

ANÁLISE DE METODOLOGIAS ATIVAS NO ENSINO E APRENDIZAGEM DE PROGRAMAÇÃO.

ANALYSIS OF ACTIVE METHODOLOGY IN PROGRAMMING TEACHING AND LEARNING.

Lucas Lima Soares Pinto*

Alexandro Lima Damasceno**

Felipe Bastos Nunes***

RESUMO

A computação tem exercido um papel dominante nos aspectos sociais, econômicos e culturais, e sua presença nessas estruturas, bem como sua utilização para o desempenho de atividades cotidianas, demonstram claramente uma necessidade e influência nas mais diferentes camadas e grupos. Com isso, os usos de plataformas digitais tornaram-se imprescindíveis, e o desenvolvimento destas não se limita apenas a programação e construção de programas, sendo necessário outras habilidades, dentre elas a compreensão e análise de algoritmos. Contudo, seu ensino-aprendizagem não é feito de forma tão simples, conseqüente, existe uma grande taxa de desistência e evasão nos cursos de Tecnologia da Informação e Comunicação. Tendo isto em vista, o presente estudo tem como objetivo verificar os efeitos de uma nova metodologia de ensino de programação para alunos de ensino fundamental, contribuindo para o desenvolvimento de *HARD SKILLS* de forma descomplicada, utilizando metodologias mais interativas e incentivadoras para os estudantes da área.

Palavras-chave: Ensino de Programação. Gamificação. Softwares Educativos. Metodologias Ativas.

ABSTRACT

Computing has played a dominant role in social, economic and cultural aspects, and its presence in these structures, as well as its use to perform everyday activities, clearly demonstrate a need and influence in the most different layers and groups. As a result, the use of digital platforms has

* Graduando em Bacharelado em Ciência da Computação no Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. E-mail: lucas.lima.soares05@aluno.ifce.edu.br

** Mestre em Ciência da Computação pela UFERSA; Docente no Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. E-mail: alexandro.lima@ifce.edu.br

*** Especialista em Docência no Ensino Técnico pelo Centro Universitário SENAC; Docente no Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. E-mail: felipebastos@ifce.edu.br

become essential, and their development is not limited to programming and building programs, requiring other skills, including understanding and analyzing algorithms. However, its teaching-learning is not done in such a simple way, therefore, there is a high rate of dropout and evasion in Information and Communication Technology courses. With this in mind, the present study aims to verify the effects of a new programming teaching methodology for elementary school students, contributing to the development of HARD SKILLS in an uncomplicated way, using more interactive and encouraging methodologies to area students.

Keywords: Programming Teaching. Gamification. Educational Software. Active Methodologies.

Data de submissão para publicação:2022.

Data de aprovação para publicação: 2022.

Aqui pode ser indicado o endereço eletrônico, DOI, suportes e outras informações relativas ao acesso do documento.

1 INTRODUÇÃO

O advento da tecnologia nas últimas décadas, promoveu verdadeiras transformações socioeconômicas e culturais. A popularização e a democratização da Internet transformou as interações dos homens entre si e com o meio. Acontecimentos antes consideradas como “futurísticos” - biometria, reconhecimento facial, Inteligência Artificial(IA) e Internet of Things ou Internet das coisas(IOT), por exemplo, passaram a fazer parte do cotidiano, tornando-se essenciais nas mais diferentes camadas e grupos da sociedade.(FERREIRA, 2003)

Nesse sentido, várias inovações também se popularizaram e têm grande peso na revolução digital dos últimos anos, para além do uso da web. Podem ser citadas a tendência mobile, com a consolidação do uso de dispositivos móveis smartphones e wearables, bem como os aplicativos, que otimizam as tarefas do dia a dia, e a modernização das formas de pagamento, como Internet banking, carteiras digitais e o Pix.

No background de toda essa evolução, existe a soma dos trabalhos de cientistas e pesquisadores, que garante cada vez mais modernidade e segurança para essas tecnologias. Como pilar dessa transformação está a prática da programação, que não se limita apenas ao desenvolvimento e construção de programas, se fazendo necessário também o desenvolvimento de habilidades como a compreensão e análise de algoritmos. Nessa perspectiva, (WING, 2006) explica que o pensamento computacional é um conjunto de habilidades para a resolução de problemas que são desenvolvidas através do estudo de conteúdos oriundos da Ciência da Computação.

Nesse primeiro contato de alunos que ingressam eixo de Tecnologia da Informação e Comunicação (TIC), tem o objetivo de desenvolver o raciocínio lógico e a aplicação prática dos conceitos, fatores que propiciarão a capacidade de resolução de problemas e tomada de decisão, como explica (RAABE; SILVA, 2005). Contudo, esse processo de ensino-aprendizagem não é uma tarefa fácil, em especial para alunos que não tiveram contato com a tecnologia anteriormente,

por exemplo. Essas dificuldades são um dos principais fatores reprovação, e conseqüentemente desistência, nas disciplinas de programação. O resultado é uma grande evasão dos cursos na área de computação.(MONCLAR; SILVA; XEXÉO, 2018; SANTIAGO; KRONBAUER, 2017)

Estudos atuais apontam a significativa contribuição do ensino de programação para estudantes ainda em idade escolar. (SCAICO et al., 2013; BERSSANETTE; FRANCISCO, 2021; GOULART et al., 2019) Graças ao novo modelo de educação básica, os alunos podem optar por estudar programação na escola, desenvolvendo desde cedo habilidades necessárias que conduzirão a jornada nos estudos relacionados às TIC.

“A área de Matemática, no Ensino Fundamental, centra-se na compreensão de conceitos e procedimentos em seus diferentes campos e no desenvolvimento do pensamento computacional, visando à resolução e formulação de problemas em contextos diversos.”(BRASIL, 2018)

Em um primeiro momento, o ensino-aprendizagem de programação deve ser orientado por uma metodologia que mantenha os alunos engajados e motivados, para que consigam avançar nos desafios sem desanimar, mantendo-se interessados em aprender. Nesse sentido, grandes organizações têm investido em projetos que ensinam programação na educação básica.(OLIVEIRA, 2018)

(SANTOS; COSTA, 2006) argumentam que o crescimento de dados e tecnologias, requer profissionais aptos a aliar dinamismo e inteligência, a fim de alcançar as melhores soluções para problemas enfrentados no processo de ensino e aprendizagem nas TIC. Para obter resultados significativos, existe uma necessidade frequente de atualizações das didáticas de ensino de forma geral, para transformar processos abstratos em concretos, com o auxílio de softwares, que possibilitam uma melhor compreensão dos conceitos abordados.

Existem diversas ferramentas a serem exploradas, de modo que desabrochem o entusiasmo dos alunos em aprender sobre programação. Principalmente, por acontecer em ambiente laboratorial onde os alunos se encontram em situação favorável, pois os mesmos têm acesso a computadores, Internet e o auxílio dos professores ou colegas mais experientes caso exista alguma dúvida no manuseio.

Neste contexto, esta pesquisa tem como objetivo analisar e comparar ferramentas que facilitam o ensino e a aprendizagem de programação para estudantes da Educação Básica. Além disso, visa desenvolver o pensamento computacional, tendo por objetivo tornar possível a implementação de estratégias e soluções para problemas de maneira eficaz. Ações como o Codecademy, Code.org e a Khan Academy, têm difundido o debate em torno da necessidade de inserir a temática da iniciação em programação no cotidiano de jovens e adolescentes, contando com a ajuda de ferramentas como MIT App Inventor, Scratch e Code.org.

Para reportar o trabalho realizado, o resultado deste artigo encontra-se organizado da seguinte forma. Na Seção 2 é apresentado o referencial teórico, assim como, algumas ferramentas exploradas durante a pesquisa. Já na Seção 3 estão dispostos trabalhos relacionados que realizaram pesquisas com o mesmo intuito. Na Seção 4 é apresentada a metodologia em que a pesquisa foi realizada. A Seção 5 expõe a forma em que o estudo de caso foi elaborado. Na

Seção 6 os resultados do estudo de caso são difundidos. E por fim na Seção 7 são descritas as conclusões.

2 REFERENCIAL TEÓRICO

Esta seção se refere à conceitos de programação, assim como, a problemática que será apresentada durante o trabalho e a via escolhida para a solução do problema. Serão retratadas algumas ferramentas utilizadas durante a pesquisa.

2.1 ENSINO TRADICIONAL DE PROGRAMAÇÃO

A introdução à programação geralmente é apresentada ao estudante de forma estruturada, para que ele possa abstrair e compreender os conceitos, visando entender como funcionam as didáticas. Esse modelo de ensino é conhecido como “método tradicional”.(BORGES, 1998; BARANAUSKAS et al., 1999).

(CARVALHO, 2012) retrata, que o método tradicional não é suficiente para motivar os estudantes. Embora a metodologia utilizada se valha de recursos inclusivos como por exemplo, aulas práticas em laboratórios de informática e ferramentas multimídia.

A Figura 1 apresenta um modelo usado em diversas áreas do conhecimento, mas neste estudo ilustra o uso em cursos de graduação em Tecnologia da Informação e Comunicação.

MÉTODO TRADICIONAL DE ENSINO

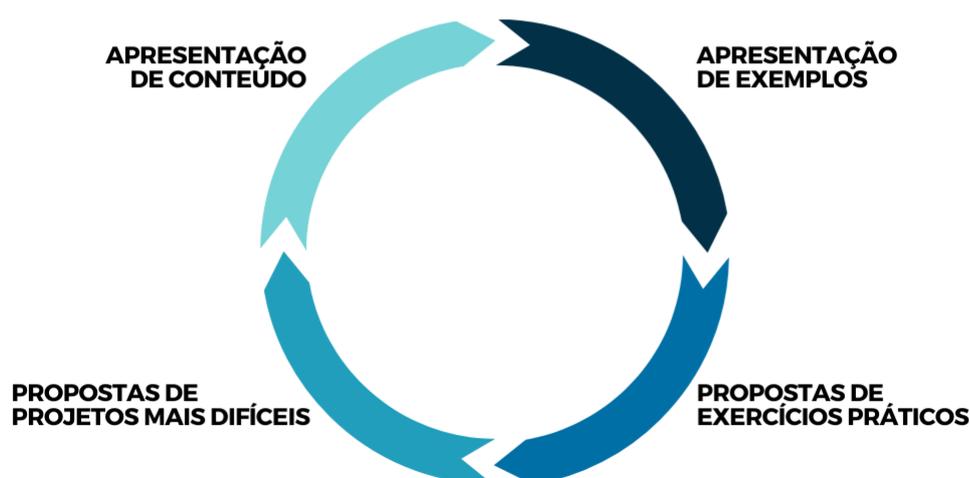


Figura 1 – Ciclo do método tradicional de ensino.

2.2 DIFICULDADES NO ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO

Segundo (RAPKIEWICZ et al., 2006), disciplinas de base para o ensino-aprendizagem de programação em cursos da área de tecnologia, abordam princípios de lógica de programação, de modo que os estudantes possam desenvolver a capacidade de analisar e resolver problemas propostos no decorrer da disciplina, fazendo isso na forma de algoritmos. (ARIMOTO; OLIVEIRA, 2019) recorda que a união de conceitos abordados e a carga de conhecimentos incorporada não são fáceis de entender, principalmente àqueles que estão iniciando na área.

