



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
TOCANTINS
CAMPUS DIANÓPOLIS
CURSO DE LICENCIATURA EM COMPUTAÇÃO

INAÊ RIBEIRO DOS SANTOS

**ANÁLISE DE APLICATIVOS MÓVEIS COMO FERRAMENTAS DE INCLUSÃO
PARA O ENSINO DE PROGRAMAÇÃO**

DIANÓPOLIS
2022

INAÊ RIBEIRO DOS SANTOS

**ANÁLISE DE APLICATIVOS MÓVEIS COMO FERRAMENTAS DE INCLUSÃO
PARA O ENSINO DE PROGRAMAÇÃO**

Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Licenciatura em
Computação do *Campus* Dianópolis Instituto
Federal do Tocantins, como exigência à obtenção
do título de licenciado em Computação.
Orientador: Professor Dr. Lucas Arruda Ramalho

**DIANÓPOLIS
2022**

**Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins**

R484a Ribeiro Dos Santos, Inaê
ANÁLISE DE APLICATIVOS MÓVEIS COMO FERRAMENTAS DE
INCLUSÃO PARA O ENSINO DE PROGRAMAÇÃO / Inaê Ribeiro Dos
Santos. – Dianópolis, TO, 2022.
32 f. : il. color.

Trabalho de Conclusão de Curso (Licenciatura em Computação) –
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, Campus
Dianópolis, Dianópolis, TO, 2022.

Orientador: Dr. Lucas Arruda Ramalho

1. Aplicações Móveis, Ensino. 2. Interface Gráfica de Usuário. 3. Lógica
de Programação.. I. Arruda Ramalho, Lucas. II. Título.

CDD 004

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins de estudo e
pesquisa, desde que citada a fonte.

**Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos pelo(a)
autor(a).**

INAÊ RIBEIRO DOS SANTOS

**ANÁLISE DE APLICATIVOS MÓVEIS COMO FERRAMENTAS DE INCLUSÃO
PARA O ENSINO DE PROGRAMAÇÃO**

Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Licenciatura em
Computação do *Campus* Dianópolis Instituto
Federal do Tocantins, como exigência à obtenção
do título de licenciado em Computação.

Aprovado em: _____/_____/_____

BANCA AVALIADORA

Prof. Dr. Lucas Arruda Ramalho
IFTO – *Campus Dianópolis*

Prof^a. Msc. Andrea Barboza Proto Sardi
IFTO – *Campus Dianópolis*

Prof. Msc. Tássio Guerreiro Antunes Virgínio
IFTO – *Campus Dianópolis*

DIANÓPOLIS-TO

2022

“Inovar não é pensar fora da caixa é ter certeza que ela não existe.”

Aguinaldo Marques COO da CYKLO (2022)

AGRADECIMENTOS

Agradeço primeiramente a Deus, tudo nele posso sem ele nada posso. Agradeço aos meus pais que viram a educação não como um luxo, mas como um investimento. Agradeço também ao meu orientador Lucas Arruda Ramalho pela dedicação e comprometimento ao me orientar com maestria nesse belo trabalho aqui apresentado. Por fim, agradeço ao IFTO que me deu o prazer de conviver com colegas estudantes e professores durante esses anos. E boa sorte a mim, boa sorte a vocês que um dia terão o prazer do meu trabalho ler.

RESUMO

Este trabalho tem a finalidade de analisar as ferramentas móveis Grasshopper, Dcoder e Replit, focando nos pontos fortes e fracos de suas interfaces. Os principais desafios na utilização estão ligados à sua adaptação ao tamanho reduzido das telas touchscreen dos smartphones que são usados como teclado e visualização, o que traz desvantagem em relação a um computador comum. Por isso, este trabalho apresenta como tema a análise de tecnologias móveis para o ensino de programação, com o objetivo principal de apresentar um estudo quali-quantitativo em relação aos editores de código e sistemas disponíveis para prática de programação através dos dispositivos móveis. Sendo assim, pretende-se investigar como as tecnologias móveis podem auxiliar os estudantes da área da tecnologia nas disciplinas de programação. Podemos concluir que este trabalho propõe uma pesquisa significativa em relação à disciplina de lógica de programação, por considerar esta uma disciplina chave para entrada e permanência de alunos nos cursos de computação. Durante o processo de pesquisa outros aplicativos de programação para uso em dispositivos móveis serão testados e analisados. Espera-se com isso que os estudantes e professores da área da tecnologia conheçam os aplicativos existentes, suas vantagens e desvantagens.

Palavras-chave: Aplicações Móveis, Ensino, Interface Gráfica de Usuário, Lógica de Programação.

ABSTRACT

This work aims to analyze the Grasshopper, Dcoder and Replit mobile tools, focusing on the strengths and weaknesses of their interfaces. The main challenges are linked to its adaptation to the reduced size of touchscreens on smartphones that are used as a keyboard and visualization, which brings a disadvantage in relation to a common computer. Therefore, this work presents as its theme the analysis of mobile technologies for programming teaching, with the main objective of presenting a quali-quantitative study in relation to the compilers and systems available for programming practice through mobile devices. Therefore, it is intended to investigate how mobile technologies can help technology students in programming disciplines. We can conclude that this work proposes a significant research in relation to programming logic, as it is considered a key subject for students to enter and remain in computing courses. During the research process, other programming applications for use on mobile devices will be tested and analyzed. It is expected that students and teachers in the area of technology know the existing applications, their advantages and disadvantages.

Keywords: Mobile Applications, Teaching, Graphical User Interface, Programming Logic.

LISTA DE ILUSTRAÇÕES

Figura 1 - Screenshots do Dcoder	10
Figura 2. Screenshots do teclado do Dcoder.....	10
Figura 3 - Screenshots do Replit	10
Figura 4. Screenshots do teclado do Replit.....	10
Figura 5 - Screenshots do Grasshopper	10
Figura 6. Screenshots do teclado do Grasshopper.....	10

LISTA DE TABELAS

Tabela 1 - Avaliação descritiva e heurística	10
--	----

LISTA DE ABREVIATURAS

ANATEL - Agência Nacional de Telecomunicações

CEO -Chief Executive Officer

IBGE - Instituto Brasileiro de Geografia e Estatística

ID -Inclusão Digital

IDE - Integrated Development Environment

IHC - Interface Homem-Computador

IHM - Interface Homem-Máquina

UFPR - Universidade Federal do Paraná

USP - Universidade de São Paulo

SUMÁRIO

1. INTRODUÇÃO	9
2 REVISÃO DA LITERATURA	11
2.1 Inclusão digital	11
2.2 Tecnologias mobile	11
2.3 Desafios da disciplina de lógica de programação	12
2.4 Interface Homem-Máquina (IHM)	13
3 METODOLOGIA DA PESQUISA	15
4 PROCESSO DE ANÁLISE DAS FERRAMENTAS	17
4.1 Dcoder	17
4.2 Replit	18
4.3 Grasshopper	20
4.4 Resultados e discussões	22
<i>4.3.1 Tabela de avaliação descritiva e heurística</i>	22
5 CONSIDERAÇÕES FINAIS	28
REFERÊNCIAS	29

1. INTRODUÇÃO

A disciplina de lógica programação é a porta de entrada dos estudantes no mundo do desenvolvimento de software e nos cursos de computação. No trabalho desenvolvido na Universidade Federal do Paraná (UFPR), Krzyzanowski (2019) foi apresentada uma análise em relação à evasão e retenção nas disciplinas de lógica de programação. Na análise, 33% dos ingressantes evadem nos primeiros semestres e destes, 79% tiveram a evasão provocada pela reprovação nesta disciplina. Já Bosse e Gerosa (2015), realizaram na Universidade de São Paulo (USP) uma análise do índice de reprovação na disciplina através dos anos de 2010 a 2014. Na pesquisa, a média de retenção encontrada foi de 30%, sendo que em vários casos ultrapassam os 50%. Esses dados mantêm-se constantes ao longo dos anos e mais de 25% dos alunos aprovados neste período fizeram 2 ou mais vezes a disciplina de lógica de programação, reforçando a característica de retenção da mesma.

