

TryCatch: um modelo de arquitetura visando a aprendizagem baseada em problemas

Trabalho de Conclusão do Curso de
Tecnologia em Sistemas Para Internet

Guilherme Parzianello¹
Orientador(a): Silvia de Castro Bertagnolli¹

¹ Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Rua Cel. Vicente, 281, Porto Alegre — RS — Brasil
glhrm_@live.com, silvia.bertagnolli@poa.ifrs.edu.br

***Resumo.** Estudos têm demonstrado a eficácia do uso da tecnologia como recurso para auxiliar o ensino de programação, tornando o aprendizado mais atrativo e diminuindo índices de reprovação e evasão. Buscando aliar os resultados positivos e suprir lacunas identificadas em pesquisas semelhantes, este trabalho visa o desenvolvimento de um modelo de arquitetura para a criação de atividades pedagógicas utilizando a metodologia ativa de Aprendizagem Baseada em Problemas. Para tanto, foi desenvolvido um estudo de caso com atividades voltadas para o ensino de Programação Orientada a Objetos (POO). Como método de trabalho, foram realizadas pesquisas exploratória e bibliográfica, identificados os requisitos e esboçado o projeto do sistema com diagramas UML e protótipos. A arquitetura auxiliará os estudantes a compreenderem o conteúdo por meio de um quiz, além de facilitar o acompanhamento de seu desempenho por parte do docente.*

1. Introdução

A Programação Orientada a Objetos (POO) é um dos paradigmas mais utilizados, atualmente, para o desenvolvimento de software, sendo parte da grade curricular de praticamente qualquer curso superior voltado à área — incluindo os do IFRS (Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul), instituição onde o presente trabalho está sendo desenvolvido.

No entanto, estudos (HENRIQUE; REBOUÇAS, 2015; SOUZA et. al, 2016; MOURA et. al, 2017) demonstram que disciplinas de cursos que envolvem o ensino de linguagens de programação chegam a concentrar mais da metade das reprovações de alunos entre toda a grade curricular. Entre os possíveis motivos está a dificuldade de aprender os conceitos e aplicá-los na resolução de problemas, pois exige do estudante alta capacidade de raciocínio lógico e abstração (especialmente no caso da POO), o que pode resultar em falta de motivação e possível desistência da disciplina, levando a elevados índices de evasão.

Buscando contornar esse cenário, diversas abordagens têm sido adotadas para tornar o aprendizado mais atrativo e lúdico, como o uso de ferramentas de software, jogos, metodologias e robótica (SILVA et al., 2015; RICHTER; BERNARDI; CORDENONSI,

2019). Os resultados observados em iniciativas do tipo foram, em geral, positivos, destacando-se maior facilidade de assimilação do conteúdo pelos alunos após o uso da ferramenta.

Em uma revisão de literatura realizada por Richter, Bernardi e Cordenonsi (2019) sobre o ensino de programação mediado por tecnologias educacionais, constatou-se que, dos 136 estudos analisados, apenas 12 abordaram a POO, dos quais três utilizaram jogos como estratégia pedagógica. Para os autores, a escassez de estudos na área denota a existência de um espaço a ser preenchido com novas metodologias para o ensino desse paradigma. Dentre as possíveis abordagens recomenda-se a aproximação do aluno com sua realidade e a diversificação do ferramental tecnológico.

Os pontos observados vêm ao encontro das propostas das metodologias ativas de educação, inicialmente propostas por John Dewey. Esse paradigma inverte o cenário tradicional de ensino centrado no professor e na simples passagem de conteúdo para o aluno, de forma a alçar este ao protagonismo do processo de aprendizagem. Na prática, o docente assume o papel de mediador entre o discente e os conceitos a serem assimilados, elaborando questões ou problemas que possibilitem ao estudante construir o conhecimento de forma mais autônoma (BERTAGNOLLI et al., 2020). Além disso, a popularidade dos dispositivos móveis entre os jovens, não raro vista como um obstáculo ao aprendizado em sala de aula, pode tornar-se uma ferramenta para que o aluno busque as informações necessárias à resolução dos problemas de aprendizagem propostos.

Considerando o exposto, este trabalho tem como objetivo o desenvolvimento de um modelo de arquitetura para dispositivos móveis vinculada à Aprendizagem Baseada em Problemas. Nesse modelo de metodologia ativa, o estudante é estimulado a resolver um desafio dado por determinado contexto com o apoio de materiais de consulta. A arquitetura terá o formato de um *quiz* cujas questões podem ser livremente associadas aos recursos previamente citados.

2. Fundamentação teórica

Neste capítulo serão explorados dois pilares que dialogam com os princípios e objetivos do sistema proposto: o conceito de aprendizagem ativa com foco na Aprendizagem Baseada em Problemas e o potencial de aplicabilidade do *mobile learning* baseado em dados sobre o uso de dispositivos móveis e internet pela população brasileira.

