PROGLING: aplicativo educacional para ensino de linguagens de programação

SANTOS, Lucas Teixeira[1]; DAIBERT, Marcelo Santos[2]
TREVIZANO, Waldir Andrade[2]; BAÍA. Joás Wesley[2]

[1] Graduando em Ciência da Computação - UNIFAGOC
[2] Docente do curso de Ciência da Computação - UNIFAGOC

lucastsantos.developer@gmail.com
daibert@unifagoc.edu.br
waldir@unifagoc.edu.br
joas.baia@unifagoc.edu.br

RESUMO

Este artigo apresenta o aplicativo Prog:ling, que foi desenvolvido com o objetivo de ensinar linguagensde programação de maneira interativa e gamificada. A demanda por programadores qualificados continua crescendo, impulsionada pelo avanço tecnológico e pela necessidade de inovação. Aplicativosde ensino por meio de jogos tornaram-se populares por oferecerem uma abordagem prática e interativa para aprendizado. Inspirado no sucesso do aplicativo Duolingo, o Prog:ling utiliza a gamificação como estratégia de motivação e engajamento dos usuários. O aplicativo oferece lições interativas e um caminho progressivo para o aprendizado, além de feedback personalizado e sugestõesde melhoria com base no desempenho do usuário. O desenvolvimento do Prog:ling seguiu uma metodologia ágil, utilizando tecnologias como ReactNative e NestJS. O objetivo é facilitar o aprendizado de programação para iniciantes, tornando-o acessível, interativo e divertido. O artigo também aborda o ensino de programação e a eficácia da gamificação como metodologia de ensino. OProg:ling visa contribuir para a formação de novos programadores e popularizar o conhecimento em programação.

Palavras-chave: Aprendizagem. Programação. Aplicativos de ensino. Gamificação. Prog:ling.

INTRODUÇÃO

A aprendizagem de programação tem se tornado cada vez mais importante em nossa sociedade, não apenas para profissionais de tecnologia da informação, mas também para pessoas de diversas áreas, que buscam aprimorar suas habilidades e melhorar suas perspectivas de carreira (Gibson; Ostashewski, 2021; Pears; Seidman; Bailey, 2019).

Pesquisas indicam uma crescente demanda por programadores qualificados em váriospaíses. Podem-se destacar pesquisas como a da instituição governamental *Bureau of Labor Statistics* realizada em 2020, a quaç aponta que até 2029 acontecerá um aumento na demanda de até 22% de programadores nos Estados Unidos. No mercado brasileiro, também é possível destacar pesquisas como a da empresa Brasscom, realizada em 2021, que prevê uma demanda por profissionais relacionados diretamente à área de desenvolvimento de softwaree serviços de TIC e TI, alcançando 797 mil vagas entre os anos de 2021 e 2025. Ambas as pesquisas apontam que o crescimento ao longo dos anos acontece por conta da velocidade no avanço tecnológico, a transformação digital em empresas de diversos setores e a necessidade de inovação constante. Além disso, novas especializações e carreiras estão

aparecendo com o avanço tecnológico, como a inteligência artificial, a análise de dados, a segurança cibernética e a computação em nuvem. Vagas para essas carreiras se somam à demanda por profissionais na área, criando um cenário promissor para aqueles que buscam se especializar nessas áreas emergentes.

Apesar de a atratividade na área de desenvolvimento de software ganhar notoriedade por conta da oportunidade gerada pela demanda por esses profissionais no mercado, profissionalizar-se nessa área é um grande desafio, uma vez que exige um conjunto de habilidades e conhecimentos técnicos em constante atualização. As linguagens de programação e as tecnologias utilizadas estão em constante evolução, demandando dos profissionais um aprendizado contínuo e uma adaptação rápida às mudanças do setor.

Nesse contexto, aplicativos de ensino de linguagens de programação têm se tornado uma opção cada vez mais popular e viável (Li; Chen, 2019; Klašnja-Milićević; Vesin; Ivanović, 2020). Esses aplicativos oferecem uma abordagem prática e interativa para aprenderas linguagens de programação, permitindo que os usuários pratiquem e adquiram experiência em um ambiente simulado. Além disso, eles fornecem recursos educacionais, tutoriais e exercícios para auxiliar no processo de aprendizado.

O Duolingo, um aplicativo de aprendizado de línguas estrangeiras, é um exemplo bem-sucedido de como a gamificação pode ser usada para motivar os usuários a aprender. Esse aplicativo utiliza uma abordagem lúdica e interativa para ensinar línguas estrangeiras, o que tem resultado em um alto índice de engajamento e retenção de conhecimento (Arsalan, 2020;Borrás-Gené; Ruiz-Corbella; Martí-Parreño, 2021).

Inspirado pelo modelo do Duolingo, foi proposto o desenvolvimento de um aplicativo que segue a mesma metodologia, porém voltado para o ensino de linguagens de programação.