Os desafios da disciplina passam da mudança do *mindset* de resolução de problemas até a simples disponibilidade de computadores na casa dos estudantes. Este último fator auxilia que os acadêmicos sejam capazes de praticar em casa o que aprenderam na disciplina. Entretanto, mesmo atualmente, várias casas dos brasileiros não têm condições de adquirir uma máquina adequada para a tarefa.

“O IBGE define o rendimento domiciliar per capita como a soma da renda de cada morador dividido pelo total de residentes. Pensionistas, domésticos e familiares estão inclusos na conta. O Tocantins aparece no gráfico com a renda correspondente a R\$ 1.055,60.” (Renda domiciliar per capita no Tocantins é de R \$1.055,60, revela IBGE, Gazeta do Cerrado, Tocantins, 04/03/2020).

“O estudo mostra que os preços dos computadores foram impactados pelo ajuste do ICMS no começo de janeiro e ficaram cerca de 20% mais caros. No primeiro trimestre do ano passado (2020), um desktop custava, em média, R \$3.146,00 e um notebook, R\$3.692,00. No primeiro trimestre de 2021, os preços médios foram para R \$3.842,00 e R \$4.450,00, alta de 22,1% e 20,5%, respectivamente, segundo o IDC Brasil.” (Venda e preço médio de PCs crescem no 1º trimestre de 2021, noomis FEBRABAN TECH,02/07/2021)

Diante do exposto, a pesquisa sobre as tecnologias mobile para o ensino de programação surgiu do fato de que a realidade econômica enfrentada por muitos estudantes de computação dificulta a aquisição de computadores com tecnologia compatível para utilização de compiladores (entre outros softwares necessários para um curso superior em computação). Por isso, é importante avaliar se os estudantes que não possuem um computador desktop/notebook podem ser incluídos

no processo ensino-aprendizagem de programação igualmente ao utilizar ferramentas de desenvolvimento de programação adaptadas a tecnologia *mobile*.

A disciplina muitas vezes exige prática de exercícios como atividade extraclasse para garantia de fixação dos recursos de programação. Entretanto, a realização dessas atividades nos dispositivos móveis é muito diferente dos computadores desktop ou notebook. As principais limitações dos dispositivos móveis são ligadas ao tamanho reduzido de suas telas *touchscreen* que são usadas como teclado e visualização, o que é uma desvantagem em relação aos computadores desktop ou notebook. Por isso, o objetivo deste trabalho é apresentar um estudo quali-quantitativo em relação aos compiladores e sistemas disponíveis para prática de programação através dos dispositivos móveis. Os aplicativos analisados são Dcoder, Replit e Grasshopper. Sendo assim, pretende-se investigar como as interfaces dos aplicativos podem trazer benefícios aos estudantes da área da tecnologia nas disciplinas de programação.

Este trabalho está organizado em três capítulos. Sendo no capítulo 2, “Revisão Literária”, onde aborda se a fundamentação teoria da temática discutida; no capítulo 3, “Metodologia da pesquisa”, faz se um detalhamento de como será a realizada a pesquisa e os seus métodos utilizados para obter os dados desejados; no capítulo 4, “ Processo de análise das ferramentas”, apresenta todo o processo da pesquisa até a obtenção dos dados para análise da comprovação da pesquisa, para concluir traz se algumas considerações acerca dos dados obtidos na pesquisa.

2 REVISÃO DA LITERATURA

2.1 Inclusão digital

A inclusão digital é uma ferramenta social que tem como principal objetivo deixar a tecnologia acessível à sociedade. Desta forma a inclusão permite que o indivíduo possa obter oportunidades profissionais em uma área que possui boa empregabilidade e alta demanda.

Conforme citado por Abdalla (2005), “Entendemos por Inclusão Digital (ID) a possibilidade de acesso dos cidadãos de uma sociedade às tecnologias de comunicação e informação, que incluem, entre outras, os computadores e serviços de internet. Segundo a Fundação Getúlio Vargas [FGV, 2003] este acesso pode ser dividido em três tipos: acesso ao capital físico (computador, periféricos etc.), capital humano (i.e., aulas de informática, educação básica etc.) e capital social (internet e outras formas de associativismo). Assim, para que a inclusão digital seja completa e sustentável é preciso que o cidadão se aproprie das tecnologias de informação de forma ampla, consciente e autônoma. Num país como o Brasil a inclusão digital tem o forte potencial de promover inclusão social, possibilitando que uma considerável camada da população participe ativamente da vida política, social e cultural do país. Além de melhorar sua capacitação para atuar no mercado de trabalho.”

Ao falar e até mesmo trabalhar com inclusão, se faz necessária atenção para não causar exclusão dos indivíduos. Considerando este trabalho, a inclusão deve deixar todos os indivíduos em iguais condições de acesso à educação e ao aprendizado de lógica de programação.

2.2 Tecnologias mobile

A tecnologia mobile, que também pode ser chamada de tecnologia móvel, dá acesso às informações e desenvolvimentos tecnológicos por meio dos dispositivos móveis, celulares e smartphones. Segundo Ledesma (2013, p. 45) “Mobile learning é uma das derivações da educação a distância ou e-Learning. O conceito mobile learning pode ser traduzido para português por aprendizagem móvel ou entendido como integração das tecnologias móveis em contexto educativo.” A tecnologia mobile é parte do princípio de tornar o seu dispositivo móvel em uma ferramenta de trabalho/estudo uma vez que ele é uma ferramenta de baixo custo e portátil.

Ledesma (2013, p. 45), afirma que “A definição de uma teoria específica ou de teorias e metodologias mais adequadas à integração do mobile learning ainda é uma área em exploração e desenvolvimento, no entanto, conforme vamos experimentando, vamos (re)adaptando algumas metodologias que permitam conduzir ao desenvolvimento de literárias na área da informação, comunicação/interação, produção/criação e segurança neste ambiente de aprendizagem digital tão interativo.”

