



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO AMAZONAS  
CAMPUS MANAUS-DISTRITO INDUSTRIAL  
CURSO DE BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

GABRIEL FELIPE BATISTA DOS SANTOS

CAPACITAÇÃO DE ALUNOS DE ESCOLAS PÚBLICAS PARA DESENVOLVIMENTO DE APLICATIVOS  
PARA IOS

MANAUS - AM  
2023

GABRIEL FELIPE BATISTA DOS SANTOS

CAPACITAÇÃO DE ALUNOS DE ESCOLAS PÚBLICAS PARA DESENVOLVIMENTO DE APLICATIVOS  
PARA IOS

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Campus Manaus - Distrito Industrial, Curso de Bacharelado em Engenharia de Controle e Automação, como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Vitor Bremgartner da Frota.

Manaus-AM  
2023

### Dados Internacionais de Catalogação na Publicação (CIP)

S231c Santos, Gabriel Felipe Batista dos.  
Capacitação de alunos de escolas públicas para desenvolvimento de aplicativos para IOS / Gabriel Felipe Batista dos Santos. — Manaus, 2023.  
47f.: il. (color.).

Monografia (Graduação) — Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial, Curso de Engenharia de Controle e Automação, 2023.

Orientador: Prof.º Vítor Bremgartner da Frota, Dr.

1. Curso de Programação. 2. Cultura Maker. 3. Aprendizagem Ativa. 4. Ensino Médio. I. Frota, Vítor Bremgartner da. II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 629.89

Elaborada por Oziane Romualdo de Souza (CRB11/ nº 734)



## TERMO DE APROVAÇÃO

CAPACITAÇÃO DE ALUNOS DE ESCOLAS PÚBLICAS PARA DESENVOLVIMENTO DE APLICATIVOS  
PARA IOS

por

GABRIEL FELIPE BATISTA DOS SANTOS

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 15 de março de 2023 como requisito parcial para a obtenção do título de Bacharel em Engenharia de Controle e Automação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Vitor Brengartner da Frota  
Prof.(a) Orientador(a)

Profa. Me. Priscila Silva Fernandes  
Membro titular

Prof. Dr. Alyson de Jesus dos Santos  
Membro titular

## RESUMO

Existem muitos cursos de programação presentes no mercado para o público jovem. Porém a maioria dos cursos são básicos, utilizando softwares que não são usados no mercado de trabalho e com conteúdo ínfimo que não prepara o estudante para trabalhar no mercado. O objetivo do presente trabalho é investigar de que forma a cultura *maker* aliada com a plataforma de desenvolvimento *Xcode* pode facilitar o processo de ensino-aprendizagem da matéria de programação e preparar o estudante para o mercado de trabalho. Como estudo de caso para aplicar a cultura *maker* no ensino de programação, criamos um curso de programação com metodologia ativa, onde em cada aula era proposto aos alunos o desafio de desenvolverem um aplicativo com os conhecimentos recém adquiridos. Participaram desse curso 40 alunos de primeiro ano do ensino médio de três escolas públicas. Em seguida, todos os alunos assistiram um total de 20 aulas, além de horas extras para desenvolverem os projetos propostos. No final, 35 alunos terminaram o curso e desenvolveram 19 aplicativos, onde 12 aplicativos estão na Apple Store, o que mostra a viabilidade da proposta.

Palavras-Chave: Curso de Programação. Cultura Maker. Aprendizagem Ativa. Ensino Médio.

## **ABSTRACT**

There are many programming courses on the market for young people. However, most courses are basic, using software that is not used in the job market and with minimal content does not prepare the student to work in the market. The objective of this work is to investigate how the maker culture combined with the Xcode development platform can facilitate the teaching-learning process of programming and prepare students for the job market. As a case study to apply the maker culture in programming teaching, we created a programming course with an active methodology, where in each class the students were proposed the challenge of developing an application with the newly acquired knowledge. Forty first-year high school students from three public schools participated in this course. Then, all students attended a total of 20 classes, in addition to overtime to develop the proposed projects. In the end, 35 students completed the course and developed 19 applications, 12 of which are on the Apple Store, which shows the viability of the proposal.

**Keywords:** Programming Course. Maker Culture. Active Learning. High School.

## LISTA DE FIGURAS

Figura 1. Local das Aulas.....	23
Figura 2. Aplicativo My Name.....	25
Figura 3. Aplicativo SUM.....	27
Figura 4. Aplicativo MaiorMenor.....	28
Figura 5. Aplicativo FatoreFibo.....	29
Figura 6. Aplicativos Desenvolvidos.....	34
Figura 7. Percentual de masculino e feminino.....	40
Figura 8. Percentual das séries dos alunos.....	41
Figura 9. Relação dos Alunos com Conhecimentos em Programação	42
Figura 10. Relação dos Alunos com Todos os Aplicativos no GitHub	44

## LISTAS DE TABELAS

Tabela 1. Espaço <i>Maker</i> x Laboratório de Informática.....	17
Tabela 2. Categorias de Desempenho.....	37
Tabela 3. Categorias de Potencial.....	38
Tabela 4. Relação aplicativos e conteúdos de programação utilizados	42

## LISTAS DE QUADROS

Quadro 1. Matriz Nine Box.....	37
Quadro 2. Categorias de Avaliação dos Aplicativos.....	39

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>15</b>
2.1	MOVIMENTO MAKER.....	15
2.2	XCODE.....	18
2.3	ENSINO MÉDIO NO BRASIL.....	19
2.3	TRABALHOS CORRELATOS.....	20
<b>3</b>	<b>MATERIAIS E MÉTODOS.....</b>	<b>22</b>
3.1	FERRAMENTAS.....	23
3.2	PROCEDIMENTOS METODOLÓGICOS.....	24
<b>3.2.1</b>	<b>Aulas.....</b>	<b>25</b>
<b>3.2.2</b>	<b>Avaliação.....</b>	<b>36</b>
<b>4</b>	<b>APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS.....</b>	<b>40</b>
4.1	CARACTERIZAÇÃO DOS PARTICIPANTES.....	40
4.2	ANÁLISE DO APRENDIZADO DOS ALUNOS.....	42
4.3	RESPOSTAS DOS ALUNOS.....	44
4.4	RESPOSTAS DOS AVALIADORES.....	44
<b>5</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>46</b>
	<b>REFERÊNCIAS.....</b>	<b>48</b>

## 1 INTRODUÇÃO

De acordo com a Base Nacional Comum Curricular (BNCC) homologada em 2018, a aprendizagem de qualidade é um objetivo que o Brasil deve buscar continuamente, em especial o Ensino Médio, no qual os índices de aprendizagem, repetência e abandono são preocupantes (MEC, 2018).

A sociedade reflete sobre questões do processo educativo: por que aprender, o que aprender, como ensinar, como avaliar o aluno e como criar redes de aprendizado colaborativo. No cenário global, ser comunicativo, criativo, participativo, resiliente, ter senso analítico e crítico requer muito além do que o acúmulo de conhecimento. Requer o desenvolvimento de competências para aprender a aprender, saber lidar com a informação cada vez mais disponível, atuar com discernimento e responsabilidade nos contextos das culturas digitais, aplicar conhecimentos para resolver problemas, ter autonomia para tomar decisões, ser proativo para identificar os dados de uma situação e buscar soluções, conviver e aprender com as diferenças e as diversidades (MEC, 2018).

Nesse contexto, deve-se assumir uma visão global do estudante e promover uma educação voltada para o seu aprendizado e desenvolvimento de suas singularidades e diversidades. Portanto, deve-se promover conhecimentos sintonizados com as necessidades e os interesses dos estudantes, além de abranger os desafios da sociedade contemporânea.

Implementar a Cultura *Maker* nos Ensinos Fundamental e Médio e na educação em geral torna-se importante, pois o atraso na educação formal no Brasil está diretamente ligado a deficiência do ensino no país. Portanto, o resultado do ensino de baixa qualidade pode gerar alunos com dificuldades para entrar em uma universidade ou para se manter na mesma. Na escola, o aluno deveria se tornar apto para iniciar uma vida universitária, contudo sabe-se que isso não ocorre por vários fatores, entre eles, um Ensino Médio ineficiente. (LEITE; FONSECA; LIMA, 2016).

Sendo assim, ao se trabalhar com a Cultura *Maker* para o desenvolvimento de algum projeto interdisciplinar, os alunos serão introduzidos em um contexto de formação científica que permitirá que ele desenvolva e aplique os conceitos adquiridos nas aulas para a produção tecnológica (MEIRA; RIBEIRO, 2016).

A utilização da Cultura *Maker*, portanto, para ensino da Linguagem de Programação Swift no *Xcode* é um projeto de grande importância, pois sabe-se que o ensino de programação pode ser considerado difícil por conta da defasagem de conteúdo técnico e desmotivação por parte dos alunos. Além disso, a interpretação de problemas do cotidiano para o computador não acontece com facilidade. Portanto, alcançar a atenção e o interesse dos estudantes para aprender o conhecimento teórico e prático na área de programação ainda é um grande desafio no processo de ensino, pois na grande maioria dos cursos o conhecimento só é repassado em forma de slides e listas para fazer, tornando assim o aprendizado árduo.

Apesar desses fatores, o ensino de programação tem sido apresentado de forma engessada o que compromete a individualização de aprendizagem de cada aluno. Portanto, a matéria de programação é visto como pouco atrativo e desmotivador, onde tal fator faz com que os alunos tenham uma deficiência na sua formação científica-cultural, isto porque não se utiliza de elementos instigantes no processo de aprendizagem (MOURÃO, 2018).

Segundo Mourão (2018), há ensino teórico principalmente nas escolas públicas de ensino fundamental e médio, mas por falta de recursos, na maioria das vezes não há aula prática. Então na aula de programação, que muitas vezes não existe e quando existe os alunos perdem o interesse, principalmente aqueles que não têm muita habilidade com números e cálculos.

Nesse contexto, as perguntas norteadoras desse trabalho podem ser apresentadas da seguinte forma: De que modo deverá ser formatado o curso de programação para ter baixo índice de desistência e alto rendimento de aprendizagem? Quais conteúdos de programação deverá ser abordado para está apto a trabalhar no mercado de trabalho? Que interface de desenvolvimento deverá ser utilizada para ensinar programação? E como deverá ser a dinâmica em sala de aula para os alunos aprenderem a trabalhar em equipe?