O Prog:ling é um aplicativo que tem como objetivo ensinar linguagens de programaçãode uma forma simples e interativa. Ele se baseia em uma metodologia de ensino progressiva, na qual os usuários aprenderão gradualmente conceitos básicos e avançados de programação, à medida que avançam em um ambiente virtual envolvente. O aplicativo contará com uma variedade de lições interativas e um caminho para o usuário seguir, baseado nas linguagens que está aprendendo, o que tornará a aprendizagem de programação mais atraente e acessível para os usuários.

Este artigo descreve o desenvolvimento do aplicativo Prog:ling, abordando a metodologia de ensino utilizada e sua eficácia em ajudar os usuários a aprenderem linguagensde programação.

Objetivos

O objetivo principal deste projeto é desenvolver um MVP (Mínimo Produto Viável) de aplicativo educacional para dispositivos móveis que ofereça uma abordagem interativa e divertida para o ensino de linguagens de programação, inspirada na metodologia bem-sucedida do Duolingo. Através desse aplicativo, pretende-se tornar o processo de aprendizagem de programação mais acessível e envolvente, permitindo que os usuários escolham a linguagem de programação de sua preferência e acompanhem um caminho estruturado para aprendê-la. O aplicativo também incentivará os usuários a superar desafios por meio de elementos de

gamificação, fornecerá feedback personalizado com base no desempenho do usuário.

ENSINO DE PROGRAMAÇÃO

De acordo com Souza, Batista e Barbosa (2016, p. 48), os principais problemas no ensino e na aprendizagem de programação incluem as dificuldades dos alunos em aprender os conceitos, a dificuldade na aplicação desses conceitos durante a construção de programase a falta de motivação. Para abordar esses problemas, foram identificadas tendências de pesquisa, como a utilização de visualização de programas e algoritmos, a utilização de jogos eo desenvolvimento de ambientes pedagógicos específicos.

Pesquisas científicas têm evidenciado a eficácia das tecnologias educacionais no ensino. De acordo com estudos realizados por Ling, Harnish e Shehab, em 2014, a utilização de aplicativos destinados ao ensino pode melhorar significativamente a compreensão e a aplicação prática dos conceitos ensinados para os alunos. Além disso, de acordo com a pesquisa conduzida por Rapkiewicz *et al.,* o uso de jogos de forma lúdica propicia flexibilidade e criatividade, fazendo o aluno explorar, pesquisar, encorajando o pensamento criativo, ampliando o universo, saciando a curiosidade, alimentando a imaginação e estimulando a intuição, e tudo isso contribui para o aprendizado" (Rapkiewicz *et al*., 2006, p. 4). Esse trecho foi retirado do artigo "Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao Uso de Jogos Educacionais" (Rapkiewicz *et al*., 2006), que defende o uso de jogos educacionais como uma ferramenta para o ensino de algoritmos e desenvolvimento de software.

Gamificação como metodologia de ensino

Ao longo dos anos, o ensino tem passado por uma evolução constante. Novas metodologias têm sido criadas e implementadas com o objetivo de tornar o processo de aprendizagem mais eficiente e atrativo. Essa evolução é especialmente notável na área de tecnologia da informação, que tem passado por uma expansão rápida nos últimos anos, comocitado anteriormente.

A gamificação é uma metodologia em destaque que consiste no uso de conceitos e estratégias implementadas por jogos para apoiar o ensino-aprendizagem. Artigos como "EduGamification: uma metodologia de gamificação para apoiar o processo ensino- aprendizagem" defendem que o uso dessa metodologia, aliada a outras, pode aumentar o desempenho acadêmico e o engajamento do aluno (Fabrício *et al*., 2019).

O aplicativo Duolingo, que foi utilizado como referência para este projeto, é um exemplo notório de sucesso no uso dessa metodologia. Sua abordagem lúdica, interativa e por vezes competitivas é uma das razões para tal êxito. O aplicativo compara o desempenho do usuário com o de outros por meio de um ranking, estimulando uma maior motivação paracontinuar aprendendo e alcançar os objetivos propostos pela plataforma.

MÉTODO DE DESENVOLVIMENTO DO PROG:LING

Para o desenvolvimento deste projeto, primeiramente foi realizada uma etapa de estudo de mercado e análise das necessidades dos usuários, incluindo a análise da

concorrência e a coleta de feedback de potenciais usuários.

Com base nesse estudo, foram identificadas oportunidades para a criação de um aplicativo de ensino de linguagens de programação. Em seguida, foram definidos os principais recursos que serão integrados na primeira versão do aplicativo. que será um MVP (*Minimum ViableProduct*) com as seguintes funcionalidades:

1. Disponibilização das linguagens de programação aos usuários;
2. Recursos de gamificação como por exemplo a vida do jogador e a ofensiva;
3. Implementação de conteúdos que precisam ser ensinados em cada linguagem pormeio de exercícios interativos;
4. Implementação da progressão através das fases do jogo, adicionando conteúdos eexercícios em cada fase;
5. Implementação de feedback nos exercícios.

Após a definição das funcionalidades, o processo de branding[1] foi realizado, no qual o nome e o logotipo do aplicativo foram escolhidos. Usando a logo como referência, foi desenvolvido um documento das cores principais do aplicativo[2]. Em seguida, ocorreu o desenvolvimento do *design* das telas do aplicativo[3], utilizando uma plataforma de *design* e prototipação Figma[4]. O design é uma etapa importante, uma vez que proporciona umavisualização mais clara para o desenvolvimento do *frontend* (parte do sistema que funciona no lado do cliente ou usuário) e da API (Interface de programação de aplicação). Com o protótipo finalizado, foi dado o início à etapa de desenvolvimento.