Segundo Higuchi (2011), pode se dizer que o aparelho celular está no centro desta expansão tecnológica móvel, pois poucas tecnologias tiveram uma curva de difusão tão rápida. No Brasil, em 2011, as pesquisas apontavam que havia mais do que um celular por habitante, segundo a Teleco – Inteligência em Telecomunicações, para um grupo de 100 mil habitantes, 104,7 mil possuem aparelhos celulares. Conforme a ANATEL - Agência Nacional de Telecomunicações, em 2009, o país já era considerado como o quinto no mercado mundial e a maioria dos usuários costuma utilizar o sistema pré-pago (82%).

Sendo assim, podemos dizer que apesar da popularidade dos aparelhos e aplicativos móveis, seu uso no ensino dentro do Brasil ainda sofre constantes experimentações e adaptações. A análise do potencial das ferramentas móveis no ensino-aprendizagem de lógica de programação se torna necessária para fortalecer o processo de inclusão digital e profissional dos estudantes com menores condições de acesso a computadores.

2.3 Desafios da disciplina de lógica de programação

A disciplina de lógica de programação tem como objetivo principal em sua ementa ensinar os conceitos de algoritmo e suas formas de representação, suas competências sendo focadas em aplicar os conceitos fundamentais de algoritmos computacionais e a manipulação de dados. Tendo como habilidades desenvolver o conhecimento a respeito dos fundamentos básicos de programação.

Segundo Lopes (2002), a lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo. O aprendizado desta técnica é necessário, para quem deseja trabalhar com desenvolvimento de sistemas e programas. Algoritmo é uma sequência de passos finitos com o objetivo de solucionar um problema.

O cenário que os cursos de tecnologia enfrentam é a evasão provocada pela grande retenção na disciplina de lógica de programação. Estas disciplinas estão presentes nos primeiros períodos de cursos técnicos e superiores na área de computação. De acordo com Janice Inês et al. (2008) apud Rocha (2003) as disciplinas de lógica, algoritmos e programação iniciam com uma média de 50 alunos, e em poucos meses constata-se que a taxa de reprovação (ou desistência) chega a 60%. Segundo Janice Inês et al. (2008) apud Galhardo (2006), os altos índices de reprovação estão relacionados diretamente a forma como ocorre o processo de ensino-aprendizagem.

Os principais fatores da evasão por repetência estão ligados aos seguintes pontos: processo de ensino-aprendizagem, compreensão de programas, aprendizagem da sintaxe, dificuldade com a língua inglesa, interpretação de texto, falta de lógica e capacidade de resolver problemas e falta de

uma máquina para praticar em casa. A prática de programação em casa, força ao aluno o desenvolvimento de autonomia no pensamento e da criatividade no formato das soluções propostas por ele. Quando o estudante não pratica em casa, ele tende a criar uma dependência nas soluções propostas pelo professor, o que inibe cada vez mais a confiança e pensamento crítico próprio.

2.4 Interface Homem-Máquina (IHM)

A Interface Homem-Máquina (IHM) é um mecanismo facilitador entre pessoas e máquinas, de uma forma geral tem o trabalho de apresentar ao usuário de uma forma clara e objetiva as funcionalidades de um determinado sistema UX design, verificar se sua interface UI design e demais funcionalidades são intuitivas ao uso do usuário.

A IHM surgiu por conta do avanço da dita computação pessoal, que hoje é a popular tecnologia digitais. Cavallin (2015) apud Carroll (2009), o marco principal que efetivou o surgimento do IHM foi o surgimento da computação pessoal e a consequente popularização das tecnologias digitais. Na mesma linha de pensamento vem Becker (2008), que caracteriza a invenção e a difusão do computador pessoal como um momento decisivo na história da IHM, no qual milhares de usuários começaram a interagir com os computadores.

Rocha (2021) apud EAO (2016), afirma que as IHMs abrangem todos os elementos que uma pessoa vai tocar, ver, ouvir ou usar para executar funções de controle e receber um retorno sobre essas ações. A IHM em relação a usabilidade, User Experience (UX) e User Interface (UI) dos dispositivos móveis está relacionada em oferecer características e funcionalidades que prendam a atenção do usuário.

Para Nielsen, a usabilidade hoje tem importância muito maior do que no passado, pois, em dispositivos móveis, por exemplo, a usabilidade para prender o foco do usuário não é algo importante, mas vital para a sua aplicação. Neste sentido, a usabilidade é a qualidade que caracteriza o uso de um sistema interativo. Ela se refere à relação que se estabelece entre usuário, tarefa, interface, equipamento e demais aspectos do ambiente no qual o usuário utiliza o sistema (FEIJÓ, 2013 apud CYBYS, 2007, p. 23).

Como o objetivo de apresentar uma metodologia de avaliação de interfaces, Jacob Nielsen criou em 1994 o modelo de avaliação das 10 heurísticas de Nielsen que realiza uma análise qualitativa da usabilidade, e também de forma quantitativa atribui uma nota de severidade para cada uma das métricas. Este trabalho criou diferentes métricas que buscam avaliar as características da interface que auxiliam o usuário na programação de código. Entretanto, a avaliação utilizará as notas de severidade como critério quali-quantitativo aliado à análise qualitativa. Assim, será possível descrever as características de cada interface e através das notas será possível compará-las.

No que compete a adaptação de aplicações dos computadores para os smartphones, a usabilidade será melhor atingida quando a adequação não se limitar apenas na redução de informações e botões, mas sim no entendimento que a mídia utilizada apresenta características de uso diferentes. Assim, haverá o surgimento de adaptações que encaixam no perfil de uso dos celulares e permitam uma interface agradável e de fácil interação.

3 METODOLOGIA DA PESQUISA

A metodologia a ser utilizada foi a quali-quantitativo, que de acordo com o pesquisador Gil (2008), por meio desse tipo de abordagem quali-quantitativa é utilizado os métodos quantitativos e os qualitativos para assim mensurar e analisar os procedimentos da pesquisa.

Desse modo, nesta pesquisa será levantado e quantificado os principais recursos dos aplicativos Grasshopper, Dcoder e Replit, que apresentam soluções de programação e aprendizagem adaptados às tecnologias móveis.

A pesquisa se caracteriza como bibliográfica por ter utilizado teorias retiradas de artigos científicos e livros de natureza eletrônica, online e impressa que tratam sobre o tema o uso de tecnologias móveis como ferramenta de inclusão para o ensino de programação, pois conforme Gil (2008), esse método de pesquisa é caracterizado por utilizar materiais como livros e artigos científicos já prontos, proporcionando para o pesquisador acesso a diversidade de informações.

De acordo com o objetivo a pesquisa se classifica como descritiva, que segundo afirma Gil (2008), tem como objetivo descrever as características de um grupo específico ou de determinado fenômeno, nesta ocasião obtendo dados por meio de uma análise descritiva de certas ferramentas tecnológicas.

Essa pesquisa estudará as ferramentas Grasshopper, Dcoder e Replit, para uma análise que trará as vantagens e desvantagens do uso das ferramentas, proporcionando aos estudantes melhores perspectivas de estudo de programação, a análise realizada foi baseada em teste e dificuldades já relatadas no ensino de programação. Para avaliação das ferramentas serão feitos comparativos entre ambas, baseada na avaliação heurística de usabilidade.