2.1 Metodologias ativas

Moran (2018) observa que a aprendizagem ativa opõe-se aos métodos tradicionais de ensino — orientados à memorização, repetição e controle — ao associar-se à aprendizagem reflexiva, tornando mais visíveis os conhecimentos e competências adquiridos no processo de aprendizado. Segundo o autor, a aprendizagem é mais significativa quando o aluno participa ativamente e o professor assume o papel de orientador do que quando o resultado é medido pela diferença entre o conteúdo explicado pelo professor e o que foi assimilado pelo aluno.

Na Aprendizagem Baseada em Problemas (ABProb), o processo de aprendizagem é guiado pela solução de uma situação-problema, na qual o aluno é motivado a analisar o contexto, resgatar conhecimentos prévios e realizar pesquisas com a mediação do professor. O contexto problemático deve ser adequado ao perfil da turma, atrair o interesse e a

curiosidade dos alunos e estar relacionado aos conteúdos curriculares e objetivos da aprendizagem (BERTAGNOLLI et al., 2020).

As metodologias ativas estão intrinsecamente ligadas a modelos de ensino flexíveis e híbridos, especialmente levando-se em conta o contexto conectado e digital atual. Além das possibilidades ampliadas de pesquisa e comunicação, as tecnologias digitais facilitam a monitoração das etapas do processo e a visualização dos resultados (MORAN, 2018). É, portanto, desejável que a aplicação das metodologias ativas ocorra num ambiente que proporcione suporte tecnológico para sua plena realização.

2.2 Mobile learning

Barcelos, Tarouco e Bercht (2009, pp. 3–4) caracterizam o *mobile learning* como uma “onda tecnológica” decorrente de “um cenário de ‘computação pervasiva’ pautado pela mobilidade global do usuário, conectividade ubíqua, independência de dispositivo e ambiente computacional do usuário disponível em qualquer lugar, a qualquer tempo”, o qual é válido tanto para a Educação a Distância (EaD) quanto para o contexto presencial.

Na época do estudo em questão, o uso de *smartphones* no Brasil ainda era incipiente, o que justifica a preocupação dos autores com a compatibilidade entre os sistemas que estavam sendo desenvolvidos para o ensino de algoritmos e os aparelhos produzidos por diferentes fabricantes. Com o lançamento do primeiro *smartphone* Android, ainda em 2009, o cenário mudou: não só houve uma crescente popularização desse tipo de dispositivo ao longo da década que se seguiu como observou-se uma simplificação do processo de desenvolvimento de aplicações para dispositivos móveis a partir do uso de uma plataforma comum.

Dados do Comitê Gestor da Internet no Brasil (CGIBR, 2020) mostram que, em todo o país, o telefone celular é o principal dispositivo usado para acessar a internet (99%), sendo o único meio utilizado por 58% dos usuários, número que chega a 85% ao se considerar as classes D e E. Além disso, o total de domicílios com acesso à internet sem computador dobrou entre 2017 e 2019, somando 34% da realidade brasileira; nesse grupo, é possível concluir que o acesso à internet se dá exclusivamente pelo celular, já que o dispositivo está presente em 93% dos domicílios. A pesquisa ainda demonstra que 37% dos usuários de internet engajam-se em jogos *online*, especialmente os mais jovens.

De qualquer forma, as expectativas em relação ao uso de dispositivos móveis como recurso de apoio à educação, lançadas em um contexto radicalmente diferente, hoje se refletem em possibilidades reais. A ampliação do acesso ao conhecimento em lugares estranhos à sala de aula tradicional (independentes, inclusive, de uma localização fixa) e o baixo custo aliado à usabilidade dos aparelhos, que já estão presentes em outras esferas da vida dos brasileiros, são apontados como vantagens da aprendizagem por dispositivos móveis (O’MALLEY, 2003; DIAS, 2008 *apud* BARCELOS; TAROUCO; BERCHT, 2009).

3. Trabalhos relacionados

Esta seção dedica-se a analisar cinco ferramentas propostas em estudos desenvolvidos com o objetivo de aliar o uso de tecnologias diversas ao ensino de programação. Ao fim, buscou-se comparar a presença ou não de alguns pontos de interesse vinculados à finalidade do presente trabalho, sendo eles: (a) utilização de Metodologias Ativas; (b) disponibilidade para uso em

dispositivos móveis; (c) abordagem de POO com utilização de Java; (d) registro de aplicação no Ensino Superior; (e) desenvolvimento realizado especialmente para o estudo; (f) possibilidade de o professor acompanhar o desempenho do aluno; (g) jogo em formato de *quiz*. Os resultados são apresentados no Quadro 1.

