



Evento: XXI Jornada de Extensão

ESTUDO DE SISTEMAS DE DESENVOLVIMENTO DE MOBILE APPS PARA ENSINO DE PROGRAMAÇÃO NA REDE BÁSICA DE ENSINO¹

STUDY OF MOBILE APP DEVELOPMENT SYSTEMS CONCERNING THE TEACHING OF PROGRAMMING TO PRIMARY AND SECONDARY EDUCATION

Gustavo Hanke Finkler², Gabriel Bauer Vieira³, Ivana Giordani Cassol⁴, Natália Krein⁵, Marcos Ronaldo Melo Cavalheiro⁶, Gerson Battisti⁷

¹ Projeto de extensão Programe o seu Futuro

² Estudante do curso de Engenharia Química da UNIJUI, Bolsista PROAV, gustavo.finkler@sou.unijui.edu.br

³ Estudante do curso de Ciência da Computação da UNIJUI, Bolsista PIBEX, gabriel.vieira@sou.unijui.edu.br

⁴ Estudante do curso de Administração da UNIJUI, Bolsista PIBEX, ivana.cassol@sou.unijui.edu.br

⁵ Estudante do curso de Engenharia de Software da UNIJUI, Bolsista PIBEX, natalia.krein@sou.unijui.edu.br

⁶ Professor coordenador do projeto de extensão Programe o seu Futuro, mrmc@unijui.edu.br

⁷ Professor orientador do projeto de extensão Programe seu Futuro, battisti@unijui.edu.br

RESUMO

O projeto de extensão Programe o seu Futuro tem como objetivo aproximar estudantes da Educação Básica do universo da computação, através de oficinas onde há o aprendizado de uma linguagem de programação. Durante o primeiro semestre de 2021, notou-se a necessidade de acrescentar outros programas às oficinas. Este resumo expandido tem como objetivo analisar cinco sistemas de programação, através de critérios definidos, para auxiliar na decisão da escolha do próximo a ser ministrado. Dentre os cinco, o programa Kodular se mostrou o mais adaptado às necessidades do projeto, por conta de sua gama de ferramentas, gratuidade de uso e adaptação da linguagem de programação para as necessidades das oficinas. O programa Thinkable também se mostrou adequado para o uso nas oficinas.

Palavras-chave: Programação. Ensino Básico. Kodular. Aplicativo *mobile*.

INTRODUÇÃO

A Tecnologia da Informação é a área da ciência responsável pelo desenvolvimento tecnológico através de meios computacionais. Ao longo dos últimos anos, esta foi tomando uma grande parcela das necessidades do mercado de trabalho (VIEIRA, 2021). O projeto de extensão Programe o seu Futuro, desenvolvido por docentes e discentes da UNIJUI, tem como objetivo apresentar e aproximar a rede de Ensino Básico do ramo da computação. Semanalmente, estudantes do 9º ano do Ensino Fundamental ao 2º ano do Ensino Médio interagem e aprendem sobre o desenvolvimento de algoritmos através de oficinas ministradas pelos integrantes do projeto.



Nos encontros que ocorreram durante o primeiro semestre de 2021, a plataforma de criação de algoritmos Scratch (SCRATCH, 2007) foi utilizada como meio de ensino. Esta foi escolhida por sua *interface* de fácil compreensão e praticidade para o desenvolvimento das atividades práticas, uma vez que não é necessário se utilizar da escrita literal dos algoritmos, pois a criação dos códigos é realizada por blocos de comando. Além disso, há, na plataforma, a possibilidade de seleção de idioma, permitindo aos estudantes que a utilizem totalmente em Português, evitando quaisquer impedimentos relacionados à língua inglesa, habitualmente usada para desenvolver linguagens de programação.

Notou-se, com o andamento das oficinas de Scratch, que os estudantes tanto adquiriram notável capacidade de entendimento quanto demonstraram bastante interesse em utilizar novas plataformas. Visto que aplicativos de celulares podem ser desenvolvidos em diversos níveis de complexidade, programas de criação de *mobile apps* se tornam candidatos para a plataforma que sucederá o Scratch. Neste resumo expandido, está definido o processo de escolha de uma plataforma de desenvolvimento de *apps* para os próximos encontros do projeto de extensão.