No sentido de responder esses questionamentos, hipotetiza-se que o curso deverá ser baseado em aprendizagem com desafios com foco na individualização de cada aluno. Por um lado, os conteúdos de programação deverá contemplar desde o básico, variáveis e constantes, até a publicação do aplicativo em uma loja virtual. Por outro lado, a interface de desenvolvimento deverá ser o *Xcode*, por ser desenvolvido por uma empresa que tem o serne na

educação de programação de computadores e o mercado de desenvolvedores Swift está com uma alta demanda. E por fim, a dinâmica na sala de aula ocorrerá em duplas que mudarão todas as aulas para trabalharem em equipes e com diferentes pessoas.

Assim, o presente trabalho tem como objetivo geral investigar de que forma a cultura *maker* aliada com a plataforma de desenvolvimento *Xcode* pode facilitar o processo de ensino-aprendizagem da matéria de programação.

Deste objetivo geral decorrem cinco outros objetivos específicos, quais sejam: a) Apropriar-se das técnicas existentes de ensino-aprendizagem da cultura *maker* com foco em programação; b) Elencar técnicas de ensino e aprendizado para apoiar o desenvolvimento dos alunos; c) Desenvolver um curso de programação para adolescentes do ensino médio utilizando a aprendizagem baseada em desafios; d) Aplicar o curso nos estudantes; e) Validar qualitativamente o artefato proposto;

A educação como era tratada nos séculos passados se difere de hoje, o fluxo de informações aos quais os estudantes são expostos é maior do que nas últimas décadas, a internet proporcionou a busca por conteúdos sobre os mais diversos assuntos (COUTINHO; LISBÔA, 2011).

São diversas as ferramentas de ensino encontradas no mercado, entre elas o *Xcode*, um conjunto de ferramentas para desenvolver aplicativos para *Mac*, *Iphone*, *Apple Watch* e *Apple TV* com um fluxo de trabalho unificado.

Apesar de existirem cursos de programação de qualidade, é raro encontrar um curso destinado ao ensino introdutório de computação até a hospedagem do aplicativo na *apple store* para adolescentes de escolas públicas do ensino médio. Visto isso constatou-se a necessidade de desenvolver um curso de programação em *swift*, que de forma lúdica ensinasse conceitos introdutórios, intermediários e avançados de computação para adolescentes, buscando assim o desenvolvimento do raciocínio lógico, tomada de decisões e desenvolvimento de habilidades interpessoais.

Dentro do cenário descrito anteriormente, o aprendizado baseado em desafios surge como uma opção de abordagem democrática, que por meio do uso de métodos colaborativos e gerenciamento individual dos alunos permitem o desenvolvimento de um modelo que busca refletir a dificuldade de cada aluno e união da classe como um todo. Assim é possível fugir do ensino engessado e

focar na fixação do conteúdo através da prática e colaboração entre os discentes.

## 2 REFERENCIAL TEÓRICO

O referencial teórico busca expor os principais conceitos e tecnologias que serão utilizados nesse trabalho, começando pela cultura maker, expondo as principais tecnologias de programação utilizadas no trabalho e por fim um resumo do aspecto educacional do Brasil.

### 2.1 MOVIMENTO MAKER

Há alguns anos, tirar do papel uma ideia, desenvolver, registrar, produzir e introduzir um novo produto no mercado levava muito tempo e, em muitos casos, este processo tornava-se inviável. Hoje, o avanço da tecnologia e da comunicação resultou na aproximação de pessoas com os mesmos interesses, que mesmo geograficamente distantes, podem trocar experiências e desenvolver projetos em conjunto (MENEZES, 2019).

O Movimento Maker é fundamentado da filosofia do “Do it Yourself” (DiY) e do “Do it with Others” (DiwO) e tem sua base a ideia de que pessoas comuns podem construir, consertar, modificar e fabricar os mais diversos tipos de objetos e projetos com suas próprias mãos (CORDOVA; 2016).

De acordo com Anderson (2012), os makers têm em comum o uso de ferramentas digitais para criar produtos e o compartilhamento de informações e colaboração em comunidades online. O trabalho em conjunto e a troca de experiências fazem do Movimento MAKER uma cultura de código aberto, onde qualquer pessoa pode pegar uma ideia na internet, recriá-la ou modificá-la e colocá-la novamente na rede para que outras pessoas possam fazer novamente o uso da mesma.

Para o mesmo autor, o movimento MAKER é fundamentado na cultura MAKER que tem como pilares de sustentação o incentivo à criatividade, à sustentabilidade ao consumir menos e utilizar os recursos que já estão disponíveis, à colaboração com o trabalho em equipe e o reaproveitamento de ideias e, por fim, à escalabilidade que ajuda a produzir em escala com um custo reduzido.

Medeiros (2016) afirmam que a sociedade atual necessita de uma escola colaborativa, que inspire a criatividade e o empreendedorismo de seus alunos,

elementos estes capazes de transformação. Com base neste propósito, a cultura MAKER tem ação direta do aluno na construção de soluções criativas, visando à resolução de problemas através da manipulação de objetos reais e apresentando-se como uma prática inovadora, mostrando uma nova forma de ver os bens de consumo.

Esse movimento pode ser aplicado em várias áreas da sociedade, como por exemplo na educação. Atualmente, na maioria das escolas, o conhecimento vem sendo aplicado de forma estruturada pelo educador, impedindo que o aluno desenvolva suas capacidades por si só. Na abordagem maker, o estudante tem o poder de através de desafios, resolução de problemas e experimentação, desenvolver soluções e aprimorar conhecimentos em diversas áreas. Engajar os alunos nas atividades e metodologias escolares tem sido um desafio, e o conceito da cultura maker traz inovação ao meio acadêmico e à aprendizagem ativa, onde o aluno, de forma prática e colaborativa, envolve-se no processo de ensino ao invés de apenas absorver de forma passiva o conteúdo que lhe é passado (MAGENNIS, 2005).

Por meio da teoria do construtivismo de Piaget, Papert propôs a teoria denominada de construcionismo para definir o conceito de conhecimento. Ele explica que por meio da ação concreta e da criação de produto palpável (criado pelo aluno), o estudante adquire o conhecimento, exerce seu protagonismo e adquire o saber gerado a partir da prática. Para Azevêdo (2019, p. 67), “quando o aluno é o protagonista no processo de ensino e aprendizagem torna-se possível proporcionar a eles uma aprendizagem mais significativa”. Dentro dessa ideia, o professor tem um papel de mediador e o aluno constrói o conhecimento que anteriormente era empregado na teoria.

A ideia de Papert é colocar o professor como desafiador do estudante e inseri-lo em situações que mobilizem o pensamento científico do aprendiz, para que, ele possa desenvolver estratégias e solução do problema proposto, ou seja, para o construcionismo o aluno é um descobridor de resultados, no qual, o conhecimento é construído e não, que o conhecimento seja repassado.

A cultura maker traz uma inovação para o modelo tradicional de ensino ao criar espaços de aprendizagem pedagógicos que dar autonomia para o aluno aprender, construindo ou transformado. Esses espaços trazem para a educação várias possibilidades no qual desprende o aluno do modelo habitual e o insere

no ambiente de construção (a tabela 1 mostra a diferença entres esses ambientes de ensino), possibilitando uma aprendizagem ampla. Ao analisarmos as propriedades da cultura *maker* é possível construir espaços *maker* gastando pouco, pois, a cultura *maker* prega a sustentabilidade, incentiva o baixo consumo, evita o desperdício e valoriza o uso de recursos que estão disponíveis (SANTOS, 2021).

Tabela 1 – Espaço *Maker* x Laboratório de Informática

<b>Aspectos</b>	<b>Espaço <i>maker</i></b>	<b>Laboratório de Informática</b>
Postura do aluno	Protagonista de sua aprendizagem tomando decisões e conduzindo a escolha dos projetos	Executor das atividades, usuários das aplicações que forem sugeridas pelo professor
Postura do professor	Facilitador das trajetórias dos alunos, mediador e coautor	Supervisão, mediação e acompanhamento das atividades
Papel da tecnologia	Prover condições para construção de objetos e artefato	Prover informações e atividades
Trabalho em grupo	Favorecido pelo espaço com mesas redondas e projetos em grupo	Restrito a duplas ou trios. Espaço para discussão é limitado.
Potencial criativo	Praticamente irrestrito	Restrito ao que os softwares usados permitirem criar
Aprendizagens	Diversificadas e não necessariamente ligadas ao currículo. Decorrentes de criar e socializar considerando restrições de recursos.	Geralmente curriculares, ligadas a algum conteúdo de definido pelo professor
Significado do Erro	O erro é parte do processo. É um problema a ser solucionado na próxima versão	É evitado. Corresponde a uma falha do aluno. Algo a ser eliminado.

Fonte: Raabe, Gomes (2018).

De acordo com Rossi, Santos e Oliveira (2019, p. 5), “o professor pode proporcionar um ambiente de aprendizagem que estimule o aluno a criar, comparar, discutir, rever, perguntar e ampliar ideias de forma, simples e de baixo custo”. E a partir desse pensamento pode-se considerar os espaços maker propício para o professor estigar o aluno a adquirir o conhecimento.

Para os mesmos autores, essa filosofia defende uma educação, no qual, o aluno não é visto como um receptor de informações, o estudante, deve ser inserido no ambiente educacional como atuante no processo da construção do seu conhecimento. A base dessa pedagogia é a interação (ação e reação) do aluno com o meio, com o objeto e com outras pessoas.

## 2.2 XCODE

Quando o iPhone foi lançado não havia nenhum ambiente nativo de desenvolvimento aberto ao público. A Apple afirmou que não era preciso um e que as aplicações criadas para ele deveriam ser baseadas na web e construídas usando CSS, JavaScript e HTML. Isso não foi o suficiente para acalmar os desenvolvedores que queriam acesso direto ao hardware em si. Apenas alguns meses depois deste anúncio, a comunidade open source conseguiu um feito que muitos diziam ser impossível. Os desenvolvedores ganharam acesso irrestrito ao aparelho usando técnicas de engenharia reversa e quebraram o código de segurança, o chamado jailbreak. Logo em seguida estavam criando uma ferramenta de desenvolvimento de código aberto para desenvolver os seus próprios aplicativos diretamente no iPhone. Detectando isso como uma ameaça em meados de 2008 a Apple lançou o seu próprio ambiente de desenvolvimento para iPhone integrado ao já existente XCode (TESSARI, 2010).