Quadro 1 — Comparativo entre as ferramentas analisadas

	Robocode	Treine POO	Lord of Code	JES	CLinClass	TryCatch
Possibilita o uso de alguma metodologia ativa						x
Disponível para dispositivos móveis		x			x	x
Foco em POO com uso de Java	x	x	x		x	x
Aplicado no Ensino Superior		x	x		x	x
Desenvolvido especialmente para o estudo		x	x		x	x
Permite que o professor acompanhe o desempenho do aluno			x		x	x
Jogo em formato de <i>quiz</i>		x	x		x	x

Fonte: Elaborado pelo autor (2021).

A ferramenta Robocode, desenvolvida pela AlphaWorks/IBM, foi utilizada em um estudo envolvendo graduandos em Engenharia Eletrônica e de Telecomunicações da Universidade Federal de Uberlândia (UFU). O objetivo era programar um robô de forma a colocá-lo em batalha com robôs criados por outros desenvolvedores. Segundo os autores, a iniciativa apresentou bons resultados, promovendo o amadurecimento do raciocínio lógico e interesse pela programação por parte dos secundaristas e a capacidade de liderança dos alunos de graduação (AMARAL; SILVA; PANTALEÃO, 2015).

No caso do aplicativo Treine POO, o elemento lúdico esteve presente por meio da gamificação, incluindo recursos como pontuação, *rankings* e conquistas de medalhas. A ferramenta apresentava questões de múltipla escolha relacionadas à POO divididas em três níveis de dificuldade. Ao fim, o desempenho do usuário poderia ser comparado ao de outras pessoas. Por ter sido desenvolvida como um protótipo para avaliação da usabilidade e experiência do usuário, não foram apresentados resultados referentes ao impacto da ferramenta no aprendizado dos alunos (MOURA et al., 2019).

De forma semelhante, a ferramenta Lord of Code propunha, a partir da análise de um código em Java, a identificação do resultado do algoritmo ou do trecho que apresentava erro.

Os acertos seriam convertidos em pontos, que poderiam ser trocados por sugestões de resposta nas questões seguintes. O recurso previa uma evolução que incluiria a definição de fluxos de perguntas e seus respectivos pré-requisitos de acordo com o conteúdo abordado. Assim, caso o usuário errasse uma questão referente a determinado conteúdo, seria direcionado a um fluxo alternativo de forma a reforçar os pré-requisitos para aquela questão. Por meio da atribuição de um perfil de acesso específico, o professor seria responsável pela definição desses fluxos e poderia analisar os registros para identificar deficiências pontuais de aprendizagem de cada aluno (SOUZA et al., 2016).

Seguindo na linha de ferramentas baseadas em perguntas e respostas, o aplicativo para dispositivos móveis CLinClass foi aplicado em uma turma da disciplina de POO de um curso de Ciência da Computação. A principal diferença identificada em relação aos trabalhos anteriormente expostos foi a abordagem colaborativa, dado que a ferramenta foi projetada visando a reunião dos alunos em pequenos grupos para responder aos desafios. Além disso, o aplicativo possuía um perfil de acesso para o professor, o qual poderia incluir materiais de apoio e visualizar o registro de respostas. O estudo verificou que o índice de reprovação na disciplina foi o menor em sete semestres, caindo 55% em relação ao semestre anterior (MACHADO et al., 2018).

Por fim, em um projeto de ensino aplicado a uma turma de nível técnico em Feira de Santana (BA), optou-se por utilizar o ambiente JES (Jython Environment for Students), voltado à manipulação de imagens com foco no ensino de programação com a linguagem Python. Os conceitos de POO foram abordados de forma gradual a cada aula, acompanhados de atividades para aplicação prática na ferramenta escolhida. Entre os pontos positivos observados, os autores ressaltaram a contextualização da atividade à realidade dos alunos e a resolução de desafios como aspecto motivador (ARAÚJO; BITTENCOURT; SANTOS, 2017).

O TryCatch tem como propósito seguir as evidências apontadas nos trabalhos citados e suprir as carências identificadas, especialmente quanto ao uso de uma metodologia ativa de aprendizagem vinculada a um jogo educacional em formato de *quiz*. Outro diferencial é o foco em dispositivos móveis, de forma que o usuário possa facilmente acessar o aplicativo por meio de *download* na loja oficial (Google Play Store) em um smartphone ou tablet.