Tecnologias utilizadas

O aplicativo *Prog:Ling* teve sua concepção realizada através da integração de diversastecnologias. Nesta seção do texto, serão abordados os passos seguidos e as tecnologias usadas no desenvolvimento do aplicativo.

O React Native[5] foi a escolha para o desenvolvimento móvel, a fim de manter uma compatibilidade entre diferentes dispositivos e permitir o desenvolvimento de uma base de
código única para iOS e Android, que são os sistemas operacionais mais comuns nos celulares atualmente (Global Stats, 2023), juntamente da flexibilidade e a rica comunidade de desenvolvedores do *framework,* que já está consolidado no mercado, o que permite criar uma experiência de usuário nativa e responsiva; além disso, a linguagem *Typescript* utilizada para desenvolver no *framework* é a mesma linguagem utilizada no *backend*, o que proporciona uma curva de aprendizagem menor, já que muitas das bibliotecas podem ser reaproveitadas.

O *backend* do Prog:Ling foi construído usando o *framework* Node.js[6], especificamenteutilizando NestJS[7]. Essa escolha se deve principalmente à velocidade,

[1] Branding - https://drive.google.com/file/d/1flLPR_5Ctb6CEZZsPCVDGISchUKZhfiy/view?usp=drive_link.
[2] Documento de cores - https://drive.google.com/file/d/1q7bgb10rIB6I_lkDVvOLlwCGTWaATS- G/view?usp=sharing
[3] Design das telas do aplicativo - https://www.figma.com/file/KmqkmU8JN3BLOjbycSZFw3/Prog%3 ALing?type=design&node-id=0-1&mode=design&t=kJUicEjZOAIIgP41-0
[4] Figma - https://www.figma.com/
[5] React Native - https://reactnative.dev/
[6] NodeJS - https://nodejs.org/en/
[7] NestJS - https://nestjs.com/

à praticidade e à robustez do framework ao criar APIs RESTful e GraphQL[8], tornando possível gerenciar os dados e a lógica de negócios de maneira eficiente. Além disso, o uso de TypeScript[9] proporcionou um desenvolvimento mais seguro e fácil de manter, uma vez que é a mesma linguagem do *framework* utilizado no desenvolvimento móvel.

O MySQL[1010] foi adotado como sistema de gerenciamento de banco de dados, garantindo a capacidade de armazenar e recuperar dados de forma eficaz e escalável. Sua confiabilidade e capacidade de escalonamento tornaram-no uma escolha sólida para lidar como grande volume de informações relacionadas ao progresso do usuário e ao conteúdo das atividades as quais ele vai realizar durante o uso do aplicativo.

Estrutura do sistema

O Prog:Ling é uma aplicação de duas camadas composta por um *backend* e um aplicativo móvel, ambos interagindo através de uma API RESTful. A estrutura segue o modelocliente-servidor, com o aplicativo móvel atuando como o cliente e o servidor NestJS fornecendo os serviços e os dados necessários.

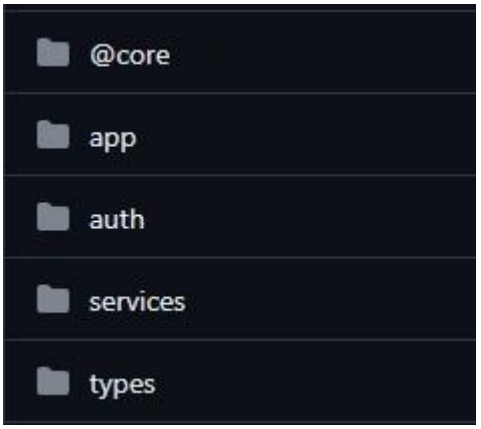
Backend

O backend foi desenvolvido seguindo princípios de arquitetura limpa (*Clean Architecture*) e consiste em módulos que abrangem autenticação de usuário, exercícios,

linguagens de programação, módulos e níveis os quais o usuário irá consumir para seu aprendizado.

A estrutura de pastas do *backend* do aplicativo Prog:Ling foi cuidadosamente organizada para garantir uma arquitetura limpa e de fácil manutenção. Esta estrutura compreende as pastas na Figura 1, a seguir:

Figura 1 - Estrutura de pastas do Servidor do aplicativo Prog:Ling



Fonte: dados da pesquisa (2023).

1. Core: A pasta "Core" é o núcleo do *backend* e abriga toda a lógica e regras de

[8] GraphQL - https://graphql.org/
[9] TypeScript - https://www.typescriptlang.org/
[10] MySQL - https://www.mysql.com/

negócio essenciais da aplicação. As camadas as quais essa pasta abriga foram projetadas com um baixo nível de acoplamento, o que significa que as funcionalidades nela contidas podem ser desenvolvidas e testadas independentemente, facilitando a escalabilidade e a manutenção do sistema.