METODOLOGIA

Diante da variedade de possíveis sistemas de desenvolvimento de aplicativos *mobile*, foram selecionados cinco programas para a análise comparativa: Adalo (ADALO, 2019); AppInventor (APPINVENTOR, 2010); Bubble (BUBBLE, 2012); Kodular (KODULAR, 2018); e Thinkable (THUNKABLE, 2015). Esta seleção foi feita com base em uma busca pelas plataformas mais conhecidas de programação de aplicativos de celular que se utilizem de algum desenvolvimento *no code* ou *low code*; ou seja, plataformas cujo desenvolvimento de algoritmos se baseia em uma *interface* mais gráfica e mais acessível, que por evitar a escrita tradicional de códigos atende melhor a usuários que estão no processo de aprendizagem de desenvolvimento de aplicativos.

Outra vantagem das plataformas *no code* e *low code* é que o desenvolvimento do algoritmo por este meio é bem mais rápido que a programação tradicional. Isso se mostra bastante útil para criações pequenas ou protótipos de grandes projetos, como por exemplo os que partem do conceito de produto viável mínimo, ou MVP (PIUNNO, 2021). No caso das oficinas de programação, essa agilidade facilita aos estudantes que assimilem o conteúdo



ensinado, uma vez que o objetivo mais essencial das oficinas é desenvolver a capacidade de raciocínio e pensamento lógico, independente da linguagem utilizada.

Para análise dos programas selecionados, nove critérios foram listados e comparados individualmente. Em ordem de prioridade, são eles: adaptação da linguagem de programação às necessidades das oficinas; limitações do uso não-pago; requisitos computacionais; quantidade de ferramentas dentro do programa; estética da interface; possibilidade de utilização do programa em outras plataformas além do computador; idioma; quantidade de recursos para aprendizado da linguagem de programação; e atividade da comunidade e fóruns, caso existam. Estes critérios foram selecionados com base na experiência adquirida durante o primeiro semestre de 2021, com as oficinas que foram lecionadas através do Scratch; levando em consideração as dificuldades encontradas neste período, bem como as facilidades vindas da própria plataforma utilizada.

RESULTADOS E DISCUSSÃO

O resultado da pesquisa está apresentado de forma sucinta na tabela 1. Os resultados de cada avaliação foram comparados com o desejado de uma plataforma ideal, e em seguida definidos conforme três possibilidades: Suficiente, Mediano e Insatisfatório.

Tabela 1 - Avaliação de Diferentes Programas de Desenvolvimento de Mobile Apps

CRITÉRIOS	ADALO	APPINVENTOR	BUBBLE	KODULAR	THUNKABLE
Linguagem de Programação	Insatisfatório	Suficiente	Insatisfatório	Suficiente	Suficiente
Limitações do Uso Gratuito	Insatisfatório	Suficiente	Insatisfatório	Suficiente	Mediano
Requisito Computacional	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
Quantidade de Ferramentas	Insatisfatório	Mediano	Mediano	Suficiente	Suficiente
Estética da Interface	Suficiente	Suficiente	Suficiente	Suficiente	Suficiente
Diversidade de Plataformas	Insatisfatório	Insatisfatório	Insatisfatório	Insatisfatório	Insatisfatório
Idioma do Programa	Mediano	Mediano	Mediano	Suficiente	Mediano
Recursos de Aprendizado	Insatisfatório	Mediano	Suficiente	Suficiente	Suficiente
Atividade da Comunidade	Insatisfatório	Mediano	Insatisfatório	Mediano	Suficiente
Avaliação Geral	Insatisfatório	Mediano	Insatisfatório	Suficiente	Suficiente

Fonte: autoria própria

No que concerne à adaptação da linguagem para as oficinas, O AppInventor, o Kodular e o Thinkable se utilizam da mesma escrita, que é baseada em blocos de comando, semelhante à usada pelo Scratch. Ambas as plataformas Adalo e Bubble, no entanto, tem a



criação de algoritmos fortemente baseada em *no code*, ou seja, evitando quando possível conceitos considerados fundamentais para a codificação literal, podendo fazer com que o estímulo ao pensamento lógico do estudante seja inferior ao desejado. Esta avaliação, acompanhada de altas limitações para o uso gratuito das plataformas, são suficientes para justificar o descarte do Adalo e do Bubble como candidatos ao ensino nas oficinas do projeto.