Por consequência disso, disciplinas que envolvem o ensino-aprendizagem de programação são uma das principais causadoras do aumento na taxa de evasão e reprovação em cursos da área de computação. (BERSSANETTE; FRANCISCO, 2021; SILVA; MELO; TEDESCO, 2018; GIRAFFA; MORA, 2013)

O processo de ensino-aprendizagem de programação é muito amplo. Por esta razão, é muito difícil de identificar a origem dos problemas relacionados ao mesmo. Em função dessas dificuldades observadas, professores e pesquisadores buscam identificar e estudar as causas, com intenção de desenvolver diversas ideias, voltadas principalmente para os estudantes ingressantes em programação, tendo em vista tornar o ensino de programação mais efetivo. (BERSSANETTE; FRANCISCO, 2021)

(JÚNIOR; RAPKIEWICZ, 2004) aponta outro problema importante, referindo-se a capacidade dos estudantes em absorver as abstrações que são necessárias durante o desenvolvimento do ensino-aprendizagem de programação. Além de tudo, grande parte dos estudantes necessita de muito esforço para desenvolver raciocínio lógico, algo que é de suma importância posteriormente para o desenvolvimento de programas. Esse nível de dificuldade acarreta aos estudantes apatia, baixa auto-estima e desmotivação. (RODRIGUES, 2002; JÚNIOR; RAPKIEWICZ, 2004; DELGADO et al., 2004)

2.3 USO DE TECNOLOGIAS PARA O ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO

Dentro desse contexto, (RODRIGUES; CORREIA et al., 2017) afirmam que um dos principais conceitos direcionados ao ensino-aprendizagem é a discussão sobre estilos de aprendizagem e a criação de metodologias de ensino correlacionadas. De acordo com (AMARAL; BARROS, 2007), o desenvolvimento de metodologias que utilizam tecnologias estando aliadas aos conceitos de estilo de aprendizagem podem formar uma ferramenta educacional. Deste modo, graças ao avanço tecnológico torna-se possível obter uma rica gama de possibilidades, desde ferramentas advindas da tecnologia, até funcionalidades e metodologias, assim obtendo novos meios para o desenvolvimento do conhecimento.

Vale ressaltar que o fator principal é a busca pelo conhecimento proveniente do estudante, para que o mesmo esteja apto a desenvolver-se profissionalmente. Algumas capacidades necessárias para isso são I) ter habilidade de abstrair o problema e haver desenvolvido raciocínio lógico para o resolvê-lo, II) entender a sintaxe e semântica da linguagem de programação escolhida,

III)compreender o funcionamento do ambiente de programação e IV)reconhecer os processos de compilação e testes do código fonte. (RODRIGUES; CORREIA et al., 2017)

2.3.1 Jogos e Simulações no ensino e aprendizagem de programação.

(MORENO; MAYER, 2007) afirma que ambientes interativos para aprendizagem proporcionam uma maneira mais eficiente para que os estudantes aprendam. Um bom exemplo disso são jogos e simulações, que por sua vez proporcionam um melhoramento na aprendizagem, pois os usuários as consideram formas mais interativas e motivadoras.

Também é possível observar que com o auxílio de jogos os estudantes podem adquirir interesse e incentivo, pois vai além de algo lúdico, proporciona encorajamento ao usuário, já que ele tem interesse em avançar durante o jogo, dessa forma absorvendo mais conhecimento.(OLIVEIRA, 2015)

De acordo com (VALENTE et al., 1998), sistemas que possuem uma abordagem para a aprendizagem construcionista, se diferem estruturalmente e conceitualmente das abordagens feitas por Tutores Inteligentes(TIS), este sistema é referido como Ambientes interativos de aprendizagem. (VALENTE et al., 1998) ainda fala que, o aprendizado torna-se mais efetivo quando o estudante constrói o próprio conhecimento, sem a necessidade de ser ensinado de forma tradicional, através de leitura ou mesmo uma sequência organizada de atividades.

Conforme (SILVA, 2007), a experiência proveniente dos jogos, estabelece vínculos diferentes dos observados por meio de trocas de informações, a interação com o jogo faz com que o usuário imagine, planeje e execute atividades propostas. Em virtude dessa imersão, é possível absorver o conjunto de informações orientadas através da situação em que se encontra no jogo, fazendo isso de forma mais natural, dentro do contexto estabelecido.

2.4 METODOLOGIAS ATIVAS

Metodologias ativas tratam-se de meios que relacionam o processo de ensino e aprendizagem em função das capacidades particulares dos alunos de forma ativa, onde o ensino não é limitado à habilidade de lecionar do professor, assim como, é necessário levar o aluno efetivamente ao saber.(PAIVA et al., 2016)

"Compreende-se que a aprendizagem necessita do saber reconstruído pelo próprio sujeito e não simplesmente reproduzido de modo mecânico e acrítico."(PAIVA et al., 2016)

Exemplos de metodologias ativas em suas formas mais simples e comumente utilizadas por professores são o ensino por meio de solução de problemas e o ensino por meio de projetos. Tais métodos influenciam os alunos a participarem de forma ativa ao ler, perguntar, fazer, ouvir ou ensinar.(BARBOSA; MOURA, 2013)

2.5 FERRAMENTAS

Nesta seção serão apresentados softwares e plataformas que foram pesquisadas com o intuito de servirem como ferramentas para o auxílio do ensino-aprendizagem de programação, os mesmos serão apresentados de forma geral durante essa seção.

2.5.1 *CronApp*

O CronApp é uma plataforma web e *mobile* para aprendizagem de programação, ele utiliza um sistema de programação através de blocos drag and drop (arrastar e soltar, em português) com o objetivo de facilitar a utilização dos usuários iniciantes, fazendo o uso de uma abordagem intuitiva e dinâmica, assim motivando o estudante.

Além de programação em blocos, a plataforma também disponibiliza a possibilidade de visualizar e editar o código que está sendo “montado”, motivando o usuário a conhecer como é a linguagem de programação por trás de tais conceitos. O mesmo, ainda oferece como alternativa diversas linguagens de programação para a escolha do estudante. No entanto, a maioria dessas funcionalidades estão disponíveis apenas para aqueles que fizerem uma assinatura do conteúdo do site.

(GOMES; SANTOS, 2019) dissertam que o CronApp é uma ferramenta que possibilita o desenvolvimento de aplicações, sistemas web ou mobile robustos de forma rápida, feito em camadas que contribuem para que o projeto possa atingir uma codificação avançada. O CronApp ajuda o usuário nesse processo, já que o mesmo possui uma vasta gama de funcionalidades que trazem facilidade ao criar projetos.

Visto que, o intuito do presente trabalho é facilitar o acesso de ferramentas que auxiliem no ensino-aprendizagem de programação, esta ferramenta torna-se pouco atrativa, devido principalmente a necessidade de que os estudantes tenham que pagar para ter acesso pleno.

2.5.2 *Blockly.games*

O Blockly.games é uma plataforma WEB que visa estimular o aprendizado de lógica de programação através do uso de blocos, esta ferramenta utiliza uma metodologia orientada como um jogo, onde o usuário tem algo parecido com missões que precisam ser concluídas para que possa avançar de fase, a cada fase aumenta-se a dificuldade e são acrescentados novos conceitos de programação.

De acordo com (FIGUEIREDO; LIMA; FILHO, 2019) o Blockly é utilizado como uma forma de programação visual, no caso, a programação é feita através de blocos, ele também fala que seu uso na educação tem demonstrado que ao utilizar blocos de programação os erros de sintaxe das linguagens de programação tradicionais são evitados, dessa forma tornando a experiência do usuário mais agradável, promovendo mais foco, para que assim o usuário consiga desenvolver melhor seu algoritmo e também o seu pensamento computacional.

Todavia, o Blockly foi criado com o foco em ensinar crianças que não tiveram nenhuma experiência com programação. Dito isto, são atividades simples, fáceis e didáticas. Conforme o avanço das fases, o usuário começa a escrever em linguagem de programação propriamente dita, e a ferramenta fica mesclando entre fases, a programação em blocos e em linguagem de programação.

2.5.3 Scratch

O Scratch também se trata de uma plataforma web e Desktop para o ensino-aprendizagem de programação para estudantes de todas as idades, feito para que os usuários programem suas próprias histórias, animações ou até mesmo jogos. A abordagem utilizando a programação em blocos é algo bem atrativo, principalmente por sua facilidade de uso, especialmente para novatos na programação.

Existem várias pesquisas que utilizaram o Scratch como ferramenta para o auxílio do ensino-aprendizagem de programação como por exemplo (RODRIGUES; CORREIA et al., 2017; JUNIOR; FRANÇA, 2017), além disso, também existe uma comunidade integrada ao site para que educadores possam compartilhar suas experiências e aprender com experiências de outros educadores, assim como os avanços obtidos com seus alunos.

(WANGENHEIM; NUNES; SANTOS, 2014) comenta sobre o uso da ferramenta para o ensino de alunos do ensino fundamental, o uso do SCRATCH torna a aprendizagem mais interativa e interessante graças a possibilidade dos alunos interagirem com imagens, sons, gráficos e etc. Dessa forma criando uma espécie de história.

(WANGENHEIM; NUNES; SANTOS, 2014) não deixa de ressaltar que os estudantes que utilizam o SCRATCH para estudar conseguem aprender e pensar de forma criativa, também possibilita um desenvolvimento para que eles consigam trabalhar em equipe de forma mais eficaz e adquirem pensamento sistemático para a resolução de problemas.

2.5.4 Codecademy

O Codecademy é uma plataforma web, em seu site possui uma metodologia online e interativa, na qual oferece aulas de codificação com diversas linguagens de programação, porém, seus recursos de forma gratuita são limitados, aos estudantes que quiserem se aprofundar e conhecer cada vez mais sobre programação tem que assinar um plano para obter acesso a mais conteúdos.

A ferramenta possui interface interativa que incentiva o usuário, também premia os estudantes com emblemas e fornece um feedback para os mesmos, para que assim haja incentivo aos estudantes que estão progredindo e aprendendo.

No entanto, assim como o CronApp, o Codecademy não se encaixa no objetivo principal do presente trabalho, devido sua plataforma ter a necessidade de ser paga para disponibilizar todos os recursos. Além disso, o site encontra-se em inglês, o que dificultaria a aprendizagem, principalmente daqueles estudantes que não conseguem entender a língua estrangeira.

2.5.5 *Robocode*

Robocode é uma plataforma Desktop que funciona em qualquer sistema operacional que tenha instalado o ambiente de execução Java a partir da versão 8. Ela tem como objetivo codificar uma batalha entre robôs em uma arena virtual. A ferramenta incentiva o estudante a aprender a programar para que possa otimizar o seu robô, principalmente àqueles estudantes que têm interesse em jogos, já que as ações de cada robô são determinadas conforme o usuário programa sua IA para a batalha.

No entanto, a plataforma aceita apenas duas linguagens de programação para modificar o código dos robôs, sendo elas Java e .NET, o que acaba tornando a ferramenta pouco atrativa, principalmente para os estudantes novatos na programação, pois os mesmos buscam linguagens mais fáceis de aprender, especialmente no início onde é preciso aprender conceitos da lógica de programação.

Um fato interessante é que a plataforma foi desenvolvida pela Alpha Works, que pertence a IBM, e que essa divisão tem como objetivo divulgar novas tecnologias de desenvolvimento. (AMARAL et al., 2015) reforça o que já foi dito durante essa subseção, e contribui também falando que a movimentação dos robôs, assim como suas ações se baseiam em ângulos pré-definidos, isto é, necessitam de uma aplicação direta de conceitos matemáticos.

(AMARAL et al., 2015) também deixa claro que após o estudo os alunos contemplados conseguiram desenvolver raciocínio lógico e também foram capazes de aplicar conceitos matemáticos simples para a resolução de possíveis problemas, tudo isso dentro de um ambiente de ensino interativo.

2.5.6 *CodeCombat*

CodeCombat é uma plataforma WEB que visa ensinar estudantes a programar, utilizando uma metodologia criativa, que é uma espécie de Role Playing Game(RPG) programável, onde conforme as fases e objetivos são alcançados e concluídos novos conceitos são apresentados e novos desafios são propostos. Dito isto, a ferramenta utiliza duas linguagens de programação, Python e JavaScript.

Sendo o Python uma linguagem mais amigável para os programadores novatos e também por ser utilizada por grandes corporações como o Google, recomendada para alunos mais jovens ou mesmo iniciantes. Já o JavaScript, é a linguagem da WEB, e é utilizado em quase todos os sites. Dessa forma, fica a critério do estudante a escolha da linguagem que preferir.

(PASSOS et al., 2020) conclui ao fim de seu estudo que foi possível que os alunos que utilizaram o CodeCombat para aprender sobre programação desenvolveram uma afinidade pela metodologia empregada, principalmente por haver uma boa interface, explicações e ajudas, tudo isso durante o decorrer das atividades apresentadas no jogo. (PASSOS et al., 2020) também comenta que é notório o engajamento dos alunos e que os mesmo se sentem motivados a aprender programação.

2.5.7 MIT App Inventor

O MIT App Inventor é uma plataforma que traz facilidades para os estudantes que estão começando a programar, através da sua metodologia intuitiva e a utilização da linguagem de blocos, torna-se mais fácil entender os conceitos de lógica de programação, assim como, o aprendizado de programação em si.