2. App: A pasta "App" consiste em módulos que fazem referência aos módulos internos da pasta "Core". Cada módulo nesta camada contém controladores que expõem rotas da API, permitindo que o cliente do aplicativo interaja com a plataforma. Além disso, esta pasta abriga serviços que utilizam as lógicas internas da pasta "Core" para executar tarefas específicas. Isso permite uma separação clara entre a interface de comunicação com o cliente e a lógica de negócio subjacente.

3. Auth: A pasta "Auth" é responsável por armazenar todas as configurações relacionadas à autorização da aplicação. O NestJS é utilizado aqui para estabelecer políticas de autenticação, garantindo que apenas usuários autorizados tenham acesso a determinadosrecursos.

4. Services: A pasta "Services" abriga serviços gerais que são utilizados em toda a aplicação. Um exemplo notável é o serviço responsável por fornecer uma instância do Prisma[11], umaferramenta de manipulação de banco de dados, para que as operações de banco de dados possam ser executadas de forma eficaz e consistente em toda a aplicação.

5. Types: A pasta "Types" é responsável por manter as tipagens gerais utilizadas em toda a aplicação. Isso inclui a definição de novos tipos de dados, bem como a sobrescrita de tipagens existentes. Um exemplo prático disso é a adição do atributo "user" à tipagem dainterface "*Request*" do Express[12], permitindo que o sistema lide com informações do usuário de forma eficiente.

Essa estrutura de pastas bem definida e organizada contribui para a clareza, manutenção e escalabilidade do sistema, permitindo o desenvolvimento aconteça de maneiraeficiente, além disso, a arquitetura escolhida isola as regras de negócio e a lógica da aplicaçãoem uma camada com baixo acoplamento, o que significa que cada componente pode ser trabalhado independentemente, sem afetar outras partes do sistema. Essa abordagem de design mantém o código coeso, facilitando a compreensão, a manutenção, criação de testes automatizados e a expansão do aplicativo ao longo do tempo.

Banco de dados

A estrutura do banco de dados é fundamental na organização e gestão de um projeto,uma vez que facilita a visualização humana das complexas estruturas do banco. Isso facilita no entendimento, criação de novas funcionalidades e correção de problemas de forma eficaz. Este diagrama foi gerado utilizando a notação DBML de forma automática usando uma biblioteca de conversão de esquema do Prisma.

Do diagrama apresentado na Figura 2 da estrutura do banco de dados, as principais tabelas são:

1. Tabela *Student*: Armazena informações sobre os estudantes, incluindo identificação única, e-mail, nome, senha, data de criação e data de atualização. Cada registro nesta tabela está associado a um registro na tabela *StudentProgress*,

[11] Prisma - https://www.prisma.io/
[12] Express - https://expressjs.com/

responsável por rastrear o progresso de aprendizado do estudante.

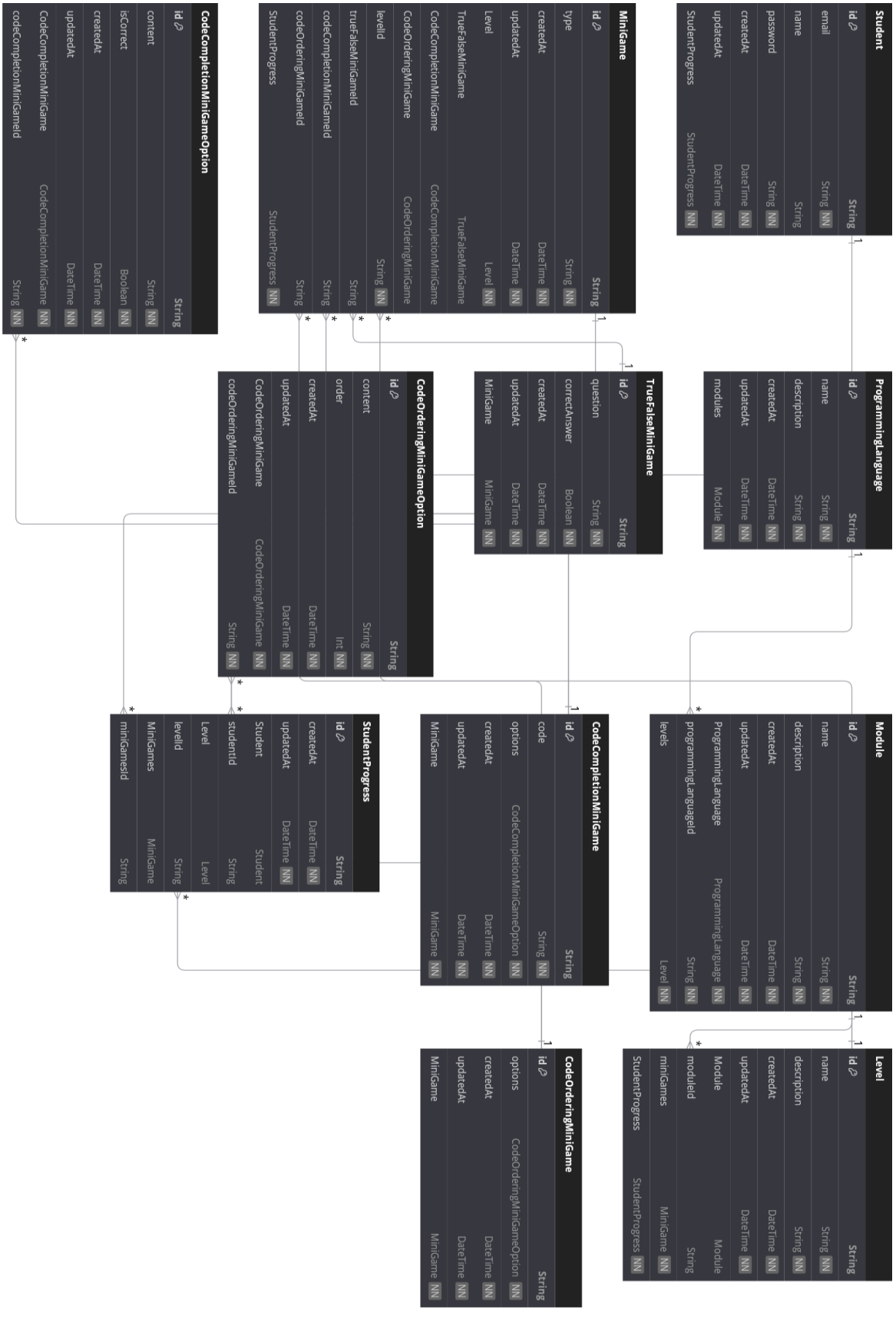
2. Tabela *ProgrammingLanguage*: Representa as linguagens de programação disponíveis no aplicativo, com atributos como identificação única, nome, descrição, data de criação e data de atualização. Estabelece uma relação com a tabela *Module* para categorizar os módulos deensino de cada linguagem.