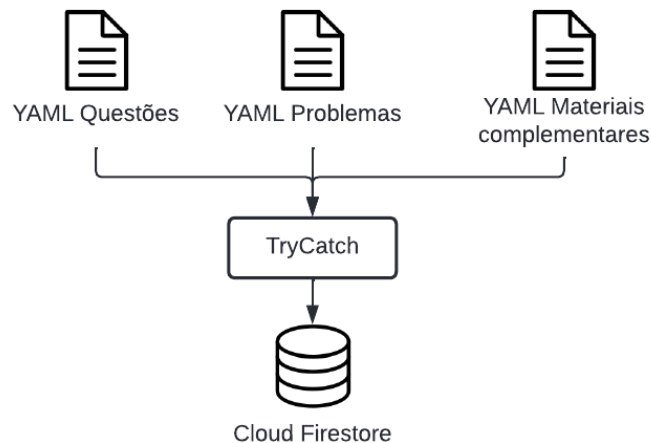
4. Sistema proposto

Inicialmente, o TryCatch foi idealizado para ser um jogo de aprendizagem. Porém, com o andamento das pesquisas, percebeu-se que, para construir um jogo no formato de *quiz*, seria necessário propor uma solução que possibilitasse não só exibir perguntas e respostas ao estudante, mas também viabilizar ao docente relacionar problemas do mundo real e recursos pedagógicos, visando uma aprendizagem que não se concentrasse em respostas baseadas na memorização.

Assim, com base em estudos identificou-se que a proposta ideal para o problema seria a proposição de um modelo de arquitetura, pois, como argumenta Sommerville (2011, p. 103), “o projeto de arquitetura está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema”. Ainda segundo o autor, a ideia é que a arquitetura proposta facilite a comunicação entre os envolvidos na produção do software, simplifique a análise do sistema e possibilite a reutilização da mesma em outros contextos.

Para D’Anuniação (2021), a definição de arquitetura é elástica, podendo abranger a estrutura do sistema, o relacionamento entre seus elementos, suas características e propriedades físicas e lógicas, sua finalidade, entre outros. Para tanto, é possível adotar determinados pontos de vista e níveis de abstração para delimitar os aspectos mais relevantes e apresentar somente o necessário para os envolvidos. A Figura 1 esquematiza o modelo de arquitetura proposto para o presente trabalho.

Figura 1 — Proposta de arquitetura do TryCatch



Fonte: Elaborado pelo autor (2022).

O TryCatch compreende um modelo de arquitetura que possibilita ao professor criar um *quiz*, definindo conteúdos, níveis de dificuldade, associando materiais complementares e estabelecendo problemas que devem ser resolvidos. A ideia é que, a partir de um problema real, o professor elabore questões vinculadas a ele e disponibilize materiais pedagógicos que auxiliarão o estudante a resolvê-lo. Destaca-se que esse modelo foi pensado de modo a incentivar que o estudante reflita sobre o problema e não use apenas a aprendizagem mecânica para responder às questões propostas.

Cada conteúdo poderá ter um número indefinido de níveis, que, por sua vez, abrigará um conjunto de questões. Estas terão formatos diversos, como múltipla escolha, verdadeiro ou falso e preenchimento de lacunas. O professor poderá cadastrar uma turma e solicitar aos alunos que se vinculem a ela por meio do compartilhamento de um código de acesso. Dessa forma, o professor terá acesso a todas as suas turmas e ao desempenho de cada aluno.

Quanto aos relatórios de desempenho, cabe ressaltar que podem ser segmentados de diferentes formas: por aluno, no qual o docente pode visualizar as respostas individuais para cada questão; por turma, no qual é exibida a média de acertos para cada questão; e por ordem de desempenho em uma turma, em que os alunos são ordenados conforme a média individual de acertos, facilitando a identificação dos estudantes que exigem maior acompanhamento pelo professor.

O Apêndice A apresenta uma descrição resumida de cada caso de uso exposto na Figura 2, visando explicitar o seu funcionamento.

Figura 2 — Diagrama UML de casos de uso do TryCatch



Fonte: Elaborada pelo autor (2022).

Outro recurso usado para a modelagem do sistema foi a prototipação das interfaces gráficas, o que permitiu materializar alguns casos de uso descritos. A Figura 3 (a) ilustra a ação de iniciar o jogo e responder às questões; a Figura 3 (b) um dos modos de se visualizar a trajetória de aprendizagem do aluno, por parte do docente; e a Figura 3 (c) a geração do código de compartilhamento após a criação da turma.