A ferramenta permite que mesmo aqueles que acabaram de embarcar no mundo da programação possam desenvolver um aplicativo para o sistema operacional Android presente na maioria dos smartphones atuais. O intuito da plataforma é principalmente o ensino de programação e desenvolvimento para amadores, dessa forma tornando-se uma grande aliada para educadores que buscam incentivar seus alunos. Em contrapartida, assim como algumas ferramentas citadas, o MIT App Inventor também só possui tradução parcial para o português.

2.5.8 Code.org

O Code.org é uma plataforma WEB que tem como objetivo incentivar e ensinar a todos sobre lógica de programação e seus conceitos, sem deixar de lado as linguagens de programação propriamente ditas. O site tem uma interface interativa e fácil de se entender, além de possuir planos de aula previamente feitos e disponibilizados aos usuários, sua logística de aula é muito parecida com jogos educativos que conforme avanço das fases, são acrescentados novos conceitos e o grau de dificuldade tende a aumentar.

A ferramenta usa programação através de blocos, mas também é possível ver a forma em que cada bloco é escrito com determinada linguagem de programação escolhida pelo estudante, por ser uma plataforma voltada principalmente para crianças, os desafios são de fácil compreensão e acessíveis ao público em geral, independente de sua idade. Uma ferramenta de grande importância para o desenvolvimento de iniciantes.

2.5.9 Programming Hero

O Programming Hero é um aplicativo *mobile* disponibilizado tanto para Android quanto para IOS (Iphone Operational System), foi selecionado pela plataforma Code.org como aplicativo de aprendizagem para organização e promoção de programação.

A ferramenta simula um jogo e sua metodologia é muito parecida com aplicativos de aprendizagem de línguas estrangeiras, a forma que o Programming Hero trabalha lembra bastante o aplicativo Duolingo.

O Programming Hero é disponibilizado de forma gratuita, mas também possui planos pagos com cursos que podem ser de interesse do estudante. Porém, a linguagem nativa da ferramenta é o inglês, não havendo disponibilidade de tradução para o português, o que pode ser uma barreira para quem está começando na programação.

2.5.10 *LEGO Educacional (LEGO SPIKE PRIME)*

O kit LEGO SPIKE PRIME é um conjunto de robótica voltado para educação, ele permite que os usuários possam criar e programar robôs, utilizando a mesma ideia que o LEGO tradicional traz, onde é possível criar pequenas estruturas com blocos montáveis, só que no caso são criados pequenos robôs.

A programação do robô criado é feita através do aplicativo LEGO Education SPIKE, esse aplicativo é multiplataforma, podendo ser utilizado de forma mobile (Android ou IOS), também pode ser utilizado via DESKTOP (Windows ou Mac OS) ou então de forma WEB. Independente da plataforma que será utilizada, a programação pode ser feita de três formas diferentes, sendo elas: linguagens de blocos (podendo ser blocos de ícones ou blocos de palavras) ou então através da linguagem de programação Python.

2.5.11 *Arduino*

O arduino é um microcontrolador para o desenvolvimento de projetos eletrônicos. O mesmo é constituído por hardware e software, vale ressaltar que seu software e hardware são livres, podendo ser utilizado e/ou modificados por qualquer pessoa caso necessário.

A plataforma arduino encontra-se nas versões WEB e DESKTOP, pode-se enfatizar que devido ao seu baixo custo e facilidade no manuseio ele é indicado para qualquer pessoa, desde os iniciantes até aqueles que já possuam experiência na área de programação ou eletrônica.(ALVES et al., 2013)

Por ser um microcontrolador programável ele dispõe em sua plataforma essa possibilidade, utilizando de sua linguagem de programação própria. Podendo definir as ações que serão tomadas conforme a utilização de equipamentos e a decisão do usuário.

Aos estudantes que estão ingressando na área de programação, o arduino pode ser uma boa opção para começar a programar, principalmente se o mesmo tiver inclinação para a eletrônica. No entanto, por mais que o arduino seja um projeto open source a aquisição de seus componentes continua sendo paga, podendo tornar-se um empecilho aos alunos.

3 TRABALHOS RELACIONADOS

No presente capítulo, serão apresentados trabalhos relacionados onde houve uma pesquisa sobre o uso de metodologias para auxiliar no ensino e aprendizagem de programação.

3.1 *CODE: O ensino de linguagens de programação educativas como ferramentas de ensino/aprendizagem.*

(DANTAS; COSTA, 2013) expõe em seu trabalho como o avanço de novas tecnologias tem influenciado no cotidiano das pessoas, e mais, as facilidades que elas trazem permitem que as pessoas consigam acompanhar as informações de forma rápida, além de inserir-se no contexto digital. No decorrer do trabalho, (DANTAS; COSTA, 2013) argumenta sobre o ensino através

da plataforma CODE. Eles realizaram um estudo sobre o projeto CODE e impactos causados pelo mesmo, fazendo com que as habilidades desenvolvidas com o aprendizado de programação se relacionem com habilidades fundamentais na construção do conhecimento do estudante. De acordo com (DANTAS; COSTA, 2013) em concordância com (BARANAUSKAS et al., 1999), a programação traz inúmeras contribuições para o contexto educacional, já que ao programar é necessário a utilização de uma linguagem de programação para solucionar um problema, e é a mesma ação que o estudante faz para resolver uma tarefa em outras áreas de conhecimento.

Ao fim, (DANTAS; COSTA, 2013) pôde concluir que a utilização da plataforma CODE para o ensino de programação tem abordagens eficientes, e que qualquer pessoa pode aprender com ela, independente de sua área. Assim, desenvolvendo habilidades de programação como ler e escrever códigos, sem contar com as habilidades de raciocínio lógico, conceitos matemáticos, resolução de problemas de forma eficiente, dentre outras habilidades advindas da prática do processo de programar.

3.2 Jogos com o propósito para o ensino de programação.

(MONCLAR; SILVA; XEXÉO, 2018) argumenta que alunos de disciplinas introdutórias de cursos relacionados a TIC costumam enfrentar dificuldades para entender o funcionamento da programação. A maior parte dessas dificuldades estão relacionadas com a falta de domínio sobre o conteúdo, principalmente associados à falta de conhecimento em relação a conceitos abstratos, raciocínio lógico, resolução de problemas e abstração de alto nível.

Do mesmo modo em que tais dificuldades aumentam, as taxas relacionadas à inscrição de alunos em cursos da área diminuem na mesma medida em que as taxas relacionadas à evasão aumentam. O meio encontrado por (MONCLAR; SILVA; XEXÉO, 2018) para tentar solucionar esses problemas, foi a utilização de jogos com a finalidade de fazer com os alunos construam conhecimento através de experiências realizadas através de jogos.

(MONCLAR; SILVA; XEXÉO, 2018) entende que a utilização de jogos com a finalidade de ensinar programação pode ser um grande facilitador durante o ensino e aprendizagem do mesmo; Desenvolvendo a lógica de programação, abstraindo seus conceitos e seu funcionamento.

3.3 Uma taxonomia para ambientes de aprendizado baseados no computador.

(BARANAUSKAS et al., 1999), visa trabalhar com atividades lúdicas com o intuito de captar a atenção de quem está aprendendo, de forma que o estudante sinta-se motivado a continuar nesse processo de aquisição de conhecimento. Ele baseia seu estudo utilizando a experiência obtida através de 38 alunos universitários iniciantes provenientes dos cursos de computação e engenharia da universidade de Salvador. O autor faz uso de atividades lúdicas para evidenciar e complementar aquilo que é visto na teoria, ajudando na formação e abstração dos conceitos vistos em aula.

De acordo com (BARANAUSKAS et al., 1999), os resultados gerados através desse experimento foram bem satisfatórios, pois foi notório que o ensino aliado a essa metodologia

lúdica puderam tornar a compreensão dos alunos mais fácil. Porém, os alunos sentiram falta da visualização de códigos-fonte durante esse processo.

3.4 A utilização da ferramenta scratch como auxílio na aprendizagem de lógica de programação.

(NETO, 2013) fala sobre os desafios que alunos iniciantes passam para que possam aprender disciplinas introdutórias ao mundo da programação. Ele utiliza como base a turma anterior ao estudo, que cursava técnico de Informática no ano de 2012. Foi notado pelo autor a grande dificuldade apresentada por esses alunos, também tornou-se evidente o desinteresse dos mesmos em continuar na área.

(NETO, 2013) tem como objetivo contornar esse desinteresse utilizando uma metodologia motivadora, utilizando a ferramenta SCRATCH como auxiliadora. Durante o trabalho (NETO, 2013) demonstra e explica sobre o funcionamento da plataforma SCRATCH, e argumenta que graças a sua interatividade com o usuário faz com que o interesse do mesmo aumente.

Ao fim, (NETO, 2013) observa que o SCRATCH é uma valiosa ferramenta para introduzir os alunos ao mundo da programação, porém, ela não pode ser utilizada de forma isolada, pois a ferramenta usa de uma abordagem de arrastar e conectar blocos, o que difere bastante da programação habitual. Mas, o SCRATCH obteve sucesso significativo em cativar e iniciar estudantes em conceitos que ainda não haviam presenciado em seu cotidiano.

Dessa forma (NETO, 2013), obteve êxito ao conseguir motivar os alunos a continuarem na disciplina, e conseqüentemente na área. Portanto, a utilização do SCRATCH apresenta uma maneira diferenciada no ensino de conceitos computacionais, e pode amenizar o problema de compreensão, além de contribuir ricamente para a formação do profissional na área.

3.5 Percepção de docentes que lecionam programação de computadores quanto à formação pedagógica.

(BERSSANETTE; FRANCISCO, 2021) leva em consideração o advento das tecnologias, assim como a relevância de computadores na sociedade. Acarretando em habilidades relacionadas à programação, de modo que há uma necessidade de desenvolver tais habilidades na educação formal através do ensino de programação de computadores.

(BERSSANETTE; FRANCISCO, 2021) em concordância com (MONCLAR; SILVA; XEXÉO, 2018) definem que as dificuldades relacionadas ao ensino e aprendizagem de programação podem ser consideradas como gargalos existentes em cursos de TIC.

Existem diversos estudos que visam contribuir para a solução desse problema. No entanto, nenhum deles pôde ser considerado como uma solução completa. Tendo isso em vista (BERSSANETTE; FRANCISCO, 2021) avalia a percepção de docentes que lecionam em disciplinas relacionadas a programação, quanto ao ensino e aprendizagem de programação, como também a formação pedagógica. Com o objetivo de tornar mais eficaz a busca de uma solução completa para esse problema.

3.6 Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando gamification.

Como supracitado, (SILVA; MELO; TEDESCO, 2018) também elucida através de seu estudo as adversidades que são enfrentadas por alunos ao ingressarem em cursos de TIC, principalmente relacionadas à programação. Dificuldades comumente encontradas e citadas por alunos que estão relacionadas a abstração de conceitos de programação e a lógica de programação.

Em consequência disso há uma alta taxa de desmotivação nos alunos. Por conta disso, se torna necessário que docentes e discentes da área busquem por soluções para o mesmo.

A forma escolhida por (SILVA; MELO; TEDESCO, 2018) para tentar solucionar esse problema é a utilização de jogos com o intuito de motivar e engajar os alunos a aprenderem, essa prática é conhecida como gamificação. Diante do atual estado da arte (R3) através de seu estudo, tem como finalidade investigar a influência positiva da gamificação em promover engajamento dos alunos em relação a programação.

Em virtude dos trabalhos apresentados é possível concluir que as utilizações de ferramentas nos estudos levantados anteriormente tornam-se validadas como funcionais, e por conta disso o presente trabalho está embasado nessa metodologia, utilizando ferramentas com o intuito de ensinar a estudantes sobre conceitos de programação e por consequência disso os ensinar a programar.

4 METODOLOGIA

Neste tópico será retratada a metodologia utilizada neste estudo, através de uma pesquisa qualitativa, com o intuito de apresentar a aplicação de modelos de software como método auxiliar no ensino-aprendizagem de disciplinas de algoritmo ou de programação.

A metodologia utilizada tem como natureza a pesquisa bibliográfica, visto que de acordo com (LIMA; MIOTO, 2007).