3. Tabela Module: Contém informações sobre os módulos de ensino dentro de cada linguagem,incluindo identificação única, nome, descrição, data de criação e data de atualização. Estabelece relações com a tabela *ProgrammingLanguage* para indicar a linguagem associada e com a tabela *Level* para representar os diferentes níveis de dificuldade nos módulos.

4. Tabela *Level*: Armazena dados sobre os níveis de dificuldade dentro dos módulos de ensino, incluindo identificação única, nome, descrição, data de criação e data de atualização. Estabelece uma relação com a tabela *Module* para identificar a qual módulo pertence. Está associada à tabela *MiniGame*, que representa os jogos educacionais disponíveis em cada nível.

5. Tabela *MiniGame*: Representa os diversos jogos educacionais presentes no aplicativo, com atributos como identificação única, tipo de jogo, data de criação e data de atualização. Possuireferências a outras tabelas específicas para cada tipo de jogo, como *TrueFalseMiniGame*, *CodeCompletionMiniGame* e *CodeOrderingMiniGame*. Está relacionada com a tabela *Level* para indicar o nível ao qual o jogo pertence. Também está associada à tabela *StudentProgress*,que monitora o progresso do estudante em cada jogo.

Figura 2 – Diagrama do banco de dados do aplicativo Prog:Ling



Fonte: dados da pesquisa (2023).

Api

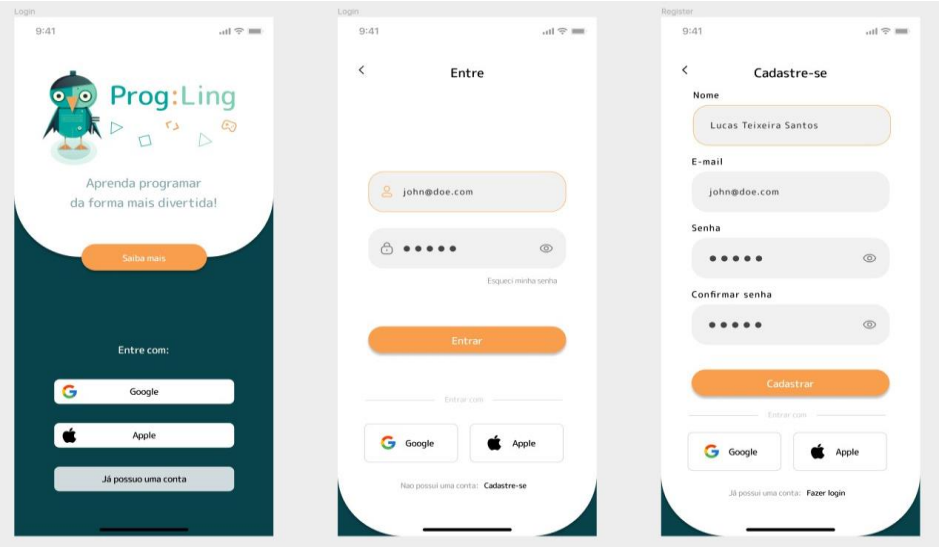
A API, sigla para *Application Programming Interface*, que significa Interface de Programação de Aplicação, consiste em um conjunto de rotas que executam partes específicas do código do *backend* e retornam um resultado para o cliente que a solicitou. Em outras palavras, ela age como um facilitador na comunicação entre o *backend* e o *frontend*.