De acordo com Barros (2017), Jakob Nielsen e Rolf Molich em 1990, desenvolviam as heurísticas de avaliação de usabilidade que posteriormente seriam aprimoradas no ano de 1994 por Nielsen dando origem ao que hoje é conhecido como as 10 heurísticas de Nielsen.

Como dito acima a heurística são divididas em 10 requisitos de teste de usabilidade sendo eles: visibilidade do estado do sistema, compatibilidade do sistema com o mundo real, controle e liberdade do usuário, consistência e padrões, prevenção de erros, reconhecimento em vez de lembrança, flexibilidade e eficiência de uso, estética e design minimalista, ajudar usuários a reconhecer, diagnosticar e corrigir erros, ajudas e documentação.

Conforme Barros (2017) apud Nielsen (1993), define usabilidade como sendo um requisito de qualidade de software que permitam um sistema computacional ser fácil de utilizar e aprender, características essenciais para obter qualidade. Quanto melhor for a usabilidade de um sistema mais

simples e satisfatório é utilizá-lo, logo sua utilização passa a ser mais produtiva uma vez que o desgaste é significativamente reduzido.

Nessa pesquisa, o método de quantificação das heurísticas foi utilizado. Entretanto, como a análise das interfaces é voltada especificamente para ensino-aprendizagem de programação, analisamos as características que fazem parte da utilização dos estudantes no processo de programação. São elas:

- Teclado Virtual - Se a qualidade e usabilidade do teclado virtual, quais teclas e atalhos estão disponíveis no auxílio ao programador;
- Armazenamento na Nuvem - Para o compartilhamento de programas criados entre o dispositivo móvel e computadores disponíveis na instituição de ensino;
- Interatividade da Interface - Se a forma de interação serve a um programador iniciante;
- Destaque de Botões e Controles - Se há visibilidade dos botões e controles na interface reduzida dos aplicativos;
- Adaptação da Interface ao tamanho da Tela - Se área de edição, tamanho de fonte são adequados à tela;
- Simbologia - Se os símbolos utilizados são intuitivos ao programador;
- Acessibilidade a menus e funções - Se menus e controles estão acessíveis durante o uso;
- Funções de autocompletar - Se a função de autocompletar existe e se ajuda o programador iniciante;
- Interface resultante da compilação - Se o resultado da compilação de código pode ser visualizado adequadamente.

4 PROCESSO DE ANÁLISE DAS FERRAMENTAS

4.1 Dcoder

O Dcoder é uma IDE mobile criada pelo CEO Ankush Chugh que disponibiliza de um editor de código, o mesmo dá a possibilidade do usuário programar em oito linguagens sendo elas C, C#,CPP, Python, PHP, Java, Ruby, Objective-C entre outras. Na figura 1 apresentamos a interface da ferramenta, o que se pode destacar é que a mesma possui uma interface interativa e intuitiva para quem já tem experiência em programação.

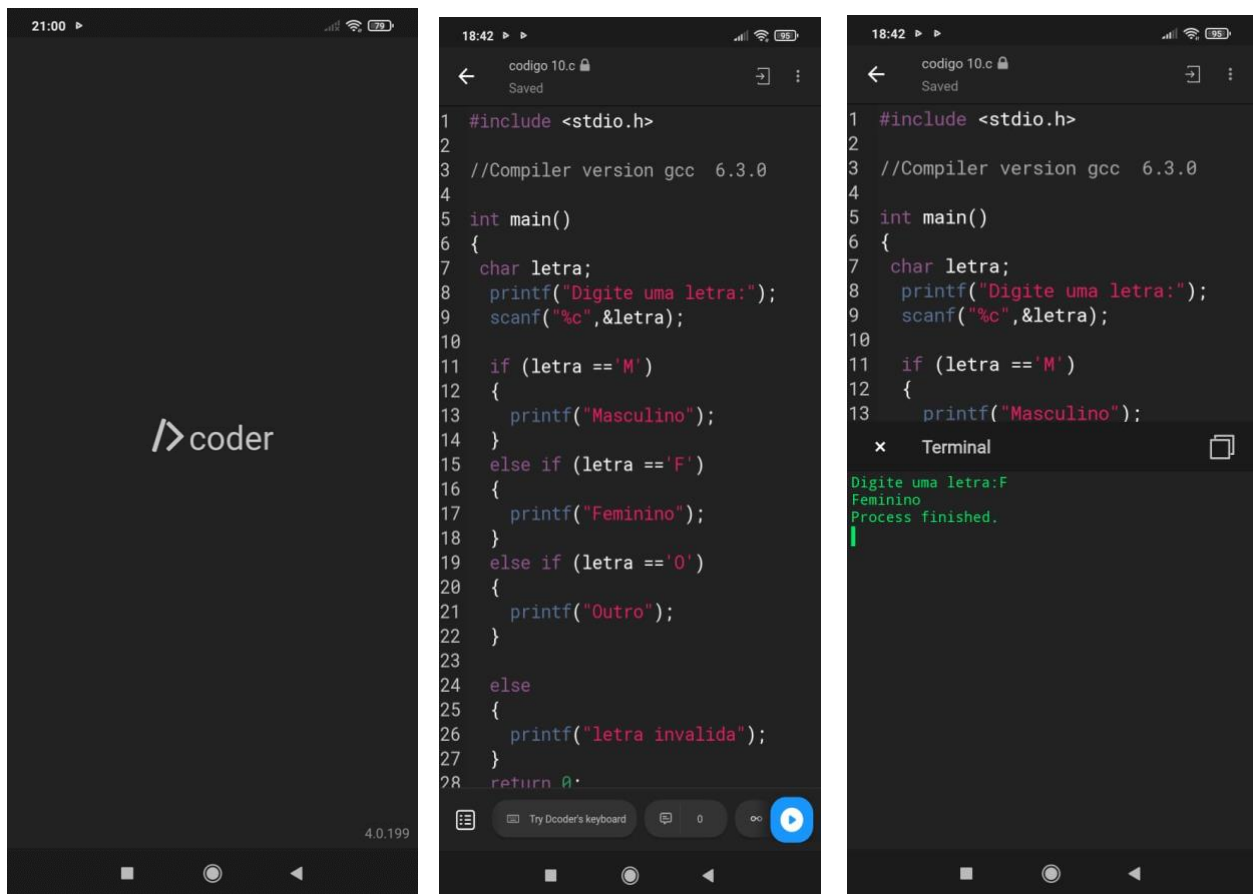


Figura 1. Screenshots do Dcoder

Analisando a imagem podemos perceber que a ferramenta possui a característica do design dos compiladores de um computador. Uma coisa negativa é que por ser uma versão de um editor de código de computador suas telas foram reduzidas para versão mobile, o que acaba prejudicando o usuário na hora do seu trabalho. A vantagem é que seus ícones e botões remetem aos dos aplicativos que os usuários usam no seu dia a dia, com se pode ver na imagem o botão de executar parecido com o do play.