O Xcode possui diversas ferramentas muito úteis, entre as mais importantes podemos citar o compilador da Apple LLVM, que além de compilar o código, fornece uma ferramenta inteligente de para completar o código; o *Interface Builder (IB)*, no qual trabalhamos na parte visual do aplicativo, construindo nossas telas; uma ferramenta para debug; o *Organizer*, que auxilia no gerenciamento de projetos e dispositivos cadastrados para testes, além de apresentar a documentação auxiliar; o *Instruments*; uma ferramenta para análise

de desempenho do código; e por fim o iOS *Simulator*, um simulador para iPhone e iPad no qual podemos verificar nossos aplicativos (MACHADO, 2018).

Para o mesmo autor, cada aplicativo desenvolvido pode e deve ser testado não só no simulador, mas também em dispositivos reais. Além do mais, apesar do simulador possuir funcionalidades importantes como toque, gestos e GPS, algumas funcionalidades como as do acelerômetro ou bússola, não foram disponibilizadas, desta forma faz-se extremamente necessário os testes em dispositivos reais.

### 2.3 ENSINO MÉDIO NO BRASIL

De acordo com a Base Nacional Comum Curricular (BNCC, 2018), o ensino médio é a etapa final da Educação Básica, direito público subjetivo de todo cidadão brasileiro. Todavia, a realidade educacional do País tem mostrado que essa etapa representa um gargalo na garantia do direito à educação. Para além da necessidade de universalizar o atendimento, tem-se mostrado crucial garantir a permanência e as aprendizagens dos estudantes, respondendo às suas demandas e aspirações presentes e futuras. Para responder a essa necessidade de recriação da escola, mostra-se imprescindível reconhecer que as rápidas transformações na dinâmica social contemporânea nacional e internacional, em grande parte decorrentes do desenvolvimento tecnológico, atingem diretamente as populações jovens e, portanto, suas demandas de formação. Nesse cenário cada vez mais complexo, dinâmico e fluido, as incertezas relativas às mudanças no mundo do trabalho e nas relações sociais como um todo representam um grande desafio para a formulação de políticas e propostas de organização curriculares para a Educação Básica, em geral, e para o Ensino Médio, em particular.

Para o mesmo documento, para formar esses jovens como sujeitos críticos, criativos, autônomos e responsáveis, cabe às escolas de Ensino Médio proporcionar experiências e processos que lhes garantam as aprendizagens necessárias para a leitura da realidade, o enfrentamento dos novos desafios da contemporaneidade (sociais, econômicos e ambientais) e a tomada de decisões éticas e fundamentadas. O mundo deve lhes ser apresentado como campo aberto para investigação e intervenção quanto a seus aspectos políticos, sociais,

produtivos, ambientais e culturais, de modo que se sintam estimulados a equacionar e resolver questões legadas pelas gerações anteriores – e que se refletem nos contextos atuais – abrindo-se criativamente para o novo.

Garantir a consolidação e o aprofundamento dos conhecimentos adquiridos no Ensino Fundamental é essencial nessa etapa final da Educação Básica. Além de possibilitar o prosseguimento dos estudos a todos aqueles que assim o desejarem, o Ensino Médio deve atender às necessidades de formação geral indispensáveis ao exercício da cidadania e construir “aprendizagens sintonizadas com as necessidades, as possibilidades e os interesses dos estudantes e, também, com os desafios da sociedade contemporânea” (BNCC, 2018).

## 2.4 TRABALHOS CORRELATOS

Para a concepção metodológica da oficina, foi necessário primeiramente analisar alguns trabalhos correlatos que serviram de base para a execução do trabalho junto aos alunos.

Foi analisada a proposta do trabalho de Duarte, Silveira e Borges (2017), cujo objetivo principal era estabelecer a importância de utilizar a lógica da programação como soluções de problemas cotidianos, presentes na vida de todo estudante. O diferencial deste trabalho, é que o mesmo foi realizado em duas escolas. A primeira era uma escola pública situada na cidade Limeira-SP e contou com a participação de 11 alunos do sétimo ano. Foram necessárias duas semanas, sendo três aulas por semana, com duração de uma hora cada. Na segunda (uma escola particular, situada na cidade de Campinas-SP), teve como participantes dezesseis alunos do sexto, sétimo e oitavo ano. Foram necessárias seis semanas, tendo uma aula por semana de uma hora. A metodologia de realização da oficina foi igual para ambas escolas. Foram elencados os seguintes temas necessários para capacitação dos alunos no entendimento da lógica de programação: (i) introdução, em que houve a explicação do termo “Algoritmo” e “Programa”; (ii) algoritmo no dia-a-dia, em que foi explanada a aproximação de conceitos abstratos para metáforas do mundo real; (iii) sequências de passos, que pretendeu demonstrar as formas de representar a sequência de passo descritas nos algoritmos; (iv) laços de repetições, cujo

objetivo era explicar formas de otimizar a descrição de trabalhos repetitivos; (v) depuração, em que era demonstrado que um algoritmo pode estar errado ou incompleto, fazendo com que fosse necessário analisá-lo para encontrar o erro; e; por fim (vi) revisão e condição, em que foi explorado o conceito de condição. Para obtenção dos resultados, foram desenvolvidos três questionários, sendo, o primeiro realizado no início da oficina para identificar o quão envolto na tecnologia e nos assuntos que seriam abordados os alunos estavam. O segundo foi realizado no último dia de aula visando analisar a perspectiva dos alunos sobre o conteúdo aprendido, a metodologia utilizada, interesse dos alunos para o aprendizado, entre outras. Já o terceiro questionário, também foi aplicado no último dia de aula, e teve como finalidade identificar quais foram as mudanças que ocorreram com os alunos no decorrer do curso. Como resultados, os autores, afirmam que houve uma melhora em diversos quesitos analisados durante o decorrer da oficina. Além disso, os alunos se mostraram interessados com a dinâmica proposta por essa abordagem. Por fim, os autores afirmam que é possível o aprendizado por meio da metodologia utilizada na oficina, e que a mesma pode ser estendida para mais níveis de ensino, “podendo torná-la, inclusive como uma matéria optativa nas escolas” (DUARTE; SILVEIRA; BORGES, 2017).

Outro estudo analisado foi o de Berssanette (2020), que consiste no estudo de metodologias ativas de aprendizagem e teoria da carga cognitiva para construção de caminhos no ensino de programação de computadores. O estudo propôs um curso de qualificação para docentes com foco em metodologias ativas e um caderno de planos de aula para o ensino de programação.

Por último, o estudo analisado foi o de Silva, Souza e Silva (2016), que utilizou a ferramenta Scratch para o ensino de programação de 26 alunos do quinto ano do Ensino Fundamental I de uma escola na capital da Região Norte do país. Foram realizadas cinco aulas expositivas e práticas que foram executadas por meio da elaboração e desenvolvimento de um quiz sobre meio ambiente. Como principais resultados, foi notório o entusiasmo por parte dos participantes, havendo uma grande aceitação em aprender programação por meio da ferramenta. Ao questionar os alunos sobre dar continuidade ao aprendizado, a resposta foi unânime, todos pretendiam continuar com o aprendizado de programação.

### 3 MATERIAIS E MÉTODOS

O trabalho foi desenvolvido por meio de pesquisa quanti-qualitativa que, segundo Souza e Kerbaui (2017), é um tipo de pesquisa que incorpora elementos de ambas abordagens qualitativa e quantitativa.

Segundo Silva (2010, p. 6), a abordagem qualitativa trabalha com valores, crenças, representações, hábitos, atitudes e opiniões. Ela aprofunda a complexidade de fenômenos, fatos e processos; passa pelo observável e vai além dele ao estabelecer inferências e atribuir significados ao comportamento.

Já a pesquisa quantitativa, conforme Fonseca (2002, p. 20), se centra na objetividade. Influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros. A pesquisa quantitativa recorre à linguagem matemática para descrever as causas de um fenômeno, as relações entre variáveis, etc. Os resultados da pesquisa quantitativa podem ser quantificados.

Deste modo o tipo de pesquisa que será utilizado será uma pesquisa tanto quantitativa como qualitativa, para desenvolvimento do projeto, pois o mesmo se enquadra nas definições desses dois tipos de pesquisa.

Outra forma de classificar a pesquisa é utilizando a metodologia definida pelos autores Gil (2002), Silva e Menezes (2005) e Tasca et al. (2010), que se constitui da identificação dos diferentes tipos de pesquisa, sua natureza, seus objetivos e seus procedimentos. Baseado nesta metodologia foi realizado a identificação para determinar a melhor opção para se seguir de forma adequada ao objeto de pesquisa, sendo este a investigação de que forma a cultura *maker* aliada com a plataforma de desenvolvimento *Xcode* pode facilitar o processo de ensino-aprendizagem da matéria de programação. Apresentando os tipos de pesquisa que mais se enquadram com o projeto, sua natureza e os procedimentos aplicados.

Portanto, a natureza da pesquisa do projeto será aplicada, pois para o seu desenvolvimento será necessária uma aplicação prática, para verificar sua eficiência e na resolução de um problema específico, sendo este a dificuldade dos alunos em aprender programação.

Quanto aos objetivos da pesquisa será descritiva, que tem como a finalidade de descrever os fatos e fenômenos de uma realidade determinada. Entretanto será realizado em conjunto a pesquisa explicativa, pois busca identificar os fatores para determinar a ocorrência dos fenômenos, utilizando de resultado, para mostrar as vantagens que serão obtidas ao se utilizar essa forma alternativa de ensinar programação.

Quanto aos procedimentos trata-se de uma pesquisa experimental e bibliográfica. Experimental, porque será implementado um modelo alternativo de ensinar programação, dessa forma a pesquisa será submetida a revisão para associar a outros cursos de programação presentes no mercado que mais se adequam e assim assegurar a eficácia no que se propõe. Com relação à pesquisa bibliográfica terá papel importante em mostrar como os cursos de programação atuais se comportam e como eles podem ser melhorados.

