



**PROFEPT**

MESTRADO PROFISSIONAL EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL  
Farroupilha

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
FARROUPILHA  
CAMPUS JAGUARI  
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO PROFISSIONAL E  
TECNOLÓGICA**

**RODRIGO THOMAS**

**O ENSINO DE PROGRAMAÇÃO NA EDUCAÇÃO PROFISSIONAL E  
TECNOLÓGICA: DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL  
POR MEIO DA CONSTRUÇÃO DE APLICATIVOS NO APP INVENTOR**

Jaguari  
2022

**RODRIGO THOMAS**

**O ENSINO DE PROGRAMAÇÃO NA EDUCAÇÃO PROFISSIONAL E  
TECNOLÓGICA: DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL  
POR MEIO DA CONSTRUÇÃO DE APLICATIVOS NO APP INVENTOR**

Dissertação apresentada ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo campus Jaguari do Instituto Federal Farroupilha, como parte dos requisitos para a obtenção do título de Mestre em Educação Profissional e Tecnológica.

Orientador: Profº Dr. Adão Caron Cambraia

Jaguari  
2022

Ficha catalográfica

Dados Internacionais de Catalogação na Publicação (CIP)

T454e Thomas, Rodrigo  
O ensino de programação na educação profissional e tecnológica: desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor / Rodrigo Thomas. - Jaguari, 2022.  
174 f. : il.

Orientador: Adão Caron Cambraia  
Dissertação (Mestrado) – Programa de Pós-graduação em Educação Profissional e Tecnológica, Instituto Federal de Educação, Ciência e Tecnologia Farroupilha, 2022.

1. Pensamento computacional. 2. Sequência didática.  
3. Aplicativos móveis. I. Cambraia, Adão Caron. II. Título.

CDU: 004.4

Elaborada por:  
Márcia Della Flora Cortes CRB10/1877

**RODRIGO THOMAS**

**O ENSINO DE PROGRAMAÇÃO NA EDUCAÇÃO PROFISSIONAL E  
TECNOLÓGICA: DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL  
POR MEIO DA CONSTRUÇÃO DE APLICATIVOS NO APP INVENTOR**

Dissertação apresentada ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo Instituto Federal Farroupilha, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Aprovado em 10 de março de 2022.

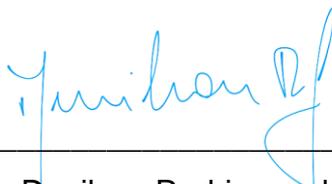
**COMISSÃO EXAMINADORA**



Prof. Dr. Adão Caron Cambraia  
Instituto Federal Farroupilha - Orientador



Prof.ª Dr.ª Sandra Elisabet Bazana Nonenmacher  
Instituto Federal Farroupilha



Prof. Dr. Denilson Rodrigues da Silva  
Universidade Regional Integrada do Alto Uruguai e das Missões

**RODRIGO THOMAS**

**SEQUÊNCIA DIDÁTICA PARA O DESENVOLVIMENTO DO PENSAMENTO  
COMPUTACIONAL POR MEIO DA CONSTRUÇÃO DE APLICATIVOS NO APP  
INVENTOR**

Produto Educacional apresentado ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo Instituto Federal Farroupilha, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Validado em 10 de março de 2022.

**COMISSÃO EXAMINADORA**



Prof. Dr. Adão Caron Cambraia  
Instituto Federal Farroupilha - Orientador



Prof.ª Dr.ª Sandra Elisabet Bazana Nonenmacher  
Instituto Federal Farroupilha



Prof. Dr. Denilson Rodrigues da Silva  
Universidade Regional Integrada do Alto Uruguai e das Missões

Dedico este trabalho à minha família, em especial aos meus pais que, apesar das limitações e dificuldades, nunca mediram esforços para possibilitar uma educação qualificada aos filhos.

## **AGRADECIMENTOS**

À Deus pelo dom da vida, por me guiar e ser a luz no caminho.

À minha família, meus pais que, em toda sua simplicidade, muitos valores me ensinaram e serviram de inspiração para os momentos mais difíceis. Aos meus irmãos, que mesmo distantes fisicamente, em pensamentos sempre estiveram comigo durante essa trajetória.

À minha esposa, Fabiane, pelo apoio incondicional, incentivo, paciência, compreensão e por todo amor.

Ao meu orientador, professor Adão, por todo conhecimento compartilhado, pela maestria na condução das orientações, pela disponibilidade e auxílio durante os encontros com os alunos, meu respeito e reconhecimento.

Aos colegas e docentes do ProfEPT, pela parceria e aprendizagens proporcionadas durante esse período ímpar. Os levo na memória com muito carinho.

Aos professores da banca, que gentilmente aceitaram o convite, pelas relevantes contribuições que tornaram este trabalho mais significativo.

Por fim, à todos que me incentivaram e ajudaram a tornar esse sonho possível, meu profundo agradecimento.

*A verdadeira habilidade competitiva é a habilidade de aprender.  
Não devemos aprender a dar respostas certas ou erradas, temos de aprender a  
solucionar problemas.*

*(Seymour Papert)*

## RESUMO

O pensamento computacional é uma habilidade que contribui na formulação, compreensão e resolução de problemas da vida cotidiana. Trata-se, portanto, de uma aptidão importante para todas as pessoas (WING, 2006). Assim, a presente pesquisa traz como problema: de que forma a construção de aplicativos móveis por meio da programação baseada em blocos no App Inventor proporciona o desenvolvimento do pensamento computacional e contribui com a aprendizagem de programação na Educação Profissional e Tecnológica? No intuito de responder a esse questionamento, propomos uma sequência didática pautada nos Três Momentos Pedagógicos sobre o ensino de programação de aplicativos móveis. A metodologia da pesquisa é qualitativa, assumindo o formato de estudo de caso com grupos de alunos ingressantes dos cursos superiores de Licenciatura em Computação do Instituto Federal Farroupilha - Campus Santo Augusto e Campus Santo Ângelo/RS. Os dados da pesquisa foram gerados por meio de questionários de percepção inicial e final e testes na ferramenta CodeMaster dos aplicativos construídos pelos participantes, visando identificar a ocorrência do desenvolvimento do pensamento computacional. O produto educacional resultante da pesquisa é constituído pela sequência didática, dividida em cinco aulas e disponibilizada através de um *website*, com o intuito de contextualizar o conhecimento à realidade dos educandos, desenvolver habilidades relativas ao pensamento computacional e construir aplicativos móveis no App Inventor. Os resultados do estudo de caso apontam uma maior motivação dos alunos para a aprendizagem de programação e indícios do desenvolvimento do pensamento computacional a partir da resolução dos desafios propostos durante as aulas.

**Palavras-chave:** Pensamento Computacional; Sequência Didática; Aplicativos Móveis.

## ABSTRACT

Computational thinking is a skill that contributes to the formulation, understanding and problem solving of everyday life. It is, therefore, an important skill for all people (WING, 2006). Thus, the present research brings as a problem: how does the construction of mobile applications through programming based on blocks in App Inventor provide the development of computational thinking and contribute to the learning of programming in Vocational and Technological Education? In order to answer this question, we propose a didactic sequence based on the Three Pedagogical Moments on the teaching of mobile application programming. The research methodology is qualitative, taking the form of case study with groups of students entering the undergraduate courses in Computer Science at Instituto Federal Farroupilha - Campus Santo Augusto and Campus Santo Ângelo/RS. The research data were generated through initial and final perception questionnaires and tests in the CodeMaster tool of the applications built by the participants, aiming to identify the occurrence of the development of computational thinking. The educational product resulting from the research consists of the didactic sequence, divided into five classes and made available through a website, in order to contextualize knowledge to the reality of students, develop skills related to computational thinking and build mobile applications in App Inventor. The results of the case study point to a greater motivation of the students to learn programming and evidence of the development of computational thinking from the resolution of the challenges proposed during the classes.

**Keywords:** Computational Thinking; Didactic Sequence; Mobile Applications.

## LISTA DE FIGURAS

Figura 1 – Distribuição das unidades que compõem a Rede Federal de Educação Profissional, Científica e Tecnológica .....	28
Figura 2 – Eixos dos conhecimentos da área de Computação .....	37
Figura 3 – Mandala do Currículo de Referência em Tecnologia e Computação na Educação Básica.....	44
Figura 4 – Pilares do Pensamento Computacional .....	47
Figura 5 – Similaridade entre figuras (Reconhecimento de Padrões) .....	50
Figura 6 – Mapa do Transporte Metropolitano (Abstração) .....	51
Figura 7 – Janela “Designer” do App Inventor .....	56
Figura 8 – Janela “Blocos” do App Inventor.....	57
Figura 9 – Etapas da Sequência Didática .....	70
Figura 10 – Relação entre as práticas do pensamento computacional e o conceito (e subconceitos) de algoritmos e programação.....	72
Figura 11 – Resultado da avaliação de um projeto no CodeMaster .....	73
Figura 12 – Subdimensões dos itens da rubrica CodeMaster para App Inventor.....	74
Figura 13 – Página inicial do <i>website</i> do Produto Educacional .....	78
Figura 14 – Apresentação do <i>website</i> do Produto Educacional .....	79
Figura 15 – Seção “Sequência Didática” do <i>website</i> .....	81
Figura 16 – Segmento da aula 1 sobre os pilares do pensamento computacional ...	84
Figura 17 – Segmento da aula 2 sobre a escrita de algoritmos .....	86
Figura 18 – Segmento da aula 3 sobre estruturas de seleção .....	88
Figura 19 – Segmento da aula 4 sobre estruturas de repetição.....	90
Figura 20 – Segmento da aula 5 sobre várias telas no App Inventor .....	92
Figura 21 – Seção “Publicações” do <i>website</i> .....	94
Figura 22 – Seção “Contato” do <i>website</i> .....	95
Figura 23 – Seção “Sobre o site” .....	96
Figura 24 – Aplicativos desenvolvidos pelos alunos no App Inventor durante a realização da sequência didática .....	106
Figura 25 – Divulgação do questionário de percepção final.....	108
Figura 26 – Acompanhamento de atividades no App Inventor e entrega de teclado adaptado a aluno com baixa visão .....	117

Figura 27 – Resultado da atividade colaborativa de construção das partes de um algoritmo para o cálculo da média entre dois números ..... 119

## LISTA DE GRÁFICOS

Gráfico 1 – Respostas dos alunos do Campus Santo Augusto à pergunta: “Em que ano você nasceu?” .....	63
Gráfico 2 – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Em que ano você nasceu?” .....	63
Gráfico 3 – Respostas dos alunos do Campus Santo Augusto à pergunta: “Qual(is) equipamento(s) você possui em casa?” .....	64
Gráfico 4 – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Qual(is) equipamento(s) você possui em casa?” .....	64
Gráfico 5 – Respostas dos alunos do Campus Santo Augusto à pergunta: “Se você possui acesso à internet, qual(is) equipamento(s) utiliza para acessar?” .....	65
Gráfico 6 – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Se você possui acesso à internet, qual(is) equipamento(s) utiliza para acessar?” .....	65
Gráfico 7 – Respostas dos alunos do Campus Santo Augusto à pergunta: “Quais conceitos você estudou na disciplina de algoritmos?” .....	66
Gráfico 8 – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Quais conceitos você estudou na disciplina de algoritmos?” .....	66
Gráfico 9 – Respostas dos alunos do Campus Santo Augusto à pergunta: “Você já ouviu ou leu algo a respeito do termo “Pensamento Computacional”?” .....	67
Gráfico 10 – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Você já ouviu ou leu algo a respeito do termo “Pensamento Computacional”?” .....	68
Gráfico 11 – Respostas dos alunos à pergunta: “Você acredita que é importante aprender sobre o Pensamento Computacional?” .....	110
Gráfico 12 – Respostas dos alunos à pergunta: “De maneira geral, o que você achou das atividades desenvolvidas?” .....	110
Gráfico 13 – Respostas dos alunos à pergunta: “Você acredita que as atividades contribuíram para o seu entendimento dos conceitos abordados na disciplina de Algoritmos?” .....	111
Gráfico 14 – Respostas dos alunos à pergunta: “Você acredita que as atividades contribuíram no seu processo de construção do Pensamento Computacional?” ....	111
Gráfico 15 – Respostas dos alunos à pergunta: “O que você achou da ferramenta de programação visual com blocos MIT App Inventor?” .....	112
Gráfico 16 – Respostas dos alunos à pergunta: “Você conseguiu desenvolver os	

aplicativos propostos ao final de cada atividade?” .....	112
Gráfico 17 – Respostas dos alunos à pergunta: “Você acredita que os conteúdos se tornam mais interativos e/ou lúdicos a partir do desenvolvimento de aplicativos móveis?” .....	113
Gráfico 18 – Respostas dos alunos à pergunta: “Você gostaria que essas atividades fossem incorporadas à disciplina de Algoritmos de forma permanente (para as turmas ingressantes)?” .....	113

## LISTA DE TABELAS

Tabela 1 – Reprovação de alunos na disciplina de Algoritmos: 2015 à 2018. ....	24
Tabela 2 – Número de alunos participantes e de aplicativos desenvolvidos durante as aulas da sequência didática. ....	100
Tabela 3 – Avaliação dos aplicativos desenvolvidos no desafio da aula 1 pelos alunos do Campus Santo Augusto. ....	101
Tabela 4 – Avaliação dos aplicativos desenvolvidos no desafio da aula 2 pelos alunos do Campus Santo Augusto. ....	102
Tabela 5 – Avaliação dos aplicativos desenvolvidos no desafio da aula 3 pelos alunos do Campus Santo Augusto. ....	103
Tabela 6 – Avaliação dos aplicativos desenvolvidos no desafio da aula 3 pelos alunos do Campus Santo Ângelo. ....	103
Tabela 7 – Avaliação dos aplicativos desenvolvidos no desafio da aula 4 pelos alunos do Campus Santo Augusto. ....	104
Tabela 8 – Avaliação dos aplicativos desenvolvidos no desafio da aula 4 pelos alunos do Campus Santo Ângelo. ....	104
Tabela 9 – Avaliação dos aplicativos desenvolvidos no desafio da aula 5 pelos alunos do Campus Santo Augusto. ....	105

## LISTA DE ABREVIATURAS E SIGLAS

- ABNT – Associação Brasileira de Normas Técnicas
- ACM – Association for Computing Machinery
- BBC – *British Broadcasting Corporation*
- BNCC – Base Nacional Comum Curricular
- CAS – *Children's Aid Society Of London and Middlesex*
- CEFET – Centro Federal de Educação Tecnológica
- CIEB – Centro de Inovação para a Educação Brasileira
- COVID-19 – Coronavírus
- CSTA – *Computer Science Teachers Association*
- CT – *Computational Thinking*
- EduComp – Simpósio Brasileiro de Educação em Computação
- EPT – Educação Profissional e Tecnológica
- IBGE – Instituto Brasileiro de Geografia e Estatística
- IFFar – Instituto Federal de Educação, Ciência e Tecnologia Farroupilha
- IFs – Institutos Federais
- IFSC – Instituto Federal de Santa Catarina
- ISTE – *International Society for Technology in Education*
- MEC – Ministério da Educação
- MIT – *Massachusetts Institute of Technology*
- MP – Momento Pedagógico
- NAPEAD – Núcleo de Apoio Pedagógico à Educação a Distância
- NVDA – *NonVisual Desktop Access*
- OCDE – Organização para a Cooperação e Desenvolvimento Econômico
- PC – Pensamento Computacional
- PNAD – Pesquisa Nacional por Amostras de Domicílios
- PPC – Projeto Pedagógico de Curso
- PROFEPT – Mestrado Profissional em Educação Profissional e Tecnológica
- SBC – Sociedade Brasileira de Computação
- SBIE – Simpósio Brasileiro de Informática na Educação
- TI – Tecnologia da Informação
- TCLE – Termo de Consentimento Livre e Esclarecido
- TICs – Tecnologias da Informação e Comunicação

UFRGS – Universidade Federal do Rio Grande do Sul

UFRPE – Universidade Federal Rural de Pernambuco

UNISINOS – Universidade do Vales dos Sinos

UNIVESP – Universidade Virtual do Estado de São Paulo

UTFPR – Universidade Tecnológica Federal do Paraná

VPL – *Visual Programming Language*

3MP – Três Momentos Pedagógicos

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>19</b>
<b>2. FORMAÇÃO INTEGRADA, PENSAMENTO COMPUTACIONAL E INTERAÇÃO PEDAGÓGICA</b> .....	<b>27</b>
2.1 A formação integrada nos Institutos Federais .....	27
2.2 Três Momentos Pedagógicos como organizadores do processo didático....	33
2.3 Pensamento computacional como alternativa de formação integrada .....	36
2.3.1 Pilares do pensamento computacional .....	47
2.3.1.1 Decomposição .....	48
2.3.1.2 Reconhecimento de padrões .....	49
2.3.1.3 Abstração .....	50
2.3.1.4 Algoritmos .....	51
2.4 Computação móvel, App Inventor e pensamento computacional.....	52
<b>3. CAMINHOS METODOLÓGICOS</b> .....	<b>59</b>
3.1 Metodologia da pesquisa .....	59
3.2 Local de pesquisa e amostra .....	60
3.2.1 Caracterização dos sujeitos.....	61
3.3 Construção da sequência didática .....	69
3.4 Avaliação do pensamento computacional através do CodeMaster .....	71
<b>4. PRODUTO EDUCACIONAL</b> .....	<b>75</b>
4.1 <i>Website</i> da sequência didática.....	76
4.1.1 Seção “Página inicial” .....	77
4.1.2 Seção “Apresentação” .....	79
4.1.3 Seção “Sequência Didática” .....	80
4.1.3.1 Aula 1 .....	82
4.1.3.2 Aula 2.....	84
4.1.3.3 Aula 3.....	87
4.1.3.4 Aula 4.....	88
4.1.3.5 Aula 5.....	90
4.1.4 Seção “Dissertação” .....	92
4.1.5 Seção “Publicações” .....	93
4.1.6 Seção “Contato”.....	95
4.1.7 Seção “Sobre o site” .....	96

<b>5. RESULTADOS</b> .....	<b>97</b>
5.1 Dinâmica das aulas.....	97
5.2 Avaliação dos aplicativos no CodeMaster.....	98
5.3 Questionário de percepção final.....	108
5.4 Manifestações dos alunos e percepções do pesquisador.....	114
<b>6. CONSIDERAÇÕES FINAIS</b> .....	<b>121</b>
<b>REFERÊNCIAS</b> .....	<b>125</b>
<b>APÊNDICE A - PRODUTO EDUCACIONAL</b> .....	<b>135</b>
<b>APÊNDICE B - QUESTIONÁRIOS</b> .....	<b>162</b>
<b>APÊNDICE C - TCLE</b> .....	<b>173</b>

## 1. INTRODUÇÃO

Hoje, cada vez mais as tecnologias digitais estão inseridas no cotidiano das pessoas. Os avanços tecnológicos atingem praticamente todos os setores e, com isso, surge a necessidade do desenvolvimento de novas competências (pensamento computacional, pensamento crítico, colaboração, criatividade, autonomia, entre outras). As discussões a respeito das habilidades necessárias para o século XXI também perpassam o ambiente educacional, uma vez que a presença cada vez mais frequente da tecnologia tem influência direta na aprendizagem. A educação e a tecnologia servem de apoio uma à outra e podem evoluir em conjunto. Embora haja uma ideia equivocada de que a educação possa estar atrasada em relação à tecnologia, há inúmeros casos de inovações técnicas promovidas a partir da educação (UNESCO, 2013).