Na questão da acessibilidade aos menus e funções é que uma vez que o usuário já teve um contato com um editor de código ele facilmente terá um bom desempenho, pois a ferramenta tem uma semelhança com o compilador de um computador.

Como apresentado na figura 2, temos evidência que seu teclado possui um recurso que facilita o desempenho do usuário em relação a está utilizando de uma ferramenta mobile. Um desses recursos é o TAB que está localizado na parte superior do teclado, sabemos que o TAB na programação é um recurso muito utilizado nos comandos.

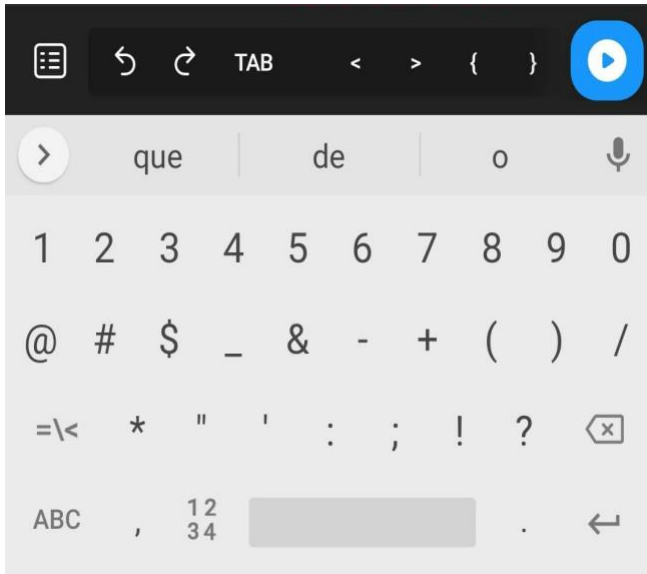


Figura 2. Screenshots do teclado do Dcoder

4.2 Replit

O Replit é uma IDE criada pelos programadores Amjad, Frai Masad juntamente com a designer Haya Odeh e com cofundação de Masad, Max Shawabkeh no ano de 2016.

A IDE Replit ela permite que o usuário possa criar projetos, editar e testar o mesmo, o bom dessa ferramenta é que o usuário pode programar em diversas linguagens de programação, vale lembrar que o usuário pode compartilhar seus códigos e projetos com outros desenvolvedores.

Na figura 3 apresentamos a interface da ferramenta, onde podemos visualizar que a mesma possui uma interface interativa e intuitiva, mas vale destacar que é para quem já tem experiência em programação. Ao analisar com um olhar crítico percebermos que o usuário terá que ter um certo domínio de lógica e inglês, para ter um melhor desempenho na ferramenta.

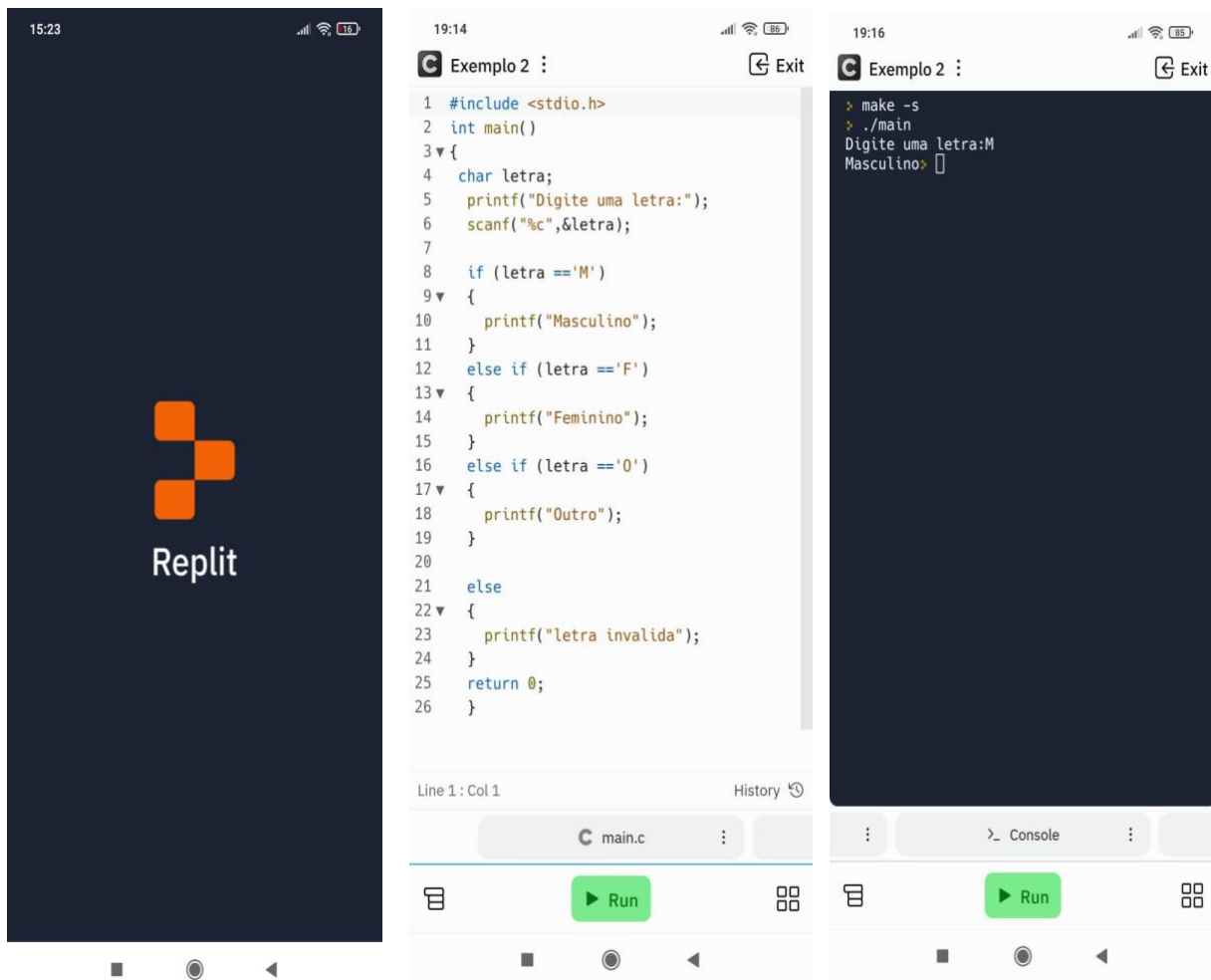


Figura 3. Screenshots do Replit

Nesta figura 4 apresentamos o teclado da ferramenta dando destaque aos recursos que ele disponibiliza e dos que ele deveria disponibilizar. O seu diferencial está em sua parte superior onde ele dispõe dos recursos mais utilizados nas linguagens de programação, por exemplo as chaves e colchetes, logo em seguida em outra aba ele disponibiliza algumas funções específicas que muito se utiliza na programação. Um diferencial em relação aos outros é o seu botão central que está localizado na segunda aba onde o mesmo tem a finalidade de ver o histórico.