A pesquisa bibliográfica terá como base trabalhos científicos de estudo relacionado a ensino de programação para jovens e adolescentes, a grade curricular dos cursos e trabalhos relacionados ao projeto que será desenvolvido.

### 3.1 FERRAMENTAS

Para apoiar os processos de ensino e de aprendizagem da programação na educação do ensino médio com o intuito de ir do zero até a publicação do aplicativo na Apple Store é necessário um ambiente confortável e ferramentas que sejam fáceis de ensinar para envolver o interesse dos alunos. Neste sentido, foram utilizadas dez Macs e o ambiente de desenvolvimento Xcode.

Figura 1 – Local das Aulas



Fonte: Próprio Autor (2022).

Os *Macs* são essenciais por manterem os alunos engajados, pois muitos não tem contato com essa tecnologia e profissionais de desenvolvimento IOS utilizam os *Macs* no dia a dia.

O *Xcode* é a principal ferramenta de desenvolvimento para IOS, utilizando-se de interface gráfica e códigos simples, os alunos estudam e verificam como pode ser aplicado os conhecimentos adquiridos.

### 3.2 PROCEDIMENTOS METODOLÓGICOS

O presente estudo foi realizado com 40 alunos do ensino médio de 3 escolas públicas subdividas em 3 turmas – duas a noite e uma de manhã.

A metodologia de ensino utilizado nesse trabalho será dividida em dois aspectos – comportamentais e técnicos – pois auxiliará de forma eficaz o aprendizado dos alunos e a identificação de possíveis dificuldades por cada um. Podendo essa metodologia de ensino ser flexível, voltada para o aluno e fácil de ser gerenciada.

Pelo aspecto comportamental, todas as aulas eram realizadas sorteios para descobrir em qual computador os alunos iriam ficar e também quem seria a sua dupla em cada aula. Assim os alunos têm a oportunidade de se tornarem flexíveis por trabalharem em locais diferentes e a aprender a trabalhar com diferentes tipos de pessoas.

Já pelo aspecto técnico, todas as aulas devem ter dez minutos de teoria e uma hora e vinte minutos de prática, totalizando uma hora e meia semanais. A metodologia de ensino é a aprendizagem baseada em projetos, onde a cada aula era realizado um aplicativo juntamente com o professor e no final de cada lição é passado um desafio para o aluno realizar até a próxima aula. Com a metodologia ativa espera-se que o aluno consiga ao final do curso publicar um aplicativo na Apple Store.

### 3.2.1 Aulas

Nesta seção será descrita o conteúdo de cada aula, assim como os desafios propostos para cada aluno realizar no decorrer da semana.

#### 3.2.1.1 Variáveis e Constantes

A maioria dos cursos de programação começa com a definição de programação. Porém, os conceitos iniciais eram dados em uma aula de 10 minutos, onde explica-se:

- o que é programação
- por que programar
- variáveis e constantes

No primeiro dia de aula, os alunos fazem o seu primeiro aplicativo para treinar os conceitos de variáveis e constantes.

Figura 2 – Aplicativo My Name



Fonte: Próprio Autor (2022).

O aplicativo é o “My Name”, onde o título é uma constante, a pergunta logo a seguir é uma variável e quando apertar o botão “Answer”, no lugar da pergunta aparece o nome da pessoa que desenvolveu o aplicativo.

O desafio semanal é realizar uma programação para a variável mostrar a pergunta novamente.

### 3.2.1.2 Git e GitHub

Primeiramente, o Git é um software de controle de versões que tem como objetivo gerenciar arquivos texto, principalmente códigos de programação. O funcionamento dessa ferramenta é semelhante aos sistemas de nuvens (Google Drive, Dropbox, Mega, etc), em que podemos compartilhar as edições de um script com uma equipe, além de retomar antigas versões de nossos arquivos. Resumidamente, o Git examina as mudanças realizadas nos arquivos de um diretório e vai guardando essas informações para compartilhar com outros usuários e/ou possibilitar a reversão de alterações realizadas. Além de ser uma ferramenta extremamente útil, o Git se conecta ao serviço de hospedagem e armazenamento de versões chamado GitHub. A partir do GitHub é possível contribuir com projetos de qualquer lugar do mundo, além de compartilhar os nossos próprios códigos com fins de divulgação e/ou colaboração (FAVALESSO, 2020).

Assim, nesta aula acontece a primeira ruptura com outros cursos de programação, pois a maioria dos cursos de programação não ensinam esse assunto para os seus alunos e os que ensinam utilizam algum software para facilitar o uso da ferramenta.

Nesta aula, os alunos aprendem a usar o terminal para acessar diretórios e arquivos, a criar uma conta no GitHub e a hospedar o aplicativo que desenvolveram na primeira aula.

O intuito do assunto Git e GitHub ser a segunda aula do curso é para os alunos utilizarem esses comandos no decorrer de todo o curso. Assim, os alunos aprenderam os comandos básicos de terminal e versionamento do código, além de construir um portfólio para utilizarem no seu currículo ao terminarem o curso.

O desafio dessa aula é programar um aplicativo utilizando conceitos de constantes e variáveis e alocar no GitHub.

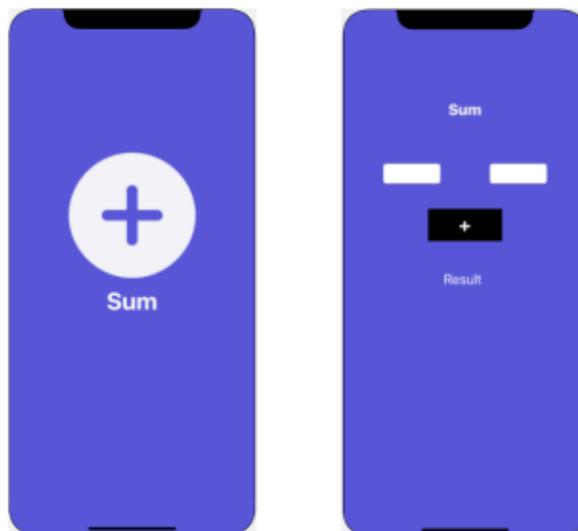
### 3.2.1.3 Opcionais e Operadores

Uma variável opcional pode, ou não, conter um valor. É uma funcionalidade para garantir maior segurança e produtividade. Esse assunto é

focado em desenvolvedores Swift e apresenta para os alunos os conceitos de condicionais.

Já com os assuntos de operadores, é apresentado somente os operadores matemáticos e os alunos durante a aula desenvolver o aplicativo SUM.

Figura 3 – Aplicativo SUM



Fonte: Próprio Autor (2022).

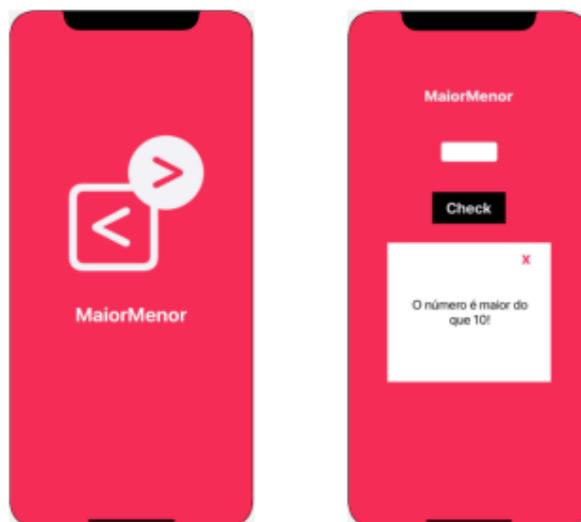
Onde com um clique de um botão e a inserção de dois valores é apresentado a soma daqueles valores. Ao final da aula é proposto para os alunos a confecção de um aplicativo de calculadora.

#### 3.2.1.4 Condicionais

Nesta aula, os alunos aprendem condicionais e programam o aplicativo “MaiorMenor”.

Onde ao inserir um número, o aplicativo relata se aquele número é maior ou menor do que 10.

Figura 4 – Aplicativo MaiorMenor



Fonte: Próprio Autor (2022).

O desafio dessa aula é programarem um aplicativo para calcular o IMC (Índice de massa corporal) e relatar se está acima do peso.

#### 3.2.1.5 Applcon e AutoLayout

Nesta aula aprende-se a confeccionar o ícone do aplicativo e também a redimensionar os tamanhos dos elementos e o posicionamento deles na tela do celular, portanto, o aplicativo desenvolvido poderá ser usado em qualquer dispositivo, sendo iPhone ou iPad.

O desafio dessa aula é selecionar um dos aplicativos alocados no GitHub e adicionar um ícone, como também realizar o AutoLayout.

#### 3.2.1.6 Segues

Segues são mecanismos de navegação entre telas. O usuário ao fazer uma ação pode sair de uma tela e adentrar em outra. Esse assunto é essencial no aprendizado dos alunos, pois a maioria dos aplicativos atuais é constituído por várias telas de navegação.

Nesta aula, os alunos programam o aplicativo “Navegar”.

Em um clique de um botão, eles adentram em outra tela e podem sair a qualquer momento.

O desafio dessa aula é unir dois aplicativos que já realizaram nas aulas passadas no mesmo aplicativo, cada um em telas diferentes.

### 3.2.1.7 Loops

Esse é um assunto bastante comum em cursos de programação, o que não deixa de ser necessário nas aulas. Este tópico é dividido em duas aulas.

Na primeira aula, os alunos aprenderão sobre o “for” e confeccionarão o aplicativo “Contador”.

Onde com o passar do tempo, o aplicativo mostra a contagem em segundos. O desafio desta aula é programar um aplicativo que mostra a tabuada.

Na segunda aula, os alunos aprenderão sobre o “while” e confeccionarão o aplicativo “Divisores”.

Onde ao inserir um número, o aplicativo mostrará ao usuário os divisores comuns. O desafio desta aula é programar um aplicativo que mostra os números primos a partir de um número inserido.

### 3.2.1.8 Funções

Outro tópico bastante comum em cursos de programação. Onde o aluno, nesta aula, aprenderá o que são funções e como utilizá-las na programação.