A API criada para o aplicativo Prog:Ling segue os conceitos arquiteturais RESTful e permite o gerenciamento dos recursos relacionados aos usuários, às linguagens de programação, aos módulos, aos níveis e aos jogos. Além disso, a API implementa mecanismos de autenticação e autorização dos usuários, utilizando tokens JWT (*JSON Web Token*), e padroniza os métodos HTTP para operações CRUD, tornando a compreensão e a documentação mais acessíveis.

Mobile

O desenvolvimento do mobile começou a partir do design e prototipação, ambos foram realizados na plataforma web colaborativa de design de interfaces *Figma* a qual possuiferramentas para design e prototipação de interfaces para aplicações para celulares, páginas web, televisores e outros dispositivos. Com as cores do aplicativo e logo definidos no processode *branding* e utilizando de sites na internet como *Dribble*, *Muzli* e *Behance* para inspiração, primeiramente foi desenvolvido o sistema de design do aplicativo que possui regras de espaçamento, fontes de texto, tamanhos de fonte, ícones, cores e diversas outras regras e com o isso as primeiras telas de login e cadastro, elas podem ser observadas na Figura 3.

Figura 3 – Telas de login e cadastro do aplicativo Prog:Ling



Fonte: dados da pesquisa (2023).

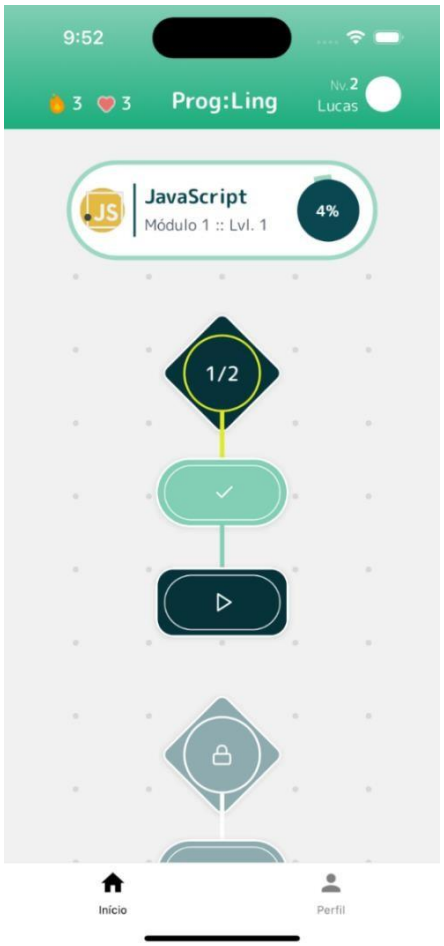
De acordo com um artigo da UX Movement (https://uxmovement.com/thinking/why-rounded-corners-are-easier-on-the-eyes/, 2011), que explica os benefícios dos cantosarredondados para a usabilidade e a estética, cantos arredondados são mais fáceis de processar visualmente, pois exigem menos esforço cognitivo e se adequam melhor ao movimento natural dos olhos. Além disso, cantos arredondados são mais orgânicos e seguros, diferentemente dos cantos com

ângulo reto que lembram uma resposta de esquiva aos cantosafiados e pontiagudos, que podem representar uma ameaça. Por conta disso, o design foi pensado para ter o mínimo possível de componentes com cantos pontiagudos o que o torna mais convidativo aos olhos.

Após a tela de login e cadastro, para manter a complexidade no mínimo possível, a tela inicial foi pensada com o mínimo de elementos e seguindo as regras de arredondamento doscomponentes citadas anteriormente e com nenhuma tela entre a inicial e as de autorização, no entanto, vale ressaltar que, em versões posteriores, o ideal seria ter telas de questionário para configurar o aplicativo para cada usuário da aplicação já com a linguagem de programação que ele decidiu iniciar, as notificações já permitidas e configurações a respeito do tempo que o usuário ira dedicar por dia no aplicativo; dessa forma, o aplicativo seria capazde lembrar o usuário de fazer exercícios no aplicativo todos os dias para sua progressão constante.

Na tela inicial da Figura 4, é possível observar, no topo da tela, em verde, o componente de nome *AppBar;* nele é possível localizar o nome do aplicativo, a quantidade de corações e investidas do usuário, o nome dele e seu nível atual. Em seguida, no corpo da tela, estão os módulos e níveis de cada modulo. O formato das figuras que representam os módulos e níveisfoi pensado para lembrar Fluxogramas que são diagramas que todo desenvolvedor aprende durante sua jornada de aprendizado.

Figura 4 – Tela principal do aplicativo Prog:Ling



Fonte: dados da pesquisa (2023).

Na parte superior do corpo da tela inicial na Figura 4, está a linguagem que o usuário está aprendendo, assim como o progresso dele, o módulo em que ele está e o nível desse módulo. As figuras geométricas losango e retângulo representam módulos e níveis respectivamente. Um nível concluído ou bloqueado tem sua borda totalmente arredondada e cor diferente para melhor identificação. Os níveis e módulos bloqueados possuem uma cor menos chamativa, já que o usuário ainda não tem acesso a eles, enquanto os níveis desbloqueados e concluídos possuem cores mais chamativas para atrair a atenção do usuário.