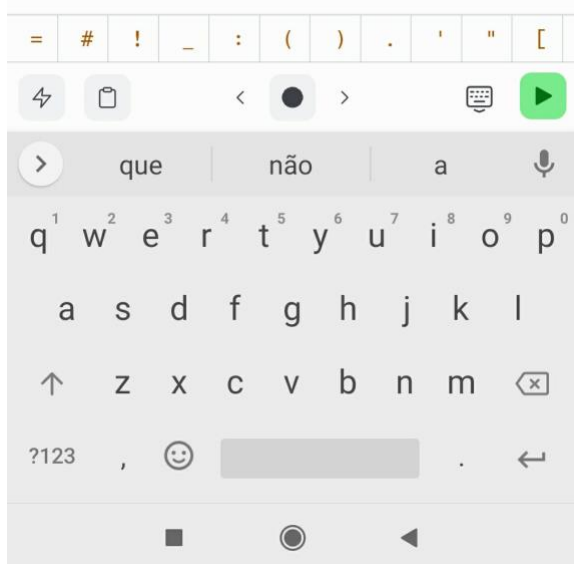


Figura 4. Screenshots do teclado do Replit

4.3 Grasshopper

O Grasshopper é uma ferramenta voltada ao ensino e apresenta uma jornada de conhecimento que promete ensinar lógica de programação e os principais recursos da linguagem Javascript.

O Grasshopper foi criado por uma equipe de programadores da oficina do google para produtos experimentais, ele foi criado com o intuito de ajudar as pessoas a codificar de uma maneira mais fácil e dinâmica.

Na figura 5 apresenta se a ferramenta, que com um primeiro olhar podemos perceber que sua interface é interativa e intuitiva, vale ressaltar que ela é de fácil entendimento para o usuário que não tem experiência em programação, seu ponto negativo é que como pode ser visto ele tem um aspecto de jogos infantis o que faz o usuário achar que é uma jogo é não uma ferramenta para o ensino de programação.

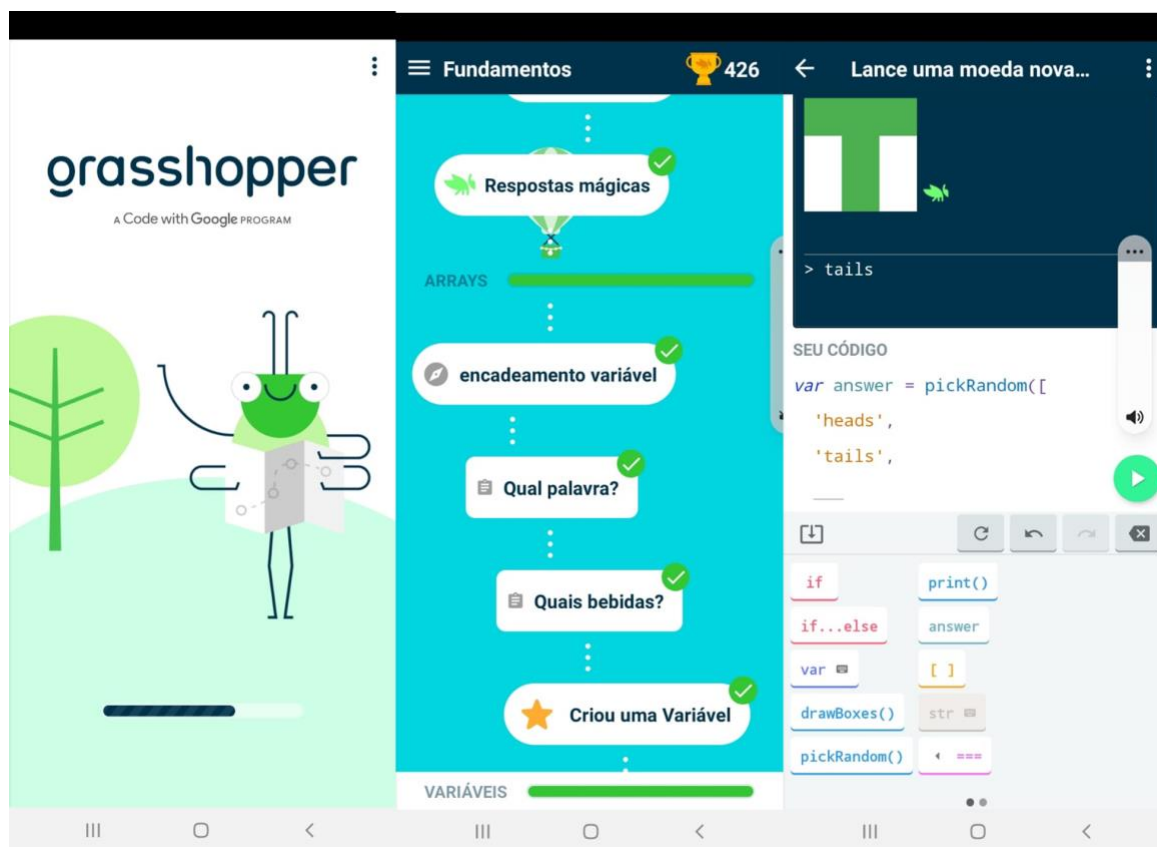


Figura 5. Screenshots do Grasshopper

Na figura 6 está representado o teclado da ferramenta, como pode ser visto o seu teclado comparado aos demais é diferente, essa diferença se dá ao fato de suas teclas serem especiais dando ao usuário a possibilidade de com um só clique fazer a estrutura um comando, isso a torna uma ferramenta principal para o usuário que está iniciando no campo da programação.

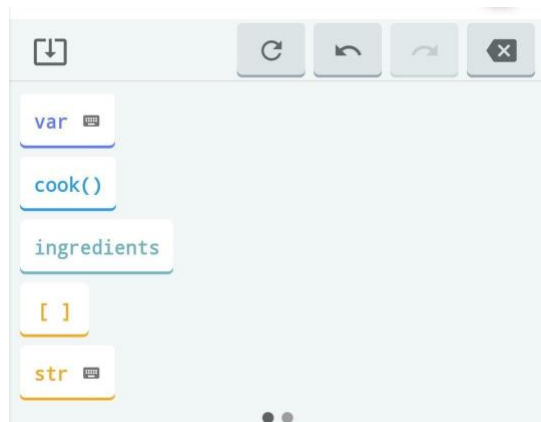


Figura 6. Screenshots do teclado do Grasshopper

Os três aplicativos analisados foram escolhidos como amostragem. Existem outros aplicativos que podem ser utilizados para inclusão na lógica de programação que apresentam características similares ao Dcoder, Replit e Grasshopper.

4.4 Resultados e discussões

Nesta seção são apresentadas as fases de avaliação dos aplicativos. Que estarão em uma tabela onde na mesma serão elencados os conceitos a serem avaliados acerca de cada ferramenta, estes resultados estão apresentados na tabela acompanhados de uma descrição e classificação de cada um dos aplicativo. A classificação seguirá o formato das heurísticas de Nielsen atribuindo uma nota de severidade entre 0 a 4 para as heurísticas que se aplicam a ação de usuário proposta.