A capacitação de um cidadão num mundo permeado de tecnologia requer a formação de produtores de novas soluções de tecnologias da informação (CSTA, 2016). Nesse sentido, a escola não pode estar alheia a esse processo, uma vez que a sociedade contemporânea tem sido influenciada pela cultura digital em decorrência do crescente acesso à Internet, principalmente por meio de dispositivos móveis. Segundo Nunes *et al.* (2017), os estudantes nasceram no meio social tecnológico e estão acostumados com a linguagem hipermidiática dos jogos digitais, redes sociais e ambientes virtuais.

Isso faz com que os discentes esperem pela inserção das tecnologias digitais na educação desde o ensino fundamental, seja como ferramenta de aprendizagem, seja como objeto de estudo. O papel da escola, diante desse contexto, é possibilitar aos educandos um ambiente que favoreça a aprendizagem das novas tecnologias de forma inovadora e criativa, visando reduzir a exclusão digital existente, principalmente, entre as classes sociais de menor renda. Essa demanda passa, também, pelo papel desempenhado pelo professor, a partir da exploração de novas possibilidades de ensino com a finalidade de expandir as perspectivas e

acompanhar as realizações dos alunos, utilizando as tecnologias de forma adequada para melhorar a interação com os estudantes e promover a aprendizagem (ALVES *et al.*, 2017).

Dessa forma, os conhecimentos da área da computação são importantes não somente para a formação de profissionais de informática, mas para a capacitação de profissionais das mais variadas áreas que se fazem valer dos conceitos, técnicas e ferramentas da computação para desenvolver novas tecnologias, aprimorar processos e economizar tempo e recursos. Assim, é fundamental a introdução das habilidades da área da Ciência da Computação nos diferentes níveis de ensino, a começar pela Educação Básica.

Uma das principais e precursoras iniciativas, nesse sentido, pode ser atribuída a Seymour Papert que, ainda na década de 80, propunha através da perspectiva construcionista, o uso da linguagem LOGO como uma abordagem pedagógica para o ensino dos conceitos de computação. Segundo Papert (1986, p. 187), “a verdadeira alfabetização computacional não é apenas saber como usar o computador e as ideias computacionais. É saber quando é apropriado fazê-lo.”

Em vista disso, Papert pode ser considerado um dos precursores da difusão da computação na educação, capacidades essas que atualmente são conhecidas como pensamento computacional (PC<sup>1</sup>). Apesar de não definir o termo, as iniciativas de Papert foram muito importantes e serviram de base para a conceituação do PC. A definição mais difundida foi proposta por Jannete Wing que, em 2006, conceituou a combinação do pensamento crítico com os fundamentos da computação como um processo para resolver problemas, denominado pensamento computacional (WING, 2006). Assim, o desenvolvimento do PC pressupõe a compreensão de problemas e a capacidade de formulação de soluções computacionais ou não, que superem o simples domínio da TICs.

Com relação à relevância dessa temática, Boucinha *et al.* (2017) afirmam que o pensamento computacional pode ser comparado com a alfabetização do século XXI, uma vez que inúmeros postos de trabalho passaram a necessitar de habilidades básicas de codificação. Estudos estimam que em 2022, só no Brasil, haverá um déficit de 408 mil profissionais de TI, sendo que a demanda de profissionais vem crescendo e sendo um dos obstáculos à expansão da Indústria

---

<sup>1</sup> Nesse trabalho, utilizamos a abreviatura PC para o termo Pensamento Computacional e não *Personal Computer* como normalmente adotado na literatura.

Brasileira de Software e Serviços de TI (VILLELA, 2013).

Apesar da união de entidades profissionais, educadores e órgãos públicos com vistas a promover metodologias e ferramentas para o desenvolvimento do pensamento computacional em ambientes formais e informais de aprendizagem, aproximando os conhecimentos de computação ao cotidiano das pessoas, a exemplo de iniciativas como: Grasshopper<sup>2</sup>, Code.org<sup>3</sup>, Exercism<sup>4</sup>, Codecademy<sup>5</sup>, freeCodeCamp<sup>6</sup>, no ambiente escolar brasileiro ainda há uma carência enorme de ações que introduzam as habilidades da Ciência da Computação no processo de resolução de problemas em conjunto com as demais disciplinas do currículo escolar.

A falta de processos de ensino e de aprendizagem de Computação no Ensino Fundamental e Médio acaba por refletir na dificuldade que os alunos ingressantes do Ensino Técnico ou Superior, na área de computação, apresentam ao se depararem com o ensino de programação. O estudo de linguagens de programação exige um considerável esforço cognitivo e a falta de uma base mínima de conhecimentos computacionais torna esse aprendizado mais complexo. Giraffa e Müller (2017) ressaltam ainda que os alunos possuem lacunas de formação que vão desde a interpretação de enunciados, passando pela falta de competências e habilidades matemáticas, até limitações no uso da memória para lembrar o que já foi aprendido. Sendo essas capacidades requisitos fundamentais para os cursos de Computação, acabam por gerar um elevado índice de reprovação e, por consequência, evasão e desistências. (SILVEIRA *et al.*, 2018).

O processo de ensino e aprendizagem de programação perpassa por diferentes aspectos, desde a maneira como a informação é transmitida pelo educador, passando pelo contexto sociocultural dos educandos, a metodologia utilizada, além de outros fatores humanos e sociais. Ortiz *et al.* (2018) chamam atenção para a importância de se levar em consideração o contexto dos alunos para a construção de uma aprendizagem significativa, no intuito de gerar engajamento e alcançar maior aproveitamento do conteúdo.

Entre as metodologias existentes que objetivam a aprendizagem contextualizada, destacamos os Três Momentos Pedagógicos (3MP) como uma

---

<sup>2</sup> <https://grasshopper.app/>

<sup>3</sup> <https://www.code.org>

<sup>4</sup> <https://exercism.io/>

<sup>5</sup> <https://www.codecademy.com/>

<sup>6</sup> <https://www.freecodecamp.org/>

proposta para o ensino com o objetivo de compreender o mundo em que os estudantes vivem (DELIZOICOV; ANGOTTI; PERNAMBUCO, 2011). Com uma dinâmica baseada na concepção freireana (dialógica e problematizadora) onde os temas a serem trabalhados surgem da realidade dos educandos, os 3MP se apresentam como uma ferramenta pedagógica para a compreensão de conceitos científicos de forma contextualizada.

Em vista disso, adotamos a metodologia dos 3MP neste trabalho com a perspectiva de colaborar com a formação do conhecimento, a partir de uma abordagem não tradicional para o desenvolvimento de competências e habilidades nos educandos. Assim, o problema desta pesquisa é: como a construção de aplicativos móveis por meio da programação baseada em blocos (App Inventor<sup>7</sup>), aliada aos 3MP, proporciona o desenvolvimento do pensamento computacional e colabora com a aprendizagem de programação?

Assim, a pesquisa teve por objetivo desenvolver o pensamento computacional por meio de uma sequência didática pautada nos Três Momentos Pedagógicos (DELIZOICOV; ANGOTTI; PERNAMBUCO, 2011) sobre o ensino de programação de aplicativos móveis com a ferramenta App Inventor.

A importância do ensino de programação está diretamente relacionada às necessidades de uma sociedade em rede (CASTELLS, 2016), na qual boa parte das atividades econômicas e pessoais envolve recursos computacionais, principalmente softwares que possam auxiliar na realização de tarefas rotineiras, gerenciamento de recursos e que forneçam informações estratégicas. O atendimento dessas demandas passa pela formação de bons profissionais de computação, começando pelo desenvolvimento do raciocínio lógico e crítico, preferencialmente desde a Educação Básica.

A resolução de problemas é uma das habilidades a serem desenvolvidas a partir da aprendizagem de programação. Por meio da programação o aluno amplia a capacidade de organizar e memorizar ideias, desenvolvendo o PC na solução de problemas reais. O conjunto de saberes necessários ao ato de programar associados à lógica de programação caracteriza o chamado pensamento computacional (ARAÚJO *et al.*, 2015). Contudo, para o desenvolvimento do PC, a programação é apenas uma alternativa dentre várias outras, como: computação

---

<sup>7</sup> Mais informações em: <https://appinventor.mit.edu>

desplugada, robótica educacional, modelagem e resolução de problemas.

As dificuldades enfrentadas pelos discentes em desenvolver o pensamento computacional e, conseqüentemente, as habilidades necessárias para programar, acaba por afetá-los, levando à desmotivação, reprovação e, em alguns casos, a evasão dos cursos de computação. Bosse e Gerosa (2015), em um estudo a respeito de reprovações e trancamentos em disciplinas de introdução à programação, demonstram que o índice de reprovação é de 30%, ultrapassando 50% em alguns casos, sendo que 25% dos alunos aprovados fizeram duas ou mais vezes a disciplina. Nessa perspectiva, Hoed e Ladeira (2016) destacam evidências de que a reprovação em disciplinas iniciais do curso relacionadas a algoritmos aumenta a probabilidade do aluno evadir, seja pelo fato da disciplina ser pré-requisito ou por ser fundamental para o sucesso em alguma outra disciplina relacionada a paradigmas e técnicas de programação. Krzyzanowski *et al.* (2019), por sua vez, identificaram que “em média, 62,5% dos alunos que reprovaram na primeira disciplina de Algoritmos e Programação por nota e 70% dos que reprovaram por frequência acabaram trancando o curso” (KRZYZANOWSKI *et al.*, 2019, p. 27). Ao mesmo tempo, metade daqueles que conseguem aprovação, mantêm-se matriculados no curso. Tais resultados confirmam que possivelmente as disciplinas introdutórias são determinantes na permanência dos estudantes.

A partir da consciência do desafio que envolve o processo de aprendizagem dos conceitos de lógica de programação nos cursos da área de computação, abrem-se portas para que professores e alunos desenvolvam e apliquem diferentes métodos de ensino. Giraffa e Müller (2017) apontam o não entendimento de enunciados, exercícios dissociados de problemas reais e pouca disponibilidade para estudos extraclasse entre os fatores para o abandono e desistência nas disciplinas iniciais nos cursos de programação. Para Souza *et al.* (2018), as dificuldades no aprendizado de lógica podem ser provenientes de vários fatores, inclusive das metodologias de ensino tradicionais onde os conteúdos são apresentados apenas de forma expositiva aos alunos. Porém, aprender a programar requer mais do que a simples assimilação de conteúdo, é necessário compreender um problema, pensar nas diferentes maneiras de solucioná-lo e ser capaz de representar a solução computacionalmente por meio de uma linguagem, o que nos leva a pensar sobre a importância de se propor e avaliar outras metodologias para o ensino dessa disciplina.

No escopo deste estudo, investigamos os cursos superiores de Tecnologia em Sistemas para Internet (TSI) e Licenciatura em Computação (LC), do eixo de Informação e Comunicação dos *campi* Santo Augusto e Santo Ângelo do Instituto Federal Farroupilha, onde, logo no início dos cursos, são ministradas disciplinas voltadas à programação, como Algoritmos e Lógica. A partir desses componentes curriculares muitos alunos apresentam dificuldades de aprendizagem, possivelmente por não possuírem um aporte teórico adequado ou ainda não terem se familiarizado com a dialética computacional, uma vez que poucas escolas de Educação Básica trabalham o pensamento computacional em seu itinerário formativo.

As dificuldades enfrentadas pelos alunos e relatadas por docentes dos cursos superiores do eixo de Informação e Comunicação podem ser confirmadas a partir da análise de desempenho dos discentes da disciplina de Algoritmos do curso de Tecnologia em Sistemas para Internet. O curso oferta 35 vagas em turno noturno, sempre no primeiro semestre de cada ano, tendo em vista a periodicidade anual de oferta dos cursos superiores de graduação. Durante quatro anos, 2015 à 2018, o índice médio de reprovação por nota é superior à 22% dos alunos, enquanto a reprovação por faltas atinge quase 26%, chegando a um índice médio total de 48% de discentes reprovados durante esse período, conforme é possível visualizar na Tabela 1.

**Tabela 1** – Reprovação de alunos na disciplina de Algoritmos: 2015 à 2018.

Ano	Nº de alunos da disciplina	Nº de reprovados por nota	Índice de reprovação por nota	Nº de reprovados por falta	Índice de reprovação por falta	Índice total de reprovação
2015	24	2	8,3%	4	16,7%	25,0%
2016	33	10	30,3%	8	24,2%	54,5%
2017	35	8	22,9%	9	25,7%	48,6%
2018	27	8	29,6%	10	37,0%	66,7%
<b>Índice médio total de reprovação</b>						<b>48,7%</b>

Fonte: Elaborada pelo autor com dados obtidos junto à Coordenação de Registros Acadêmicos do Campus Santo Ângelo, 2019.

No intuito de auxiliar na mitigação desse problema, o que se apresenta como a **justificativa** deste trabalho, pretende-se compreender como a tecnologia móvel, especificamente a construção de aplicativos, pode auxiliar no ensino de programação na Educação Profissional e Tecnológica e no desenvolvimento do pensamento computacional.

Além dos conhecimentos técnicos, que já são complexos, a formação de um profissional na Rede Federal de EPT também perpassa pela compreensão de diferentes aspectos sociais e da realidade em que os indivíduos estão inseridos. O entendimento da importância de uma formação integral como forma de superação da precarização das relações de trabalho e das desigualdades sociais é um dos norteadores da formação nos Institutos Federais. O conhecimento tecnológico pode representar uma alternativa de quebra desse paradigma, pois, como retrata Ricardo Antunes:

[...] quanto mais se avança na competição intercapitalista, quanto mais se desenvolve a tecnologia concorrencial em uma dada região ou conjunto de países, [...] maior é a desmontagem e a desestruturação daqueles que estão subordinados ou mesmo excluídos desse processo, ou ainda que não conseguem acompanhá-lo, quer pela ausência de base interna sólida, [...] quer porque não conseguem acompanhar a intensidade do ritmo tecnológico hoje vivenciado (ANTUNES, 2009, p. 35).

Nesse sentido, uma formação sólida, ampla e integral, pautada na contextualização, interdisciplinaridade e no compromisso com a transformação social pode completar as lacunas de formação dos educandos e dotá-los de habilidades e aptidões que viabilizem sua inserção no mundo do trabalho, de forma consistente e criativa, sendo capazes de desempenhar atividades que busquem o rompimento do status quo, não apenas reproduzindo o sistema capitalista atual, mas o modificando (INSTITUTO FEDERAL FARROUPILHA, 2014).

Dentro desse contexto, a proposta de pesquisa compreendeu desenvolver uma Sequência Didática, pautada nos Três Momentos Pedagógicos, no intuito de auxiliar no aprendizado de programação e na formação do PC. A proposta visou estimular o processo de ensino-aprendizagem através do desenvolvimento de aplicativos para dispositivos móveis, fazendo com que os alunos se sintam envolvidos e comprometidos com o processo de aquisição do conhecimento por meio do diálogo e da reflexão.

A metodologia para a construção da Sequência Didática partiu de um estudo de caso com alunos do primeiro semestre do curso superior de Licenciatura em Computação. Por meio de encontros online, foram desenvolvidos aplicativos em uma linguagem de programação visual em blocos, com posterior análise dos artefatos de código numa ferramenta web para avaliação do pensamento computacional. Também foram utilizados questionários para o entendimento do perfil dos alunos e suas percepções quanto às atividades desenvolvidas, além de diário de campo para

registro das percepções do pesquisador.

Portanto, esse trabalho procurou contribuir no ensino de programação aos discentes ingressantes, especialmente conhecimentos relacionados ao pensamento computacional e ao desenvolvimento de aplicativos móveis, e socialmente pelo incentivo à formação integral orientada pela práxis, incentivando a auto-organização dos estudantes, a autonomia, o trabalho coletivo e o pensamento crítico-reflexivo, de forma a perceber a tecnologia como um constructo humano e que precisa ser desenvolvida a partir das necessidades locais de cada instituição.

Nesse sentido, a pesquisa teve por **objetivo geral** identificar as contribuições e/ou limites da construção de aplicativos móveis com o App Inventor, por meio de uma sequência didática pautada nos 3MP, para a aprendizagem de programação, bem como para o desenvolvimento do pensamento computacional. Quanto aos **objetivos específicos**, se pretendeu analisar estudos sobre o desenvolvimento do pensamento computacional e as relações com a formação integrada; identificar as dificuldades apresentadas pelos alunos no ensino de programação; desenvolver uma sequência didática com enfoque na programação para dispositivos móveis pautada nos Três Momentos Pedagógicos e identificar as contribuições e limites no aprendizado de algoritmos; analisar as possíveis contribuições da tecnologia móvel a partir da construção de aplicativos para o aprendizado de programação e o desenvolvimento do pensamento computacional.

O presente trabalho se estrutura a partir do referencial teórico com o apanhado das múltiplas definições do pensamento computacional e sua conexão com a formação integral praticada na Rede Federal de Educação Profissional e Tecnológica. Também é apresentada a metodologia adotada na pesquisa, destacando o processo de avaliação do PC, e o produto educacional que se constitui numa sequência didática. Por fim, são expostos os resultados do estudo com relação a proposta de trabalho adotada e as considerações finais. Além disso, são apresentados três apêndices: o primeiro (Apêndice A) com o Produto Educacional, o segundo (Apêndice B) com os questionários respondidos pelos sujeitos do estudo; o terceiro (Apêndice C) com o Termo de Consentimento Livre e Esclarecido (TCLE), a fim de que os participantes concordassem em participar da pesquisa.

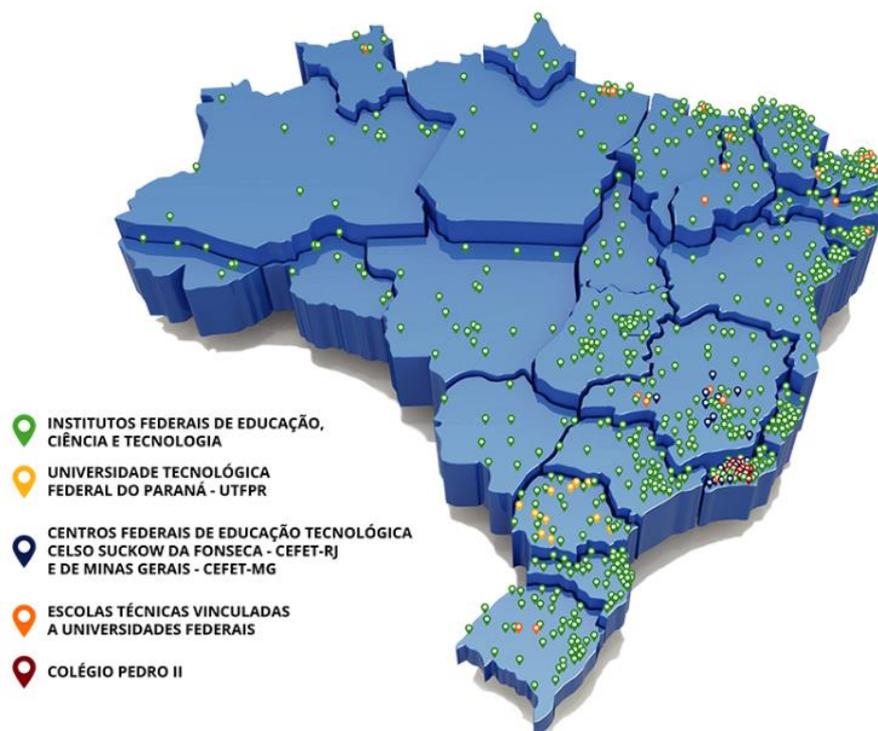
## **2. FORMAÇÃO INTEGRADA, PENSAMENTO COMPUTACIONAL E INTERAÇÃO PEDAGÓGICA**

Neste capítulo são discutidos aspectos relativos à formação integrada num curso de Licenciatura em Computação, na possibilidade da efetivação de uma educação para o mundo do trabalho, numa perspectiva politécnica e omnilateral. A revisão bibliográfica aborda as bases da formação integral nos Institutos Federais e os conceitos relativos ao pensamento computacional, com vistas a problematizar o ensino de programação em cursos de computação. Assim, apresentam-se indícios de que o PC, como habilidade capaz de auxiliar na compreensão e resolução de problemas, se aproxima das concepções de formação integral.