Os alunos desenvolverão o aplicativo “FatoreFibo”.

Figura 5 – Aplicativo FatoreFibo



Fonte: Próprio Autor (2022).

Utilizando-se dos conceitos das aulas passadas – loops, condicionais, segues – juntamente com funções, os alunos desenvolverão o aplicativo para calcular a fatorial e a sequência de Fibonacci utilizando funções.

O desafio dessa aula é programar um aplicativo para realizar o cálculo do mínimo divisor comum e do máximo multiplicador comum, cada um na sua respectiva tela e utilizando funções.

#### 3.2.1.10 Vetores

Vetor na programação é uma sequência de variáveis do mesmo tipo que são representados pelo mesmo nome, cuja a diferença está no índice que representa a localização do elemento dentro do vetor.

Nesta aula, o aluno programa o aplicativo “Colegas”.

Onde mostra na tela, uma sequência de nomes armazenadas no vetor. O desafio dessa aula é programar outra tela para inserir um novo elemento no vetor.

#### 3.2.1.9 Classes

Classes é basicamente uma definição de um tipo de dado, onde reunimos vários tipos primitivos para representar uma entidade, ou seja, aquilo que queremos.

Este é um assunto intermediário que geralmente é encontrado em cursos mais avançados. Nesta aula, os alunos aprendem o que é classe e como abstrair informações para o contexto da programação.

Continuando a aula, os alunos juntamente com o instrutor melhoram o aplicativo “Colegas”, transformando-o em uma classe com os parâmetros – nome, sobrenome e idade – e programando funções para adicionar um novo colega e mostrar as informações dos colegas registrados.

O desafio desta aula é fazer uma classe para automóveis cujo os parâmetros são – modelo, placa e cor – e as funções são para inserir um novo automóvel e mostrar os automóveis registrados.

### 3.2.1.10 Arquitetura MVC

De acordo com GAINHER (2019), o View Controller faz parte do “Padrão Model-View-Controller” (Padrão Modelo-Visão-Controlador). Este padrão de arquitetura de software separa a representação da informação da interação do usuário com um aplicativo em três partes, sendo usada para separar representações de informação internas dos modos como a informação é apresentada para e interagida pelo usuário. As responsabilidades são:

- O Model define a estrutura de dados como uma forma de modelagem, atualizando o sistema de acordo com as ações do usuário que interage com o programa. Como por exemplo, o usuário pressiona um botão, e todo o modelo deve ser atualizado à partir desta ação. Com isso, a View é atualizada;
- A View define o display do aplicativo de acordo com o modelo que foi projetado. Utilizando o mesmo exemplo, o usuário interage com um botão dentro de um aplicativo, de uma forma direta sem mexer com código ou modificando o modelo. A View manda a ação do usuário diretamente ao Controller;
- O Controller define a parte lógica da programação. Utilizando o mesmo exemplo, o usuário interage com um botão dentro de um aplicativo, e à partir disto, o que será modificado será a parte do código, que irá entender que um usuário apertou um botão e realizar ações em cima da mesma diretamente no código. O Controller irá manipular o Model à partir da ação, e também pode atualizar diretamente a View, de acordo com as mudanças pertinentes dentro do código.

Esse tipo de arquitetura é muito utilizado no ambiente de desenvolvimento Xcode. E utilizando-se do conteúdo anterior, os alunos nesta aula desenvolvem o aplicativo “ChooseStudent”.

Onde os parâmetros do estudante – nome, série e nota – são cadastrados no sistema e é programado as telas de inserir um novo estudante e outra tela para mostrar os estudantes cadastrados.

O desafio desta aula é desenvolver um aplicativo de banco onde os parâmetros são – nome do cliente, conta débito e conta crédito – são cadastrados no sistema e é programado as telas de inserir um novo cliente e outra tela para mostrar os clientes cadastrados.

### 3.2.1.11 Consumo de API

Uma API (Application Programming Interface) funciona como uma ponte entre duas aplicações. Dessa forma, uma aplicação interessada em utilizar os serviços de outra pode fazê-lo por meio de uma interface fácil de utilizar. As APIs permitem que o usuário final utilize um aplicativo, software ou até uma simples planilha, consultando, alterando e armazenando dados de diversos sistemas, sem que o usuário precise acessá-los diretamente. O intuito de uma API é trocar dados entre sistemas diferentes, na maior parte das vezes essas trocas de dados têm como objetivo automatizar processos manuais e/ou permitir a criação de novas funcionalidades (ORACLE, 2019).

Para o mesmo autor, o exemplo mais comum de serventia da API é o de um funcionário que precisa emitir notas e boletos para finalizar o pedido de um cliente. Nesse caso, a API pode conectar o sistema de gestão da empresa ao sistema de geração de boletos do banco – e ao de emissão de notas da prefeitura. Assim, o colaborador precisa apenas inserir os dados uma vez e finalizar o processo com poucos cliques.

Nesta aula, os alunos irão consumir a API do GITHUB com o aplicativo “Programe”.

Onde com a inserção do nome do usuário, o aplicativo revela a foto e quantos repositórios o programador tem.

O desafio dessa aula é programar um aplicativo que consuma outra API.

### 3.2.1.12 Banco de Dados

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados. Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser

facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados (ORACLE, 2019).

Nesta aula, os alunos o aplicativo “Carros”.

Onde ao inserir os dados – marca, ano e preço – o aplicativo armazenará as informações no banco de dados SQLite.

#### 3.2.1.13 IoT

No uso do IoT, foi desenvolvido juntamente com os alunos um aplicativo onde consumimos os dados de um esp8266 e acionamos uma lâmpada através do aparelho. Assim ampliando a visão de mundo dos estudantes para a área da computação.

#### 3.2.1.14 Últimas Aulas

As últimas aulas foram disponibilizadas para os alunos formarem as equipes e desenvolverem um aplicativo para a Apple Store. Semanalmente, havia uma reunião com o professor para esclarecimento do andamento do desenvolvimento do aplicativo e possíveis dúvidas.

Para o projeto final, deve-se utilizar os conteúdos apreendidos em sala de aula, assim como desenvolver o AutoLayout do aplicativo para funcionar em qualquer dispositivo IOS como também desenvolver um site com o fim de constatar a política de privacidade do aplicativo.

Para o projeto final, os alunos desenvolverão uma apresentação no modelo pitch, cuja apresentação durará em torno de 3 a 5 minutos com o objetivo de despertar o interesse dos avaliadores na formatura dos alunos.

#### 3.2.1.15 Encerramento

A formatura dos alunos aconteceu na metade de dezembro, onde 35 alunos formaram e apresentaram os seus trabalhos para seis avaliadores – três da área da educação e três da área da programação. Abaixo está detalhado os aplicativos dos alunos.

Figura 6 – Aplicativos Desenvolvidos



Fonte: Apple Store (2022).

O aplicativo Historiando foi desenvolvido por um trio de estudantes de escolas diferentes que programaram um aplicativo de história sobre assuntos nacionais, internacionais e continha a função de perguntas aleatórias para testar o conhecimento do usuário. Cada aluno ficou responsável por um aspecto do aplicativo – designer, lógica e confecção das perguntas. Com o aplicativo pronto, os alunos fizeram um teste de campo na escola e deram aula juntamente com o professor com o auxílio do aplicativo.

O aplicativo Fubol foi desenvolvido somente por um aluno com o intuito de testar os conhecimentos na área de futebol, porém o aplicativo não foi publicado na Apple Store por problemas na confecção das questões relacionadas ao futebol.

O aplicativo Mind Quiz foi desenvolvido por três alunos de escolas diferentes, que foi inspirado no aplicativo Brain Test: Jogos Mentais, um aplicativo de entretenimento para testar o raciocínio lógico do usuário. O aplicativo não foi publicado na apple store por problemas de modelagem no banco de dados.

O aplicativo Enricando ganhou a premiação de melhor aplicativo da noite, o aplicativo foi desenvolvido por dois alunos de escolas diferentes e tinha o foco de ensinar educação financeira para adolescente, com quizzes para mudar o *mindset* dos usuários em relação ao dinheiro e um simulador de juros para projetar o dinheiro futuro.

O aplicativo English for Kids foi desenvolvido por dois alunos onde se basearam em dois livros de inglês para desenvolverem o aplicativo. O aplicativo tem duas funções básicas – vocabulário, onde aparece a imagem para a criança escolher a resposta certa, e a gramática, para a criança preencher corretamente.

O aplicativo Chemistry Quiz foi desenvolvido por dois alunos de escolas diferentes com quatro opções, um de cada assunto da química.

O aplicativo Coding Quiz foi desenvolvido por um universitário do primeiro período onde ensina os usuários sobre assuntos diversos da área de programação. O aplicativo consistia em uma série de perguntas geradas aleatoriamente com o resultado mostrado ao final de dez questões.

O aplicativo Crazyando foi desenvolvido por um aluno de 11 anos de idade com a mesma ideia do aplicativo Mind Quis com foco no entretenimento e gera um quiz de dez questões aleatórias para testar o raciocínio lógico dos usuários. O aplicativo Crazyando não foi submetido para a apple store por problemas na configuração do banco de dados.

O aplicativo World Education foi desenvolvido por um aluno e foi o primeiro aplicativo submetido a Apple Store. O aplicativo consiste em uma série de perguntas sobre educação mundial para testar os conhecimentos dos usuários nesta curiosidades do mundo.

O aplicativo Potter Quiz foi desenvolvido por uma dupla de estudantes da mesma escola com o intuito de entreter os fãs do mundo de fantasia do Harry Potter com uma série de perguntas e resposta sobre o tema.

O aplicativo Urubuzal foi o único aplicativo que não chegou a conclusão e não foi apresentado para os avaliadores no dia, pois o aluno responsável ficou doente ao final do curso. O aplicativo era voltado para fãs do flamengo com o fim de testarem os conhecimentos do clube de forma lúdica.

O aplicativo Guerras Médicas foi desenvolvido por um aluno, onde fez uma vasta pesquisa sobre este momento histórico e desenvolveu o aplicativo para divulgar o acontecimento histórico. O aplicativo é composto por uma função que aleatoriamente escolhe dez questões para o usuário aprender sobre a história e no final é mostrado a pontuação do aluno.