A princípio, pesquisas foram feitas para obter um melhor entendimento em relação às dificuldades encontradas por alunos que estão ingressando na área de Computação, e por sua vez, nas disciplinas de algoritmo ou programação. Também tem-se conhecimento de que existe uma alta taxa de reprovação dos alunos, e por consequência disso a evasão dos mesmos, de acordo com (EDUCAÇÃO, 2022) a taxa de evasão do Campus Aracati está em 37,72%, além de que, a taxa de suspensão temporária está em 14,19% podendo evoluir para uma desistência. Levando em consideração todos os campi do estado do Ceará esses números demonstram uma taxa de evasão de 33,68% e a de suspensão temporária de 16,17%. Sendo esse um problema geral, faz-se necessário encontrar metodologias que diminuam a evasão, a suspensão temporária ou estimulem o retorno.

A partir disso, a metodologia utilizada para analisar a eficiência das ferramentas para o ensino-aprendizagem de programação, tem como base a comparação de metodologias aplicadas

por autores em trabalhos semelhantes, de modo que, serão abordados critérios positivos e negativos. Avaliações com relação a relevância de cada ferramenta serão aplicadas e estudadas dentro desse contexto.

4.1 PESQUISA DE FERRAMENTAS

Durante a pesquisa bibliográfica, o foco principal estava relacionado aos *softwares* que são utilizados durante o ensino-aprendizagem de programação ou algoritmos. Desse modo tornou-se possível definir a forma que a coleta de dados deveria tomar, e também foi determinado o modelo de análise a ser seguido.

Diversos softwares foram analisados para que fossem utilizados como ferramentas para o ensino-aprendizagem de programação, salvo que, todos esses têm o enfoque nessa finalidade. Porém, foram analisados aqueles que se encaixavam melhor com a realidade da pesquisa. Conforme a análise feita em relação às ferramentas citadas anteriormente, foram escolhidas o Code.org, MIT App Inventor, Arduíno e o LEGO SPIKE PRIME.

Software	Web	Mobile	Desktop	Português	Gratuito
CronApp	SIM	SIM	NÃO	SIM	NÃO
Blocly.games	SIM	NÃO	NÃO	SIM	SIM
Scratch	SIM	NÃO	SIM	SIM	SIM
CodeCademy	SIM	NÃO	NÃO	NÃO	NÃO
Robocode	NÃO	NÃO	SIM	NÃO	SIM
CodeCombat	SIM	NÃO	NÃO	NÃO	NÃO
MIT App Inventor	SIM	SIM	SIM	NÃO	SIM
Code.org	SIM	NÃO	SIM	SIM	SIM
Programming Hero	SIM	SIM	NÃO	NÃO	NÃO
LEGO SPIKE	SIM	NÃO	SIM	SIM	SIM
Arduino	SIM	NÃO	SIM	SIM	SIM

Tabela 1 – Comparação de Ferramentas

A Tabela 1 apresenta alguns pontos importantes que foram considerados em relação às ferramentas, sendo eles ser multiplataforma, estar em língua portuguesa, ser um auxiliador para o ensino de programação e ser gratuito. E as ferramentas melhores avaliadas foram o Scratch, Code.org e o MIT App Inventor.

O MIT App Inventor tem 8.2 milhões de usuários registrados até o mês de novembro de 2021, sendo utilizado por alunos de instituições com o intuito de solucionar problemas cotidianos apresentados em sua página WEB.

Já Code.org disponibilizou informações que até o mês agosto de 2021 a plataforma contava com 2.051.327 usuários cadastrados. Deve-se salientar que, o fato de todos eles estarem disponíveis em plataformas *online* as tornam mais atrativas, já que podem ser acessados de qualquer lugar e a qualquer hora.

Com a união da computação móvel, da Internet, da educação a distância(EAD) e de redes de comunicação sem fio, é possível obter um impulsionamento no ensino-aprendizagem partindo

da ideia de que não existem barreiras geográficas, tampouco temporais que possam dificultar ao estudante aprender.(PASSOS; CAMARÁ, 2016)

Esse conceito é definido como u-learning de acordo com (PASSOS; CAMARÁ, 2016), que é uma forma de educação ubíqua onde faz analogia à crença cristã, que define Deus como onisciente e onipresente, partindo deste mesmo princípio em que o estudante escolhe o local e o horário para realizar as atividades propostas.

5 ESTUDO DE CASO

O estudo de caso será realizado com alunos do ensino fundamental da rede municipal de educação dos municípios do Ceará.

Este estudo tem como objetivo apresentar uma metodologia de ensino de programação que utiliza ferramentas computacionais que auxiliem no ensino-aprendizagem de programação/computação tornando-o mais fácil e interativo, mostrando quais ferramentas se destacam em cada situação, não deixando de lado os benefícios propiciados aos estudantes que não tem em sua grade curricular a necessidade do ensino de programação, como é o caso dos alunos do ensino fundamental.

Os estudantes contemplados com o estudo foram observados por professores ou monitores que estiverem ministrando o módulo, e dessa forma podendo avaliar quais benefícios foram apresentados após essa prática. Esses benefícios vão desde o aumento da participação dos estudantes nas aulas, até o desenvolvimento do raciocínio lógico que aumentou o desempenho dos mesmos, principalmente em disciplinas cuja base é a matemática.

O estudo aconteceu durante um minicurso ofertado no Instituto Federal de Educação, Ciência e Tecnologia do Ceará, campus Aracati. A ordem de realização dos módulos ocorreu da seguinte forma: no primeiro módulo foi realizado um encontro com os alunos, para que fosse explicado o objetivo do estudo de caso que foi realizado, e apresentar a forma em que o mesmo ocorreu no decorrer das aulas, também apresentando conceitos relacionados a computação e programação.

No módulo 2 foi apresentado o Code.org, suas funcionalidades, atividades e objetivos. Dando ênfase aos conceitos de programação e se aprofundando no conteúdo com o decorrer das atividades.

No módulo 3 os alunos utilizaram o MIT App Inventor aprendendo sobre suas funcionalidades através de atividades e objetivos. Foi dada continuidade às atividades com a finalidade de tornar mais concretos os conceitos programacionais aprendidos.

No módulo 4 os alunos aprenderam sobre Python, e colocaram em prática conceitos vistos anteriormente. Visando a autonomia, os participantes recebem uma pequena introdução sobre a linguagem, são apresentados a um problema e em seguida estimulados a realizar suas próprias pesquisas para solucioná-lo.

No módulo final os alunos estudaram sobre robótica utilizando o Arduino e o LEGO SPIKE PRIME, desenvolvendo circuitos e projetos, que serão programados através das platafor-

mas respectivas de cada ferramenta.

Espera-se, ao fim desse estudo de caso, obter materiais para responder às seguintes questões: Os objetivos de aprendizagem foram atingidos ao fim do estudo? O método utilizado trouxe motivação e interesse aos alunos a aprenderem lógica de programação? Quais benefícios externos foram trazidos pelo método?

5.1 Questionários

O questionário foi criado com o intuito de avaliar a estratégia para o ensino-aprendizagem de conceitos computacionais (em especial a programação) na forma de um minicurso lecionado para adolescentes que estão em seus últimos anos do ensino fundamental. Visa-se contribuir para a ampliação de sua visão sobre o campo acadêmico ao qual podem seguir. Visto que, a instituição onde o minicurso é ofertado dá a possibilidade de seguir na área.

O questionário foi desenvolvido para que fosse possível a coleta de dados em uma forma mais centralizada, fazendo com que os dados não ficassem tão dispersos, de modo que contribuísse para sua coleta e análise. Esse questionário foi elaborado baseando-se em trabalhos realizados por (MEDEIROS; SILVA; ARANHA, 2013), (ANDRADE et al., 2020) e (WANGENHEIM; NUNES; SANTOS, 2014).

No apêndice A será apresentado como o questionário foi realizado. Os mesmos foram respondidos por professores ou instrutores que ministraram o mini curso, para que assim pudesse haver a coleta de dados concernente as ferramentas utilizada, assim como a metodologia que foi utilizada.

5.2 Efeitos Esperados

5.2.1 Desenvolvimento de autonomia

A autonomia nesse contexto, se trata da participação do aluno de forma ativa na construção do seu conhecimento, tornando-se o ator principal. A autonomia é um dos principais efeitos que foram evidenciados durante as pesquisas realizadas em relação ao atual estado da arte.(PAIVA et al., 2016)

5.2.2 Desenvolvimento de pensamento crítico

O desenvolvimento do pensamento crítico evidencia-se através das habilidades que o aluno tem em tomar decisões equilibradas e assertivas, tornando o seu progresso mais eficaz. A crítica ao conhecimento ajuda os alunos a aprimorar o seu desenvolvimento.(MARTINS; OLIVEIRA; KOMOSINSKI, 2005)

5.2.3 Pensamento cognitivo real e potencial

(VALE; NEVES et al., 2018) consideram a cognição como a habilidade do aluno de processar informações através da percepção. Que é produzida por meio de sentidos humanos,

sendo algum específico ou por combinações deles.

O desenvolvimento cognitivo real é evidenciado de forma individual, de modo que o aluno consegue se desenvolver de forma autônoma. Enquanto o desenvolvimento cognitivo potencial necessita da ajuda de terceiros, como em uma relação de aluno e professor.(VALE; NEVES et al., 2018)

5.2.4 *Estimulo criativo*

Conforme observado por (ESTEVES, 2019), a criatividade acontece quando há uma interação entre o ambiente, aptidões e processos, a fim de que seja criado algo novo e útil em determinado contexto social. A criatividade também pode ser considerada um atributo relevante para que haja a resolução de problemas.

5.2.5 *Instrução em pares*

A instrução em pares trata-se da prática em que os alunos interagem entre si, com o intuito de explicar uns aos outros os conceitos que estão sendo estudados.(CHICON; QUARESMA; GARCÊS, 2018)

5.2.6 *Aumento de foco*

O aumento de foco está ligado ao interesse que é expressado pelo aluno ao desenvolver as práticas propostas pelo professor. Tornando os alunos mais motivados durante suas práticas e aumentando sua produtividade.(BORGES, 2000)

6 RESULTADOS

Para que fosse viável acompanhar os resultados do estudo de caso foram realizadas observações que tornaram possíveis as verificações de motivação, interesse, comportamento e desenvolvimento dos alunos contemplados de modo geral. Ao todo foram analisadas 12 turmas diferentes.

A grande maioria dos alunos foram escolhidos sem a interferência deste trabalho, já que as secretarias de educação de cada município usaram seu próprio critério de decisão que se distinguiu de uma localidade para outra, muitas delas escolhendo alunos de turmas e escolas diferentes. Porém, houve casos onde a secretaria de educação escolheu motivada pelo interesse dos alunos.

O estudo utilizou duas abordagens para a coleta de dados, a abordagem 1 foi feita através do formulário visto na seção 5.1. O mesmo foi enviado aos instrutores e monitores a fim de que eles o respondessem ao término de cada módulo do minicurso. As perguntas dispostas no formulário foram desenvolvidas com a ajuda de outros trabalhos com a mesma finalidade. Desse modo foi possível obter os dados necessários para a análise.

Já a abordagem 2 se deu através de observações do autor, onde as turmas foram analisadas durante todo o minicurso e os dados puderam ser coletados, além dos dados por observação também foram realizadas entrevistas com os instrutores ao fim de cada módulo do minicurso para que eles pudessem contribuir comentando sobre sua visão de cada turma.

De acordo com a abordagem 2, no início os alunos contemplados mostravam-se ignorantes quanto ao conteúdo ministrado, não só em relação às práticas referentes a programação, mas também no que diz respeito ao manejo de computadores. Por mais que todos tivessem smartphones, poucos deles já haviam utilizado computadores, o que tornou um pouco mais lenta a realização das atividades elaboradas pelos instrutores que tinham a necessidade do uso de computadores.

Os dados analisados através da abordagem 1 estão dispostos nas tabelas 2 e 3 que serão apresentadas no decorrer da seção e estarão descritas nos textos que as seguem.

Com o auxílio do estudo foi possível analisar as ferramentas e assim identificar a que melhor se adequa para a finalidade do estudo. Ao utilizar a soma dos resultados obtidos no questionário foi possível classificar as ferramentas da seguinte forma: Python, MIT App Inventor, Code.org e Robótica/LEGO respectivamente, como é possível visualizar na tabela 2 de acordo com as seguintes perguntas.