Para inicializar um nível e aprender o conteúdo que ele guarda, basta o usuário tocar na figura que representa o nível que está desbloqueado, representado por uma cor esverdeadaescura e botão de jogar no centro. Pressionando-o, ele será direcionado para a tela apresentaçãodo nível ,que possui o título dele no topo da página, uma imagem da linguagem e uma breve descrição do que ele aprendera nesse nível (Figura 5). Vale ressaltarque níveis já concluídos pelo usuário não podem ser revisitados a fim de diminuir a complexidade do desenvolvimento.

Figura 5 – Tela de apresentação do nível aplicativo Prog:Ling

Fonte: dados da pesquisa (2023).

Por fim, na parte inferior da página de nível, na Figura 5, observa-se um botão, o qual, ao ser pressionado,dará início aos jogos que irão ensinar o conteúdo que o nível se propõe a ensinar ao usuário.

Existem quatro tipos de jogos:

1. Markdown (linguagem de marcação para formatação de textos);
2. Verdadeiro ou falso;

3. Ordenação de código;
4. Complete o código.

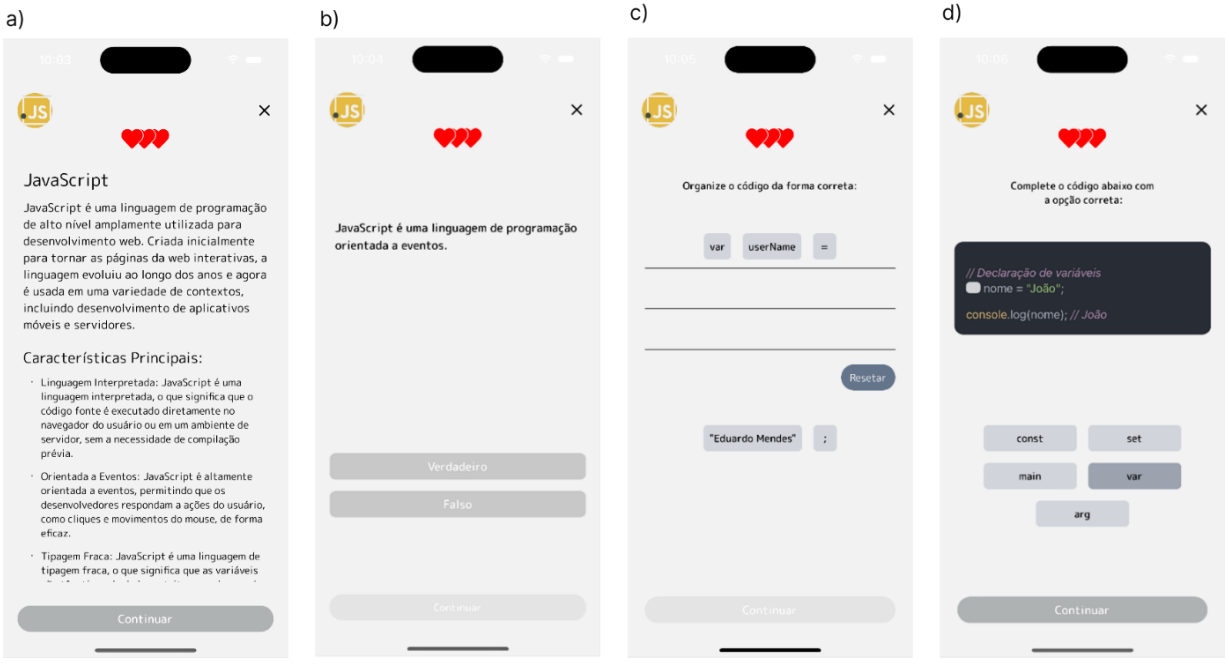
Markdown é uma linguagem de marcação leve, usada para formatar texto de maneirasimples e legível, e foi escolhida pela facilidade de estruturar os textos do banco de dados daaplicação. O jogo do tipo Markdown (Figura 6.a), é considerado um jogo da perspectiva técnica,uma vez que tanto o *backend*, quanto o mobile o tratam como tal. No entanto, não é possívelenviar respostas através dele, pois seu propósito é apresentar texto no formato Markdown aousuário, com o intuito de ensinar algo sobre o módulo.

O jogo Verdadeiro ou Falso, Figura 6.b, como o próprio nome sugere, consiste em umaafirmação, e o objetivo do usuário é identificar se ela é verídica ou não.

A Ordenação de Código, Figura 6.c, é um jogo em que o usuário encontra partes de código separadas em botões que podem ser clicados. Ao clicar, o código é direcionado para uma linha na parte superior da tela, na ordem em que o usuário pressionou. O objetivo do usuário é selecionar o código na ordem correta, de acordo com a sintaxe da linguagem.

Por fim, o jogo Complete o Código, Figura 6.d, disponibiliza no topo da tela um códigoem que falta uma parte; e um número determinado de opções que podem ou não preencher a lacuna corretamente. O objetivo do usuário é escolher a opção que completa corretamente o código, preenchendo a parte que está faltando.