### **2.1 A formação integrada nos Institutos Federais**

Planejar o presente e projetar o futuro da educação profissional passa, necessariamente, pela preservação e resgate dos caminhos percorridos. Na história da Educação Profissional e Tecnológica (EPT), se destaca o período compreendido após 2003 em que ocorreram dois momentos de expansão importantes da educação pública: 1) até 2010, relativo a fase II do plano de expansão da Rede Federal de EPT, em que o número total de campi de Institutos Federais passou de 140 para 354 e o número de municípios atendidos de 120 para 321; 2) até 2014, correspondente a fase III, constituíram a rede de EPT um total de 562 campi com atendimento de 512 municípios brasileiros (BRASIL, 2014). Isso demonstra a recente popularização e democratização da EPT no Brasil, que atingiu, em 2019, 661 unidades compostas por 38 Institutos Federais, 02 Centros Federais de Educação Tecnológica (Cefet), a Universidade Tecnológica Federal do Paraná (UTFPR), 22 escolas técnicas vinculadas às universidades federais e o Colégio Pedro II, conforme a Figura 1.

**Figura 1** – Distribuição das unidades que compõem a Rede Federal de Educação Profissional, Científica e Tecnológica



Fonte: BRASIL, 2019.

Conforme Pacheco (2015), a partir de 2003, foram implementadas políticas que, em partes, se contrapuseram às concepções neoliberais e que possibilitaram acesso à educação e permanência para milhões de jovens e adultos da classe trabalhadora. Através da Lei nº 11.195, de 18 de novembro de 2005, foi revogada a proibição de criação de novas unidades de ensino profissional federais. A interiorização e a ampliação do acesso à Educação Profissional e Tecnológica culminaram com a promulgação da Lei nº 11.892, de 29 de dezembro de 2008, que criou os Institutos Federais de Educação Ciência e Tecnologia (IFs). Trata-se de instituições novas no Brasil que atuam em Educação de Jovens e Adultos, Educação Profissional e Tecnológica, Licenciaturas, Bacharelados e Pós-Graduação, constituindo-se como principal identidade dos IFs o currículo integrado.

É importante salientar que essa expansão esteve acompanhada de outros cenários conquistados por uma política nacional de distribuição de renda, geração de empregos e elevação da renda média do trabalhador. Vinculado a esse ambiente houve a consequente necessidade de mão de obra qualificada para os novos postos de trabalho. Assim, a Rede Federal de EPT surgiu com a intenção de articular trabalho, ciência e cultura na perspectiva da emancipação humana, derrubando as

barreiras entre ensino técnico e científico. A diversidade de cursos ofertados e a atuação junto à população e empresas locais, com intuito de potencializar cada região em termos de trabalho, cultura e lazer, marcaram a nova fase de ensino público, gratuito e de excelência introduzida pelos IFs.

O currículo integrado traz a concepção de um Ensino Médio Integrado ao Ensino Técnico buscando superar o histórico conflito em torno do papel da escola, de formar para a cidadania ou para o trabalho produtivo. Na Licenciatura em Computação, também entendemos como relevante o currículo integrado, pois é constituída por diferentes disciplinas que precisam ser articuladas para a constituição do professor de Computação, visando uma transformação no processo de uso e produção tecnológica. Segundo Cambraia e Zanon (2018), nas licenciaturas se percebe uma cisão entre quem pensa e quem faz o currículo, o que exige o desenvolvimento de um processo formativo crítico e integral, pois não se trata de dois cursos diferentes, mas sim, de uma demanda de integração curricular que proporciona essa formação omnilateral<sup>8</sup>. A elaboração de uma proposta curricular com esse viés busca, segundo Ciavatta (2005, p. 85):

garantir ao adolescente, ao jovem e ao adulto trabalhador o direito a uma formação completa para a leitura do mundo e para a atuação como cidadão pertencente a um país, integrado dignamente à sua sociedade política. Formação que, neste sentido, supõe compreensão das relações sociais subjacentes a todos os fenômenos.

Historicamente, a Educação Profissional e Tecnológica caminha alinhada ao ideal de uma sociedade mais justa, pautada na igualdade política, econômica e social, e ligada ao mundo do trabalho com vistas a superação das desigualdades existentes entre o trabalhador assalariado que produz e o proprietário dos meios de produção. Esses pressupostos se justificam, na visão de Pacheco (2015, p. 10), na medida em que “não basta incluir em uma sociedade desigual [...]. O conceito de inclusão tem de estar vinculado ao de emancipação, quando se constroem também os princípios básicos da cidadania como consciência, organização e mobilização”. Em outras palavras, o sucesso escolar perpassa pelo ambiente de sala de aula e envolve a comunidade, fazendo da educação um instrumento a serviço da

---

<sup>8</sup> A formação omnilateral, defendida pelos clássicos do marxismo (Marx, Engels e Gramsci), destaca o trabalho enquanto princípio educativo, superando a oposição entre formação intelectual e formação técnica. Apresenta-se como preceito básico para a emancipação humana a partir de uma formação integral pautada na igualdade de oportunidades, na autonomia, no pensamento crítico e na consciência da realidade, aliando a instrução e a atividade prática com vistas a superar o caráter alienante do trabalho na sociedade capitalista.

democracia e inclusão.

Nesse sentido, a formação integrada, numa perspectiva politécnica, é fundamental para promover a emancipação dos setores excluídos da sociedade, formando cidadãos para o mundo do trabalho e ressaltando o protagonismo daqueles que fazem educação.

A indissociabilidade entre ensino, pesquisa e extensão, uma das características do currículo integrado, reforça a busca por metodologias que aproximem ciência e tecnologia como forma de romper com o modelo da racionalidade técnica, de conhecimento fragmentado que ainda separa teoria da prática. A pesquisa como princípio educativo e científico e a extensão como aproximação da comunidade demonstram essa intencionalidade de superação (PACHECO, 2015).

Com a formação integrada pode-se remeter ao sentido de completude, que trata a educação como uma totalidade social. Segundo Ciavatta (2005, p. 84),

[...] queremos que a educação geral se torne parte inseparável da educação profissional em todos os campos onde se dá a preparação para o trabalho: seja nos processos produtivos, seja nos processos educativos como a formação inicial, como o ensino técnico, tecnológico ou superior.

Essa visão que aborda o trabalho como princípio educativo procura incorporar a dimensão intelectual ao trabalho produtivo, passando a formar trabalhadores que possam desempenhar funções de direção e não apenas tarefas manuais. Ademais, objetiva formar cidadãos integrados à sociedade, capazes de pensar, agir e refletir sobre a mesma.

O estímulo para que o aluno tenha senso crítico e reflita a respeito da disciplina e seus conceitos vêm ao encontro de um modelo educacional mais participativo e eficiente. Freire (1996, p. 26) afirma que “o educador democrático não pode negar-se o dever de, na sua prática docente, reforçar a capacidade crítica do educando, sua curiosidade, sua insubmissão.”. O desenvolvimento de uma formação mais reflexiva e crítica passa pelo conhecimento da situação real vivida pelo aluno e se dá através da aproximação da realidade complexa, valorizando a interdisciplinaridade e a articulação entre teoria e prática. A utilização das tecnologias da informação pode contribuir para a aprendizagem que fomenta reflexões críticas e científicas, tendo como base o protagonismo dos alunos na relação destes com o conhecimento (FREIRE, 2006).

A proposta de educação trabalhada nos Institutos Federais refuta a visão de

formação voltada apenas para o mercado, uma vez que procura, a partir do potencial transformador da escola, propiciar um ambiente que amplie a compreensão de mundo, contribuindo para a formação de seres humanos com raciocínio crítico-reflexivo e autonomia intelectual. Segundo Freire:

[...] uma das tarefas mais importantes da prática educativa-crítica é propiciar as condições em que os educandos em suas relações uns com os outros e todos com o professor ou a professora ensaiam a experiência profunda de assumir-se. Assumir-se como ser social e histórico, como ser pensante, comunicante, transformador, criador, realizador de sonhos, capaz de ter raiva porque é capaz de amar. (FREIRE, 1996, p. 46).

Porém, desenvolver a criticidade não é um processo simples e requer uma abordagem educacional capaz de promover a autonomia e ampliar os horizontes dos sujeitos das práticas pedagógicas. O currículo integrado procura garantir o entendimento dos fundamentos que explicam os processos naturais e sociais, ou seja, exige que os conceitos científicos sejam aprendidos nas suas raízes epistemológicas. Para Ciavatta (2005, p. 94), a formação integrada “exige que se busquem os alicerces do pensamento e da produção da vida além das práticas de educação profissional e das teorias da educação propedêutica que treinam para o vestibular”. Com relação à elaboração de uma proposta curricular na perspectiva da formação integrada, Ramos (2005, p. 114) apresenta como finalidade da formação:

possibilitar às pessoas compreenderem a realidade para além de sua aparência fenomênica. Sob essa perspectiva, os conteúdos de ensino não têm fins em si mesmos e nem se limitam a insumos para o desenvolvimento de competências. Os conteúdos de ensino são conceitos e teorias que constituem sínteses da apropriação histórica da realidade material e social do homem.

Nesse sentido, a autora apresenta dois fundamentos filosóficos que fundamentam a organização curricular: a) “a concepção de homem como ser histórico-social que age sobre a natureza para satisfazer suas necessidades e, nessa ação, produz conhecimentos”, e b) “que a realidade concreta é uma totalidade, síntese de múltiplas relações”. Em outras palavras, o processo de apropriação social mediado pelo trabalho produz conhecimento, o que colabora com a compreensão da realidade que o currículo integrado pode proporcionar. O que reforça a integração entre trabalho, ciência, tecnologia e cultura como dimensões indissociáveis.

Essas dimensões também são discutidas por Kuenzer (2007) que ressalta a integração entre ciência, trabalho e cultura como um dos princípios para a organização curricular que pretende atender as necessidades dos que vivem do

trabalho. Inclusive, destaca que essa integração

demanda uma formação científico-tecnológica e sócio-histórica que verdadeiramente integre os conhecimentos científicos que fundamentam os processos sociais e produtivos contemporâneos, as formas de comunicação e os conhecimentos sócio-históricos. Para que tal ocorra, o eixo do currículo deverá ser o trabalho compreendido como práxis humana e como práxis produtiva, a partir do que não há dissociação entre educação geral e formação para o trabalho (KUENZER, 2007, p. 50).

Para isso, a autora chama a atenção para a organização curricular que intenta a educação científico-tecnológica específica, de modo que “atenda às especificidades regionais, viabilizando a movimentação do pensamento do geral para o específico e vice-versa, [...] sem que haja momentos separados no currículo para as partes geral e diversificada” (KUENZER, 2007, p. 60). Assim, o trabalho na Licenciatura em Computação precisa superar a ideia de uma formação calcada nas tecnologias de “última geração” apenas e proporcionar, por meio do ensino da computação, uma formação integrada.

A computação, nesse caso, ultrapassa os limites da simples aplicação técnica e amplia-se ao proporcionar uma formação que traz novas possibilidades de compreensão, interpretação, explicação e transformação do mundo. Assim, o desenvolvimento do PC por meio do ensino de programação não tem a preocupação de apenas formar para o exercício do trabalho na expectativa de atender um mercado tecnicista e instrumental, mas para entender o trabalho como princípio educativo onde o ser humano é produtor da sua própria realidade, sendo capaz de transformá-la. Outrossim, formar um trabalhador que atue intelectualmente de maneira crítica, criativa e autônoma, capaz de acompanhar as mudanças e se educar permanentemente.

O PC se apresenta como uma proposta para contribuir com a formulação e resolução de problemas e o pensamento crítico. A possibilidade de desenvolver habilidades essenciais para a vida, independentemente da área em que o estudante irá atuar, com foco em criar soluções para questões sem resoluções pré-definidas, aproxima o PC das concepções de formação integral. As modificações nas formas de ensinar e aprender exigem do ambiente escolar a articulação do pensamento e ação, teoria e prática, contemplando o universo da ciência, das tecnologias e as relações culturais e sócio-históricas.

## 2.2 Três Momentos Pedagógicos como organizadores do processo didático

Para contribuir com uma formação integrada, propõe-se a utilização dos Três Momentos Pedagógicos (3MP) enquanto ferramenta educativa para organizar o processo didático e orientar o planejamento da sequência didática. Para Zabala (1998, p. 18), as SD's são "(...) um conjunto de atividades ordenadas, estruturadas e articuladas para a realização de certos objetivos educacionais, que têm um princípio e um fim conhecidos tanto pelos professores como pelos alunos". A sequência didática tem o papel de indicar a função que cada uma das atividades possui, procurando apresentar um caminho possível para o desenvolvimento do pensamento computacional e a aprendizagem de programação a partir da construção de aplicativos móveis, deixando claro a pertinência e a ênfase a ser atribuída a cada conteúdo.

As primeiras abordagens dessa dinâmica foram realizadas por Delizoicov (1982), em sua dissertação de mestrado, onde o autor relata uma prática educacional em escolas do primeiro grau da Guiné Bissau com a primeira adaptação da concepção freireana<sup>9</sup> num contexto de educação formal. Porém, segundo Muenchen e Delizoicov (2012), a dinâmica conhecida como 3MP só passou a ser disseminada na década de 80 a partir dos livros "Metodologia do Ensino de Ciências" e "Física", ambos dos autores Demétrio Delizoicov e José Angotti.

Originalmente, os 3MP foram pensados enquanto estruturantes do contexto das aulas, ou seja, sendo utilizados para estruturar currículos críticos, discutindo o que é ensinado e não apenas como é ensinado. Na concepção inicial, os 3MP eram conhecidos como "roteiro pedagógico" e se dividiam em: Estudo da Realidade (ER), Estudo Científico (EC) e Trabalho Prático (TP). Após a sua utilização em vários projetos, os quais extrapolaram sua concepção inicial, e por diferentes autores, com incorporações no roteiro e mudanças na nomenclatura, surgiu o que, atualmente, conhecemos por Três Momentos Pedagógicos (3MP) (MUNCHEN, 2010).

Nesse sentido, Munchen (2010, p. 156) afirma:

---

<sup>9</sup> Segundo Delizoicov(1983, p. 85), trata-se de uma concepção de educação problematizadora ou dialógica que "é realizada pelo professor com o aluno, e se contrapõe à educação que Paulo Freire chama de "educação bancária", realizada pelo professor sobre o aluno. [...] considera o educando como sujeito da ação educativa, e não como objeto passivo desta, o que implica que a sua participação no processo deve ocorrer em todos os níveis, inclusive na definição conjunta do conteúdo programático".

Ao longo dos anos, os 3MP foram sendo revistos e extrapolaram sua utilização inicial, tornando-se um parâmetro para o processo como um todo, fundamentalmente pelo aspecto dinâmico. Assim, pode-se hoje destacar mais uma utilização não conjecturada inicialmente, que é a elaboração de material didático para cursos de graduação à distância (livro do aluno) e a utilização como estruturadores/organizadores das discussões em eventos, além da proposição e publicação dos livros *Física e Metodologia do Ensino de Ciências*.

Em nosso estudo, todo o processo de elaboração da SD foi guiado pelos Três Momentos Pedagógicos, porém limitamos sua utilização à metodologia de trabalho de sala de aula (estruturação das aulas) numa visão mais reduzida e simples de uma ferramenta didático-pedagógica, articulando os momentos com o intuito de ensinar um determinado conteúdo com a perspectiva de diálogo entre os participantes. Portanto, adotamos a concepção trazida por Muenchen e Delizoicov (2014), que caracterizam os 3MP por:

1) **Problematização inicial (PI)**: por meio de situações reais que os alunos experienciam, desafiá-los a expor o que pensam para entendê-los melhor e, assim, problematizar e criar a necessidade de conhecer o que ainda não conhecem.

Nesse momento os alunos são instigados a participar e apresentar o seu posicionamento, cabendo ao professor o papel de coordenar as discussões e apresentar questionamentos sobre o tema. Espera-se que o aluno sinta a necessidade de buscar novos conhecimentos para elucidar as dúvidas e obter respostas para a problematização. Segundo Delizoicov e Angotti (1994, p. 54):

São apresentadas questões e/ou situações para discussão com os alunos. Sua função, mais do que simples motivação para se introduzir um conteúdo específico, é fazer a ligação desse conteúdo com situações reais que os alunos conhecem e presenciam, para as quais provavelmente eles não dispõem de conhecimentos suficientes para interpretar total ou corretamente.

A problematização, referida no primeiro momento, implica em diálogo, onde existe um problema a ser resolvido que estimula a curiosidade do educando, fazendo com que ele se sinta sujeito do processo, podendo se expressar e tendo a sua opinião considerada pelos demais participantes. Esse processo desvela os conceitos a serem discutidos durante a disciplina.

2) **Organização do conhecimento (OC)**: desenvolvimento de atividades, com a orientação docente, sobre os conhecimentos necessários para entender a problematização inicial. O professor aborda o conteúdo para aprofundamento e conhecimento sobre o tema discutido, gerando a percepção de outras explicações a partir do conhecimento científico.

O objetivo deste momento é ampliar os conhecimentos sobre a temática a partir da leitura e discussão de textos, seminários, exercícios, atividades práticas individuais e em grupos, entre outros, propiciando o desenvolvimento de conceitos e relações para a compreensão do tema. Delizoicov e Angotti (1994, p. 55) apontam que:

O conteúdo programado é preparado em termos instrucionais para que o aluno o aprenda de forma a, de um lado perceber a existência de outras visões e explicações para as situações e fenômenos problematizados, e, de outro, a comparar esse conhecimento com o seu, para usá-lo para melhor interpretar aqueles fenômenos e situações.

**3) Aplicação do conhecimento (AC):** sistematizar o conhecimento elaborado pelo aluno, analisando e interpretando as situações iniciais e outras que possam ser compreendidas pelo mesmo conhecimento. O conhecimento científico é utilizado para reinterpretar as situações problematizadoras, ou seja, o aluno deve ser capaz de empregar o conhecimento em situações reais ou outras não diretamente ligadas ao motivo inicial.

Neste último momento, as problematizações podem ser retomadas para verificar se os conhecimentos da OC (segundo MP) foram de fato assimilados pelos alunos. Delizoicov (1982, p. 150) afirma que “podemos também ampliar o quadro das informações adquiridas ou ainda abranger conteúdo distinto da situação original”. Em vista disso, aqui nos referimos a este momento como de “**Recontextualização do Conhecimento (RC)**”, por considerarmos que o termo “aplicação” deixa margem ao risco de incorrer na racionalidade técnica, num caráter positivista e simplista.