- I. Os objetivos de aprendizagem foram alcançados?
- II. Foi possível concluir o conteúdo programado?
- III. A ferramenta utilizada facilita a aprendizagem da prática computacional/programação ?
- IV. A ferramenta utilizada engaja os alunos a aprenderem?

	PYTHON	MIT APP INVENTOR	CODE.ORG	ROBÓTICA/LEGO
I	96%	84%	80%	84%
II	100%	0%	36,36%	33.33%
III	100%	94%	90%	84%
IV	100%	94%	96%	84%

Tabela 2 – Engajamento/Proposta

Os dados dispostos na tabela 2 foram obtidos através da abordagem 1 e a porcentagem vista foi dada em relação ao total de turmas realizadas, por exemplo o índice I de python mostra que 96% das turmas conseguiram alcançar os objetivos que foram estipulados para esse módulo, no entanto o conteúdo que havia sido programado pôde ser finalizado em todas as turmas.

O python foi o que obteve o melhor desempenho nos dados vistos na tabela 2. Como dito anteriormente 80% das turmas conseguiram ser plenas no alcance de aprendizagem previsto e o conteúdo pretendido chegou a conclusão em todas as turmas.

Em 100% das turmas foi possível observar que o python facilita a aprendizagem computacional e de programação. Mediante a abordagem 2 foi possível contribuir para essa conclusão,

já que o python é uma linguagem de programação propriamente dita, e que a mesma nesse caso de fácil compreensão. O que fez com que os alunos obtivessem um desempenho melhor e conseguissem aprender mais sobre o assunto.

Como apresentado na tabela 2 é possível identificar que 100% das turmas sentiu-se engajadas no propósito de aprender mais sobre a programação, assim como o python. De acordo com a abordagem 2 os alunos apresentavam-se tão engajados que foi possível finalizar o conteúdo programado rapidamente e não satisfeitos continuaram em tal intensidade que buscavam novas formas de deixar seus algoritmos mais robustos, assim adquirindo mais conhecimento.

A ferramenta MIT App Inventor (MAI) foi a segunda melhor avaliada seguindo os dados que foram apresentados na tabela 2, nela é possível observar que em 84% das turmas foram alcançados os objetivos de aprendizagem almejados, mas em contrapartida o conteúdo que havia sido estipulado para acontecer neste módulo não pôde ser concluído em nenhuma das turmas realizadas.

No entanto, de acordo com a abordagem 2, foi observado que esse valor em relação à conclusão de conteúdo pode ser mais atrativo do que se imagina. Pois, a não conclusão se deve ao fato dos alunos estarem tão engajados em partes específicas do módulo que acabaram atrasando a programação do instrutor. De tal modo que o mesmo teve que se adaptar às preferências de cada turma, e assim desenvolver melhor as práticas às quais os alunos estavam envolvidos.

Conforme visto na tabela 2 o MAI facilitou a aprendizagem computacional e de programação em 94% das turmas, o que é um valor bastante expressivo visto que se trata de uma ferramenta.

Assim como observado na abordagem 2, o MAI facilita a aprendizagem devido utilizar programação em blocos onde os alunos não precisam aprender uma linguagem de programação convencional para que possam desenvolver seus algoritmos, sendo necessário somente clicar e arrastar aquilo que for preciso, o que torna essa construção mais fácil. Mas para que essa aplicação funcione o aluno precisa organizar os blocos de forma lógica, e de modo eles conseguem praticar e aplicar conceitos computacionais e a lógica de programação que os mesmos vêm desenvolvendo no decorrer do minicurso e através de pesquisas e conteúdos proporcionados pelos instrutores.

A tabela também mostra que em 94% das turmas pode-se notar o alto grau de engajamento ao utilizarem o MAI. Assim como já havia sido salientado anteriormente, de acordo com a abordagem 2 os alunos mostraram-se muito entusiasmados ao utilizá-lo e conseguiram aprofundar-se nele mesmo sem conseguir completar o conteúdo planejado pelo instrutor.

Dando continuidade a próxima ferramenta melhor avaliada é o Code.org que obteve em 80% das turmas um ótimo resultado em relação ao alcance de objetivos. A ferramenta se assemelha ao MAI já que o Code.org também usa a programação em blocos, o que torna mais atrativo aos que estão iniciando na programação. Mas o mesmo também tem suas peculiaridades, principalmente por ter uma dinâmica baseada em jogos onde conforme o avanço das fases novos conceitos são expostos para que os alunos aprendam e possam utilizá-los de forma que otimizem seus algoritmos.

A taxa de conclusão do conteúdo programado do módulo Code.org é de apenas 36,36%

das turmas. Todavia, isso se dá em consequência do funcionamento do Code.org, por ser uma plataforma bastante intuitiva e dinâmica os alunos acabam se aprofundando em módulos internos, e dessa forma aprendem se divertindo. Contudo a aula que havia sido formulada pelo instrutor é atrasada e o mesmo tem que se adaptar a essa realidade, assim como observado na abordagem 2.

De acordo com a tabela 2, em 90% dos casos foi possível observar que o Code.org contribui e facilita na aprendizagem computacional e de programação. Conforme visto na abordagem 2 os alunos conseguiram compreender conceitos e lógica de programação de uma forma simples e divertida, uma vez que se sentem mais confortáveis e menos pressionados com uma aprendizagem parecida com jogos que estão acostumados.

Assim como visto na tabela 2, em 96% das turmas houve um alto índice de engajamento dos alunos ao utilizarem a ferramenta Code.org. Como já discutido anteriormente, por assemelhar-se a um jogo é possível perceber através da abordagem 2 que os alunos conseguem manter-se bastante interessados, e também buscam avançar os níveis da melhor maneira possível, já que a plataforma mostra quando o algoritmo pode ser otimizado, fazendo com que aprendam formas mais efetivas de desenvolver seus problemas, os motivando a aprender e melhorar.

O módulo seguinte, de Robótica/LEGO (R/L) alcançou os objetivos de aprendizagem em 84% das turmas. Nesse módulo são realizadas atividades em duas etapas, a primeira é feita utilizando um microcontrolador chamado Arduíno e são criados circuitos simples, para que os alunos possam entender um pouco mais sobre robótica, o mesmo é programável devido a esse microcontrolador e essa programação é feita através de software e linguagem próprias.

Ainda nesse módulo é apresentado o LEGO SPIKE PRIME, também utilizado para a robótica e é uma forma mais simples dos alunos desenvolverem seus protótipos, já que assim como o LEGO clássico que é conhecido mundialmente, o SPIKE PRIME também só precisa que as peças sejam escolhidas e montadas, como foi evidenciado através da abordagem 2.

Assim como a maioria dos módulos, a taxa de conclusão do conteúdo programático de R/L é igualmente baixa, com apenas 33,33% das turmas atingindo a sua completude. No entanto, isso não é um problema, visto que o intuito principal, que é o de incentivar os alunos a aprenderem sobre robótica e programação, pôde ser alcançado.

Dentre todos os módulos o R/L é o que apresenta o pior desempenho em facilitar a prática computacional e de programação, principalmente na etapa em que se utiliza o Arduíno, que se mostrou pouco intuitivo. Isto se deu porque os participantes precisam aprender uma nova linguagem de programação, e como ainda não possuem uma base robusta sobre conceitos e lógica de programação isso torna-se algo desinteressante para os alunos, como observado pela abordagem 2.

Em contrapartida, quando os alunos utilizam o LEGO eles ficam eufóricos e tem mais interesse no conteúdo, sem falar que a plataforma que é utilizada para programar os protótipos usa de programação em blocos para desenvolver o algoritmo, algo que os mesmo estão mais habituados a fazer no decorrer do minicurso.

O módulo R/L só conseguiu atingir plena motivação em 84% das turmas. De acordo com a abordagem 2, isso se dá principalmente por conta da dificuldade dos alunos nas práticas com o

Arduíno e ao criarem os seus circuitos, isso acabou os desmotivando. Porém, eles voltam a se interessar quando começam as atividades utilizando o LEGO. Quando eles começam a ver que seus projetos estão começando a “ganhar vida” os alunos voltam a sentir-se motivados e fazem o melhor possível para conseguir atingir o desafio proposto pelo instrutor.

Foram analisados também efeitos esperados com o decorrer do estudo de cada ferramenta, visando classificar cada uma da melhor forma possível. Utilizando os dados de efeitos obtidos e unindo aos dados de engajamento e proposta foi possível ordenar as ferramentas da seguinte forma: Code.org, MIT App Inventor, Python e Robótica/LEGO respectivamente. Assim como mostrado na tabela 3, abaixo também se encontra o índice com seus respectivos efeitos esperados que serão representados na tabela 3.

- I. Desenvolvimento de autonomia.
- II. Desenvolvimento de pensamento crítico.
- III. Desenvolvimento cognitivo real.
- IV. Desenvolvimento cognitivo potencial.
- V. Estímulo criativo.
- VI. Instrução em pares.
- VII. Aumento de foco.

	CODE.ORG	MIT APP INVENTOR	PYTHON	ROBÓTICA/LEGO
I	71,5%	78,6%	28,5%	21,5%
II	43,0%	85,7%	28,5%	14,5%
III	71,5%	85,7%	7,15%	0%
IV	78,6%	85,7%	35,7%	43,0%
V	43,0%	35,7%	0%	28,5%
VI	71,5%	43,0%	50,0%	35,7%
VII	71,5%	28,5%	14,5%	7,15%

Tabela 3 – Efeitos esperados

Conforme visto na tabela 3, há uma ótima atuação do Code.org na geração de efeitos esperados, o mesmo conseguiu gerar todos os efeitos, em algum nível. Desenvolvendo-os na sua maioria acima de 70% das turmas do módulo. Esse excelente resultado deve-se à forma em que o Code.org atua, visto que, ele age de forma guiada pela plataforma em si, além de também haver o conteúdo programado pelo professor.

De acordo com observações feitas através da abordagem 2, tornou-se claro que as taxas referentes ao desenvolvimento de autonomia(DA), desenvolvimento cognitivo real(DCR) e aumento de foco(AF) estão ligadas entre si. Os alunos do módulo Code.org conseguiram mostrar-se autônomos em relação às atividades propostas, desenvolvendo a capacidade de resolver

problemas de forma independente e desse modo também demonstrando sua alta capacidade de foco.

Porém, em determinadas situações os alunos do módulo Code.org mostravam-se um pouco mais dependentes do auxílio do professor, monitor ou até mesmo de outros alunos. No entanto, após as orientações os mesmos conseguiam avançar e superar novos desafios, desempenhando uma elevada taxa de desenvolvimento cognitivo potencial(DCP) e também de instrução em pares(IP).

No entanto, o módulo Code.org também apresentou duas taxas igualmente baixas, o desenvolvimento de pensamento crítico(DPC) e o estímulo criativo(EC). Esse baixo desempenho está atrelado a forma em que o Code.org trabalha, visto que, ele é uma plataforma guiada que dispõe suas atividades em forma de fase, aumentando a complexidade conforme se avança. Não dando brechas para que os alunos demonstrem tanta criatividade. Além disso, a plataforma também mostra quando o algoritmo pode ser otimizado, de modo que o aluno decide se continua expressando sua criatividade ou se otimiza o mesmo.

O MAI assemelha-se com o Code.org, pois o mesmo conseguiu obter valores notáveis na maioria dos efeitos. Esse resultado ocorre graças à metodologia aplicada durante o módulo MAI, onde o professor quem orienta a forma que os alunos deverão fazer as atividades. Além de que a plataforma proporciona diversas possibilidades aos alunos.

O MAI conseguiu desenvolver altas taxas em relação ao DA, DPC, DCR E DCP. Conforme as atividades eram expostas aos alunos, os mesmos mostravam-se autônomos e conseguiam desenvolver aplicações propostas pelo professor, demonstrando também o DCR. Não passando despercebido que quando os alunos tinham dificuldades eles buscavam orientações de monitores ou professores, que os instruíam e os alunos conseguiam superar essas dificuldades, avançando nas atividades.

Em virtude das atividades propostas e pela plataforma do MAI ter uma vasta gama de funcionalidades, os alunos puderam obter uma alta taxa de DPC, conseguindo definir qual o melhor caminho a ser seguido durante as atividades.