Figura 6 – Images dos tipos de jogos do aplicativo Prog:Ling



Fonte: dados da pesquisa (2023).

Cada um dos jogos mencionados permite que o usuário submeta uma resposta, que pode estar correta ou incorreta, com exceção do Markdown. No caso de uma resposta correta,o usuário é direcionado para o próximo jogo, enquanto em caso de uma resposta incorreta, uma "vida" é subtraída. Quando o contador de vidas atinge zero, o jogador perde a partida e retorna à tela inicial. O contador de vida é representado por um ícone de coração que fica notopo da tela do jogo e no topo da tela

no canto esquerdo da tela inicial. Após um período determinado, as vidas são reiniciadas. O usuário só pode iniciar um novo nível caso o contadorde vida não esteja zerado.

Além do elemento "vida" da gamificação, também foi implementado o elemento "investida", representado pelo ícone de fogo. A investida é um incentivo para que o usuário mantenha sua atividade diária no aplicativo, e funciona da seguinte maneira:

Ao entrar pela primeira vez no aplicativo, o contador é inicializado em um. A partir desse ponto, toda vez que o usuário acessar o aplicativo diariamente, o contador será incrementado em um, até atingir o número máximo de cinco. Após isso, o contador passará amudar de cor seguindo a seguinte sequência: 1º dia: +1 de investida; 2º dia: +1 de investida;3º dia: +1 de investida;4º dia: +1 de investida;5º dia: +1 de investida; 6º dia: A cor do ícone de fogo muda para laranja;7º dia: A cor do ícone de fogo muda para verde; 8º dia: A cor do ícone de fogo muda para azul; 9º dia: A cor do ícone de fogo muda para roxo; 10º dia: A cor do ícone de fogo muda para dourado. A partir do 11º dia: A cor do ícone de fogo permanece dourada.

Para que a investida aumente, o usuário deve acessar o aplicativo de forma frequentee realizar pelo menos um exercício. No entanto, se o usuário em qualquer um dos dias não abrir o aplicativo ou abri-lo, mas não realizar nenhum exercício, sua contagem de investida será resetada.

CONSIDERAÇÕES FINAIS

O projeto do aplicativo Prog:Ling foi realizado através de várias etapas: análise de requisitos, *design* e prototipação, *branding*, planejamento da arquitetura, desenvolvimento *backend* e *mobile* e, por fim, o desenvolvimento e inserção no banco de dados dos conteúdos que serão consumidos pelos usuários para seu aprendizado. Cada etapa foi executada de forma a atingir os objetivos definidos de criar um MVP de um aplicativo para ensino de linguagens de programação, baseando-se na metodologia do aplicativo Duolingo de forma adaptada para o propósito do projeto.

O resultado atingido foi um aplicativo com uma interface simples com pouco mais de dez páginas, o que permite que o usuário consiga se cadastrar e em seguida entrar no aplicativo usando a conta cadastrada, selecionar uma das duas linguagens de programação disponíveis no aplicativo e seguir um caminho proposto para aprender dois módulos básicos disponíveis.

As linguagens de programação, módulos e exercícios são amostras, uma vez que não foi possível implementar todo o conteúdo e um conjunto maior de linguagens e exercícios devido ao tempo disponível para a realização do projeto. Em vista disso, a implementação demais conteúdo para o aplicativo e a realização de testes para comprovar a eficácia do métodode ensino proposto são considerações importantes para futuras etapas do projeto uma vez que podem proporcionar percepções de novas funcionalidades ou melhorias, que podem serintegradas para melhor aproveitamento do conteúdo por parte do usuário.

REFERÊNCIAS

LING, C.; HARNISH, D.; SHEHAB, R. Educational apps: using mobile applications to enhance student learning of statistical concepts. **Human Factors and Ergonomics in Manufacturing & ServiceIndustries**, v. 24, n. 5, p. 532–543, 2014.

PINTO, Fabrício; SILVA, Paulo**.** EduGamification: uma metodologia de gamificação para apoiar o processo ensino-aprendizagem. *In:* Workshop sobre Educação em Computação (WEI), 27, 2019, Belém. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 414-428. ISSN2595-6175. DOI: https://doi.org/10.5753/wei.2019.6647.

RAPKIEWICZ, C. E.; FALKEMBACH, G.; SEIXAS, L.; DOS SANTOS ROSA, N.; DA CUNHA, V. V.; KLEMANN, m. **Estratégias pedagógicas no ensino de algoritmos e programação associadasao uso de jogos educacionais.** UENF-Universidade Estadual Norte Fluminense Darcy Ribeiro, CCINTED-UFRGS, FAETEC, Colégio Mauá-RS, 2006.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. **Problemas e dificuldades no ensino e naaprendizagem de programação:** um mapeamento sistemático, 2016.

UX Movement. **Why Rounded Corners are Easier on the Eyes,** 15 nov. 2010. Disponível em: https://uxmovement.com/thinking/why-rounded-corners-are-easier-on-the-eyes/. Acesso em: 2023.