O aplicativo Ápeiron foi desenvolvido por uma dupla de estudantes com foco em filosofia, onde o aplicativo sorteia aleatoriamente as questões e caso o aluno erre, mostra a resposta certa e no final mostra a pontuação que ficou. Um

aluno ficou responsável pelo designer e outro ficou responsável pelas questões, ambos desenvolveram o código juntos.

O aplicativo Projeto Sol foi desenvolvido somente por um aluno com o intuito de promover informações sobre o sistema solar. O aplicativo continha botões em formato de planetas e cada planeta ensinava sobre ele mesmo e no final tinha questões para o usuário testar o que aprendeu. O aplicativo não está na Apple Store sobre o pretexto do aplicativo ter poucas funcionalidades.

O aplicativo Pisco Edu foi desenvolvido por dois estudantes de escolas diferentes onde tinha a proposta de abastecer um banco de dados com várias matérias de diferentes séries, por ficar um trabalho enorme e com pouco tempo, entregaram um aplicativo com somente uma série e um assunto.

O aplicativo Helper Math foi desenvolvido por dois alunos de escolas diferentes e tinha o intuito de ajudar os alunos do nono ano na área de matemática com o ensino de trigonometria, equações, entre outros. O aplicativo não está na Apple Store por considerar o designer pouco atrativo.

O aplicativo Simple Math e Bee Operations foi desenvolvido por dois alunos e um aluno, respectivamente. Ambos os aplicativos estão na Apple Store e tem o foco de ensinar matemática básica para crianças como adição, subtração, multiplicação e divisão. Os aplicativos são compostos por uma série de questões que são sorteadas aleatoriamente e os erros são mostrados no decorrer das respostas de cada pergunta.

Por fim, o aplicativo Conectus foi desenvolvido juntamente com o professor e mais cinco alunos. O aplicativo armazena o valor ip de diferentes esp8266 e pode acionar remotamente qualquer dispositivo cadastrado no banco de dados. O aplicativo não está na Apple Store por falta de criptografia ao registrar os endereços ips no banco de dados.

### **3.2.2 Avaliação**

Nesta seção será descrita o processo de avaliação dos alunos no decorrer das aulas e o processo de avaliação dos aplicativos finais submetidos para Apple Store.

### 3.2.2.1 Avaliação dos Alunos

Para realizar a avaliação dos alunos é necessário um novo tipo de verificação para realmente comprovar se os alunos estão retendo os conteúdos, portanto utilizou-se a matriz Nine Box.

A matriz Nine Box é uma ferramenta de avaliação de desempenho de fácil aplicação. Ela revela quais alunos necessitam de ações para se desenvolverem melhor ou aqueles que estão a frente dos demais estudantes

. Quadro 1 – Matriz Nine Box

Potencial	Alto	Diamante Bruto	Estrela	Alto Potencial
	Médio	Dilema	Mantenedor	Futura Estrela
	Baixo	Ineficiente	Dúbio	Compromissado
		Baixo	Médio	Alto
		Desempenho		

Fonte: Próprio Autor (2022).

Tabela 2 – Categorias de Desempenho

Desempenho	
Baixo	Muitas Faltas
	Não consegue acompanhar as aulas
	Má vontade
	Não consegue se relacionar bem com a equipe
Médio	Só faz os aplicativos das aulas
	Não mostra nenhuma inovação
	Relacionamento OK com a equipe
Alto	Vem outros dias da semana
	Mostra inovações nos aplicativos

	Faz outros aplicativos além do previsto
	Sempre está disposto a ajudar os outros

Fonte: Próprio Autor (2022).

Tabela 3 – Categorias de Potencial

Potencial	
Baixo	Muita dificuldade em desenvolver aplicativos
Médio	Desenvolve somente os aplicativos do curso
Alto	Vai atrás de outros conhecimentos

Fonte: Próprio Autor (2022).

Portanto, o sistema não é baseado em notas, porém é baseado onde o estudante está localizado na matriz Nine Box. Para o aluno ser mantenedor, é somente comparecer as aulas e realizar as atividades juntamente com o professor. Para o aluno ser considerado Alto Potencial é necessário mostrar conteúdos aprendidos além da sala de aula e realizar os desafios semanais.

Nas segundas feiras, os alunos recebiam um feedback de aonde estavam na matriz Nine Box e como poderiam melhorar na próxima semana.

A matriz Nine Box juntamente com a análise dos GitHubs dos alunos foram fundamentais no decorrer dos alunos para detectar se os alunos estavam comparecendo as aulas e se estavam compreendendo os assuntos.

### 3.2.2.1 Avaliação dos Aplicativos

Os aplicativos submetidos para a Apple Store foram desenvolvidos em equipes e avaliados por uma representante da SEDUC (Secretaria de Estado de Educação e Desporto) e dois programadores profissionais de IOS. As avaliações seguiram o formato o quadro abaixo.

## Quadro 2 – Categorias de Avaliação dos Aplicativos

Categoria	Básico (1 ponto)	Intermediário (2 pontos)	Proficiência (3 pontos)	Domínio (4 pontos)	Pontos
<b>Conteúdo da apresentação</b>	Compartilha informações básicas, como objetivo e público-alvo.	Fornecer explicação clara do objetivo e design do app, e como ele atende às necessidades do usuário.	Fornecer explicação clara e convincente do problema a ser resolvido, da demanda de mercado, do público-alvo e de como o app foi criado para atender às necessidades do usuário.	Faz uma apresentação persuasiva sustentada por fatos que demonstram como o app atende às necessidades do usuário, vai além delas ou as redefine.	
<b>Execução da apresentação</b>	Informativa. Apresentação feita por um integrante da equipe.	Confiante e animada. Apresentação feita por mais de um integrante da equipe.	Envolvente e com bom uso de recursos visuais para sustentar a história. A equipe destaca as contribuições de cada integrante.	História contada de forma criativa e memorável. Auxílio visual envolvente e alternâncias harmoniosas entre os integrantes da equipe.	
<b>Interface do usuário (UI)</b>	Telas coerentes com o objetivo do app.	Design claro e funcional com elementos conhecidos. O protótipo sustenta as tarefas básicas de usuário.	Design elegante, conciso e agradável com uso consciente de cores, layouts e facilidade de leitura. O protótipo facilita a navegação do usuário pelo app.	O design permite que o usuário interaja facilmente com o conteúdo. O protótipo usa animações, cores e layouts para criar uma experiência transparente e envolvente.	
<b>Experiência do usuário (UX)</b>	Intenção clara. Os usuários conseguem atingir um ou mais objetivos.	Navegação padrão e estável. Caminho intuitivo pelo conteúdo do app.	Adaptável às necessidades do usuário. Trata acessibilidade, privacidade e segurança.	Inovador, encantador e surpreendente. Oferece aos usuários uma nova experiência que diferencia o app de seus concorrentes.	
<b>Conceitos de programação</b>	Algum nível de ligação entre a funcionalidade do app e o código por trás dele.	Explicação de como os conceitos gerais de programação, como tipos de dados, lógica condicional ou eventos de toque, se relacionam com o app.	Descrição de tarefas específicas de programação necessárias para criar o app. Demonstração de como o código é a força motriz da funcionalidade do app.	Explicação da arquitetura, estrutura de dados, algoritmos e recursos do app. Discussão da tomada de decisões ao desenvolver esta abordagem.	
<b>Avaliação técnica (opcional)</b> <i>Para protótipos funcionais de apps no Xcode. Os jurados devem conhecer a linguagem Swift e as melhores práticas de desenvolvimento de apps para iOS.</i>	A linguagem Swift é executada em exemplos específicos. O código é básico, sem abstração.	O código é executado sem erros em todos os casos. O código é básico, com algum grau de abstração.	O código é organizado seguindo as convenções de nomenclatura da linguagem Swift. Alto grau de abstração. As diretrizes de desenvolvimento para iOS são seguidas.	O código é bem documentado com comentários. Uso eficiente dos recursos da linguagem Swift. Emprega organização, como o uso do Model-View-Controller (modelo-visualização-controlador).	
<b>Comentários:</b>					<b>0</b> Pontuação total

Fonte: *Apple Developer* (2022).

Onde os avaliadores verificaram cada aplicativo e disseram o que poderiam melhorar em versões futuras.

## 4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

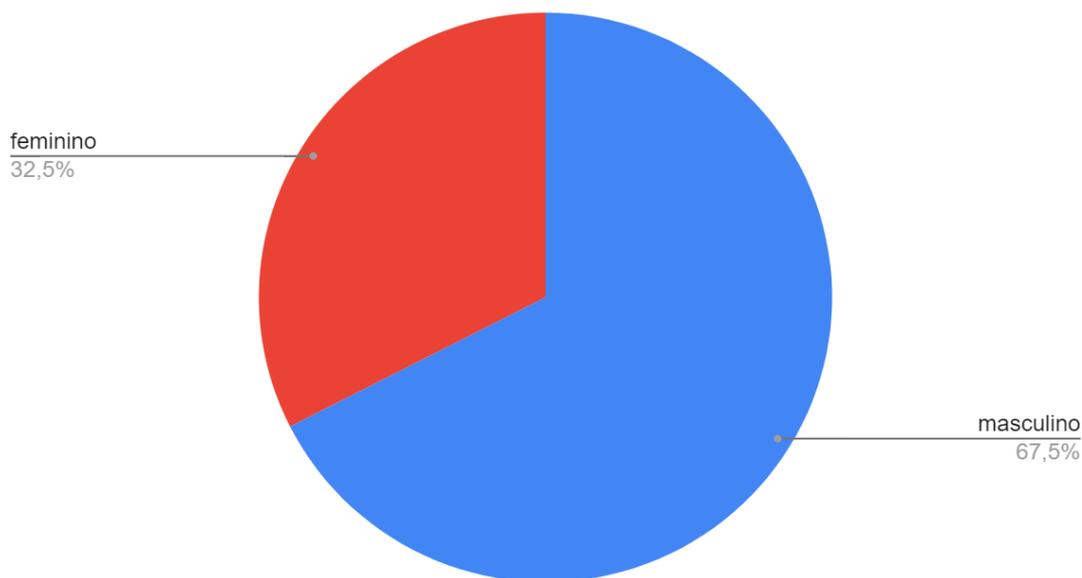
O curso de programação, como já citado, foi realizado com 40 alunos do ensino médio de três escolas públicas. O presente capítulo apresenta uma avaliação dos resultados obtidos com a execução do curso, estruturado da seguinte forma: caracterização dos alunos participantes, análise do aprendizado dos alunos, discorre se o curso cumpriu com o resultado esperado e apresenta a análise da entrevista realizada quanto com os alunos como com os avaliadores.