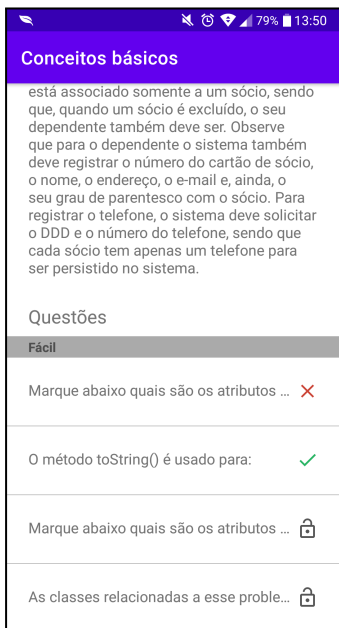
Figura 3 — Protótipos da interface gráfica do TryCatch



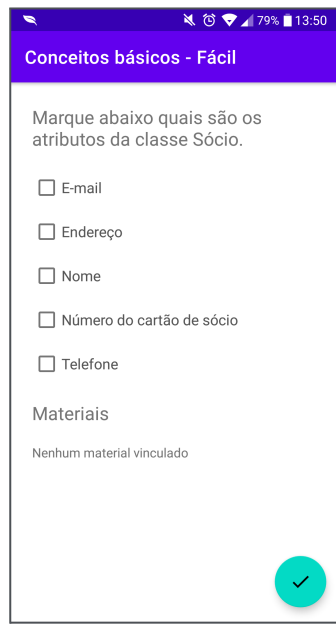
Fonte: Elaborada pelo autor (2022).

A Figura 4 apresenta algumas das interfaces gráficas desenvolvidas ao longo do trabalho. Considerando a perspectiva do estudante, é possível analisar a tela correspondente à exibição da lista de questões disponíveis em um conteúdo, em que o estudante consegue visualizar o problema e as questões que já respondeu (ícone de um cadeado e símbolo de verificação) (a), o acesso à questão (b) e o acesso aos materiais complementares (c). Por outro lado, sob a óptica do professor é possível analisar as telas relacionadas ao compartilhamento do código de acesso (d), a lista de problemas (e) e o relatório de desempenho do aluno (f).

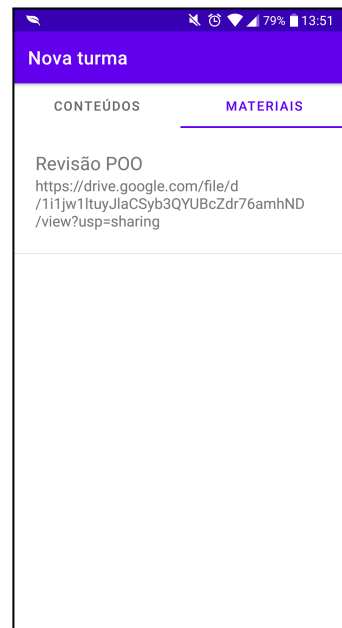
Figura 4 — Interfaces gráficas desenvolvidas



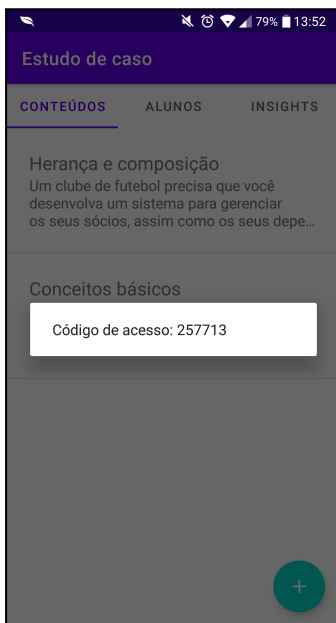
(a)



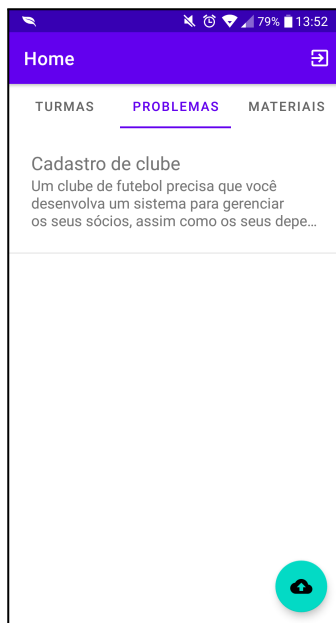
(b)



(c)



(d)



(e)



(f)

Fonte: Elaborada pelo autor (2022).

De modo a facilitar o cadastro dos recursos (questões, problemas e materiais complementares) pelo professor, optou-se por definir um template de importação em massa a partir de arquivos YAML (*YAML Ain't Markup Language*). Esse tipo de arquivo segue um formato que possui uma sintaxe breve para a declaração de pares chave-valor em comparação

com outras linguagens utilizadas em arquivos de configuração, como o JSON (*JavaScript Object Notation*) e o XML (*eXtensible Markup Language*). O Apêndice B demonstra como o arquivo pode ser utilizado para a definição de um conjunto de questões distribuídas ao longo de conteúdos e níveis de um *quiz*.

O sistema foi desenvolvido nativamente em Android, com a linguagem Java, por meio da IDE (*Integrated Development Environment*) Android Studio. Para a persistência dos dados, utilizou-se o Cloud Firestore devido à sua integração natural com a plataforma Android e possibilidade de armazenamento dos dados em nuvem.