Dentre os três outros programas em questão, somente no Kodular foi encontrada a opção de alterar o idioma para o Português, embora esta não abranja a totalidade da plataforma. Além disso, o Kodular foi a plataforma que se mostrou a mais completa referente a ferramentas e possibilidades de criação, seguida do Thunkable. Conclui-se que, dentre as cinco possibilidades, o Kodular mostrou-se a mais adequada.

Ainda assim, o Thunkable também se mostrou bastante adequado para as atividades do projeto. A diferença mais marcante deste para o Kodular é uma limitação de dez projetos por usuário, e a impossibilidade de mantê-los de forma privada sem utilizar o plano pago. Dessa forma, mesmo não sendo a melhor oportunidade, encaixa-se bem nas necessidades das oficinas e pode vir a ser utilizado no futuro.

CONSIDERAÇÕES FINAIS

O ensino da programação dentro do Ensino Básico traz à classe mais jovem a vasta gama de possibilidades de criação e desenvolvimento de algoritmos, bem como todas as possibilidades de ingresso no mercado de trabalho através deste ramo. Além disso, o raciocínio lógico e a resolução de problemas por etapas é estimulada durante todo o processo de aprendizado.

Visto que, durante o primeiro semestre do ano corrente, os estudantes das oficinas do projeto de extensão manifestaram interesse em utilizar novas plataformas de desenvolvimento, a introdução de um sistema de criação de aplicativos *mobile* mostrou-se interessante. Dentre os programas analisados, o Kodular foi escolhido como o mais adaptado para uso nas oficinas do projeto. A plataforma Thunkable também cumpriu muitos dos requisitos utilizados na análise, e poderá ser utilizada como substituta do Kodular em um momento futuro, caso haja a necessidade.

Esta avaliação levou em consideração apenas plataformas *low* e *no code*, uma vez que essas categorias foram consideradas mais adaptadas às necessidades das oficinas.



Dependendo das necessidades do projeto Programe o seu Futuro, um possível prosseguimento desta pesquisa pode levar em consideração plataformas que se utilizam da programação clássica para desenvolvimento de algoritmos.

REFERÊNCIAS BIBLIOGRÁFICAS

ADALO. Inside Adalo. c2019. Disponível em: <<https://www.adalo.com/about-us>>. Acesso em: 7 ago, 2021.

APPINVENTOR. About Us. c2010. Disponível em: <<https://appinventor.mit.edu/about-us>>. Acesso em: 7 ago, 2021.

BUBBLE. Bubble. c2012. Disponível em: <<https://bubble.io>>. Acesso em: 7 ago, 2021.

KODULAR. Kodular. c2018. Disponível em: <<https://www.kodular.io>>. Acesso em: 7 ago, 2021.

PIUNNO, Tony. Why low code is becoming the digital transformation MVP. **Hyland**, 2021. Disponível em: <<https://blog.hyland.com/digital-transformation/why-low-code-is-becoming-the-digital-transformation-mvp>>. Acesso em: 7 ago, 2021.

SCRATCH. Sobre o Scratch. c2007. Disponível em: <<https://scratch.mit.edu/about>>. Acesso em: 7 ago, 2021.

THUNKABLE. Thinkable: Build powerful, native mobile apps without coding. c2015. Disponível em: <<https://thinkable.com/#>>. Acesso em: 7 ago, 2021.

VIEIRA, Letícia. Vagas em TI superam crescimento de 2020 e aumentam 20% no primeiro trimestre. **Uol**, 2021. Disponível em: <<https://6minutos.uol.com.br/carreira/vagas-em-ti-superam-crescimento-de-2020-e-aumentam-20-no-primeiro-trimestre/>>. Acesso em: 7 ago, 2021.