0 – Não é considerado, totalmente, um problema de usabilidade

1 – Problema apenas estético: não necessita ser consertado a menos que tenha tempo extra disponível no projeto

2 - Problema menor de usabilidade: o conserto deste problema deverá ser baixa prioridade.

3 – Problema maior de usabilidade: é importante consertá-lo, para isso deverá ser dado alta prioridade

4 – Catástrofe de usabilidade: é obrigatório consertá-lo, antes do produto ser divulgado

N/A - Não se Aplica essa heurística a atividade avaliada

4.3.1 Tabela de avaliação descritiva e heurística

Características	Aplicativos em Análise		
	Dcoder	Replit	Grasshopper

Teclado Virtual	<p>Nota: 3. A presença da tecla TAB faz muita diferença para o programador. Entretanto, o processo de criação de código no teclado virtual reduzido limita a velocidade do desenvolvimento. O que faz necessária a implementação de função de autoco</p> <p>Ponto Negativo: Teclado reduzido Ponto positivo: Tecla TAB Sugestão :Inserção de função de autocompletar pode mitigar o problema do tamanho do teclado)</p>	<p>Nota: 2. O teclado virtual tenta ter todos os recursos do teclado comum, e por isso se torna muito pequeno para programação. Apesar disso, existe a função de autocompletar que auxiliam o processo.</p> <p>Ponto Positivo: Função de Autocompletar Ponto Negativo: Teclado reduzido</p>	<p>Nota 0: Seu diferencial é seu teclado com teclas especiais para inserção de números, operadores, strings e estruturas dando ao usuário a possibilidade com um só clique fazer a estrutura um comando como IF ELSE or FOR. A estrutura do teclado apresenta atalhos de acordo com o nível do programador.</p> <p>Ponto Positivo: Separação dos teclados, variáveis, operadores, estruturas, etc. Sugestão :Não existem funcionalidades a adicionar</p>
Armazenamento na Nuvem	<p>Nota: 1. Possui compartilhamento através de usuário email. Mas o compartilhamento com outras pessoas funciona melhor se elas tiverem o aplicativo.</p> <p>Ponto Positivo: Possui compartilhamento através de usuário email. Ponto Negativo: Necessita ter aplicativo Sugestão: Necessita possibilitar que outras pessoas vejam os códigos compartilhados</p>	<p>Nota: 0. Possui compartilhamento através de usuário email. Além disso, é possível encaminhar o link do código criado diretamente por Whatsapp ou Telegram, por exemplo.</p> <p>Ponto Positivo: Possui compartilhamento através de usuário email. Além disso, é possível encaminhar o link do código criado diretamente por Whatsapp ou Telegram, por exemplo. Sugestão: Não existem funcionalidades a adicionar</p>	<p>Nota: 3. Não, o aplicativo não permite a prática e desenvolvimento de código adicionais que possam ser reutilizados.</p> <p>Ponto Negativo: Não possui compartilhamento. Sugestão:A função de compartilhamento possibilita o acompanhamento dos professores.</p>

Interatividade da Interface	<p>Nota: 1. Interface exige uma certa experiência em editor de código para utilização.</p> <p>Ponto negativo: interface exige do usuário experiência em programação.</p>	<p>Nota: 1. Interface exige uma certa experiência em editor de código para utilização.</p> <p>Ponto negativo: interface exige do usuário experiência em programação.</p>	<p>Nota: 0. Por ser diferente de um editor de código padrão, não exige conhecimento prévio do usuário. Além disso, apresenta um método de interação baseado em gamificação.</p> <p>Ponto positivo: Disponibiliza uma interface intuitiva ao usuário.</p>
Destaque de Botões e Controles	<p>Nota: 1. Sim, inclusive tem o mesmo design dos editores de código de um computador. Pela presença das inúmeras funcionalidades e comandos, torna não amigável ao estudante de programação iniciante.</p> <p>Ponto negativo: Interface intuitiva mas não amigável para iniciantes.</p>	<p>Nota: 0. Sim, inclusive o seu diferencial é que em sua parte superior do teclado virtual disponibiliza os recursos mais utilizados das linguagens de programação, por exemplo as chaves e colchetes outro recurso exclusivo dele em relação aos outros é o seu botão central que está localizado na segunda aba onde o mesmo tem a finalidade de ver o histórico.</p> <p>Ponto positivo: Interface intuitiva e amigável com botões que remete a aplicativos do dia a dia.</p>	<p>Nota: 0. A presença do play como execução simplifica os comandos utilizados pelo usuário.</p> <p>Ponto positivo: Interface intuitiva e amigável com botões que remete a aplicativos do dia a dia.</p>
Adaptação da Interface ao tamanho da Tela	<p>Nota: 4. Não, por ser uma versão de um editor de código de computador suas telas foram reduzidas para versão mobile. O que deixa o acesso a botões, teclado, depuração muito restritos.</p>	<p>Nota: 4. Não, por ser uma versão de um editor de código de computador suas telas foram reduzidas para versão mobile. O que deixa o acesso a botões, teclado, depuração muito restritos.</p>	<p>Nota: 0. Como não se baseia em um editor de código comum, apresenta interface mais adequada aos celulares.</p> <p>Ponto Positivo: Separação dos teclados, variáveis, operadores, estruturas, etc.</p>

	Ponto Negativo: Tamanho da interface não facilita o acesso a botões e menus.	Ponto Negativo: Tamanho da interface não facilita o acesso a botões e menus.	
Simbologia adaptada	Nota: 0. Sim, inclusive os seus ícones se remetem a botões de aplicativos que o usuário utiliza no dia a dia. Ponto positivo: Ponto positivo: botões que remetem a aplicativos do dia a dia.	Nota: 0. Sim, inclusive os seus ícones se remetem a botões de aplicativos que o usuário utiliza no dia a dia. Ponto positivo: Ponto positivo: botões que remetem a aplicativos do dia a dia.	Nota: 0. Sim, o símbolo de execução como PLAY é amplamente utilizado em programação. Ponto positivo: Ponto positivo: botões que remetem a aplicativos do dia a dia.
Acessibilidade a menus e funções	Nota: 0. Sim, ele ainda tem um diferencial que é a semelhança com o computador. Ponto positivo: A semelhança com os editores de código desktop .	Nota: 0. Sim, ele ainda tem um diferencial que é a semelhança com o computador. Ponto positivo: A semelhança com os editores de código desktop	Nota: 0. Sim, aliás este é o quesito principal do aplicativo. Ponto Positivo: por ser uma aplicativo de gamificação seus menus e funções são bem intuitivos.
Funções de autocompletar	Nota: 4. Não, essa é uma função essencial para o desenvolvimento de programas. Ponto negativo: Não possui essa função.	Nota: 0. Sim, aliás este é o quesito principal do aplicativo. Ponto positivo: possui uma aba extra em seu teclado onde dispõe dos recursos mais utilizados nas linguagens de programação, como por exemplo as chaves e colchetes.	Nota: 0. Sim, aliás este é o quesito principal do aplicativo. Ponto Positivo: possui teclas especiais com as funções de números, operadores, strings e estruturas .
Interface resultante da compilação	Nota:0. Sim,inclusive o seu diferencial é que o seu terminal quando aberto fica logo abaixo ao	Nota: 3.Não, seu terminal fica em uma aba a parte a aba do código o que dificulta a visualização dos usuário	Nota:0,Sim, inclusive o seu diferencial é que o seu terminal quando aberto fica logo abaixo ao código, facilitando a visualização do