5. Método de trabalho e cronograma

Para o desenvolvimento deste Trabalho de Conclusão de Curso as etapas abaixo foram realizadas:

1. Pesquisa exploratória: permitiu identificar quais os caminhos já abordados no uso de jogos para o ensino de POO. Com isso foi possível perceber os seus resultados e as lacunas a serem exploradas;
2. Pesquisa bibliográfica: possibilitou compreender o papel dos jogos digitais na educação, o uso de dispositivos móveis e internet pela população brasileira, a importância da acessibilidade no desenvolvimento de sistemas;
3. Análise de requisitos: através do diagrama de casos de uso foi possível identificar as principais funcionalidades para a entrega de um produto viável mínimo (MVP) do jogo;
4. Análise e projeto: através do diagrama de classes e dos protótipos de alta fidelidade foi possível compreender como a estrutura do código foi organizada e identificar a visão da interface gráfica pelo usuário;
5. Implementação e testes: essa etapa foi desenvolvida durante a realização do TCC2 com as tecnologias mencionadas previamente;
6. Estudo de caso: elaboração de um *quiz* vinculado ao ensino de POO, no qual o conteúdo foi introduzido de forma a aumentar gradativamente a dificuldade, com o intuito de validar a arquitetura;
7. Validação da metodologia e arquitetura: o estudo de caso foi usado como um instrumento para validar se a arquitetura está funcional e atende aos princípios da ABProb.

6. Estudo de caso

Para a validação da arquitetura proposta, foi elaborado um estudo de caso vinculado ao ensino de POO, que consistiu em disponibilizar uma versão do TryCatch para os usuários interagirem sob a perspectiva do estudante e um formulário para avaliação posterior. Inicialmente, foi inserido no app um problema fictício (“Cadastro de clube”), o qual foi vinculado a um conjunto de 19 questões divididas em dois conteúdos (“Conceitos básicos” e “Herança e composição”) com três níveis de dificuldade cada. Também foram disponibilizados materiais complementares para consulta. Na sequência, criou-se um formulário online com onze perguntas usando uma Escala Likert de cinco pontos¹ e

¹ A escala usada era composta pelas alternativas: discordo totalmente, discordo parcialmente, não concordo nem discordo, concordo parcialmente e concordo totalmente.

dissertativas, das quais quatro tinham como tema a ABProb e sete o aplicativo TryCatch. Os participantes foram orientados a acessar o app, responder ao *quiz* e fazer a avaliação no formulário.

Foi registrada a participação de cinco discentes, dos quais dois responderam ao *quiz* completo, dois responderam a algumas questões e um não respondeu a nenhuma. Quanto à análise posterior, foi possível identificar alguns pontos de unanimidade nas perguntas de Escala Likert relativas à ABProb e à interação com o aplicativo. Para todos os participantes, o uso da ABProb ajudou a assimilar o conteúdo e o app mostrou-se eficaz em termos da aderência à metodologia e da arquitetura proposta. Os resultados estão sintetizados no Quadro 3, cuja escala numérica corresponde da resposta “discordo totalmente” (1) à resposta “concordo totalmente” (5).

Quadro 3 — Respostas das questões de Escala Likert do formulário de avaliação

Pergunta	1	2	3	4	5
Você acredita que o uso de um problema facilitou a compreensão do conteúdo?	0%	0%	0%	0%	100%
Você acredita que relacionar conteúdos a problemas pode ajudar no processo de aprendizagem?	0%	0%	0%	0%	100%
Você acredita que a possibilidade de visualizar ou consultar o problema no momento de responder às questões facilita relacionar as questões e o conteúdo?	0%	0%	0%	20%	80%
Você acredita que aprender com problemas é melhor do que não usar problemas para contextualizar os conteúdos?	0%	0%	0%	40%	60%
Na sua opinião, o aplicativo favorece a aprendizagem de programação?	0%	0%	0%	40%	60%
Você acredita que a vinculação de material didático no aplicativo ajuda na compreensão do conteúdo?	0%	0%	0%	20%	80%
Você acredita que o uso de níveis para organizar a complexidade das questões ajuda na compreensão do conteúdo?	0%	0%	0%	20%	80%

Fonte: Elaborado pelo autor (2022).

As respostas dissertativas forneceram insumos interessantes, especialmente em relação à interação com o aplicativo. Quanto aos problemas identificados durante o uso, dois participantes relataram dificuldade inicial para compreender o funcionamento do app e sugeriram um tutorial ou tela de ajuda para solucionar o problema. Entende-se que as instruções dadas aos participantes foram mínimas e que, idealmente, a introdução ao aplicativo deveria ser mais bem contextualizada pelo docente em sala de aula. Ainda assim, sugere-se que algumas telas sejam aprimoradas para que os usuários possam assimilar intuitivamente o propósito da ferramenta.