### 4.1 CARACTERIZAÇÃO DOS PARTICIPANTES

A partir do preenchimento do formulário pelos alunos foi possível traçar o perfil dos alunos que participaram do curso.

No gráfico abaixo, verifica-se que dos 40 estudantes, 67,5% eram do gênero masculino e 32,5% eram do gênero feminino.

Figura 7 – Percentual de masculino e feminino

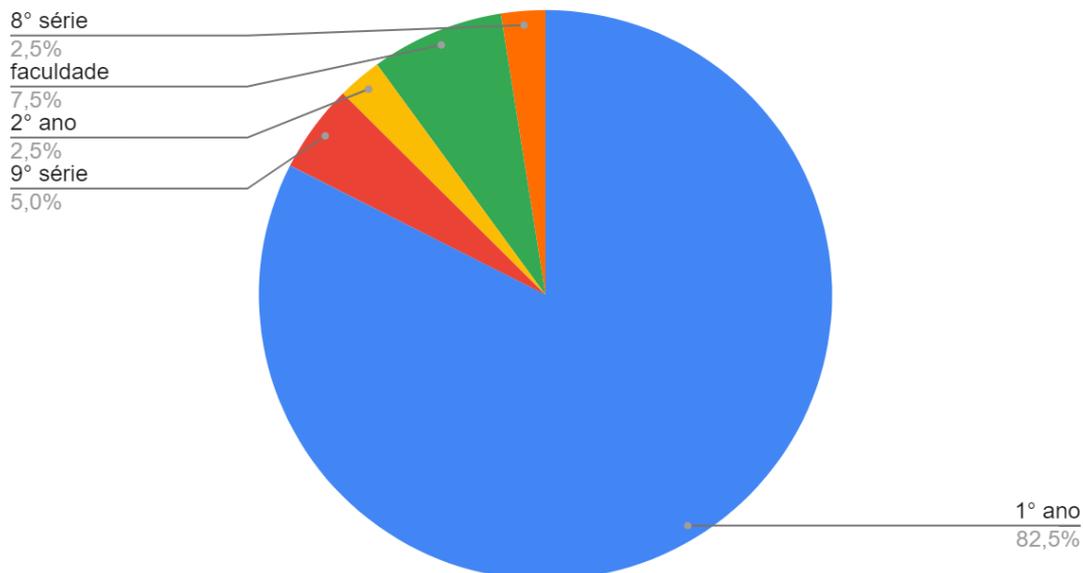


Fonte: Próprio Autor (2022).

Constatou-se também, com o gráfico abaixo, que a maioria dos estudantes são do primeiro ano do ensino médio, mas também houve procura por outras partes, onde se estabeleceu como estudantes aluno da 8ª série até

peças que cursavam faculdade com o interesse de publicar aplicativos na Apple Store.

Figura 8 – Percentual das séries dos alunos

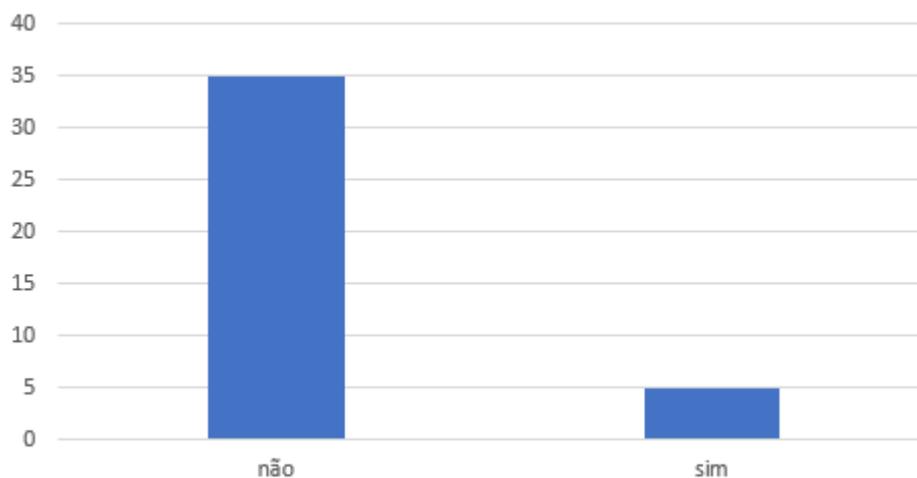


Fonte: Próprio Autor (2022).

De acordo com as respostas, apurou-se que 45,00% dos participantes possuíam interesse na área de informática, outros tinham preferências pela área de medicina, advocacia e designer de interiores. Perguntou-se também, se os mesmos já haviam realizado algum curso de informática, em que foi constatado que a maioria (90,00%) destes, já havia realizado. Também foi apurado quantos tinham feito algum curso na área de programação e constatou-se que 17,5% dos alunos tinham realizado um curso de programação antes. E por fim ficou evidenciado que 100% dos alunos nunca tinham usado Mac.

Outra variável constatada foi o nível de conhecimento dos alunos a respeito da área de programação. A qual está representada no gráfico abaixo.

Figura 9 – Relação dos Alunos com Conhecimentos em Programação



Fonte: Próprio Autor (2022).

De acordo com as repostas, cinco alunos já programaram anteriormente ao curso, o que representa 12,5% do total da amostra auferida a pesquisa. E trinta e cinco alunos, o que representa 87,5% dos alunos nunca programaram antes de começar o curso de programação.

#### 4.2 ANÁLISE DO APRENDIZADO DOS ALUNOS

Como forma de analisar se o que foi abordado durante o curso foi assimilado pelos alunos, foi solicitado aos mesmos que desenvolvessem um aplicativo para a Apple Store sem a intervenção do instrutor.

Nesta seção são apresentados os resultados obtidos por meio da análise dos projetos que cada equipe realizou de forma autônoma. Para tanto, procurou-se analisar o grau de dificuldades do aplicativo, as soluções encontradas para cada problema e os conceitos utilizados na programação. O resultado obtido é referente a apenas 35 alunos, pois 5 alunos desistiram no decorrer do curso.

Tabela 4 – Relação aplicativos e conteúdos de programação utilizados

Aplicativo	Condicionais	Loops	Funções	Arquitetura MVC	Consumo de API	Banco de Dados	IoT
Historiando	ok	ok	ok	ok	-	ok	-
Fubol	ok	ok	ok	ok	-	-	-
Mind Quiz	ok	ok	ok	ok	-	ok	-

Enricando	ok	ok	ok	ok	-	-	-
English for Kids	ok	ok	ok	ok	-	-	-
Chemistry Quiz	ok	ok	ok	ok	-	ok	-
Coding Quiz	ok	ok	ok	ok	-	-	-
Crazyando	ok	ok	ok	ok	-	-	-
Word Education	ok	ok	ok	ok	-	ok	-
Potter Quiz	ok	ok	ok	ok	-	-	-
Urubuzal	ok	ok	ok	ok	-	-	-
Guerras Médicas	ok	ok	ok	ok	-	ok	-
Ápeiron	ok	ok	ok	ok	-	-	-
Projeto Sol	ok	ok	ok	ok	-	-	-
Pisco Edu	ok	ok	ok	ok	-	-	-
Helper Math	ok	ok	ok	ok	-	-	-
Simple Mah	ok	ok	ok	ok	-	-	-
Bee Operations	ok	ok	ok	ok	-	-	-
Conectus	ok						

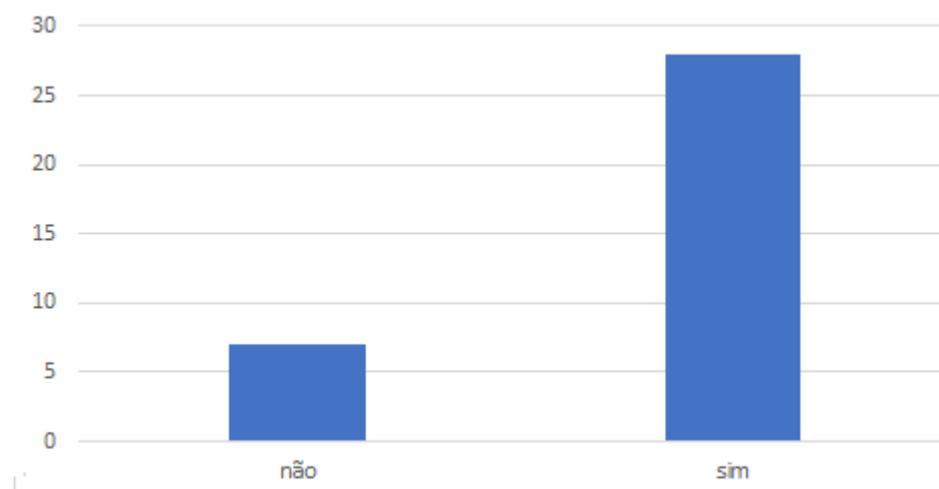
Fonte: Próprio Autor (2022).

Na tabela acima, percebe-se que todos utilizarão vários conceitos ensinados em sala de aula, porém cinco equipes usaram o banco de dados com eficiência. Isso deve ao fato de que os alunos somente tiveram um mês para desenvolver o aplicativo e os assuntos mais avançados – consumo de API e banco de dados – eram muito recentes para os alunos usarem com maestria. O aplicativo Conectus não foi contabilizado, pois o professor estava diretamente envolvido com o desenvolvimento.

Dos 18 aplicativos, apenas 6 não foram aprovados na Apple Store até o presente momento e a razão mais proeminente foi que o aplicativo tinha poucas funcionalidades. Então constata-se que os alunos aprenderam os conceitos de programação ensinados no curso e desenvolveram um aplicativo para a Apple Store.