Este momento marca a prática formativa integrada na perspectiva da educação como ato ético-político, numa intersubjetividade que, como refere Freire (2011, p. 119), “se faça, não para levar-lhes uma mensagem “salvadora”, em forma de conteúdo a ser depositado, mas, para, em diálogo, conhecer, não só a objetividade em que estão, mas a consciência que tenham desta objetividade”, com o reconhecimento “dos vários níveis de percepção de si mesmos e do mundo em que e com que estão”.

Freire (2011), ao discutir a noção de educação problematizadora, contribui no entendimento da potencialidade da formação integrada articulada com os momentos pedagógicos para favorecer relações entre sujeitos que concebem a realidade como processo em constante transformação, de forma crítica e reconstrutiva, apontando

que: “somente o diálogo que implica um pensar crítico é capaz, também, de gerá-lo”, como dimensão constitutiva do humano, pois sem a relação de diálogo “não há comunicação e sem esta não há verdadeira educação” (p. 115).

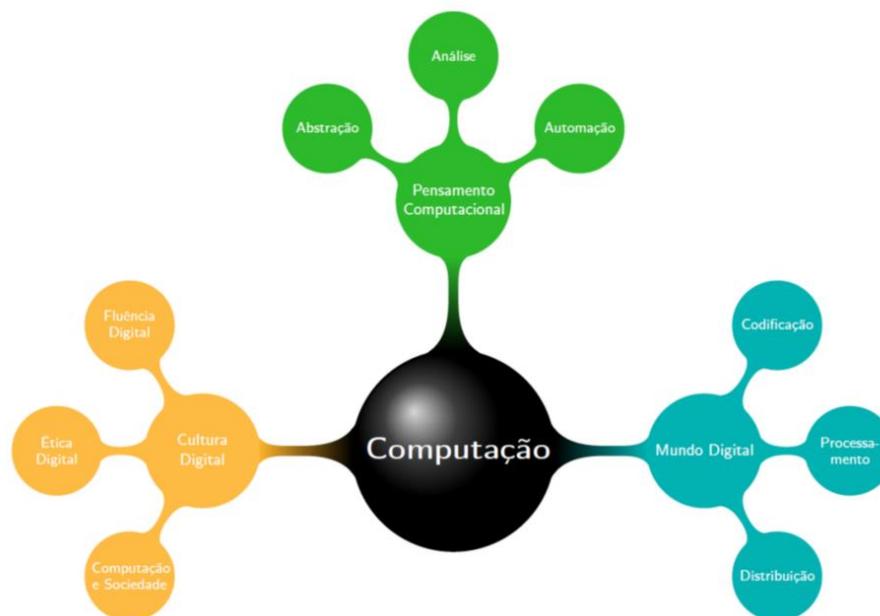
Sendo assim, os três momentos pedagógicos relacionam-se diretamente com o currículo integrado, pois não reduzem a construção do conhecimento ao treinamento para a execução de determinadas tarefas. Pelo contrário, visam superar a divisão do ser humano entre o que pensa e aquele que faz, produzida pela divisão social do trabalho. Buscam sim, a formação humana omnilateral que procura desenvolver o pensamento crítico e a autonomia intelectual. Além disso, propiciar a compreensão do mundo do trabalho como dimensão histórica e ontológica, promovendo o desenvolvimento humano em sua integralidade, em substituição à unilateralidade, e a emancipação dos sujeitos.

A importância da utilização dos 3MP para relacionar o conteúdo abordado na sequência didática com a realidade dos alunos, problematizando aspectos socioculturais, além dos técnicos e de aprendizagem, tem a intencionalidade de alcançar maior engajamento e aproveitamento do conteúdo. Ortiz *et al.* (2018), indicam em seu mapeamento a respeito de iniciativas que objetivam ensinar o pensamento computacional, que de 46 pesquisas analisadas, apenas 5 exploram o contexto sociocultural, sendo identificado como um dos pontos a serem melhorados nas pesquisas que envolvem o PC.

### **2.3 Pensamento computacional como alternativa de formação integrada**

Nos últimos anos, ocorreu uma transformação na sociedade onde o fluxo de informações cresceu consideravelmente. Pierre Levy (1999) afirma que vivemos um segundo dilúvio, entretanto hoje de informações. Com isso, exige-se novas habilidades cognitivas para navegar e sobreviver no mar de informações. Para acompanhar a evolução, a sociedade precisa se apoderar dos novos instrumentos culturais da computação, procurando ser crítica e criativa no uso da tecnologia e promovendo uma apropriação tecnológica no lugar de apenas ampliar o fluxo de informações. Nesse sentido, a Sociedade Brasileira de Computação (SBC) organiza os conhecimentos da área da computação em 3 eixos: Pensamento Computacional, Mundo Digital e Cultura Digital, conforme a Figura 2.

**Figura 2 – Eixos dos conhecimentos da área de Computação**



Fonte: SBC, 2018.

Esse tripé que define os conhecimentos fundamentais de computação para qualquer cidadão no mundo de hoje e do futuro, traz o “Mundo Digital” como o estudo das máquinas computacionais, a “Cultura Digital” que analisa os impactos da computação na sociedade e a importância do “Pensamento Computacional” para a formação de habilidades capazes de apoiar a ciência e suas áreas de conhecimento. Apesar do pensamento computacional se tratar de uma temática relativamente nova e em expansão, Wing (2006) o considera, junto com a leitura, a escrita e a aritmética, um dos pilares fundamentais do intelecto humano por descrever, explicar e modelar o universo e seus processos complexos. Nesse sentido, a Sociedade Brasileira de Computação vem estudando formas de introduzir o PC na matriz curricular da educação básica brasileira, a exemplo do que já ocorre em diversos outros países do mundo.

As primeiras ideias relativas ao pensamento computacional foram introduzidas por Seymour Papert ainda na década de 70 (PAPERT; SOLOMON, 1971), mesmo que sem a utilização do termo propriamente dito, ao desenvolver o construcionismo como uma abordagem que defendia o uso do computador enquanto uma ferramenta capaz de estimular o desenvolvimento do raciocínio do estudante na solução de problemas e para construir o seu próprio conhecimento. Em suas produções intelectuais, Papert, conhecido como pai da informática educativa, defende a importância do aprendizado a partir do fazer, da prática, ressaltando a importância

da habilidade de aprender a resolver problemas, e a partir disso, proporcionando também o desenvolvimento intelectual dos sujeitos. A base teórica e filosófica dos trabalhos de Papert foi muito influenciada pelos estudos do filósofo suíço Jean Piaget no que diz respeito a aprendizagem espontânea e natural a partir da interação do sujeito com o seu ambiente. Segundo Piaget (1990, p. 8), o conhecimento resulta “de interações que se produzem a meio caminho entre sujeito e objeto, e que dependem, portanto, dos dois ao mesmo tempo”. No entanto, a perspectiva construcionista de Papert extrapola a compreensão teórica de Piaget e traz objetivos educacionais nas interações do sujeito com os objetos e o mundo exterior (PAPERT, 1986).

A primeira citação efetiva do termo pensamento computacional se deu no livro “Mindstorms: children, computers and powerfull ideas” (PAPERT, 1980, p. 182), traduzido para português como “LOGO: Computadores e Educação” (PAPERT, 1986), enquanto Papert relata experimentos desenvolvidos em LOGO<sup>10</sup>, uma linguagem interpretativa planejada para garantir acesso à programação de computadores para estudantes sem conhecimentos matemáticos prévios. Porém, há época não foi dada a devida atenção ao termo e, tampouco, Papert ocupou-se em aprofundá-lo. O autor defendia o uso do computador de forma humanizada, enquanto ferramenta para desenvolver habilidades mentais das crianças. Nesse sentido, idealizava o ensino de programação na educação básica, afirmando que a criança “deve programar o computador, e ao fazê-lo, ela adquire um sentimento de domínio [...] e estabelece um contato íntimo com algumas das ideias mais profundas da ciência, da matemática e da arte de construir modelos intelectuais” (PAPERT, 1986, p. 17).

A popularização do termo “pensamento computacional” se deu a partir de um artigo publicado por Wing (2006), conceituando o PC como uma “distinta forma de pensamentos com conceitos básicos da Ciência da Computação para resolver problemas, desenvolver sistemas e para entender o comportamento humano, habilidade fundamental para todos”. Traz ainda que o pensamento computacional é “reformular um problema aparentemente difícil em um problema que sabemos como

---

<sup>10</sup> Seymour Papert também se refere a LOGO como uma filosofia de educação composta por uma família de linguagens de computação desenvolvidas por um grupo de pesquisadores do Laboratório de Inteligência Artificial do MIT (Massachusetts Institute of Technology, Cambridge/EUA), coordenado por ele e Marvin Minsky (PAPERT, 1986).

resolver, talvez por redução, incorporação, transformação ou simulação” (WING, 2006, p.33).

Em trabalhos seguintes, a autora refina a conceituação do termo, aproximando e complementando a forma de pensar baseada na Matemática e na Engenharia. Destaca também os processos de pensamento envolvidos na formulação de um problema e na expressão de soluções que possam ser executadas por um humano ou por uma máquina. Ainda complementa como sendo um conjunto de habilidades intelectuais e de raciocínio que indica como as pessoas interagem e aprendem a pensar (WING, 2008; WING 2014).

Outra definição de pensamento computacional é apresentada por Aho (2011) como sendo “o processo de pensamento envolvido na formulação de problemas, para que suas soluções possam ser representadas como etapas e algoritmos computacionais”. O autor também chama atenção para a importância de encontrar modelos computacionais apropriados para formular o problema e derivar suas soluções. Blikstein (2008), por sua vez, afirma que pensamento computacional é “saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano – em outras palavras, usar computadores, e redes de computadores, para aumentar nossa produtividade, inventividade, e criatividade”.

Segundo Ribeiro, Foss e Cavalheiro (2019), o pensamento computacional se aproxima do raciocínio lógico, podendo ser visto como uma generalização dele. Porém, enquanto o raciocínio lógico objetiva basicamente encontrar verdades, o PC, por sua vez, consiste num processo de transformação de entradas em saídas que não são necessariamente sentenças verdadeiras, mas qualquer coisa, se utilizando de regras ou instruções quaisquer bem definidas. Essa sequência de regras, comumente chamada de algoritmo, é o produto do raciocínio computacional e define o processo de transformação para resolver um determinado problema.

Para a Sociedade Brasileira de Computação, o pensamento computacional pode ser definido como a “habilidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática” (SBC, 2018. p. 2). Essas capacidades são necessárias para a construção de algoritmos que descrevem um processo capaz de resolver determinado problema. Um entendimento idêntico é trazido na Base Nacional Comum Curricular (BNCC), que enfatiza a solução de problemas por meio do

desenvolvimento de algoritmos como uma aprendizagem essencial do ensino fundamental para o ensino médio (BRASIL, 2018).

Uma definição mais ampla de pensamento computacional é apresentada por Brackmann (2017. p. 29) como:

uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas colaborativamente através de passos claros de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente.

Assim como abordado por Wing, essa visão ressalta a possibilidade da solução de problemas por pessoas ou máquinas, o que remete às duas modalidades de pensamento computacional mais difundidas: plugado e desplugado. Enquanto a primeira se faz valer de equipamentos e softwares específicos, como ferramentas de programação visual em blocos, a segunda pode ser ensinada sem o uso de computadores, a partir de objetos tangíveis como cartões e jogos de tabuleiro (BRACKMANN, 2017).

Cabe ressaltar que o PC não se limita a programação, sua abordagem é muito mais ampla, constituindo-se de um conjunto de processos mentais comumente utilizados por profissionais de computação com o propósito de solucionar problemas (SILVA, 2020). Mesmo em se tratando da modalidade plugada, onde a interação ocorre a partir de um ambiente computacional, não se pode reduzir o PC ao ato de codificar. A escrita de instruções computacionais, nesse caso, é precedida, no mínimo, pela compreensão do problema, sua divisão em partes menores e pelo processo de abstração que consiste em concentrar esforços nas características essenciais para a elaboração da solução, desconsiderando informações não relevantes, conforme melhor explicado nas seções seguintes deste trabalho.

Em contrapartida, a aprendizagem de programação pode contribuir com o desenvolvimento do pensamento computacional, posto que o estudo de conceitos de programação como: variáveis, operadores, estruturas condicionais e de repetição, entre outros, fomentam o raciocínio lógico, apoiam a compreensão do PC e, conseqüentemente, a sua aplicabilidade a partir da concepção de soluções computacionais para resolver problemas reais. Conforme Da Silva, Kurtz e Santos

(2020a, p. 12, tradução nossa<sup>11</sup>), a programação

permeia todas as áreas do conhecimento e orienta o desenvolvimento de diversas habilidades cognitivas do aluno, como: criatividade, raciocínio lógico, resolução de problemas, criticidade, reflexão, interpretação, concentração, pensamento algorítmico e computacional, entre outros.

Como o pensamento computacional apresenta distintas abordagens e diferentes propostas de implementação, pode-se dizer que o termo não está totalmente estabelecido, portanto ainda é um conceito em aberto e em constante evolução. Embora se encontrem várias definições para o PC, além das apresentadas nesta pesquisa, existe uma visão convergente de que sua utilização enquanto ato de pensar como um cientista da computação pode auxiliar na solução de problemas para diferentes situações e áreas. Portanto, o pensamento computacional não tem a intenção de fazer com que as pessoas pensem como computadores, mas sim que se utilizem dos fundamentos e recursos da computação, aliados à sua inteligência, para abordar (formular e resolver) problemas. As propostas de soluções, no caso os algoritmos, podem ser executadas por seres humanos, máquinas ou ambos combinados.

Diante dessa confluência de conceitos, nesta pesquisa partimos do princípio de que as habilidades e competências concernentes ao pensamento computacional podem ser estimuladas e desenvolvidas, auxiliando no entendimento de problemas reais e na elaboração de soluções computacionais ou não, sejam elas concretas ou abstratas. Desse modo, o PC se apresenta como uma capacidade que pode ser assimilada através de um ensino significativo, pautado em situações vivenciadas pelo educando, contribuindo com uma formação integrada.

O entendimento da importância de uma formação integral como forma de superação da precarização das relações de trabalho e das desigualdades sociais é um dos norteadores da formação dos Institutos Federais, como já explicado no capítulo anterior. Segundo Pacheco (2015, p. 11), o

objetivo central não é formar um profissional para o mercado, mas sim um cidadão para o mundo do trabalho – um cidadão que tanto poderia ser um técnico quanto um filósofo, um escritor. Significa superar o preconceito de classe de que um trabalhador não pode ser um intelectual, um artista.

O pensamento computacional pode em muito contribuir para essa formação integrada, possibilitando o desenvolvimento de conhecimentos que gerem novas

---

<sup>11</sup> “[...] programming permeates all areas of knowledge and guides the development of various cognitive skills of the student, to mention: creativity, logical reasoning, problem solving, criticality, reflection, interpretation, concentration, algorithmic and computational thinking, among others”.

oportunidades para a inclusão dos sujeitos, pois o PC não se vincula fundamentalmente a atender as demandas do mercado de trabalho. Trata-se de contribuir com a compreensão e transformação do mundo, por meio da constituição de um olhar crítico às tecnologias apenas como meio de transmissão de informações, visando a produção de conhecimentos que possibilitem o desenvolvimento de habilidades constituidoras de um trabalhador reflexivo.

No entanto, existem adversidades quanto à introdução do PC no ambiente escolar. Segundo Cambraia (2016), uma das dificuldades para o desenvolvimento do PC é a falta de um espaço para o professor de computação nas escolas. Algumas iniciativas desenvolvidas em cursos de Licenciatura em Computação procuram superar esse entrave e contribuir para a formação de professores e o desenvolvimento do PC nas escolas, com: gincanas, oficinas, clubes de programação e atividades interdisciplinares.

Para que esse movimento ocorra, não basta uma formação técnica em computação, afinal o saber da área específica de atuação é pressuposto básico, mas mais que isso, requer a compreensão de como esse conhecimento se transforma em ensino. Essa transposição do conhecimento científico para o conhecimento escolar exige que o PC integre o espaço da escola, atuando como ferramenta de construção de ideias e resolução de problemas que envolvem aspectos sociais, culturais e curriculares. Conforme Cambraia (2016), “o pensamento computacional científico é subversivo e refuta o adestramento que privilegia as listas de conteúdo sem significado”. A articulação dos conhecimentos científicos com significados nas comunidades escolares é possível através da aproximação da realidade do estudante, possibilitando que ele interaja, discuta e procure soluções para problemas concretos.

A importância do ensino do PC em cursos de licenciatura também é evidenciada no trabalho de Kurtz e Da Silva (2018), que apresentam a relação entre PC e a formação de professores, abordando as tecnologias como elemento organizador e potencializador da aprendizagem. Ainda, apontam a incorporação do PC em outras disciplinas e temáticas, fortalecendo o sentido da presença das suas habilidades no âmbito educacional e desfazendo a percepção de competências repetitivas, meramente técnicas ou descontextualizadas. Segundo eles, a introdução de conceitos envolvendo o pensamento computacional em programas de formação de professores

[...] é fundamental para o desenvolvimento de habilidades e competências ligadas à vivência destes sujeitos no mundo e, conseqüentemente, à sua atuação profissional no âmbito educacional. Não se trata de uma preparação para o mercado de trabalho (mesmo que seja), mas sim de viabilizar o contato e a experiência de futuros docentes com conceitos que os auxiliem a ampliar sua compreensão e, porque não, suas atitudes em relação às TICs e suas possibilidades educacionais, de modo amplo. (KURTZ; DA SILVA, 2018, p. 28).

Segundo Wing (2014), a própria definição do termo “pensamento computacional” se deu a partir do momento em que os departamentos de Ciência da Computação deixaram de contratar novos professores. Assim, a autora trouxe a discussão sobre o PC para impulsionar o interesse pela área e para incluí-lo entre os princípios e conceitos fundamentais da computação. Esse movimento fez crescer as pesquisas sobre o raciocínio computacional na educação e o importante papel do licenciado em computação nesse contexto (ZORZO *et al.*, 2017). Ainda, promoveu mudanças na grade curricular dos cursos superiores, principalmente nos semestres iniciais onde havia ênfase em programação. A iniciativa também repercutiu no ensino fundamental e médio com a introdução de atividades que objetivam o desenvolvimento do PC. Nos Estados Unidos, a Associação de Professores de Ciência da Computação (CSTA) e a Sociedade Internacional de Tecnologia em Educação (ISTE) são pioneiros na promoção do PC na Educação Básica (K-12). Essas entidades elaboraram um guia, denominado *Computational Thinking Toolkit*, para auxiliar educadores a entender, valorizar e implementar o pensamento computacional (ISTE; CSTA, 2011). O material aborda conceitos inerentes à Ciência da Computação que podem ser aplicados, preferencialmente, por atividades práticas para desenvolver a capacidade cognitiva dos estudantes, entre eles: coleta, análise, representação de dados, decomposição de problemas, abstração, algoritmos, automação, paralelismo e simulação.

