrama UML? | Lucidchart. Lucidchart. Disponível em <<https://www.lucidchart.com/pages/pt/o-que-e-uml#:~:text=um%20diagrama%20UML-.O%20que%20é%20UML%3F,tanto%20estruturalmente%20quanto%20para%20comportamentos.>>>. Acesso em 21 de set. de 2023
- MAURÍCIO, S. J.; OLIVEIRA, A.; RENHE, C. M. (2020) O Uso de Jogos para Apoiar o Ensino e Aprendizagem de Programação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31., 2020, Online. *Anais...* Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 381-390.
- MIMO. (2023) Learn to Code with Mimo: Python, JavaScript, HTML, CSS, and More, 2023. Mimo. Disponível em: <<https://mimo.org>>. Acesso em: 19 de jun. de 2023.
- MOTTA, Sérgio. (2023) Aprenda a programar com o Programming Hub. SoftDownload. Disponível em <<https://www.softdownload.com.br/aprenda->

[programar-programming-hub-android-ios.html#:~:text=Em%20resumo%2C%20Programming%20Hub%20é,atrapalhar%20um%20pouco%20o%20aprendizado.&text=Sou%20o%20criador%20dos%20sites%20SoftDownload%2C%20TopFreewares%20e%20SuperTutoriais.>.](#) Acesso em: 19 de jun. de 2023.

RIBEIRO, Ronilson. (2019) Escrita do BDD no processo de desenvolvimento. Medium. Disponível em <<https://medium.com/@ronilsonribeiro/escrita-do-bdd-no-processo-de-desenvolvimento-469ea46540db>>. Acesso em: 19 de out. de 2023.

ROCHA, S. P.; FERREIRA, B.; MONTEIRO, D.; NUNES, C. S. D.; GÓES, N. C. H. (2010) Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. *Revista Novas Tecnologias na Educação*, Porto Alegre, v. 8, n. 3, 2010. DOI: 10.22456/1679-1916.18061.

SOLOLEARN (2023). Frequently Asked Questions. Sololearn. Disponível em <<https://www.sololearn.com/faq>>. Acesso em: 18 de jun. de 2023.

WERNECK, Vivi. (2021) 6 aplicativos para aprender programação. Tecnoblog. Disponível em <<https://tecnoblog.net/responde/6-aplicativos-para-aprender-programacao/>>. Acesso em: 19 de jun. de 2023