De outro modo, constando-se as matrizes Nine Box dos alunos juntamente com o GitHub dos mesmos verifica-se o gráfico abaixo.

Figura 10 – Relação dos Alunos com Todos os Aplicativos no GitHub



Fonte: Próprio Autor (2022).

O gráfico apresenta a relação dos alunos que fizeram todos os aplicativos no decorrer do curso e dispuseram no GitHub. Assim 7 alunos dos formandos não completaram todos os aplicativos, porém 80% dos formandos, o que representa um total de 28 alunos concluíram todos os aplicativos e desenvolveram o seu portfólio.

#### 4.3 RESPOSTAS DOS ALUNOS

Foi gerado um formulário para os alunos responderem o que acharam da metodologia das aulas, o que poderia melhorar e como foi esta experiência.

De todos os alunos, 57,5% não gostaram de ficar mudando de colega para estudar, 32,5% preferiam continuar usando o mesmo computador durante todo o curso. Por outro lado, nenhum aluno reclamou sobre o conteúdo dos assuntos, mas 77,5% dos alunos gostariam de ter mais tempo para estudar. E sobre o instrutor, todos gostaram do instrutor.

#### 4.4 RESPOSTAS DOS AVALIADORES

No dia da amostra dos aplicativos para os avaliadores, que como dito antes, era um representante da SEDUC e dois profissionais de programação IOS, tivemos uma troca de ideias onde eles ficaram impressionados com o tempo curto que os estudantes tiveram para desenvolver e o que desenvolveram.

Um dos representantes comentou que era incrível que os alunos além de aprenderem a programar, também aprenderam a utilizar o GITHUB sem o uso de um software, mas direto pelo terminal.

Portanto, comprova-se que o curso de programação além de ser rico em conteúdo, também atinge o objetivo proposto que é ensinar adolescentes que não saibam programação a programar um aplicativo para a Apple Store.

## 5 CONSIDERAÇÕES FINAIS

O presente trabalho buscou investigar de que forma a cultura maker aliada a plataforma de desenvolvimento Xcode pode facilitar o processo de ensino-aprendizagem da matéria de programação, pois, conforme apresentado, várias são as vantagens de uma criança aprender programação. Dentre elas destacam-se o desenvolvimento de competências e habilidades essenciais para a vida de um indivíduo, tanto para os que desejam ingressar na área de computação, quanto para outras profissões, quais sejam: estimular a criatividade e a curiosidade; desenvolver a autonomia; desenvolver o raciocínio lógico e o pensamento computacional; desenvolver a capacidade de resolver problemas; aprender a trabalhar em equipe; e desenvolver a capacidade de interpretar problemas entre outras.

Desta forma, concebeu e implementou um curso de programação para 40 alunos do primeiro ano de três escolas públicas. Foram realizados 16 encontros com duração de uma hora e meia cada, com turmas a noite na terça e quarta das 17h às 18h30 e uma turma na quarta de manhã das 9h às 10h30. No decorrer do curso foi ensinado para os alunos a fazer aplicativos utilizando a linguagem swift na plataforma de desenvolvimento Xcode. Essa ferramenta foi escolhida por ter um painel gráfico que facilita para os alunos poderem desde o primeiro dia desenvolver um aplicativo.

Para a obtenção dos resultados da pesquisa, apresentou-se uma análise do perfil dos alunos, uma análise do aprendizado dos alunos e uma análise das entrevistas com os alunos e com os avaliadores. Além da verificação e análise dos aplicativos desenvolvidos para a Apple Store.

Foi possível notar a motivação dos alunos para o ensino de lógica de programação, a aceitabilidade da metodologia e da ferramenta utilizadas, o desenvolvimento do raciocínio lógico e do pensamento computacional, bem como o estímulo para ingresso na área da computação e pelo aprendizado em lógica de programação.

A pesquisa também mostrou que o ensino de programação para este público é possível, quando se emprega ferramentas adequadas que possibilitem os adolescentes aprender de forma divertida e com dedicação. Conclui-se, portanto, que a realização da pesquisa permitiu obter resultados positivos tanto

aos participantes quanto ao autor deste trabalho. Para o autor, a realização da pesquisa promoveu a experiência de ensinar, no qual precisou aprender sobre: formas de lecionar, preparar aulas, aprender a lidar com adolescentes, pesquisar e desenvolver metodologias de ensino para adolescentes, aprender a lidar com dificuldades encontradas, entre outras. Assim, promoveu o enriquecimento de sua formação. Para os participantes, infere-se que os mesmos conseguiram submeter, em um curto período de tempo, um aplicativo para a Apple Store, deixando claras a possibilidade e necessidade do ensino de lógica de programação para adolescentes.

Durante a pesquisa é importante relatar a dificuldade em encontrar trabalhos correlatos, o que prejudicou um maior aporte de comparações entre os resultados da pesquisa com os de outros autores. Já ao conceber o curso proposta, a falta de experiência inicial, trouxe dificuldades no desenvolvimento dos planos de aulas, da seleção de atividades e projetos que seriam realizadas, as quais foram sendo superadas a partir da observação da literatura encontrada. Além disso, algumas limitações físicas ocorreram no decorrer do curso, como falta de energia e falha no fornecimento de internet, provocando atrasos no cronograma inicial.

Como trabalhos futuros, propõem-se: que se estude de forma mais aprofundada o grau de conhecimento de programação adquiridos pelos alunos, esmiuçar o conteúdo para melhorar a performance dos alunos na retenção de conteúdo e desenvolver uma segunda parte do curso para os alunos que se graduarem estejam aptos a trabalhar no mercado de programação.

## REFERÊNCIAS

CYULIK, Lucas. **COMO APRESENTAR PROGRAMAÇÃO PARA CRIANÇAS? ENVOLVER PARA DESMISTIFICAR, VALORIZAR PARA IMPLEMENTAR.** Orientador: Dra. Marília Abrahão Amaral. 2016. 68 p. Monografia de Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Ponta Grossa, 2016.

BERSSANETTE, João Henrique. **CONSTRUINDO “NOVOS” CAMINHOS PARA O ENSINO DE PROGRAMAÇÃO POR MEIO DO USO DE METODOLOGIAS ATIVAS DE APRENDIZAGEM E TEORIA DA CARGA COGNITIVA.** 2020. 682 p. Tese TECNOLÓGICA FEDERAL DO PARANÁ, Ponta Grossa, 2020.

HIRANO, Fabio Takashi. **O USO DE GAMIFICATION NO ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS: UM ESTUDO DE CASO.** Supervisor: Dr. Marco Dimas Gubitoso. 2015. 26 p. Monografia – INSTITUTO DE MATEMÁTICA E ESTATÍSTICA DA UNIVERSIDADE DE SÃO PAULO, São Paulo, 2015.

ZANATTA, Andrei Cardozo. **PROGRAMAÇÃO DE COMPUTADORES PARA CRIANÇAS – METODOLOGIA DE CODE CLUB BRASIL.** Orientador: Dra. Eliane Pozzebon. 2015. 102 p. Trabalho de Conclusão de Curso (Bacharel em Tecnologia da Informação e Comunicação) – UNIVERSIDADE FEDERAL DE SANTA CATARINA, Araranguá, 2015.

SILVA, Leandro Orlando Sousa da. **ENSINO DE LÓGICA DE PROGRAMAÇÃO PARA ALUNOS DO ENSINO FUNDAMENTAL UTILIZANDO A FERRAMENTA SCRATCH.** Orientador: Dra. Marianne Kogut Eliasquevici. 2018. 90 p. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – UNIVERSIDADE FEDERAL DO PARÁ, Belém, 2018.

JUNIOR, Flavio Malvestiti. **ANALISE E DESENVOLVIMENTO DE TÉCNICAS DE ENSINO DE PROGRAMAÇÃO PARA JOVENS E ADOLESCENTES.** Orientador: Lilian J. Meyer Riveros. 2020. 25 p. Trabalho de Conclusão de Curso.

GARLET, Daniela; BIGOLIN, Nara Martini; SILVEIRA, Sidnei Renato. **UMA PROPOSTA PARA O ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NA EDUCAÇÃO BÁSICA.** 2016. 25 p. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – UNIVERSIDADE FEDERAL DE SANTA MARIA, Frederico Westphalen – RS, 2016.

SOUZA, Laís dos Santos. **A CULTURA MAKER NA EDUCAÇÃO: PERSPECTIVAS PARA O ENSINO E A APRENDIZAGEM DE MATEMÁTICA.** Orientador: Ma. Marcella Suarez Di Santo. 2021. 68 p. Trabalho de Conclusão

de Curso (Licenciatura em Matemática) – INSTITUTO FEDERAL DE GOIÁS, Valparaíso de Goiás, 2021.

SILVA, Regina Maria Borges. **DESIGN E CULTURA MAKER COMO FATOR EDUCACIONAL E SUSTENTÁVEL, PARA EDUCAÇÃO INFANTIL**. Orientador: Dra. Aline Teixeira de Souza. 2022. 74 p. Trabalho de Conclusão de Curso (Bacharel em Designer) – UNIVERSIDADE FEDERAL DE UBERLÂNDIA, Uberlândia, 2022.

LIMA, Diego Menezes de. **O MOVIMENTO MAKER E A EDUCAÇÃO: IMPACTO DA APLICAÇÃO DE FERRAMENTAS COMPUTACIONAIS QUE ESTIMULAM A LÓGICA E O PENSAMENTO CRIATIVO EM ALUNOS DO ENSINO FUNDAMENTAL**. Orientador: Joacir Giaretta. 2019. 108 p. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) – UNIVERSIDADE DE CAXIAS DO SUL, Bento Gonçalves, 2019.

**APPLE DEVELOPER**. Disponível em: <<https://developer.apple.com/>> Acesso em: 03 de Fevereiro de 2022.

O que é um Banco de Dados?. **ORACLE**. Disponível em: <<https://www.oracle.com/br/database/what-is-database/>>. Acesso em: 21 de Novembro de 2022.

TV A CRÍTICA, **Coding Club**, Canal 4, Manaus, 10 de Dezembro de 2022.

REDE AMAZÔNICA – REDE GLOBO, **Coding Club**, Canal 5, 10 de Dezembro de 2022.