Quanto a possíveis melhorias e novas funcionalidades, destacou-se a sugestão de apresentar os materiais complementares no próprio app em vez de direcionar para um link externo; ainda sobre este recurso, outro participante observou que não foi utilizada a formatação característica de links clicáveis. Em relação às questões, dois respondentes relataram que gostariam de visualizar a resposta correta caso errassem a questão, sendo que um deles sugeriu que fosse possível respondê-la novamente e que houvesse uma justificativa

para a veracidade da alternativa. Os pontos mencionados são válidos por estarem alinhados às melhores práticas de experiência do usuário e por convergirem com o princípio de autonomia do aluno proposto pela ABProb.

A exibição dos conteúdos e níveis foi percebida como um ponto de dúvida para dois respondentes, que aparentaram desejar uma estruturação mais fortemente ligada à progressão da dificuldade. Na configuração utilizada no estudo de caso, optou-se por não restringir o acesso às questões, mas o contrário também seria possível — dessa forma, o aluno deveria responder ao *quiz* em ordem para acessar a questão seguinte. No entanto, esse bloqueio ocorre apenas em nível de questão; como sugestão futura, poderia ser atribuída uma relação de dificuldade entre cada bloco de questões, caso o professor desejasse.

A interface do usuário também foi citada como ponto de melhoria, compreendendo a apresentação do problema e das questões em termos de apelo visual. Pelo fato de o processo de desenvolvimento do app ter sido direcionado à implementação das funcionalidades, a versão disponibilizada aos participantes é baseada no tema *default* do Material Design, com poucas customizações, o que dá margem à criação de uma identidade visual própria para o TryCatch futuramente. Ainda em relação ao aspecto visual, um respondente sugeriu que o problema pudesse conter representações para além do texto, como imagens do código ou diagramas.

Por fim, todos os participantes manifestaram apoio à adoção do aplicativo em disciplinas de programação, ressaltando a possibilidade de o aluno praticar o que está aprendendo e construir o conhecimento em seu próprio ritmo com auxílio do material disponibilizado. Destacou-se também um comentário sobre a utilização do app para avaliar o aprendizado dos alunos, funcionalidade que, embora o participante provavelmente não tivesse conhecimento, foi implementada na visão do professor por meio dos relatórios de desempenho.

7. Considerações finais

O objetivo deste artigo foi detalhar o projeto de desenvolvimento de um modelo de arquitetura para o ensino de POO. A partir das informações obtidas pela revisão bibliográfica e seleção de trabalhos relacionados, averiguou-se que o uso de jogos no formato de *quiz* como método de aprendizagem tem produzido resultados positivos e, ainda, que há uma gama de possibilidades a serem exploradas. Entre elas pode-se citar a ampliação do acesso por meio da disponibilização do sistema em forma de aplicativo para dispositivos móveis e a implementação guiada pelos princípios da Aprendizagem Baseada em Problemas. Além disso, no processo de concepção da arquitetura constatou-se a pertinência de se oferecer uma forma de acompanhamento do percurso de aprendizagem do aluno por parte do professor, o que exigiu a inclusão de novas funcionalidades.