Porém, em consequência dessa abrangência em relação a funcionalidades propiciadas pelo MAI, taxas como EC, IP e AF caíram em relação aos demais efeitos. Isso ocorreu por conta de haver um conteúdo tão abrangente que os alunos acabaram perdendo-se com a quantidade de possibilidades. Perdendo o foco do que era necessário para concluir as atividades, e ao mesmo tempo tentando seguir instruções do professor o que proporcionou pouca margem para que expressassem sua criatividade, do mesmo modo que a falta de domínio sobre a plataforma acarretou na dificuldade dos alunos se ajudarem.

Já em relação aos módulos de Python e R/L as taxas têm uma grande queda, se comparadas às taxas dos módulos anteriores. Essa redução no desenvolvimento dos efeitos ocorre por consequência da utilização de uma linguagem profissional nos módulos seguintes, além de usar equipamentos elétricos durante as práticas de R/L.

Efeitos Esperados

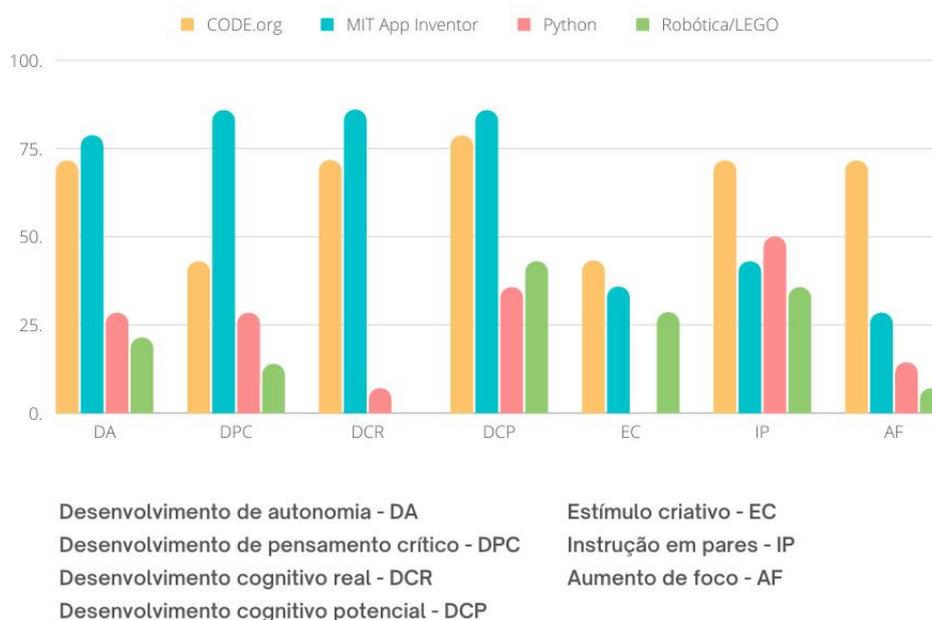


Figura 2 – Representação gráfica dos efeitos esperados.

Em decorrência do modo em que o módulo Python é aplicado houve uma queda no desempenho das turmas na geração de efeitos previstos. Assim como dito anteriormente, por tratar-se de uma linguagem profissional a utilização do Python se difere das ferramentas utilizadas outrora. Nesse módulo os alunos recebiam apenas algumas instruções do professor, para que os mesmos pesquisassem para que fosse possível resolver aquilo que era pedido pelo professor.

Os alunos deveriam buscar soluções por conta própria à medida em que as dificuldades apareciam. Tal metodologia causou insegurança nos alunos, tornando taxas como DA, DPC, DCR, EC e AF menores. Esse primeiro contato com uma linguagem de programação profissional causou certo desânimo nos alunos, que muitas vezes não conseguiram desenvolver soluções de forma independente.

No entanto, o módulo Python apresentou um aumento no desenvolvimento de efeitos que visam a cooperação de terceiros, como o DCP e a IP. Esse aumento deve-se em consequência das dificuldades proporcionadas pelo módulo, onde os alunos sentiam a necessidade de pedir ajuda ao professor, monitor ou até mesmo aos colegas que conseguiram entender o motivo da dificuldade apresentada.

O módulo de R/L foi dividido em duas etapas, a primeira o professor apresenta o Arduíno e componentes elétricos para os alunos, e posteriormente é apresentado o LEGO SPIKE PRIME para que os alunos possam prototipar das duas formas.

É possível notar que as taxas dos efeitos esperados caem bastante durante o módulo R/L

assim como no Python, por mais que as atividades sejam guiadas de um modo parecido com o MAI e o Code.org.

Dentre todas as taxas produzidas no módulo R/L a menor é a de DCR. Visto que, para os alunos a programação já é algo novo e que está no processo de aprendizagem ainda, a dificuldade para o manuseio de componentes elétricos aliados as linguagens de programação tornam as práticas ainda mais complexas, reduzindo a produção de DA, DPC, EC e AF.

Porém, houve um aumento nas taxas de DCP e IP. Isso ocorre pois mesmo em meio a tantas dificuldades apresentadas no decorrer das atividades, alguns alunos conseguiram se sobressair e avançar nos desafios, e os mesmos ajudavam os colegas que ainda não haviam conseguido, gerando desenvolvimento em tais efeitos.

6.1 TURMAS DE CONTROLE

Para que fosse possível validar a metodologia e contrastar os resultados e efeitos obtidos nos alunos através do estudo, foram realizadas turmas de controle onde foram replicadas as fases do mesmo, tendo como objetivo obter resultados melhores ou similares aos atingidos com as turmas anteriores.

Essas turmas de controle foram divididas em dois tipos: turmas com conhecimento prévio em programação(CCP) e turmas que não tinham esse tipo de conhecimento mas possuíam certo conhecimento, em relação ao uso de computadores(SCP).

6.1.1 TURMAS COM CONHECIMENTO PRÉVIO

Para as turmas de controle CCP o estudo foi feito com turmas de técnico em informática do campus IFCE Aracati. No entanto, diferente da forma que o estudo foi realizado nas turmas do ensino fundamental, não houve necessidade de aplicar todas as fases, até mesmo por conta que eles já tinham o conhecimento sobre o conteúdo.

O objetivo principal dessa turma de controle foi o de saber se eles conseguiriam concluir desafios de programação propostos, utilizando uma linguagem de programação nunca vista pelos mesmos.

Como as turmas já haviam utilizado a linguagem de programação python anteriormente. Ela não foi utilizada, no lugar dela os alunos tiveram que usar a linguagem Ruby. Assim como dito anteriormente eles precisavam pesquisar sobre o funcionamento e como utilizar a linguagem para desenvolver os desafios propostos.

A princípio foi pedido que os alunos fizessem uma breve pesquisa sobre o Ruby para que eles pudessem se basear e começar a programar, devido ao tempo reduzido que eles teriam, só foi proposto apenas um desafio.

Esse desafio consistia em desenvolver uma calculadora simples, com os seguintes requisitos:

CALCULADORA

- Entrada/Saída - valores Inteiros.
- Operações Aritméticas - Soma, Subtração, Divisão e Multiplicação.

ORGANIZAÇÃO DA CALCULADORA (ESPERADA)

- Saída - Insira o primeiro valor.
- Entrada - Valor1
- Saída - Insira a operação aritmética.
- Entrada - Operação
- Saída - Insira o segundo valor.
- Entrada - Valor2
- Saída - O resultado é ().

A princípio foi pedido aos alunos que programassem as calculadoras seguindo essas instruções para que assim eles conseguissem desbravar essa nova linguagem, pesquisando e praticando ao mesmo tempo. Por mais que esse fosse o padrão esperado, não necessariamente os alunos deveriam fazer dessa forma, ele serviu apenas como algo para eles se basearem e criarem suas próprias calculadoras.

Partindo desse princípio também foram pedidos recursos a mais, caso eles conseguissem terminar essa primeira parte e ainda houvesse tempo antes de chegar ao fim da aula.

CALCULADORA EXTRA

- Entrada/Saída - valores Reais.
- Operações aritméticas - Porcentagem, Radiciação, Potenciação e Precedência matemática.
- Leitura de expressão matemática completa - (o ideal seria tentar resolver sem utilizar nenhuma função pronta).
- Imaginação e criatividade do aluno.

Devido ao tempo reduzido que os alunos dispunham, era esperado que eles fizessem somente a primeira parte do desafio. Porém, a maioria dos alunos conseguiu fazer o desafio de forma completa, alguns deles fazendo até mais do que havia sido proposto.

Para que não houvesse perda de tempo os alunos utilizaram a plataforma online do Replit.com, onde não foi necessário a instalação de nenhum pacote, biblioteca ou nada relacionado a isso.

6.1.2 TURMAS SEM CONHECIMENTO PRÉVIO

Já com a turma SCP foi um pouco diferente, primeiro foram desafiadas a replicar a mesma proposta aplicada com a turma CCP. Para que assim fosse possível avaliar a autonomia dos alunos em pesquisarem, aprenderem e aplicarem esse conhecimento. No entanto, como já era esperado, eles não conseguiram reproduzir os mesmos resultados que as turmas CCP.

Porém a turma SCP conseguiu abstrair algumas coisas da linguagem de programação Ruby, mesmo sem conseguir fazer uma calculadora. Após essa primeira parte foram aplicadas as mesmas fases feitas no estudo com as turmas de ensino fundamental, com a diferença que os alunos dessa vez faziam parte do ensino médio.

Essa proposta tem como objetivo dar uma nova oportunidade aos alunos SCP para que eles realizem o desafio, com o diferencial deles terem feito todas as etapas do estudo. Nesse momento imaginou-se que os alunos obteriam êxito no desafio apresentado.

CALCULADORA

- Entrada/Saída - valores Inteiros.
- Operações Aritméticas - Soma, Subtração, Divisão e Multiplicação.

ORGANIZAÇÃO DA CALCULADORA (ESPERADA)

- Saída - Insira o primeiro valor.
- Entrada - Valor1
- Saída - Insira a operação aritmética.
- Entrada - Operação
- Saída - Insira o segundo valor.
- Entrada - Valor2
- Saída - O resultado é ().

Cerca de 80% alunos SCP pós etapas do estudo conseguiram concluir a calculadora esperada utilizando a linguagem de programação Python, visto que eles já haviam estudado um pouco de Ruby durante o primeiro desafio proposto optou-se em aplicar outra linguagem de programação para que pudesse ser o mais fiel possível ao desafio original aplicado às turmas CCP.

CALCULADORA EXTRA

- Entrada/Saída - valores Reais.
- Operações aritméticas - Porcentagem, Radiciação, Potenciação e Precedência matemática.

- Leitura de expressão matemática completa - (o ideal seria tentar resolver sem utilizar nenhuma função pronta).
- Imaginação e criatividade do aluno.

Os outros 20% dos alunos conseguiram fazer a calculadora extra, criando funcionalidades além das mencionadas anteriormente. Podendo dessa forma provar a eficácia de aplicar todas as fases do estudo descrito com mais detalhes na seção 6.

6.1.3 RESULTADOS DAS TURMAS DE CONTROLE

Através das turmas de controle tornou-se possível observar e coletar dados relacionados à eficácia da metodologia empregada tanto no ensino convencional que é utilizado no IFCE, campus Aracati, quanto no projeto realizado na mesma instituição mas feito com alunos de escolas municipais e de nível fundamental.

Salvo também que a forma em que o desafio foi proposto para ambas turmas CCP e SCP se diferiu um pouco, por conta do que já foi supracitado em relação à questão dos tipos de alunos testados, uma parte já tinha conhecimento relacionado à área de programação enquanto a outra no máximo sabia como utilizar computadores. Algo que também se diferencia das turmas de ensino fundamental, é que em diversas vezes os alunos do fundamental estavam tendo o primeiro contato com computadores no momento em que o mini curso é ministrado.

Primeiramente os dados relacionados às turmas CCP estão paralelos aos esperados. Como as turmas testadas nesse quesito estavam cursando o segundo semestre em diante do curso técnico em informática, os resultados obtidos foram paralelos aos que eram previstos.

Não deixando de lado que mesmo que alguns alunos tenham tido dificuldades para desenvolver esse desafio, junto com a questão de aprender uma linguagem de programação em tempo hábil para conseguir programar. Vale ressaltar que nessa turma não houve intervenção alguma de professores ou instrutores, tornando a proposta ainda mais difícil de ser realizada.