Estudos como o de Heintz, Mannila e Färnqvist (2016) e Duncan, Bell e Atras (2017) apontam que diversos países, em sua maioria desenvolvidos, incorporam essa temática desde a educação básica, sendo mais frequentemente obrigatória no Ensino Fundamental e eletiva no Ensino Médio. Também é evidenciada a necessidade de estudos e a manutenção de profissionais com conhecimentos relevantes para desenvolver uma progressão adequada das competências da computação em conjunto com as demais disciplinas que compõem o sistema educacional.

Nesse sentido, uma iniciativa brasileira interessante é apresentada pelo

Centro de Inovação para a Educação Brasileira (CIEB), fornecendo um Currículo de Referência em Tecnologia e Computação com orientações e diretrizes para inclusão da computação nas propostas curriculares da Educação Básica e associando habilidades, inclusive do pensamento computacional, com a Base Nacional Comum Curricular (BNCC). O currículo tem por referência experiências de países que já tem em seus documentos conteúdos de tecnologia e computação e sugere práticas e métodos de avaliação alinhados às habilidades da BNCC, visando auxiliar na implementação das competências gerais, a exemplo da 5ª que propõe:

“Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva” (BRASIL, 2018).

O currículo de referência é organizado por eixos, conforme proposto pela SBC e apresentado abaixo na Figura 3, e traz sugestões de práticas pedagógicas, avaliações e materiais para apoiar os docentes na compreensão e utilização de tecnologias digitais de informação (CIEB,2018).

**Figura 3** – Mandala do Currículo de Referência em Tecnologia e Computação na Educação Básica



Fonte: CIEB, 2018.

Apesar de iniciativas como a do CIEB, da nova BNCC e de simpósios e workshops relacionados à informática na educação promovidos pela SBC, a adoção do pensamento computacional nos currículos nacionais ainda é um desafio. No ano

de 2021, ocorreu o EduComp<sup>12</sup> – Simpósio Brasileiro de Educação em Computação, um evento nacional destinado a discutir essas temáticas, o que demonstra como a comunidade de pesquisadores está privilegiando estudos relacionados à aprendizagem de computação. França e Tedesco (2015) elencam alguns desses desafios, dentre os quais: preparar os futuros licenciados em Computação para que atuem com o pensamento computacional, promover uma formação em outros cursos de licenciatura (Matemática e Biologia, por exemplo) em PC e desenvolver ações educativas envolvendo o PC de modo interdisciplinar. Assim, percebe-se que os desafios também podem se caracterizar como oportunidades à disseminação do pensamento computacional.

Especificamente em relação a BNCC, concordamos com a introdução de conhecimentos da Computação na Educação Básica por meio do pensamento computacional como estratégia de produção do conhecimento e não apenas uso das tecnologias como ferramentas. No entanto, discordamos de alguns aspectos, como por exemplo: os itinerários formativos que desintegram o currículo em nome de uma capacitação para determinadas áreas (CAMBRAIA; KEMP; ZANON, 2021). Um exemplo pode ser observado na proposta do Novo Ensino Médio Paranaense, onde o estudante passa a cursar dois conjuntos de aprendizagens: a Formação Geral Básica (FGB) e os Itinerários Formativos (IF). Este segundo, tem o pensamento computacional como um dos três componentes curriculares a partir da 1ª série do Ensino Médio em 2022, divididos em seis horas-aula, duas para cada componente (PARANÁ, 2022). Apesar da iniciativa positiva de inclusão do pensamento computacional na grade curricular, a fragmentação parece distanciar o PC da sua própria concepção inicial que é de ser trabalhado em conjunto com as demais áreas do conhecimento e não de forma isolada.

No ensino superior, algumas instituições brasileiras já incluíram o pensamento computacional na matriz curricular do primeiro semestre dos cursos de graduação em computação, com é o caso da Universidade Federal Rural de Pernambuco (UFRPE)<sup>13</sup>, a Universidade do Vales dos Sinos (UNISINOS)<sup>14</sup> e a Universidade

---

<sup>12</sup> Mais informações em: <https://www.educompbrasil.org/simposio/2021>.

<sup>13</sup> A UFRPE oferta a disciplina no curso de Licenciatura em Computação.

<sup>14</sup> A Unisinos oferta a disciplina nos cursos de Ciência da Computação e Engenharia da Computação.

Virtual do Estado de São Paulo (UNIVESP)<sup>15</sup>. Todavia, as Diretrizes Curriculares Nacionais para os cursos da área da Computação (MEC, 2016) e os Referenciais de Formação para os cursos de graduação em Computação (ZORZO *et al.*, 2017), documentos que norteiam os currículos e subsidiam a elaboração dos Projetos Pedagógicos de Curso (PPC), não apresentam uma proposta concreta de inclusão do pensamento computacional nos cursos da área.

Da Silva e Falcão (2020b) argumentam que há muito a se debater sobre como integrar o PC aos currículos, em particular de Licenciatura em Computação, a fim de desenvolver as habilidades do PC, formar licenciados para atuar com o PC na educação básica e facilitar a aprendizagem de programação. Porém, em sua pesquisa com estudantes e professores de cursos de LC cuja matriz curricular contém a disciplina de PC, a maioria dos alunos relatou ter se sentido mais confiante ou preparado para cursar programação após o contato com o PC. Apesar da mudança curricular recente que ainda não possibilita avaliar mudanças significativas no desempenho dos estudantes, já aponta perspectivas otimistas quanto às contribuições do PC. Contudo, não basta a introdução de uma disciplina de pensamento computacional, é necessário uma construção dialógica para contextualizar e envolver os alunos nos estudos, a exemplo da utilização dos Três Momentos Pedagógicos, como apresentamos na sequência deste trabalho.

Como abordado por Papert (1986), o processo de aquisição de conhecimentos não pode se limitar à utilização de um software, devendo explorar a criação desse artefato, pensando nas possibilidades que o ensino de programação abre ao afetar a maneira das pessoas pensarem e aprenderem. O autor defende que aprender a comunicar-se com o computador seja um processo natural, a exemplo de aprender uma língua morando em seu país de origem, o que pode influenciar positivamente a maneira como outras aprendizagens acontecem. Essa visão justifica a importância do PC, preferencialmente desde a Educação Básica.

Assim, conforme relatado por Wing (2016), muito se evoluiu para alcançar a visão de que o PC será uma habilidade fundamental usada por todos no século XXI, constituindo uma formação cidadã e que possibilita a constituição de um sujeito crítico em relação às tecnologias. Mas, ainda existem muitos desafios a serem

---

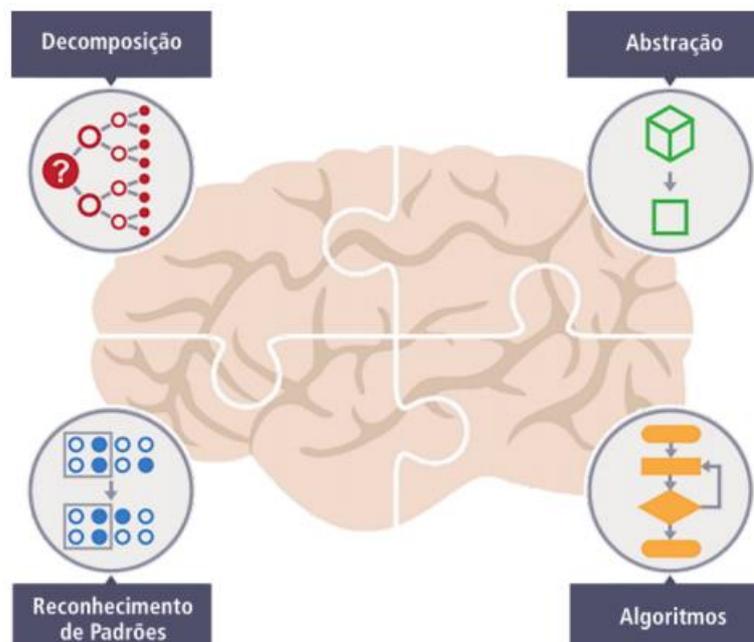
<sup>15</sup> A UNIVESP oferta a disciplina nos cursos de Bacharelado em Tecnologia da Informação e Bacharelado em Ciência de Dados.

superados, sendo um dos principais a falta de professores de ensino básico capacitados para ensinar as habilidades do PC aos alunos.

### 2.3.1 Pilares do pensamento computacional

Ao abordar o pensamento computacional, seja de maneira conceitual ou com intenção prática de solução de problemas, é comum dividi-lo em pilares ou dimensões, ou também chamadas de bases do PC. Porém, assim como na definição do termo, não existe um consenso na literatura quanto aos pilares que compõem o pensamento computacional. Segundo a SBC (2018), o PC envolve abstração, análise de informações e automação de soluções, sendo essas as três habilidades primordiais para a construção de uma solução algorítmica capaz de resolver um problema. No entanto, a maior parte dos estudos relacionados ao tema (LIUKAS 2015; BBC LEARNING, 2015; GROVER e PEA, 2013, BRACKMANN, 2017) apresentam quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos, conforme a Figura 4.

**Figura 4 – Pilares do Pensamento Computacional**



Fonte: Brackmann, 2017.

Uma quinta dimensão denominada “avaliação” é apontada no guia para

professores elaborado pelo Computer at School<sup>16</sup> (CSIZMADIA *et al.*, 2015). Nele, os autores atribuem à avaliação o papel de garantir que a solução seja adequada ao seu propósito, considerando aspectos como corretude, velocidade do algoritmo ou processo, uso econômico de recursos, usabilidade, entre outros. Apesar da relevância dessa dimensão em termos de validação da solução criada, neste estudo optamos por pensar a avaliação enquanto o domínio das demais bases do pensamento computacional e, portanto, não a adotamos como um dos pilares do PC.

Ao identificar um problema a ser resolvido a partir do pensamento computacional, é comum a utilização dos pilares na proposição da formulação da solução. O primeiro passo consiste em dividir o problema em partes menores de mais fácil análise e compreensão (decomposição). Essas partes, ao serem analisadas individualmente, possibilitam a identificação de problemas idênticos que, inclusive, já podem ter sido solucionados anteriormente, garantindo eficiência e rapidez ao processo (reconhecimento de padrões). Durante a análise, informações irrelevantes devem ser ignoradas, mantendo o foco nos detalhes necessários e importantes para a solução (abstração), normalmente composta por um conjunto de regras para resolver cada um dos subproblemas encontrados (algoritmos) (VICARI; MOREIRA; MENEZES, 2018).

### *2.3.1.1 Decomposição*

A decomposição é o processo no qual o problema complexo é dividido em tarefas menores e mais fáceis de serem gerenciadas. A separação em partes menores otimiza a compreensão do problema, o desenvolvimento de uma solução e a avaliação do resultado. Essa abordagem viabiliza a projeção de sistemas grandes e complexos, afinal a tarefa original é decomposta, desenvolvida e integrada posteriormente ao processo. Como exemplo, pode-se imaginar a elaboração de uma receita culinária ou ainda, pensando em termos computacionais, o desenvolvimento de um jogo onde diferentes pessoas criam cenários ou níveis distintos de forma independente, desde que alguns aspectos gerais sejam previamente acordados.

---

<sup>16</sup> Computer at School (CAS) é uma rede de profissionais da Inglaterra composta por professores, acadêmicos, profissionais de TI, entre outros, com a intenção de promover o ensino de informática nas escolas com base no Currículo Nacional de Computação lançado em 2014. Mais informações podem ser obtidas em: <https://www.computingschool.org.uk>.

Finalizadas as partes, os artefatos são integrados gerando um jogo único (CSIZMADIA *et al.*, 2015).

Quebrar as tarefas em fragmentos menores também facilita a identificação de erros. Soluções computacionais escritas a partir de códigos costumam subdividir as tarefas em módulos menores (funções, classes, bibliotecas) para, além de favorecer a compreensão do problema e possibilitar o trabalho em equipe, otimizar o processo de depuração (encontrar e resolver defeitos) (LIUKAS, 2015).

### 2.3.1.2 *Reconhecimento de padrões*

O processo de decompor um problema em partes menores traz clareza sobre a proposta de solução e possibilita a identificação de similaridades. Subproblemas costumam compartilhar elementos idênticos que podem ser explorados a partir de uma solução mais eficiente. Ribeiro, Foss e Cavalheiro (2019, p. 35) entendem que o reconhecimento de padrões, também chamado de generalização, “consiste em identificar padrões de comportamento e/ou dados e construir modelos genéricos que podem ser usados em vários contextos”. Sendo assim, a partir dos padrões é possível resolver novos problemas de forma rápida com base em soluções de problemas anteriores.

Algumas perguntas podem auxiliar no processo de reconhecimento de padrões, como: “isso é semelhante a um problema que resolvi anteriormente?” ou “como isso é diferente?”. Mesmo que a solução anterior não seja totalmente aderente ao novo problema, pode ser adaptada para resolver uma classe de problemas semelhantes, tornando a solução genérica (VICARI, MOREIRA e MENEZES, 2018).

Segundo Brackmann (2017), um exemplo prático do reconhecimento de padrões se dá a partir da identificação de similaridades entre figuras. Ao analisar imagens de pessoas, conforme a Figura 5, é possível observar algumas características comuns (presença de olhos, orelhas, cabelo) e outras distintas (cor e comprimento do cabelo, barba, tom de pele). No pensamento computacional, essas características recebem o nome de Padrões. Dessa forma, pode-se descrever qualquer pessoa simplesmente especificando as características, como: cabelo preto, olhos castanhos, sem barba, pele clara, etc.

**Figura 5** – Similaridade entre figuras (Reconhecimento de Padrões)



Fonte: Freepik, 2021.

### 2.3.1.3 Abstração

A abstração consiste em simplificar a realidade e envolve ignorar detalhes desnecessários, tornando os problemas mais fáceis. A habilidade primordial desse pilar está em escolher o detalhe correto a ser desconsiderado, sem perder nada que seja importante para a solução do problema. Assim, cria-se uma representação ou modelo do que se está tentando resolver apenas com os elementos relevantes. Esse modelo auxilia no entendimento do problema e nas condições para contorná-lo, desde que descrito de forma precisa a fim de que possa ser analisado e compreendido (CAS, 2020; CSIZMADIA *et al.*, 2015).

Segundo Wing (2010), a abstração é usada para permitir que um objeto represente vários, sendo o processo de pensamento mais importante do PC, pois dá o poder de dimensionar e lidar com a complexidade. Um algoritmo pode ser considerado uma abstração de um processo que recebe uma entrada, executa uma sequência de passos e produz uma saída para satisfazer determinado objetivo. A construção de sistemas computacionais também ocorre sobre camadas de abstrações, ao escrever um programa em uma linguagem visual baseada em blocos, por exemplo, não é necessário conhecer o hardware ou o sistema operacional do computador para desenvolver e executar a aplicação.

Silva (2020) aponta que a abstração possibilita que os sistemas computacionais sejam visualizados e utilizados em diferentes níveis de detalhes, e evidencia a sua relevância enquanto atividade mental superior ao propor um sistema conceitual para sustentar a compreensão e o desenvolvimento do PC com foco na

abstração enquanto habilidade mais importante. Além da abstração, dão sustentação ao Pensamento Computacional, segundo o autor, um conjunto de funções mentais superiores composto por: generalização, decomposição, pensamento algorítmico e avaliação.

Um exemplo clássico de abstração do mundo real é o mapa de um metrô ou do transporte público de uma região. Conforme é possível observar na Figura 6, as informações contidas são altamente refinadas e se restringem ao essencial para que o passageiro possa utilizar o serviço e se locomover pela cidade. Vários dados menos importantes não são incluídos no mapa, como: posição geográfica, distância e altitude. Essa representação contém precisamente as informações para planejar uma rota, nada além disso (CSIZMADIA *et al.*, 2015).

**Figura 6 – Mapa do Transporte Metropolitano (Abstração)**



Fonte: CPTM-SP<sup>17</sup>, 2020.

### 2.3.1.4 Algoritmos

Com base nos pilares anteriores, ao pensar na resolução de algum problema,

<sup>17</sup> Mapa do transporte da cidade de São Paulo elaborado pela Companhia Paulista de Trens Metropolitanos (CPTM). Mais informações em: <https://www.cptm.sp.gov.br>.

é traçado um plano, uma estratégia ou uma sequência de passos a ser executada. Essa sequência de regras a ser seguida é conhecida como algoritmo. (BBC LEARNING, 2015). As instruções lógicas contidas em um algoritmo, se seguidas com precisão, levam a solução do problema inicial, bem como podem resolver problemas semelhantes.

Csizmadia *et al.* (2015) chamam de “pensamento algorítmico” a capacidade de raciocinar em termos de regras para resolver problemas ou compreender situações. Essa habilidade é apontada pelos autores como fundamental para que alunos aprendam a escrever programas computacionais. A escrita de um algoritmo pode se dar de diferentes maneiras: em formato de diagramas, em pseudocódigo (linguagem humana) ou por meio de códigos em uma linguagem de programação (BRACKMANN, 2017). A execução da sequência de instruções pode se dar por uma pessoa ou por um computador, a depender do seu formato.

Brackmann (2017) afirma que o algoritmo é o núcleo principal do pensamento computacional em virtude da sua grande abrangência nas diferentes atividades que podem ser propostas. Ainda, segundo ele, algoritmos são soluções prontas que em sua formulação já passaram pelos processos propostos nos demais pilares do PC.

O conceito de algoritmo pode ser aplicado às mais variadas atividades. Por exemplo, ao realizar uma conta matemática de multiplicação, a partir do momento em que se identificam os passos necessários para realizar o cálculo, é possível repeti-los com diferentes números sem a necessidade de se descobrir novamente como fazer a multiplicação. Assim, automatiza-se a resolução do problema a partir da construção de um algoritmo.

## **2.4 Computação móvel, App Inventor e pensamento computacional**

Nos últimos anos tem se intensificada a disseminação da computação móvel, principalmente a partir da popularização dos celulares/smartphones, democratizando o acesso a internet e demonstrando que a população está cada vez mais conectada. Dados da Pesquisa Nacional por Amostras de Domicílios (PNAD) Contínua de 2019 apontam que 81% da população com dez anos ou mais tem telefone móvel, sendo esse o principal meio de acesso (98,6%) de quem utilizou a internet. Entre os estudantes, 92,6% do ensino privado e 64,8% da rede pública tem aparelho celular para uso pessoal, sendo que 97% deles com acesso à internet (IBGE, 2021).

A mesma pesquisa aponta ainda que além do aumento no uso dos celulares,

existe uma diminuição na utilização dos computadores de mesa, o que indica que os smartphones tem se mostrado cada vez mais completos, com maior poder de processamento e funcionalidades, se tornando o principal dispositivo computacional presente no cotidiano das pessoas. Esses números demonstram o potencial existente com a aplicação dos dispositivos móveis na educação, tanto para aprimorar as práticas de ensino nas diferentes áreas, quanto para formar profissionais capazes de explorar as capacidades destes equipamentos, a exemplo do desenvolvimento de aplicativos móveis.

Segundo Villela (2013), a computação móvel se apresenta como uma porta de entrada ao mundo de trabalho pela procura de profissionais qualificados para a produção tecnológica no Brasil. Essa produção passa necessariamente pelo desenvolvimento de aplicativos que, apesar de recente se comparado à história da programação de computadores, vem crescendo exponencialmente em função da disseminação dos dispositivos portáteis. Sua importância está relacionada à forma de interação dos usuários que passaram a utilizar a tecnologia móvel diariamente como meio de lazer, trabalho e comunicação nas mais variadas atividades.