Referências

- AMARAL, L. R.; BRAGA E SILVA, G.; PANTALEÃO, E. Plataforma Robocode como Ferramenta Lúdica de Ensino de Programação de Computadores - Extensão Universitária em Escolas Públicas de Minas Gerais. *In: Simpósio Brasileiro de Informática na Educação - SBIE*, 2016, Maceió. **Anais [...]**. Maceió: Sociedade Brasileira de Computação, 2017, pp. 200–208.
- ARAUJO, L. G. de J.; BITTENCOURT, R. A.; SANTOS, D. M. B. dos. Uma Abordagem Contextualizada para o Ensino de Programação na Educação Profissional em Informática. *In: Workshops do Congresso Brasileiro de Informática na Educação — WCBIE*, 2017, Brasília. **Anais [...]**. Brasília: Sociedade Brasileira de Computação, 2017, pp. 1021–1030.
- BACICH, L.; MORAN, J. (org.). **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: Penso, 2018.
- BARCELOS, R.; TAROUCO, L.; BERCHT, M. O uso de mobile learning no ensino de algoritmos. **RENOTE**, Porto Alegre, v. 7, n. 3, 2009, pp. 1–30.
- BERTAGNOLLI, S. de C. et al. Metodologias Ativas: possibilidades para o contexto educacional. *In: BERTAGNOLLI, S. de C.; MACHADO, R. P. (org.). Pesquisas em informática na educação: teorias, práticas e perspectivas*. Porto Alegre: IFRS, 2020, pp. 51–69.
- CGIBR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros: TIC Domicílios**. Survey on the use of information and communication technologies in Brazilian households: ICT Households 2019. Núcleo de Informação e Coordenação do Ponto BR. 1. ed. São Paulo: Comitê Gestor da Internet no Brasil, 2020. Disponível em: https://cetic.br/media/docs/publicacoes/2/20201123121817/tic_dom_2019_livro_eletronico.pdf. Acesso em: 27 jun. 2021.
- D'ANUNCIÇÃO, G. T. UFPE, 2021. Disponível em: https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/concepts/system_architecture_5F3B1E17.html. Acesso em: 28 jun. 2022.
- HENRIQUE, M. S.; REBOUÇAS, A. D. D. S. Objetos de Aprendizagem para auxiliar o ensino de conceitos do Paradigma de Programação Orientada a Objetos. **RENOTE**, Porto Alegre, v. 13, n. 2, 2015.
- MACHADO, L. D. P. et al. Uma ferramenta colaborativa para apoiar a aprendizagem de programação de computadores. **Revista Brasileira de Computação Aplicada**, v. 10, n. 1, pp. 23–29, 2018.
- MOURA, D. B. et al. Aprendendo Lógica Computacional Usando o Aplicativo AlgoFácil. *In: Anais do Congresso de Tecnologias e Desenvolvimento na Amazônia*, v. 1, p. 116–125, 2017.
- MOURA, M. P. C. et al. Protótipo de Aplicativo Educativo para o ensino de POO: Avaliação da Usabilidade e Experiência do Usuário. **RENOTE**, Porto Alegre, v. 17, n. 3, p. 305–314, dezembro, 2019.
- RICHTER, C. J.; BERNARDI, G.; CORDENONSI, A. Z. O Ensino de Programação Mediado por Tecnologias Educacionais: uma Revisão Sistemática de Literatura. **RENOTE**, Porto Alegre, v. 17, n. 1, p. 517–526, julho, 2019.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

SOUZA, M. S. C. et al. Lord of Code: uma ferramenta de apoio ao ensino de programação.
In: Simpósio Brasileiro de Informática na Educação - SBIE, 2016, Uberlândia. Anais [...].
Uberlândia: Sociedade Brasileira de Computação, 2017, pp. 1316–1320.

APÊNDICE A
CASOS DE USO DO TRYCATCH

Caso de uso	Descrição
Vincular-se a uma turma	Permite ao estudante incluir-se em uma turma por meio de um código de acesso previamente compartilhado pelo professor.
Resolver questão	Permite ao estudante iniciar o <i>quiz</i> e responder às questões.
Acessar material complementar	Permite ao estudante visualizar o repositório de materiais complementares.
Cadastrar turma	Permite ao professor criar uma turma.
Cadastrar questão	Permite ao professor cadastrar uma questão.
Cadastrar material complementar	Permite ao professor cadastrar o material que pode ser utilizado como reforço de aprendizagem.
Cadastrar problema	Permite ao professor cadastrar um problema que servirá de referência para as questões.
Vincular problema a questão	Permite ao professor vincular um problema a uma questão.
Vincular material complementar a questão	Permite ao professor vincular materiais complementares a uma questão.
Visualizar trajetória de aprendizagem do aluno	Permite ao professor visualizar o desempenho do aluno para cada questão respondida.
Gerar código de acesso à turma	Permite ao professor gerar um código que poderá ser compartilhado para o estudante vincular-se à turma.

APÊNDICE B

EXEMPLO DE ARQUIVO YAML PARA IMPORTAÇÃO DE QUESTÕES

```
subjects:
  - name: Conteúdo 1
    blocking: false
    levels:
      - name: Fácil
        questions:
          - statement: Questão de verdadeiro ou falso
            type: TRUE_OR_FALSE
            options:
              - description: Opção 1
                correct: true
              - description: Opção 2
                correct: true
          - name: Médio
            questions:
              - statement: Questão de múltipla escolha
                type: MULTIPLE_CHOICE
                options:
                  - description: Opção 1
                    correct: true
                  - description: Opção 2
                    correct: false
                  - description: Opção 3
                    correct: false
              - name: Difícil
                questions:
                  - statement: 'Questão de preenchimento: campo 1=[*], campo 2=[*]'
                    type: FILL_IN
                    options:
                      - description: Campo 1
                      - description: Campo 2
                      - description: Não utilizado
      - name: Conteúdo 2
        blocking: true
        levels:
          - name: Fácil
            questions:
              - statement: Questão de múltipla escolha
                type: MULTIPLE_CHOICE
                options:
                  - description: Opção 1
                    correct: true
                  - description: Opção 2
                    correct: false
```