Através da tabela 4 pode-se observar o desempenho de ambas as turmas CCP e SCP, e conforme a mesma esses dados serão melhor explicados abaixo:

- I. Desenvolvimento de autonomia.
- II. Desenvolvimento de pensamento crítico.
- III. Desenvolvimento cognitivo real.
- IV. Desenvolvimento cognitivo potencial.
- V. Estimulo criativo.
- VI. Instrução em pares.
- VII. Aumento de foco.

	I	II	III	IV	V	VI	VII
CCP	95%	80%	90%	0%	85%	90%	95%
SCP	100%	100%	0%	100%	50%	100%	50%

Tabela 4 – Resultados turmas de controle.

Vale ressaltar que esses efeitos observados na turma CCP já existiam, em decorrência de suas vivências em aprendizado de programação. Vale também frisar que graças ao ensino no IFCE esses alunos conseguiram acumular uma grande bagagem de conhecimento, que tornou-se evidente durante o experimento.

Como é possível identificar na tabela 4, ambas as turmas obtiveram grandes desempenhos relacionados ao DA. A turma CCP como dito anteriormente conseguiu expor aquilo que já havia aprendido, mostrando sua experiência e facilitando a aplicação do desafio.

Já com a turma SPC foi possível observar essa evolução no decorrer do minicurso. Desde o primeiro dia todos os alunos mostraram grandes capacidades em relação à sua autonomia, isso também se deve a questão dos mesmos terem optado por participar do experimento. De modo geral, em momento algum nenhuma das turmas mostrou-se desmotivada ao depararem-se com adversidades. Pelo contrário, estavam sempre dispostas a ultrapassar essas dificuldades.

Em relação ao DPC nota-se que a turma CCP era um pouco mais retida no início, quando o desafio foi apresentado aos alunos observou-se que eles seguiam as instruções sem muito questionamento. No entanto, em determinados momentos viu-se que os alunos buscavam melhores formas de resolver suas pendências.

Como em suas aulas regulares normalmente os professores dão instruções que precisam ser seguidas à risca, os alunos imaginaram que o desafio seria da mesma forma. Porém, não aconteceu desse modo. O desafio também serviu para estimular esse tipo de pensamento nos alunos, conforme a dificuldade em resolvê-lo tornava-se maior os alunos buscavam formas melhores e mais fáceis de solucionarem o mesmo.

Já a turma SCP mostrou-se diferente, como eles não estavam acostumados com nada relacionado a programação sempre buscavam formas mais fáceis ou mais eficazes de resolver aquilo que era proposto. Em razão disso, a cada etapa do minicurso havia um pouco mais de DPC na turma. Ao fim do experimento era notório que os alunos SCP haviam desenvolvido esse tipo de pensamento e que conseguiam deliberar melhor a cada decisão tomada.

Através do experimento tornou-se visível que as turmas CCP já haviam desenvolvido DCR, até mesmo porque esse dado está atrelado a autonomia, e como visto anteriormente os alunos se desempenharam bem nesse quesito. Já a turma SCP por mais que eles tenham uma alta taxa de autonomia ao realizarem as atividades, ainda assim os alunos pediam muita ajuda aos professores ou instrutores, de modo que o DCR não se evidenciou.

No entanto, a turma SPC conseguiu tirar sempre o melhor proveito possível de cada instrução. Obtendo um alto desempenho no DCP, os alunos que buscaram instruções conforme as dificuldades apareciam, e em virtude de seu pensamento crítico tornavam seus algoritmos melhores, desenvolvendo soluções mais eficazes para os problemas que apareciam.

Porém, não foi possível analisar o DCP na turma CCP, tornando sua taxa igual a 0%. Isso se dá em decorrência da forma em que o experimento foi realizado com os mesmos, não havendo brechas para ajudas externas. Já que o professor/instrutor não poderia intervir.

Como pode-se observar na tabela 4 a turma CCP tem um elevado nível criativo, algo que já havia sido trabalhado anteriormente em suas aulas regulares do curso de técnico em informática, criatividade que tornou-se evidente durante o experimento. No decorrer do desafio conforme os alunos concluíram as etapas eles tinham a liberdade para desenvolver de acordo com suas preferências, dito isso eles começaram a usar a criatividade para incrementarem seus algoritmos.

Relação de efeitos entre turmas de controle.

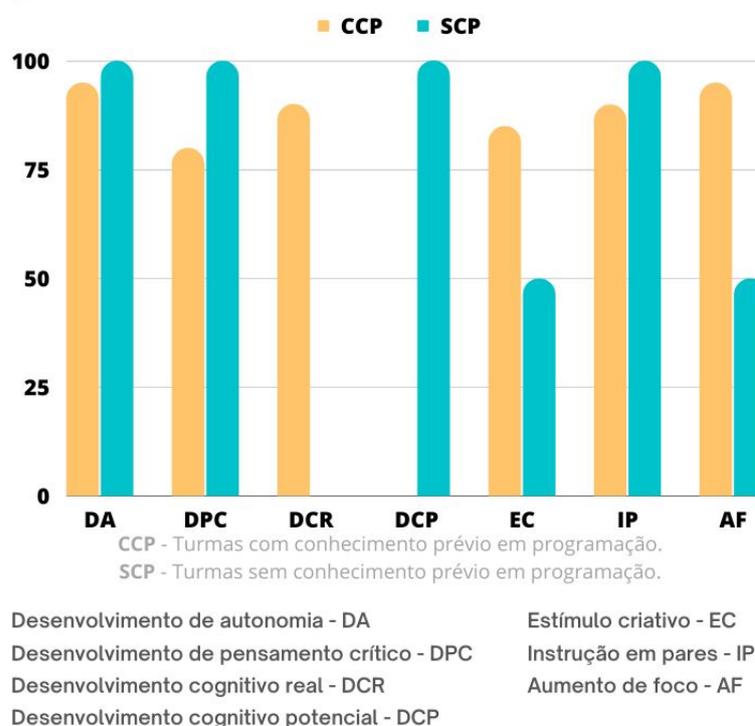


Figura 3 – Relação entre turmas CCP e SCP.

No entanto, esse índice relacionado ao EC cai em relação à turma SCP. Isso se deve principalmente ao fato dos alunos estarem tendo sua primeira interação com a área de programação, os mesmos estavam muito retidos em relação a desenvolver algo diferente do que estava no roteiro previsto. Claro que isto não se aplicou a todos os alunos SCP, alguns deles demonstraram grande potencial criativo no decorrer do minicurso saindo do que daquilo que havia sido estipulado e criando novas funcionalidades.

Observa-se que em ambas turmas CCP e SCP há um índice elevado de IP. Em razão disso pôde-se notar que os alunos estavam dispostos a ajudar um ao outro, isso também se deve ao fato de que eles se conheciam desde antes do experimento, o que nas turmas gerais da pesquisa normalmente gerou efeitos indesejados de concentração e etc.

O desempenho dos alunos torna-se melhor quando há esse tipo de interação, quando o aluno tem dificuldade em determinado ponto, outro aluno que já conseguiu resolver o ajuda e ensina aquilo que aprendeu. Dessa forma, as interrupções durante a aula para tirar dúvidas com o professor tornam-se cada vez menores.

Em relação ao AF da turma CCP, observou-se que após o desafio ter sido lançado a eles os mesmos estavam dispostos a concluí-lo. De modo que foi possível observar em 95% da turma a atuação desse efeito, assim como visto na figura 3.

Já com a turma SCP foi possível observar tal efeito em apenas 50% dos alunos. Essa taxa se dá por consequência da pouca experiência proveniente dos alunos, onde os mesmos ainda possuem muitas dúvidas e estão aprendendo a partir daquele momento. Porém, foi possível notar que com o decorrer do minicurso esse problema ia tornando-se menor, e à medida que conseguiam realizar as atividades ele adquiriam mais foco no restante dos problemas.

7 CONCLUSÃO

O processo de inicialização de estudantes no estudo de programação deve acontecer de forma que incentive os mesmos, para que assim possam construir competências suficientes de forma que consigam superar possíveis dificuldades e problemas que aparecerão no decorrer de sua jornada na programação. Visto que, essa perspectiva é de extrema importância para aqueles que estão desenvolvendo habilidades relacionadas a programação.

O atual estudo possibilita uma melhor compreensão do atual estado da arte no ensino-aprendizagem de programação. As pesquisas realizadas possibilitaram a descoberta de uma gama muito abrangente de artigos que discutem sobre problemas relacionados ao ensino-aprendizagem de programação e formas de abordá-los, com o intuito de minimizar tais dificuldades.

Durante o trabalho tornou-se evidente que a apresentação de problemas que devem ser resolvidos com a programação de forma direta causam espanto e desinteresse nos alunos que estão tendo o primeiro contato com o assunto. Assim como observado durante o processo realizado com a turma SCP, onde os alunos primeiramente foram desafiados a resolverem um problema proposto com uma linguagem de programação.

Porém, quando se utilizou uma abordagem que conceitos e linguagens de programação foram apresentados através de atividades lúdicas, os estudantes mostraram-se mais motivados e com um maior interesse ao iniciarem as atividades.

A metodologia utilizada durante o minicurso facilitou a abstração de conceitos relacionados a programação, além de ajudar os estudantes no desenvolvimento da lógica de programação, algo crucial para o ramo da programação. Também se notou que aulas que utilizam a gamificação em suas metodologias de ensino promovem um maior engajamento nos alunos. De modo que, os mesmos tendem a se esforçar mais para que possam continuar progredindo em suas atividades.

Além de que tais ferramentas possibilitaram aos alunos a visualização do funcionamento dos conceitos relacionados a programação, facilitando assim o seu entendimento. Vale ressaltar que a medida em que se avançam as fases do estudo, novos conceitos são apresentados e sua

complexidade aumenta, evoluindo até chegar na programação com uma linguagem convencional como: Python, C, Java ou a que for escolhida pelo estudante.

Foi possível observar que a linguagem visual facilitou o ensino para os alunos, porém, ela não obteve êxito ao ser utilizada de forma isolada. Durante o estudo pôde-se notar que alguns alunos podem ficar acomodados com uma linguagem de programação específica, sendo ela linguagem de blocos ou uma linguagem escrita. Visto que a programação utiliza de diversas linguagens para finalidades distintas, esse tipo de comportamento pode ser prejudicial para os estudantes.

Durante o estudo com a turma CCP foi notório que esse comportamento causa dificuldades aos alunos que estão buscando utilizar novas linguagens, por conta que cada linguagem de programação tem suas peculiaridades, os alunos tentaram usar novas linguagens da mesma forma que utilizavam as linguagens que estavam acostumados, causando assim erros de sintaxe e semântica.

Partindo desse princípio a linguagem de blocos que é utilizada na maior parte do estudo deve-se aliar à alguma linguagem de programação convencional. Visando que programas complexos normalmente utilizam de linguagens de programação escrita ao invés de blocos, desse modo os estudantes estarão se preparando para trabalhos futuros.

Graças ao presente trabalho tornou-se possível evidenciar que a metodologia empregada no minicurso aliada às ferramentas utilizadas obteve êxito em desenvolver nos alunos algumas *hard skills* que foram citadas anteriormente como efeitos esperados.

Dentre as ferramentas mencionadas e referidas durante esse estudo, as que melhor foram avaliadas são o MIT App Inventor e o CODE.org, visando a finalidade do estudo. O ambiente interativo e de fácil compreensão propiciado por ambas as ferramentas, principalmente pelo CODE.org tornou a programação mais atraente e menos complexa para os alunos. Contribuindo no processo de ensino-aprendizagem de conceitos iniciais e da lógica de programação, provando a eficácia das mesmas.

Considerando tamanha vantagem em utilizar tal metodologia aliada a essas ferramentas, torna-se possível amenizar problemas relacionados à evasão de cursos que envolvam programação. Onde as dificuldades de abstração de conceitos e da lógica de programação possam ser superados.

Também é possível afirmar que alunos que desenvolvem essa base durante sua educação básica terão mais facilidade para entender conceitos programacionais complexos caso continuem na área.

Além de que o trabalho vigente também poderá servir de base para o curso de graduação em bacharelado em ciências da computação no IFCE, campus Aracati ou para qualquer instituição que queira utilizá-lo. Devendo ser utilizado como recurso auxiliar, com a finalidade de facilitar a abstração de conceitos programação e lógica para os alunos que estiverem ingressando na área de tecnologia da informação e comunicação.