De acordo com a Unesco (2014), nos próximos anos os alunos não apenas usarão os dispositivos móveis para apoiar as tarefas de ensino, como também passarão a construir aplicativos de acordo com as suas necessidades, sendo que para isso precisarão aprender sobre os conceitos de programação e de solução de problemas. Ou seja, se espera que alunos e educadores passem a produzir soluções computacionais, não se limitando apenas a utilizá-las, o que está diretamente relacionado ao desenvolvimento do pensamento computacional. Dessa maneira, a disseminação do PC através do desenvolvimento de aplicativos móveis se destaca devido ao

“crescente interesse dos jovens pela utilização destas tecnologias e pelo fato do processo inerente do desenvolvimento de aplicações promover a aproximação dos alunos com vários princípios do pensamento computacional” (DA SILVA; KURTZ; SANTOS, 2020a, p. 11, tradução nossa<sup>18</sup>).

Normalmente, o aprendizado de programação nos cursos da área da Computação se inicia com a disciplina de Algoritmos, onde os alunos aprendem a resolver problemas através da representação descritiva da solução por um conjunto

---

<sup>18</sup> “[...] growing interest of young people in the use of these technologies, and by the fact that the process inherent in the development of applications promotes the approximation of students with various principles of computational thinking”.

de instruções lógicas (algoritmo). Diferentes ferramentas, computacionais ou não, podem ser utilizadas para auxiliar o processo de construção desses algoritmos. Na maioria das vezes, são empregados softwares que possibilitam a escrita da codificação em linguagem natural ou na forma de pseudocódigo. Cada ambiente possui uma sintaxe específica para o desenvolvimento da lógica de programação, a exemplo do VisuAlg<sup>19</sup> e do Portugol Studio<sup>20</sup> que costumam ser utilizados para fins educativos, em que o aluno pode elaborar e testar algoritmos para os problemas propostos em aula. Como já discutido anteriormente, por vezes esses softwares dificultam a aprendizagem de alunos iniciantes justamente pela necessidade de memorização de instruções particulares a determinada linguagem. Nesse sentido, Santiago e Kronbauer (2016, p. 421) evidenciam, a partir de um estudo da bibliográfica científica sobre o ensino e aprendizagem de programação, que há a “necessidade de novas propostas de metodologias instrucionais, as quais não sejam baseadas puramente em código textual”.

Como alternativa, existem as Linguagens de Programação Visual (Visual Programming Language – VPL) baseadas em blocos que procuram facilitar o processo de codificação se fazendo valer de componentes visuais que podem ser encaixados para a construção de soluções, sem exigir o domínio da sintaxe da linguagem. Seu funcionamento remete ao de um quebra-cabeça, onde as peças, no caso os blocos, devem ser conectados de forma lógica. Cada bloco possui um formato gráfico específico que limita o encaixe a determinados tipos de outros blocos, o que evita a ocorrência de erros sintáticos (WEINTROP; WILENSKY, 2015). Além disso, os ambientes de VPL também costumam possibilitar a visualização e teste imediato do que foi criado, facilitando a compreensão do problema e a identificação de possíveis erros.

Um exemplo de ambiente VPL é o Scratch<sup>21</sup> que está disponível de forma online e offline, permitindo com que alunos criem projetos sem se preocupar com a sintaxe ou a escrita de comandos das linguagens de programação tradicionais, sendo necessário apenas escolher os cenários, os atores e arrastar os blocos disponíveis para a área de programação da ferramenta. Nos últimos anos, diversos estudos nacionais foram desenvolvidos com Scratch a fim de desenvolver o

---

<sup>19</sup> Mais informações em: <http://www.apoioinformatica.inf.br/produtos/visualg>.

<sup>20</sup> Mais informações em: <http://lite.acad.univali.br/portugol>.

<sup>21</sup> Mais informações em: <https://scratch.mit.edu>.

pensamento computacional, a exemplo de: De França e Do Amaral (2013), Arantes *et al.* (2014), Rodriguez *et al.* (2015), Batista *et al.* (2016), Lisbôa *et al.* (2017), Geraldles *et al.* (2019), Santana e Oliveira (2019), dentre outros. Percebe-se um crescente interesse pelo tema e uma expressiva utilização do Scratch com alunos da educação básica, principalmente do Ensino Fundamental, em função do ambiente lúdico proporcionado pela ferramenta. Apesar das inúmeras iniciativas com o comprovado resultado positivo, atualmente não existe um aplicativo oficial do Scratch para celulares (somente para tablets), o que possibilita apenas executar a versão online do programa nesses dispositivos, mas não criar aplicativos.

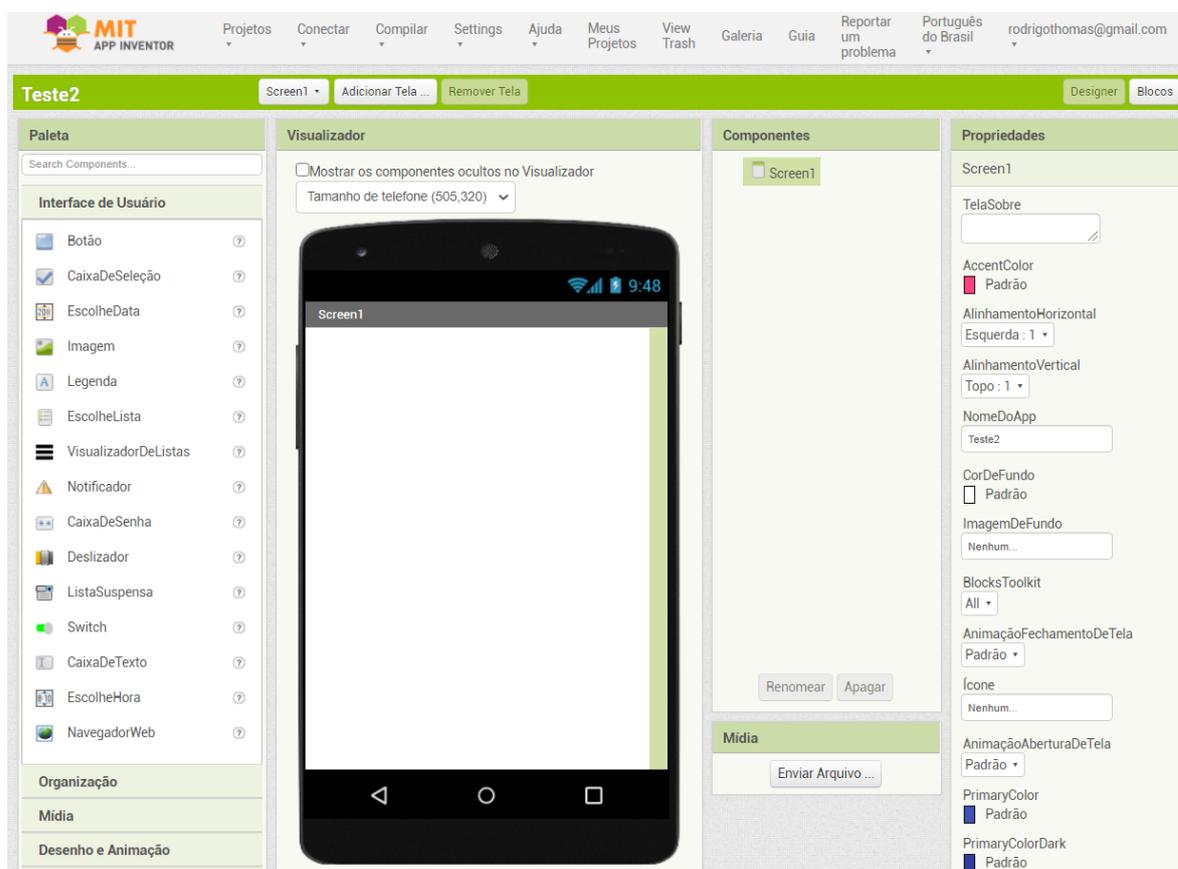
No que diz respeito ao ensino de programação para dispositivos móveis, destaca-se outra linguagem de programação visual em blocos, o App Inventor, desenvolvido pelo Google e mantido atualmente pelo Instituto de Tecnologia de Massachusetts (MIT). Trata-se de uma ferramenta de código aberto que permite a criação de aplicativos móveis, como jogos, simulações, animações, entre outros, para dispositivos com o sistema Android. De acordo com MIT (2013), “construído com a missão de democratizar a criação de aplicativos para e por todos, o MIT App Inventor é uma ferramenta gráfica que transforma os consumidores de mídia em participantes ativos e críticos do mundo móvel”. O ambiente pode ser acessado gratuitamente, de forma online, por meio de um navegador web e conta com mais de 8 milhões de usuários registrados, de 195 países, e 34 milhões de aplicativos construídos (MIT, 2020).

O funcionamento do App Inventor é similar ao do Scratch, com a possibilidade de desenvolver aplicativos apenas arrastando blocos que se encaixam segundo uma lógica de funcionalidades pré-estabelecidas, levando a uma construção abstraída, pois não requer que o aluno tenha compreensão total do código que será executado pelo aplicativo. Diante do exposto, justificamos a adoção do App Inventor enquanto ferramenta para a construção de aplicativos móveis e o desenvolvimento do pensamento computacional.

A programação de um aplicativo no App Inventor é baseada em componentes, os quais podem ser manipulados por eventos, métodos e propriedades. O comportamento dos componentes é orientado pelos eventos originados da interação do usuário com o aplicativo. A interface do App Inventor se divide em duas janelas: “Designer” e “Blocos”. A primeira se destina a criar a interface visual do aplicativo, arrastando componentes da “Paleta”, como botões, caixas de texto e seleção,

figuras, mapas, entre outros, para o “Visualizador”, conforme a Figura 7.

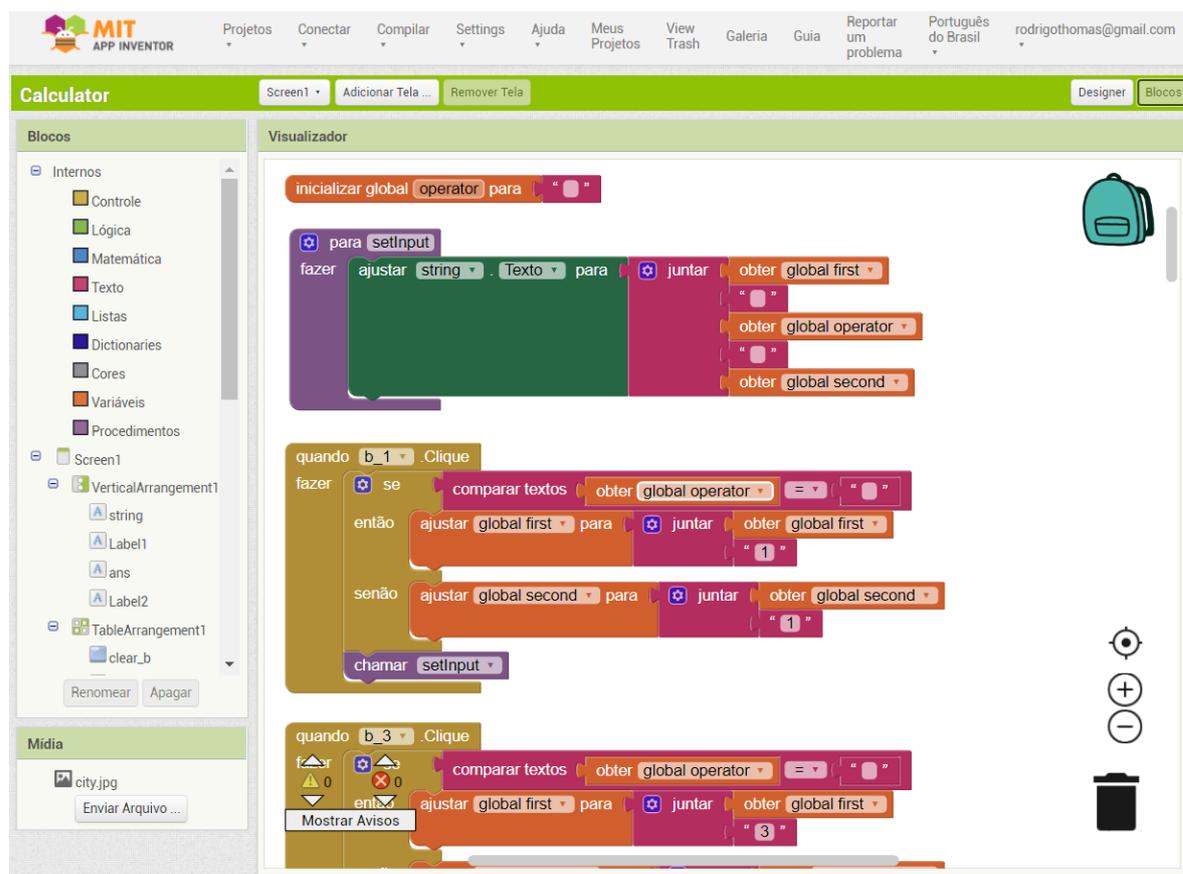
**Figura 7 – Janela “Designer” do App Inventor**



Fonte: MIT, 2020.

Para definir o comportamento dos componentes inseridos no “Designer”, é utilizada a janela de “Blocos”, conforme a Figura 8. Nela, são estabelecidas as funcionalidades do aplicativo a partir de blocos conectáveis, que podem ser eventos ou métodos. Cada bloco realiza diferentes ações, desde controles de sequência e continuidade, até operações matemáticas. O aplicativo desenvolvido pode ser executado diretamente em um dispositivo Android conectado ao computador por uma rede wireless ou via cabo, ou através de um emulador que acompanha o App Inventor.

Figura 8 – Janela “Blocos” do App Inventor



Fonte: MIT, 2020.

O uso do App Inventor parece ser menos difundido no âmbito escolar se comparado ao Scratch. Segundo Bombasar *et al.* (2015), em uma revisão sistemática de literatura para identificar as principais ferramentas utilizadas no ensino do pensamento computacional de 2006 a 2015, a ferramenta mais utilizada foi o Scratch, com 31 publicações, enquanto que o App Inventor conta com 11 publicações. Ainda assim, quando relacionado especificamente ao PC, observam-se alguns relatos interessantes de utilização desde o ensino fundamental, até o superior. A exemplo, Gomes e De Melo (2013) empregaram o App Inventor para propiciar experiências mais atrativas no ensino de lógica de programação para estudantes dos 1<sup>os</sup> e 2<sup>os</sup> anos do ensino médio. Os autores reforçam que a interface gráfica da ferramenta facilita explorar os conceitos básicos de programação de modo mais intuitivo, sem que os alunos se atenham a detalhes de sintaxe, concentrando-se na resolução dos problemas.

No trabalho de Bauer *et al.* (2017), o pensamento computacional é incentivado a partir de um projeto de extensão para o ensino de programação à

adolescentes. Os autores chamam a atenção para o crescente uso de smartphones e a importância de adaptar metodologias de ensino para agregar as tecnologias móveis à sala de aula. A partir da Aprendizagem Baseada em Problemas (ABP), o App Inventor foi apresentado a alunos do 5º ao 9º ano do ensino fundamental, resultando na elaboração do concurso “Melhor App”, onde foram premiados os três aplicativos que mais se destacaram.

Em outro estudo, Lisbôa e Karling (2019), relaram a construção de aplicativos para Android com alunos de Licenciatura em Computação como estratégia pedagógica para o desenvolvimento do pensamento computacional. Para avaliar os aplicativos construídos, foi adotada a metodologia de “avaliação por pares” onde os próprios alunos julgaram os trabalhos dos colegas. Segundo os autores, a linguagem visual contribuiu para a reflexão sobre a solução de problemas reais por meio do pensamento recursivo de forma lúdica.

O trabalho de Santos *et al.* (2017) relata a experiência do projeto “Meninas Digitais Tchê Missões” onde alunas do ensino médio foram estimuladas a desenvolver o pensamento computacional através de diferentes atividades, entre elas, um curso e atelier de programação no App Inventor. Após o curso, as alunas participaram de um desafio onde desenvolveram aplicativos socialmente úteis voltados à qualidade de vida das pessoas. Ao final, os três melhores aplicativos foram premiados com troféus e medalhas.

A partir dos trabalhos relacionados, percebe-se o potencial didático-pedagógico do App Inventor como possibilidade para o desenvolvimento do pensamento computacional. Porém, na maioria dos estudos, o foco está nas atividades práticas e no ensino de lógica de programação, sem uma preocupação com a aprendizagem significativa, conforme já relatado no trabalho de Ortiz *et al.* (2018). Como diferencial, propomos uma sequência didática pautada nos 3MP problematizando situações reais, emergidos a partir da realidade dos alunos, a fim de se obter uma aprendizagem significativa e, ainda assim, garantir a aprendizagem dos conteúdos previstos na disciplina de algoritmos.

### **3. CAMINHOS METODOLÓGICOS**

Este capítulo é destinado à descrição da metodologia de pesquisa adotada durante o trabalho, com a apresentação dos instrumentos de coleta de dados e o contexto do ambiente e dos participantes da pesquisa. Descrevemos a proposta de aprendizagem de programação e desenvolvimento do pensamento computacional por meio de uma sequência didática pautada nos 3MP. Também, apontamos as características do CodeMaster, ferramenta selecionada para a avaliação das habilidades do PC.

#### **3.1 Metodologia da pesquisa**

A presente pesquisa tem caráter exploratório, assumindo o formato de um estudo de caso. Segundo Gil (2016, p. 37), o estudo de caso “consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento”. O estudo de caso foi planejado e executado com base em Yin (2001), que define o estudo de caso como “uma maneira de se investigar um tópico empírico seguindo-se um conjunto de procedimentos pré-especificados” (YAN, 2011, p. 35).

Segundo a sua finalidade, a pesquisa é aplicada, pois procurou adquirir conhecimentos com o objetivo de aplicação em uma situação específica (GIL, 2016). Quanto à forma de abordagem do problema, trata-se de uma pesquisa qualitativa, uma vez que o suporte teórico para a elaboração desta dissertação e do Produto Educacional partiu da coleta de informações em variadas fontes relevantes ao assunto (livros, periódicos, artigos científicos, teses e dissertações). Também foram utilizados a observação direta e os artefatos digitais criados pelos alunos no App Inventor, com posterior análise através da ferramenta CodeMaster<sup>22</sup> para a validação dos resultados.

---

<sup>22</sup> Mais informações em: [http://apps.computacaonaescola.ufsc.br:8080/i18n\\_pt](http://apps.computacaonaescola.ufsc.br:8080/i18n_pt)

Além disso, foram elaborados e aplicados questionários compostos por perguntas objetivas e descritivas (abertas), sendo as objetivas analisadas em escala tipo-Likert (MARTINS; THEÓPHILO, 2007) por meio de estatística descritiva com a apresentação de resultados no formato de gráficos. A opção pelo procedimento de coleta de dados foi motivada pelas vantagens que os questionários apresentam à pesquisa, como: possibilidade de alcançar um número considerável de pessoas simultaneamente, propiciar economia de tempo com grande obtenção de dados e o retorno de respostas rápidas e precisas (MARCONI; LAKATOS, 2010). A primeira aplicação do questionário ocorreu no início da pesquisa, onde os participantes foram convidados a responder um formulário online criado no Google Formulários (Apêndice B), para um melhor entendimento a respeito do perfil dos sujeitos em relação ao processo de ensino e aprendizagem de programação, servindo de base para a elaboração da sequência didática. Após os encontros para a validação da sequência didática, um segundo questionário online foi aplicado para possibilitar a comparação de resultados entre a caracterização inicial dos sujeitos até a percepção posterior às práticas propostas. Antes de cada aplicação dos questionários, os participantes foram consultados quanto a ciência e concordância com as condições apresentadas no Termo de Consentimento Livre e Esclarecido - TCLE (Apêndice C).