	<p>código, facilitando a visualização do usuário.</p> <p>Ponto Positivo: Possui terminal na mesma aba do programa. Sugestão: Não existem funcionalidades a adicionar</p>	<p>em questão aos erros que o seu código venha a ter.</p> <p>Ponto Negativo: Possui terminal em aba separada do programa. Terminal executado em aba separada provoca problemas para o desenvolvedor iniciante</p>	<p>usuário, outro quesito exclusivo que ele disponibiliza é exibir gráficos.</p> <p>Ponto Positivo: Possui terminal na mesma aba do programa. Terminal ainda possui capacidade de apresentar resultados gráficos Sugestão: Não existem funcionalidades a adicionar</p>
--	---	--	---

Após analisar os aplicativos com base na avaliação heurística obteve-se o seguinte somatório: Dcoder nota 14, Replit nota 10 e Grasshopper nota 3. Desta forma pode-se afirmar que os aplicativos cumprem muito bem com a proposta de uma interface intuitiva ao uso do usuário sendo necessário fazer certos ajustes que são sugeridos na tabela. O Grasshopper é mais receptivo aos programadores iniciantes. Replit e Dcoder apresentam interface e usabilidade mais apropriada para programadores mais experientes.

Fazendo uma apanhado geral do que aqui foi apresentado a respeito da interface do Grasshopper, foi percebido que ele se disponibiliza de botões interativos assim se mostrando uma eficiência, pois ele facilita que o usuário fique digitando muito texto, esse é um diferencial que ele tem dos demais. Em relação a interface do Dcoder foi percebido que ele se disponibiliza a ser uma coisa mais profissional, não que isso seja uma coisa que tire sua usabilidade, e em relação aos estudantes da disciplina de lógica de programação recomenda se a eles passem primeiro por jornada de ensino que o Grasshopper disponibiliza. Já a interface do Replit por ser uma aplicação web é bastante interativa, mas como o Dcoder se o estudante for calouro e recomendo passar antes pela jornada de ensino do Grasshopper.

Com a análise obteve o seguinte resultado, que todas as ferramentas são eficientes, porém recomenda-se ao usuário iniciante na programação que ele use primeiro o Grasshopper, pois o mesmo disponibiliza uma jornada de ensino, o Dcoder e Replit são bastante semelhantes.

Aos professores das disciplinas de programação iniciais, a recomendação é a mesma. Iniciar os conteúdos com ferramentas do estilo do Grasshopper e no decorrer da disciplina inserir os editores de código padrão.

5 CONSIDERAÇÕES FINAIS

Esta pesquisa é uma ação acerca de pensar nos estudantes além da sala de aula. Fazendo com que as universidades e instituições de ensino possam cumprir um papel de inclusão e não só de geradora de conhecimento. Com o que foi pesquisado aqui neste trabalho, as instituições terão uma opção de gerar independência em seus estudantes. Desta forma a pesquisa atende um dos seus objetivos que são da autonomia ao estudante.

Recomenda-se que os professores de programação testem e conheçam as aplicações móveis ligadas a prática de programação como forma de incluir os estudantes que não possuem computador no processo de ensino-aprendizagem. Entretanto, sempre deve-se observar as características de cada aplicativo com relação a interface, recursos, adaptações. Isso aumenta a possibilidade da boa inserção dos estudantes no mundo da programação.

A pesquisa aqui propôs uma avaliação de usabilidade dos aplicativos móveis: Grasshopper, Dcoder e Replit, utilizando da avaliação heurísticas de usabilidades. O objetivo proposto na pesquisa, foi apresentar possibilidades de ferramentas móveis para o ensino de programação como forma de inclusão.

Espera-se que os resultados desta pesquisa, sejam um apoio para os professores em questão do uso das tecnologias móveis no ensino de programação.

Como trabalhos futuros seria recomendado a criação de ferramentas que englobe as características que aqui foram avaliadas, em uma criação de um único aplicativo.

REFERÊNCIAS

- ABDALLA, Débora et al. Onda Digital: A Universidade protagonizando a Inclusão Digital. In: **Anais do Workshop de Informática na Escola**. 2005.
- BOSSE, Yorah; GEROSA, Marco Aurélio. Reprovações e trancamentos nas disciplinas de introdução à programação da Universidade de São Paulo: um estudo preliminar. In: **Anais do XXIII Workshop sobre Educação em Computação**. SBC, 2015. p. 426-435.
- BARROS, A. **Avaliação da Usabilidade do Portal Conecta Apoiado pelas 10 Heurísticas Propostas por Jakob Nielsen**. Palmas, 2017.
- CAVALLIN, Rejane Vera. **Análise e identificação de formas de acesso de diferentes conexões, e aplicação de IHM (interface homem-máquina) utilizando o projeto piloto UTFPR-MD**. 2015. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.
- FEIJÓ, Valéria Casaroto; GONÇALVES, Berenice Santos; GOMEZ, Luiz Salomão Ribas. Heurística para avaliação de usabilidade em interfaces de aplicativos smartphones: utilidade, produtividade e imersão. **Design e Tecnologia**, v. 3, n. 06, p. 33-42, 2013.
- GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. Editora Atlas SA, 2008.
- HIGUCHI, Adriane Aparecida da Silva. **Tecnologias móveis na educação**. 2011. 90 f. Dissertação (Mestrado em Mestrado em Educação, Arte e História da Cultura) - Universidade Presbiteriana Mackenzie, São Paulo, 2011.
- KRZYZANOWSKI, Lucas Pedroso. **Desempenho acadêmico em disciplinas de algoritmos e programação e a evasão nos cursos de licenciatura em computação**. TCC (graduação) Universidade de São Paulo. Curso de Licenciatura em Computação, 2019.
- LEDESMA, Fernanda. **Mobile learning: Proibir ou integrar?**. Disponível em: http://cefopna.edu.pt/revista/revista_09/es_02_09_fl.htm. Acesso em 18 de nov. de 2022.
- LOPES, Anita; GARCIA, Guto. **Introdução à programação**. 15ª ed. Local de publicação: Elsevier Editora LTDA, 2002.
- Renda domiciliar per capita no Tocantins é de R\$1.055,60, revela IBGE**. Gazeta do Cerrado, 2020. Disponível em: <https://gazedo.cerrado.com.br/renda-domiciliar-per-capita-n-tocantins-e-de-r-1-05560-revela-ibge/>. Acesso em: 18 de nov. de 2022.
- Venda e preço médio de PCs crescem no 1º trimestre de 2021**. Febraban, 2021. Disponível em: <https://febrabantech.febraban.org.br/temas/infraestrutura/venda-e-preco-medio-de-pcs-crescem-no-1-trimestre-de-2021>. Acesso em: 18 de nov. de 2022.