Através do estudo foi possível demonstrar que embora existam diversas limitações que os estudantes possuam em relação à escrita, leitura e fundamentação lógico-matemática, o ensino-

aprendizagem de programação e seus conceitos é possível. Dessa forma almejando a diminuição das taxas de evasão do curso decorrente a essas dificuldades.

Demonstrou-se que na maioria dos casos, os jogos educativos e plataformas de ensino apresentam resultados significativos e tornam possível o aumento de desempenho a todos aqueles que tem interesse em aprender sobre programação.

REFERÊNCIAS

- ALVES, R. M. et al. Uso do hardware livre arduino em ambientes de ensino-aprendizagem. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, p. 162–187, 2013.
- AMARAL, L. et al. Plataforma robocode como ferramenta lúdica de ensino de programação de computadores-pesquisa e extensão universitária em escolas públicas de minas gerais. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2015. v. 26, n. 1, p. 200.
- AMARAL, S. d.; BARROS, D. M. V. Estilos de aprendizagem no contexto educativo de uso das tecnologias digitais interativas. **Consultado**, v. 1, n. 12, p. 1–32, 2007.
- ANDRADE, L. G. d. S. B. et al. Geração ze as metodologias ativas de aprendizagem: desafios na educação profissional e tecnológica. **Revista Brasileira da Educação Profissional e Tecnológica**, v. 1, n. 18, p. 8575, 2020.
- ARIMOTO, M.; OLIVEIRA, W. Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação. In: SBC. **Anais do XXVII Workshop sobre Educação em Computação**. [S.l.], 2019. p. 244–254.
- BARANAUSKAS, M. C. C. et al. Uma taxonomia para ambientes de aprendizado baseados no computador. **O computador na sociedade do conhecimento**, UNICAMP/NIED Campinas, v. 45, 1999.
- BARBOSA, E. F.; MOURA, D. G. de. Metodologias ativas de aprendizagem na educação profissional e tecnológica. **Boletim Técnico do Senac**, v. 39, n. 2, p. 48–67, 2013.
- BERSSANETTE, J. H.; FRANCISCO, A. C. de. Percepção de docentes que lecionam programação de computadores quanto à formação pedagógica. **Revista Brasileira de Informática na Educação**, v. 29, p. 133–159, 2021.
- BORGES, M. A. Uma nova abordagem para o ensino de banco de dados. In: **Anais do VI Workshop sobre Educação em Informática, XVIII CNSBC**. [S.l.: s.n.], 1998. p. 445–453.
- BORGES, M. A. Avaliação de uma metodologia alternativa para a aprendizagem de programação. In: **VIII Workshop de Educação em Computação-WEI**. [S.l.: s.n.], 2000. p. 15.
- BRASIL. Base nacional comum curricular. **Ministério da Educação**, Brasil, p. 471, 2018.
- CARVALHO, C. C. F. de. Visitas de estudo virtuais: contributos para uma outra aprendizagem da história na era da sociedade da informação. 2012.
- CHICON, P. M.; QUARESMA, C. R.; GARCÊS, S. Aplicação do método de ensino peer instruction para o ensino de lógica de programação com acadêmicos do curso de ciência da computação. **Anais do 5o SENID. Cruz Alta: UNICRUZ**, 2018.

DANTAS, R. F.; COSTA, F. E. A. d. Code: O ensino de linguagens de programação educativas como ferramentas de ensino/aprendizagem. **Simpósio Hipertexto e Tecnologias na Educação**, v. 5, 2013.

DELGADO, C. et al. Uma abordagem pedagógica para a iniciação ao estudo de algoritmos. In: **XII Workshop de Educação em Computação**. [S.l.: s.n.], 2004.

EDUCAÇÃO, C. e. T. d. C. Instituto Federal de. **Taxa de evasão, curso de Bacharelado em Ciência da Computação**. 2022. Disponível em: <<https://ifceemnumeros.ifce.edu.br/>>.

ESTEVES, M. B. **Robótica e Programação em Contexto Escolar: um estímulo à motivação e criatividade na aprendizagem**. Tese (Doutorado), 2019.

FERREIRA, D. A. A. Tecnologia: fator determinante no advento da sociedade de informação? **Perspectivas em ciência da informação**, v. 8, n. 1, 2003.

FIGUEIREDO, M. P. S. de; LIMA, E. G. de O.; FILHO, G. A. de A. C. Intervenção em turmas de jovens e adultos numa escola da rede pública do recife, multidisciplinaridade e uso da ferramenta blockly games: Um relato de experiência. In: SBC. **Anais do IV Congresso sobre Tecnologias na Educação**. [S.l.], 2019. p. 561–567.

GIRAFFA, M. M.; MORA, M. da C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: **Congressos CLABES**. [S.l.: s.n.], 2013.

GOMES, F. F. B.; SANTOS, T. S. dos. Estudo comparativo entre o xamarin e o cronapp no desenvolvimento de aplicativos nativos e híbridos. **Revista Computação Aplicada-UNG-Ser**, v. 8, n. 1, p. 13–21, 2019.

GOULART, J. L. de J. et al. Tupy online: Uma ferramenta para visualização de algoritmos. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2019. v. 8, n. 1, p. 337.

JÚNIOR, J. C. R. P.; RAPKIEWICZ, C. E. O processo de ensino-aprendizagem de fundamentos de programação: uma visão crítica da pesquisa no brasil. In: **Anais do XII Workshop sobre Educação em Computação (SBC)**. [S.l.: s.n.], 2004. v. 3, p. 4.

JUNIOR, S. M. da S.; FRANÇA, S. V. A. Programação para todos: Análise comparativa de ferramentas utilizadas no ensino de programação. In: SBC. **Anais do XXV Workshop sobre Educação em Computação**. [S.l.], 2017.

LIMA, T. C. S. d.; MIOTO, R. C. T. Procedimentos metodológicos na construção do conhecimento científico: a pesquisa bibliográfica. **Revista Katálysis**, SciELO Brasil, v. 10, p. 37–45, 2007.

MARTINS, S. W.; OLIVEIRA, A. C. M. de; KOMOSINSKI, L. J. Webpensa: Uma ferramenta para apoio ao desenvolvimento do pensamento crítico. **RENOTE**, v. 3, n. 1, 2005.

MEDEIROS, T. J.; SILVA, T. R. da; ARANHA, E. H. da S. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. **Renote**, v. 11, n. 3, 2013.

MONCLAR, R. S.; SILVA, M. A.; XEXÉO, G. Jogos com propósito para o ensino de programação. **Anais do XVII Simpósio Brasileiro de Jogos e Entretenimento Digital-SBGames**, p. 1132–1140, 2018.

- MORENO, R.; MAYER, R. Interactive multimodal learning environments. **Educational psychology review**, Springer, v. 19, n. 3, p. 309–326, 2007.
- NETO, V. d. S. M. A utilização da ferramenta scratch como auxílio na aprendizagem de lógica de programação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2013. v. 2, n. 1.
- OLIVEIRA, G. A. S. Uma abordagem semiótica para utilização de modelos tridimensionais na aprendizagem de programação orientada a objetos. Centro Universitário Campo Limpo Paulista, 2018.
- OLIVEIRA, W. K. F. d. Vila python: um jogo rpg para auxiliar a aprendizagem de programação com uso da linguagem python. Universidade Federal da Paraíba, 2015.
- PAIVA, M. R. F. et al. Metodologias ativas de ensino-aprendizagem: revisão integrativa. **SANARE-Revista de Políticas Públicas**, v. 15, n. 2, 2016.
- PASSOS, L. S. et al. Aplicação do jogo digital code combat no ensino de programação aos alunos do curso médio técnico em informática. **Revista de Ensino, Educação e Ciências Humanas**, v. 21, n. 1, p. 112–116, 2020.
- PASSOS, M.; CAMARÁ, W. U-learning: integração de técnicas de ensino-aprendizagem para o alcance da aprendizagem significativa. **SIED: EnPED-Simpósio Internacional de Educação a Distância e Encontro de Pesquisadores em Educação a Distância**, 2016.
- RAABE, A. L. A.; SILVA, J. d. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. In: SN. **XIII Workshop de Educação em Computação (WEI'2005)**. São Leopoldo, RS, Brasil. [S.l.], 2005. v. 3, p. 5.
- RAPKIEWICZ, C. E. et al. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. **RENOTE**, v. 4, n. 2, 2006.
- RODRIGUES, M. Como ensinar programação. **Informática–Boletim Informativo Ano I**, n. 01, 2002.
- RODRIGUES, P. H. B.; CORREIA, L. H. et al. Comparação entre ferramentas de ensino de programação orientada a blocos. mit app inventor vs scratch. Insitituto Federal de Educação, Ciência e Tecnologia de Goiás, 2017.
- SANTIAGO, A. D.; KRONBAUER, A. H. Um modelo lúdico para o ensino de conceitos de programação de computadores. **Revista Brasileira de Informática na Educação**, v. 25, n. 03, p. 1, 2017.
- SANTOS, R. P. dos; COSTA, H. A. X. Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática. **INFOCOMP Journal of Computer Science**, v. 5, n. 1, p. 41–50, 2006.
- SCAICO, P. D. et al. Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. **Revista Brasileira de Informática na Educação**, v. 21, n. 02, p. 92, 2013.
- SILVA, C. E. d. M. A importância dos jogos eletrônicos como elemento de apoio à produção, resgate e valorização cultural. In: **Anais do XII Congresso de Ciências da Comunicação na Região Sudeste**. [S.l.: s.n.], 2007.

SILVA, T. S. C. da; MELO, J. C. B. de; TEDESCO, P. C. d. A. R. Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando gamification. **Revista Brasileira de Informática na Educação**, v. 26, n. 03, p. 120, 2018.

VALE, M. B. d.; NEVES, S. F. et al. O uso da linguagem de programação (logo) como recurso para o desenvolvimento cognitivo na educação básica. UFRA/Campus Belém, 2018.

VALENTE, J. et al. O computador na sociedade do conhecimento, coleção informática para a mudança na educação. 1998.

WANGENHEIM, C. G. von; NUNES, V. R.; SANTOS, G. D. D. Ensino de computação com scratch no ensino fundamental—um estudo de caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 03, p. 115, 2014.

WING, J. M. Computational thinking. **Communications of the ACM**, ACM New York, NY, USA, v. 49, n. 3, p. 33–35, 2006.

APÊNDICE A - Questionário dos Professores/Monitores

1. Instrutor
2. Turma realizada.
3. Data em que foi realizada.
4. Ferramenta utilizada.
 - Code.org
 - MIT App Inventor
 - Python
 - Arduino/LEGO
 - Comentário/Contribuição
5. Quais conceitos de programação ou recursos do ambiente são aprendidos/utilizados pelos alunos durante a aula?
 - Introdução ao ensino de programação
 - Introdução ao ensino Maker
 - Algoritmo
 - Programação em Blocos
 - IDE
 - Estrutura de decisão
 - Laço de repetição
 - Linguagem de programação
 - Conceitos aritméticos em linguagens de programação
 - Do it yourself (Faça você mesmo)
 - Eletrônica
 - Notebook
 - Tablet
 - Arduino (LED's, Jumper's, Protoboard, etc.)
 - LEGO
 - Monitor ou Projetor
6. Em uma escala de 1 à 5 os objetivos de aprendizagem foram alcançados?
7. Efeitos foram observados nos alunos com as atividades?

- Desenvolvimento de autonomia.
- Desenvolvimento de pensamento crítico.
- Desenvolvimento cognitivo real.(O aluno resolve problemas de forma independente.)
- Desenvolvimento cognitivo potencial.(Aluno resolve problemas sob orientação.)
- Estimulo criativo.
- Instrução em pares.(Alunos ensinam alunos.)
- Aumento de foco.

8. Quais efeitos foram observados nos alunos com as atividades? (Dissertativa)
9. A ferramenta utilizada facilita a aprendizagem da prática computacional/programação?
10. A ferramenta utilizada engaja os alunos a aprenderem?
11. A ferramenta utilizada promove uma experiência agradável e divertida? Fale um pouco.
12. Qual a metodologia de ensino utilizada?
 - Aprendizagem baseada em problemas.
 - Pedagogia da problematização.
 - Relato crítico sobre sua experiência.
 - Exposições dialogadas.
 - Debates temáticos.
 - Leitura comentada.
 - Apresentação de filmes.
 - Dinâmicas lúdico-pedagógicas.