Por fim, utilizou-se o instrumento de diário de campo para o registro das manifestações dos alunos e das percepções do pesquisador durante a realização dos encontros online. O diário permitiu a sistematização das experiências a partir da observação reflexiva dos momentos de interação entre pesquisador-sujeitos e entre os próprios sujeitos, sucedendo nos resultados apresentados no capítulo 5.

A presente pesquisa foi submetida à Plataforma Brasil<sup>23</sup> e aprovada pelo Comitê de Ética do IF Farroupilha sob CAEE 36657020.7.0000.5574.

### **3.2 Local de pesquisa e amostra**

A presente pesquisa foi realizada em duas instituições públicas pertencentes à Rede Federal de Educação Profissional, Científica e Tecnológica, os Campus Santo Augusto e Santo Ângelo do Instituto Federal Farroupilha (IFFar), ou seja, um estudo de caso com grupos distintos.

O IFFar é uma instituição de educação superior, básica e profissional,

---

<sup>23</sup> Mais informações em: <https://plataformabrasil.saude.gov.br>

pluricurricular e multicampi, especializada na oferta de educação profissional e tecnológica nas diferentes modalidades de ensino. Atualmente o IFFar é composto por 11 Campi, mais a Reitoria localizada em Santa Maria, além de Polos de Educação à distância e Centros de Referência. Entre os cursos da área de Computação oferecidos pelos Campi, estão o Curso Superior de Tecnologia em Sistemas para Internet (Santo Ângelo) e o Curso Superior de Licenciatura em Computação (ambos os campi), sendo os alunos da licenciatura os sujeitos desta pesquisa (INSTITUTO FEDERAL FARROUPILHA, 2018).

A escolha da Licenciatura em Computação se deu pela possibilidade de realização da pesquisa em dois cursos de localidades e públicos distintos, apesar de se tratarem de instituições com características semelhantes. Ademais, conforme já abordado no capítulo 2, a partir do entendimento que a LC forma professores de computação com futura atuação profissional na educação básica, representa uma possibilidade de desenvolvimento e disseminação das habilidades do PC nas escolas. Ainda, a articulação das diferentes disciplinas do curso de LC para a constituição do professor de computação tem relação com o currículo integrado, principal característica das instituições que compõem a Rede Federal de EPT.

Sendo assim, a amostra inicial foi composta por 38 alunos do primeiro semestre dos cursos de Licenciatura em Computação, sendo 23 do Campus Santo Augusto e 15 do Campus Santo Ângelo. Esses números representam o total de respondentes do primeiro questionário logo no início do semestre letivo. No entanto, esse número foi diminuindo gradativamente durante o decorrer do semestre em função de algumas desistências e abandonos do curso, além do acúmulo de atividades e avaliações em diferentes disciplinas que, segundo relato dos discentes, impossibilitou a participação nos encontros, conforme melhor abordado no capítulo 5.

### **3.2.1 Caracterização dos sujeitos**

O contato inicial com a turma ingressante do curso de Licenciatura em Computação do Campus Santo Ângelo ocorreu no primeiro semestre letivo de 2020, de forma virtual, através de uma aula online da disciplina de algoritmos, após contato prévio com a docente ministrante da disciplina. Na ocasião, foi apresentada a proposta de pesquisa aos alunos, bem como o envio do formulário com o questionário inicial que, após alguns dias, registrou seis respostas. Porém, como o

período compreendia o início da pandemia da Covid-19<sup>24</sup>, com cancelamento das aulas presenciais no mês de março e após retomada em formato online a partir de julho, houve pouco tempo hábil para a realização das atividades antes da finalização do semestre letivo.

Dessa forma, optou-se em realizar a pesquisa com os alunos ingressantes de LC de Santo Augusto e Santo Ângelo em 2021, em formato online e remoto através de aulas no Google Meet, a partir do mês de junho, uma vez que houve atraso no início do semestre letivo ainda em função da pandemia. O envio do questionário inicial por e-mail para a turma do Campus Santo Augusto ocorreu no dia 07 de junho, com prazo de uma semana para envio das respostas. Já para a turma do Campus Santo Ângelo, o envio ocorreu no dia 24 de junho, com prazo de duas semanas para o retorno, pois as aulas com a turma iniciaram-se mais tarde. As respostas foram obtidas virtualmente a partir da própria ferramenta Google Formulários utilizada para elaboração do questionário, sem outra identificação além do nome e e-mail do aluno, utilizados somente para fins de controle de duplicidade de respostas. Logo, possibilitou-se aos entrevistados a conveniência quanto ao tempo e local para as respostas, bem como a não influência do pesquisador sobre as suas opiniões, além da garantia de um mínimo de tempo despendido para tal, uma vez que foram elaboradas perguntas simples e sucintas.

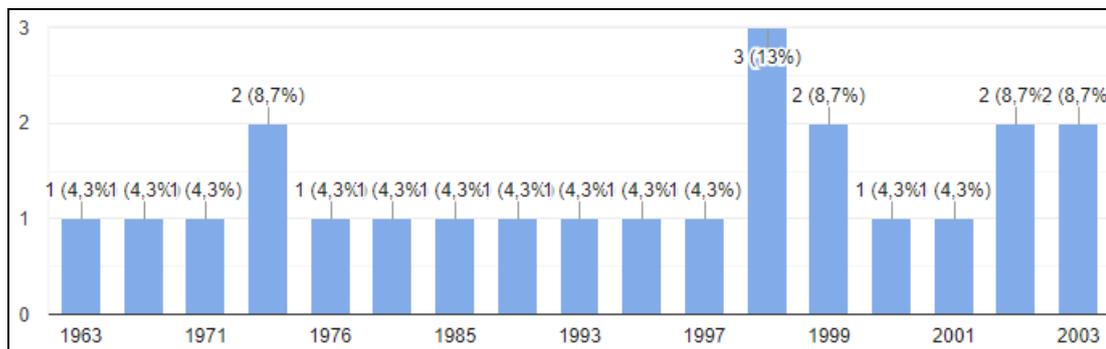
O levantamento inicial trouxe informações importantes a respeito dos sujeitos da pesquisa, desde a faixa etária até a disponibilidade de equipamentos que viabilizassem a execução das atividades, passando pelo conhecimento do pensamento computacional e o interesse em estudá-lo. Essa aproximação inicial das turmas serviu como parâmetro para o planejamento da sequência didática.

Para a identificação dos indivíduos, quando necessário, tanto na caracterização a seguir, quanto na apresentação dos resultados no capítulo 5, foram utilizadas as descrições como seguem: Aluno 1; Aluno 2; Aluno 3; e assim por diante, sem especificar o seu sexo. A começar pela idade dos alunos, observou-se uma distribuição heterogenia entre ambos os Campi. Em Santo Augusto, o aluno mais velho possui 58 anos, enquanto o mais novo 18, uma diferença de 40 anos, conforme é possível observar no Gráfico 1.

---

<sup>24</sup> Mais informações em: <https://www.paho.org/pt/covid19>

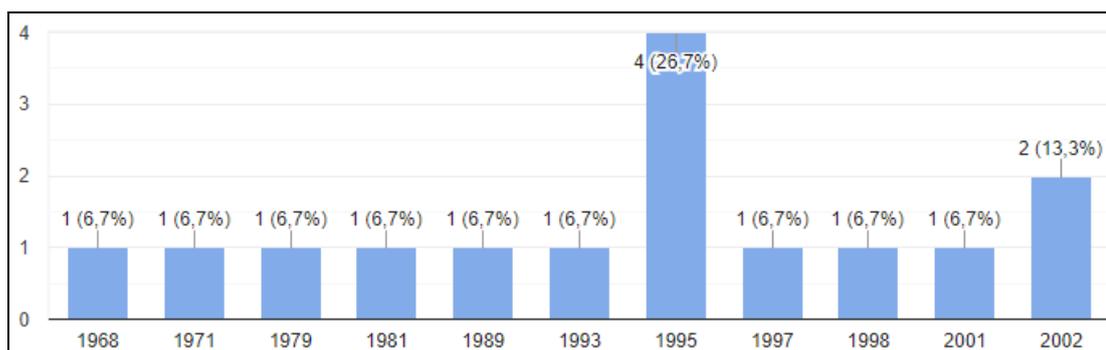
**Gráfico 1** – Respostas dos alunos do Campus Santo Augusto à pergunta: “Em que ano você nasceu?”



Fonte: Adaptado de Google Formulários, 2021.

No Campus Santo Ângelo, o aluno mais velho possui 53 anos, enquanto o mais novo 19, uma diferença menor, porém ainda considerável de 34 anos, conforme o Gráfico 2.

**Gráfico 2** – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Em que ano você nasceu?”



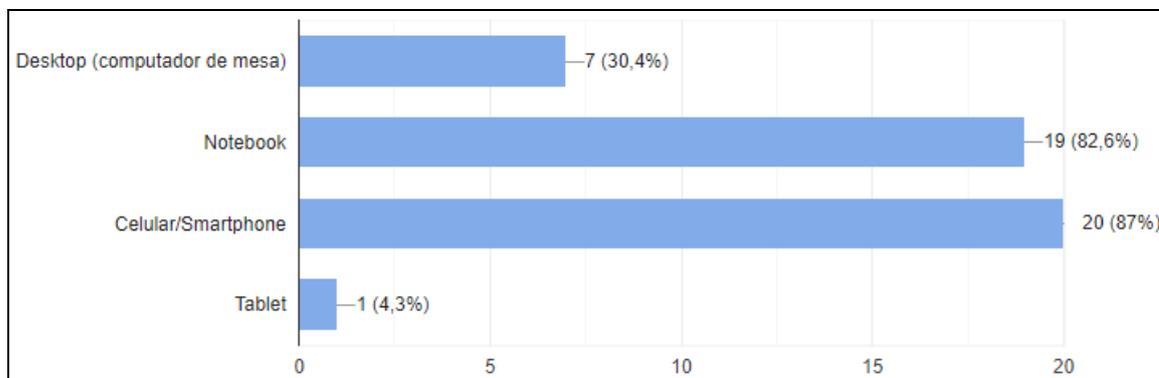
Fonte: Adaptado de Google Formulários, 2021.

Com relação ao gênero dos participantes, a maioria é do sexo masculino (quase 60%), mesmo assim a presença do público feminino é significativa e contrasta com a desigualdade frequentemente observada em cursos da área da Computação onde a presença feminina é bastante reduzida, gerando o estereótipo de que a área de TI é essencialmente masculina (RIBEIRO *et al.*, 2019).

Quando questionados a respeito dos equipamentos que possuem em casa, os alunos responderam, em sua maioria do Campus Santo Augusto (Gráfico 3) e na totalidade do Campus Santo Ângelo (Gráfico 4), dispor de smartphone. Dados semelhantes podem ser observados quanto a disponibilidade de desktop (computador de mesa) ou notebook. Essa informação foi relevante, pois demonstrou ser viável a execução desta pesquisa, na medida em que não seria possível utilizar os laboratórios de informática do IFFar, o fato dos alunos disporem de computador ou notebook para o desenvolvimento das atividades e smartphone para o teste dos

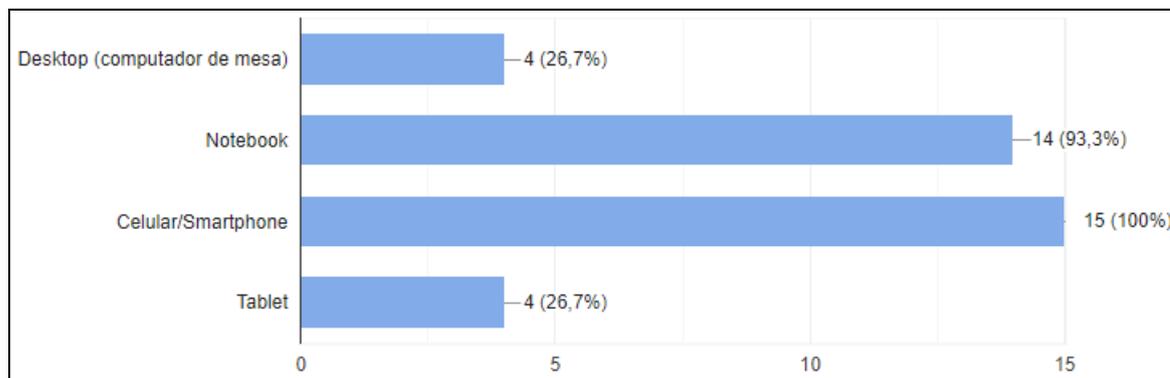
aplicativos construídos, possibilitou a realização da pesquisa em um formato online.

**Gráfico 3** – Respostas dos alunos do Campus Santo Augusto à pergunta: “Qual(is) equipamento(s) você possui em casa?”



Fonte: Adaptado de Google Formulários, 2021.

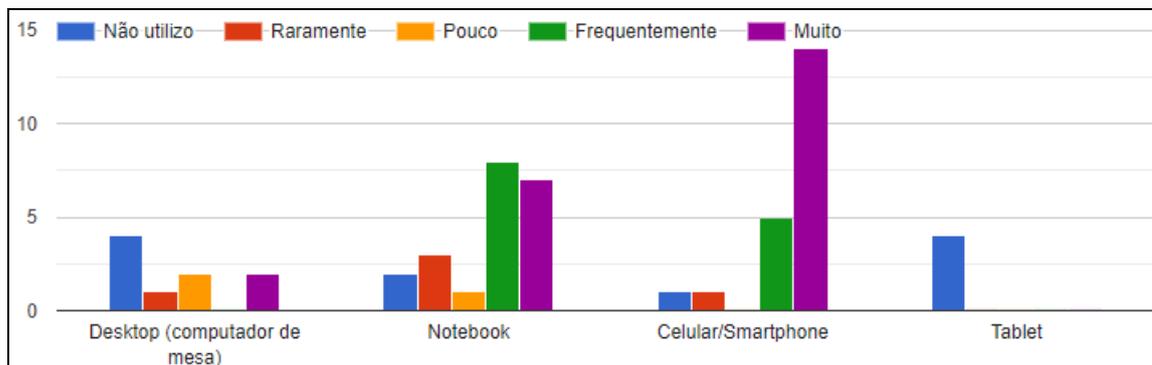
**Gráfico 4** – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Qual(is) equipamento(s) você possui em casa?”



Fonte: Adaptado de Google Formulários, 2021.

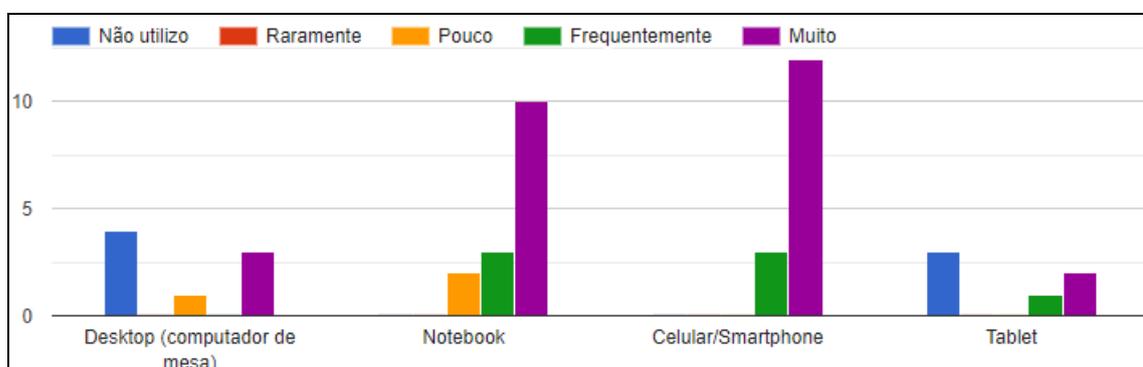
Outro fator que permitiu a execução das atividades diz respeito a disponibilidade de internet. Todos os estudantes disseram possuir conexão em casa, sendo que quase a totalidade do tipo fibra óptica, apenas um aluno de Santo Augusto informou ter conexão via rádio. Já quando indagados sobre quais equipamentos utilizam para acessar a internet, grande parte afirmou fazer muito uso do smartphone, sendo o notebook o segundo dispositivo mais utilizado, tanto no Campus Santo Augusto (Gráfico 5) quanto no Campus Santo Ângelo (Gráfico 6). Esse dado é mais um indício da viabilidade de realização da sequência didática em um formato de aulas remotas.

**Gráfico 5** – Respostas dos alunos do Campus Santo Augusto à pergunta: “Se você possui acesso à internet, qual(is) equipamento(s) utiliza para acessar?”



Fonte: Adaptado de Google Formulários, 2021.

**Gráfico 6** – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Se você possui acesso à internet, qual(is) equipamento(s) utiliza para acessar?”

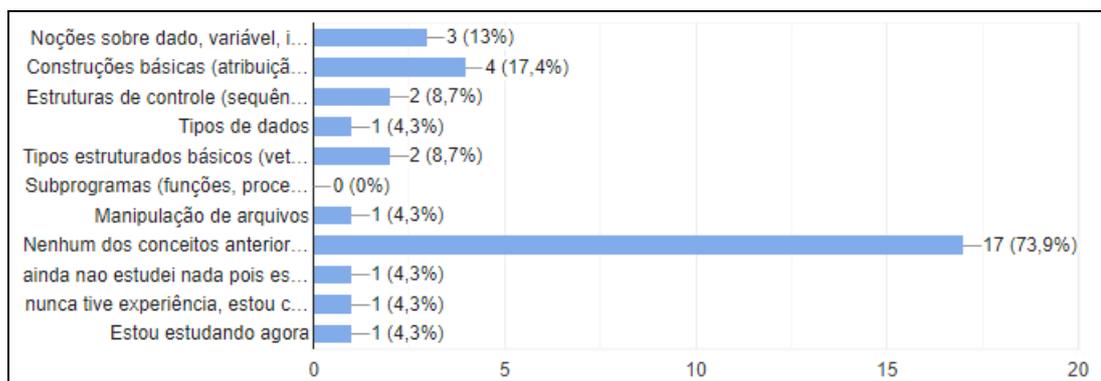


Fonte: Adaptado de Google Formulários, 2021.

Ao serem questionados sobre o que entendem por algoritmo, muitos afirmaram desconhecer o termo ou obter pouco conhecimento por estarem no início do curso. Também se podem observar algumas respostas associando o conceito à sequência de passos ou comandos para resolver uma tarefa ou problema. Por exemplo, os Alunos 4 e 15 do Campus Santo Ângelo responderam, respectivamente, “é uma sequência finita para a solução de algum problema, no caso em programação” e “creio que seja um exercício lógico que tenha por intuito principal resolver determinados tipos de problemas”.

Na questão seguinte os alunos responderam sobre quais conceitos já haviam estudado na disciplina de algoritmos. A partir deste ponto foi possível observar uma diferença entre as turmas em função do período em que a pesquisa foi executada. Dado que o estudo com a turma do Campus Santo Augusto se iniciou mais tarde, a turma de teve somente uma aula da disciplina antes do início da pesquisa, portanto a maioria apontou desconhecer os conceitos apresentados na questão, conforme o Gráfico 7.

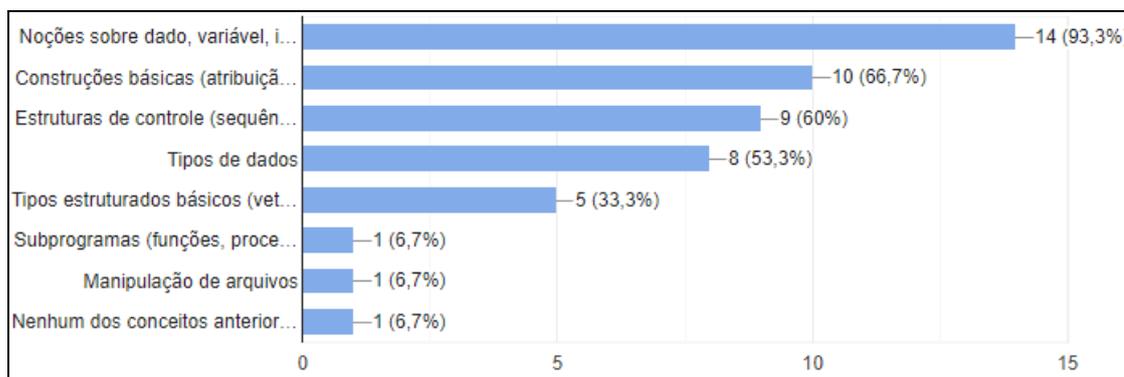
**Gráfico 7** – Respostas dos alunos do Campus Santo Augusto à pergunta: “Quais conceitos você estudou na disciplina de algoritmos?”



Fonte: Adaptado de Google Formulários, 2021.

Em contrapartida, grande parte dos alunos do Campus Santo Ângelo afirmou já possuir noções sobre variáveis, instruções, atribuições, tipos de dados e estruturas de controle, estudados nas quatro aulas anteriores da disciplina, conforme o Gráfico 8.

**Gráfico 8** – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Quais conceitos você estudou na disciplina de algoritmos?”



Fonte: Adaptado de Google Formulários, 2021.

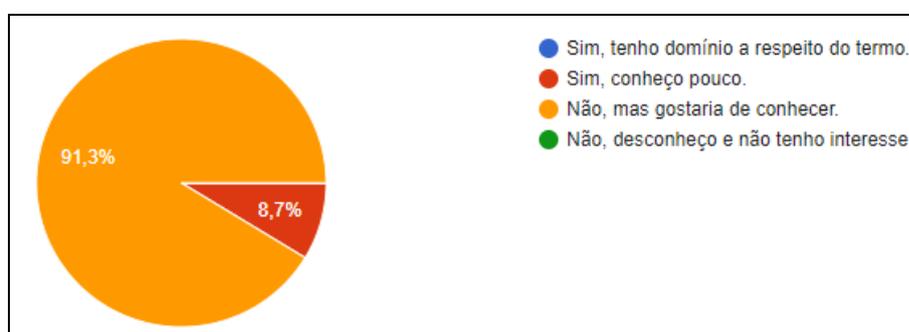
As duas perguntas seguintes refletiram o resultado da anterior, onde os sujeitos do Campus Santo Augusto demonstraram não conhecer os passos para construir um algoritmo, nesse sentido também não possuíam subsídios para a questão seguinte que solicitava a apresentação das suas dificuldades na construção de um algoritmo. Enquanto isso, boa parte dos estudantes de Santo Ângelo expuseram os passos que realizaram para elaborar um algoritmo desde a apresentação do problema pelo professor(a), no entanto, as repostas tenderam a associar os passos com a escrita de um programa computacional, especialmente na ferramenta Portugal<sup>25</sup>, utilizada nas aulas anteriores. A exemplo, o Aluno 12 afirmou:

<sup>25</sup> Mais informações em: <http://lite.acad.univali.br/portugol>

“1ª analiso o que o problema está pedindo; 2ª abro o Portugol e começo declarando as variáveis; 3ª utilizo os comandos para montar o código; 4ª executo o programa para ver se deu certo”. Com relação as dificuldades, os relatos se dividiram entre nenhuma, dúvidas quanto aos comandos a serem utilizados e quanto ao processo de declaração de variáveis, conforme manifestação do Aluno 9: “algumas vezes acabo esquecendo os comandos, mas conforme vou fazendo vai clareando novamente”. Estes aspectos apontam indícios de que as linguagens tradicionalmente utilizadas para o ensino de programação possuem uma sintaxe que torna complexa a aprendizagem, fazendo com que o aluno se preocupe mais com a memorização das instruções sintáticas do que com a lógica ou o entendimento e a resolução do problema apresentado. Esse entendimento é comprovado no estudo de Santiago e Kronbauer (2016), onde os autores afirmam que o ensino de programação baseado unicamente em sintaxe é desfavorável ao aprendizado de alunos iniciantes, justamente por não conseguirem representar por meio de modelos mentais o que está sendo memorizado.

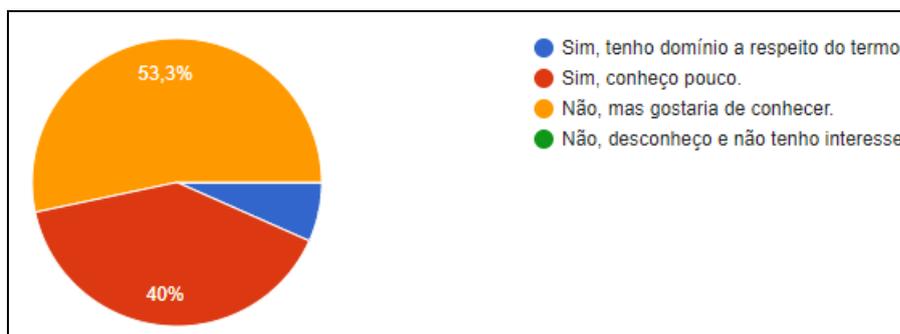
Os discentes também foram perguntados se já ouviram ou leram algo a respeito do termo “pensamento computacional”, sendo que a maioria do Campus Santo Augusto afirmou desconhecer, mas com interesse em aprender a respeito (Gráfico 9). Já entre os alunos do Campus Santo Ângelo houve uma divisão entre aqueles que conhecem um pouco e os que gostariam de conhecer (Gráfico 10). No entanto, ao analisar o entendimento daqueles que afirmam conhecer, observou-se que alguns associam o termo ao conhecimento tecnológico ou a resolução de problemas meramente computacionais.

**Gráfico 9** – Respostas dos alunos do Campus Santo Augusto à pergunta: “Você já ouviu ou leu algo a respeito do termo “Pensamento Computacional”?”



Fonte: Adaptado de Google Formulários, 2021.

**Gráfico 10** – Respostas dos alunos do Campus Santo Ângelo à pergunta: “Você já ouviu ou leu algo a respeito do termo “Pensamento Computacional”?”



Fonte: Adaptado de Google Formulários, 2021.

Outro aspecto interessante observado entre os sujeitos diz respeito a importância do aprendizado do PC. A maioria (82,6% do Campus Santo Augusto e 86,7% do Campus Santo Ângelo) acredita que estudar sobre o PC é relevante e alguns afirmam que pode contribuir na resolução de problemas, no desenvolvimento de programas e facilitar o aprendizado de outra áreas, conforme os relatos a seguir.

“No meu ver saber sobre o pensamento computacional é essencial para no futuro saber resolver um problema na máquina, até mesmo desenvolver um software.” (Aluno 4, Campus Santo Augusto).

“Uma das características mais fascinantes da infância é o desenvolvimento da plasticidade mental para se adaptar e solucionar de forma criativa os problemas apresentados. Com o passar dos anos, vejo essa capacidade ser aos poucos tolhida por sistemas educacionais engessados e antiquados. Sendo assim, acredito que o ensino dos conceitos do Pensamento Computacional deveria ser trabalhado nas unidades educacionais desde a primeira infância, para que os alunos possam resgatar essa capacidade estratégica e criativa de solucionar problemas.” (Aluno 4, Campus Santo Ângelo).

“Acredito que seja importante pois nos abre novas formas de pensamento, além de facilitar processos de aprendizado não só na área da computação.” (Aluno 6, Campus Santo Ângelo).

Por fim, com relação ao uso prévio de alguma ferramenta visual com blocos, a grande maioria dos alunos (91,3% em Santo Augusto e 80% em Santo Ângelo) afirmou nunca ter utilizado, sendo que apenas 2 já fizeram uso do Scratch e/ou do App Inventor. Porém, a maior parte (33 alunos no total) afirmou ter interesse em aprender a desenvolver aplicativos móveis para smartphone e, nesse sentido, sugeriram algumas temáticas de interesse, como: aplicativo para pessoas na 3ª idade, de ônibus com horários, localização e rotas, acessibilidade para deficientes visuais, jogos educativos, comércio de produtos naturais, entre outros.

A partir das informações obtidas, observou-se a viabilidade de realização do estudo, principalmente pela disponibilidade de smartphone entre os alunos, o que

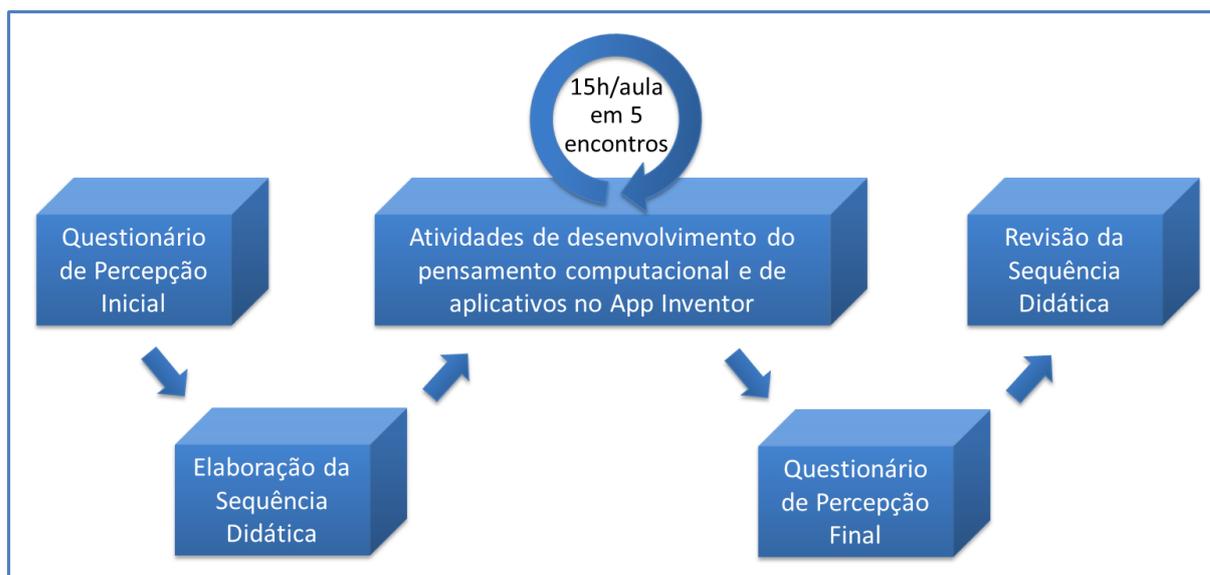
facilita o teste dos aplicativos se comparado ao uso de um emulador virtual. A falta de contato prévio com alguma ferramenta visual em blocos também pode ser considerada positiva, pois o aprendizado de um novo paradigma de programação pode motivar os alunos, além do próprio interesse em desenvolver aplicativos, já manifestado nas respostas. Destaca-se ainda a diferença entre as turmas no número de aulas de Algoritmos, o que resultou em níveis de conhecimentos distintos. Esses dados foram considerados na elaboração da sequência didática, descrita a seguir.

### **3.3 Construção da sequência didática**

O presente estudo está enquadrado na linha de pesquisa 2 – Organização e Memórias dos Espaços Pedagógicos em Educação Profissional e Tecnológica, a medida que se destinou a identificar as dificuldades dos alunos nas disciplinas de programação e a procurar uma alternativa ao método de ensino tradicional com vistas a propiciar o desenvolvimento do pensamento computacional à alunos do curso de Licenciatura em Computação por meio de uma Sequência Didática (SD) pautada nos Três Momentos Pedagógicos (3MP).

A começar pelo levantamento bibliográfico que, segundo Minayo (2002), é fundamental e exige criticidade na escolha de materiais para aprofundar os conhecimentos relevantes à pesquisa, montou-se o aporte teórico que fundamentou o desenvolvimento do Produto Educacional. Deste modo, a fim de encadear e articular as atividades de aprendizagem ao longo da disciplina de algoritmos foi elaborada uma sequência didática a respeito do ensino de programação para dispositivos móveis.

Assim, a construção da sequência didática foi dividida em nove etapas, conforme a Figura 9, sendo a primeira constituída pela aplicação do questionário de percepção inicial enviado por e-mail aos alunos. Em seguida, elaborou-se a versão inicial da SD com base nas respostas dos alunos. Após, 15 horas aula, distribuídas em cinco encontros online de 3 horas com o intuito de viabilizar o desenvolvimento do pensamento computacional e a aprendizagem de programação através da construção de aplicativos móveis com a ferramenta App Inventor. Na oitava etapa, aplicação do questionário final, também enviado por e-mail, para possibilitar a comparação de resultados entre a caracterização inicial dos sujeitos até a percepção posterior às práticas propostas. Por fim, foram realizados os ajustes necessários na SD a partir das observações proporcionadas pelas atividades de validação.

**Figura 9** – Etapas da Sequência Didática

Fonte: Elaborado pelo autor, 2021.

Primeiramente, a SD foi estruturada em formato de apresentação na ferramenta Microsoft PowerPoint, cada aula em um arquivo distinto, contendo todo o material trabalhado com os alunos. Cada um dos arquivos em formato PDF (Portable Document Format) foi compartilhado com os discentes antes do início de cada encontro para que pudessem se apropriar do tema e acompanhar o andamento da aula ministrada pelo pesquisador. Apesar de se mostrar um formato válido e produtivo para a elaboração e o desenvolvimento da SD, entendemos que o compartilhamento e a divulgação do Produto Educacional seriam potencializados se organizados em um *website*, facilitando o acesso aos interessados na temática. Portanto, a SD resultou em uma página da internet, com o endereço eletrônico: “<https://sites.google.com/view/pceappinventor>”, sob o título de “Sequência didática para o desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor”. O Produto Educacional foi elaborado através da ferramenta Google Sites<sup>26</sup> e será apresentado em detalhes no capítulo seguinte.

Em linhas gerais, o *website* contempla a apresentação do produto; a justificativa e os objetivos da pesquisa; a sequência didática dividida em 5 aulas, todas embasadas nos 3MP; um espaço para a publicação futura deste trabalho de dissertação; as publicações de artigo em evento, periódico e livro sobre a temática da pesquisa; formulário de contato para que os visitantes possam manifestar as suas

<sup>26</sup> Mais informações em: <https://sites.google.com>.

concepções sobre o produto; informações sobre o site contendo o currículo resumido dos autores com link para a Plataforma Lattes.

### 3.4 Avaliação do pensamento computacional através do CodeMaster

A exemplo do que ocorre com a definição do termo “pensamento computacional”, a sua avaliação também apresenta inúmeras abordagens distintas. A partir de uma revisão sistemática de literatura sobre a autoavaliação do pensamento computacional de jovens, Martins *et al.* (2020) demonstram a variedade de estudos, métodos e instrumentos que investigam a eficiência do ensino de atividades do PC. Nos últimos 5 anos, a revisão proposta pelos autores mapeou 11 trabalhos que evidenciaram formas de avaliar o PC em jovens de 10 a 15 anos de idade. Destes, 7 estudos realizaram intervenções com Scratch e 5 utilizaram o Computational Thinking Test (CTt) proposto por Román-González *et al.* (2017).

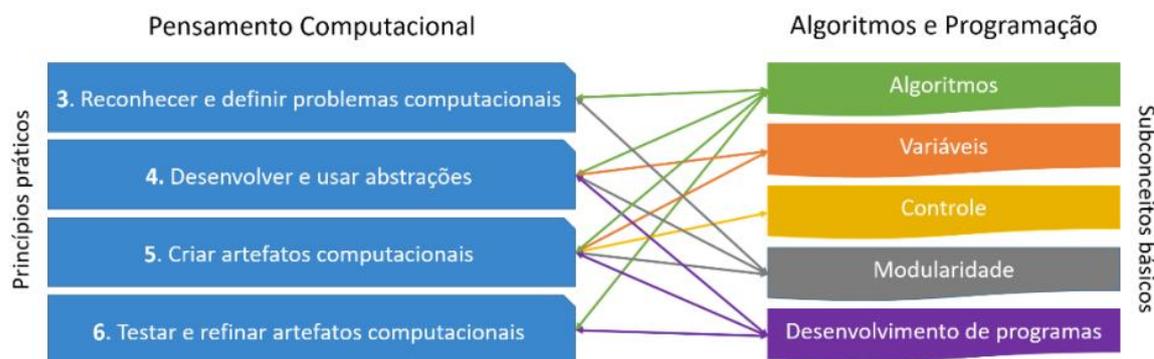
Na presente pesquisa, definiu-se pela adoção da ferramenta web CodeMaster enquanto instrumento de avaliação do pensamento computacional com base nos artefatos de código (aplicativos) criados pelos alunos. Assim, durante as 15 horas/aula desenvolvidas em cada turma, os participantes foram desafiados a construir aplicativos e encaminhar o código-fonte ao pesquisador para posterior análise no CodeMaster com o propósito de se obter uma avaliação do desenvolvimento do PC.

O CodeMaster surgiu a partir de pesquisas da iniciativa Computação na Escola<sup>27</sup> do INCod/INE/UFSC com parceria do IFSC. A ferramenta é gratuita e foi desenvolvida com o intuito de facilitar a avaliação dos alunos, em um contexto de aprendizagem baseada em problemas, sendo capaz de analisar de forma automatizada aplicativos construídos com App Inventor e Snap! (CODEMASTER, 2020). Em sua avaliação, o CodeMaster considera as competências do PC relacionadas a algoritmos e programação, analisando a resolução de tarefas de programação abertas e complexas (que podem ter mais de uma solução correta). A ferramenta se baseia nas práticas 3-6 do K-12 *Computer Science Framework* (CSTA, 2016) que apresenta as competências que os alunos devem ter ao final do Ensino Médio, relacionando os princípios práticos do PC com os conceitos de algoritmos e programação, vide Figura 10 (ALVES; WANGENHEIM; HAUCK, 2019).

---

<sup>27</sup> Mais informações em: <http://www.computacaonaescola.ufsc.br>.

**Figura 10** – Relação entre as práticas do pensamento computacional e o conceito (e subconceitos) de algoritmos e programação



Fonte: ALVES; WANGENHEIM; HAUCK, 2019.

O modelo utilizado pelo CodeMaster para avaliar a aprendizagem se baseia no uso de rubrica. A rubrica consiste em um conjunto de critérios com uma escala de níveis de desempenho (Quadro 1) utilizados para medir se os indicadores de aprendizagem foram atingidos a fim de avaliar o grau de competência do PC ao analisar os artefatos de software criados pelos alunos (ALVES, 2019).

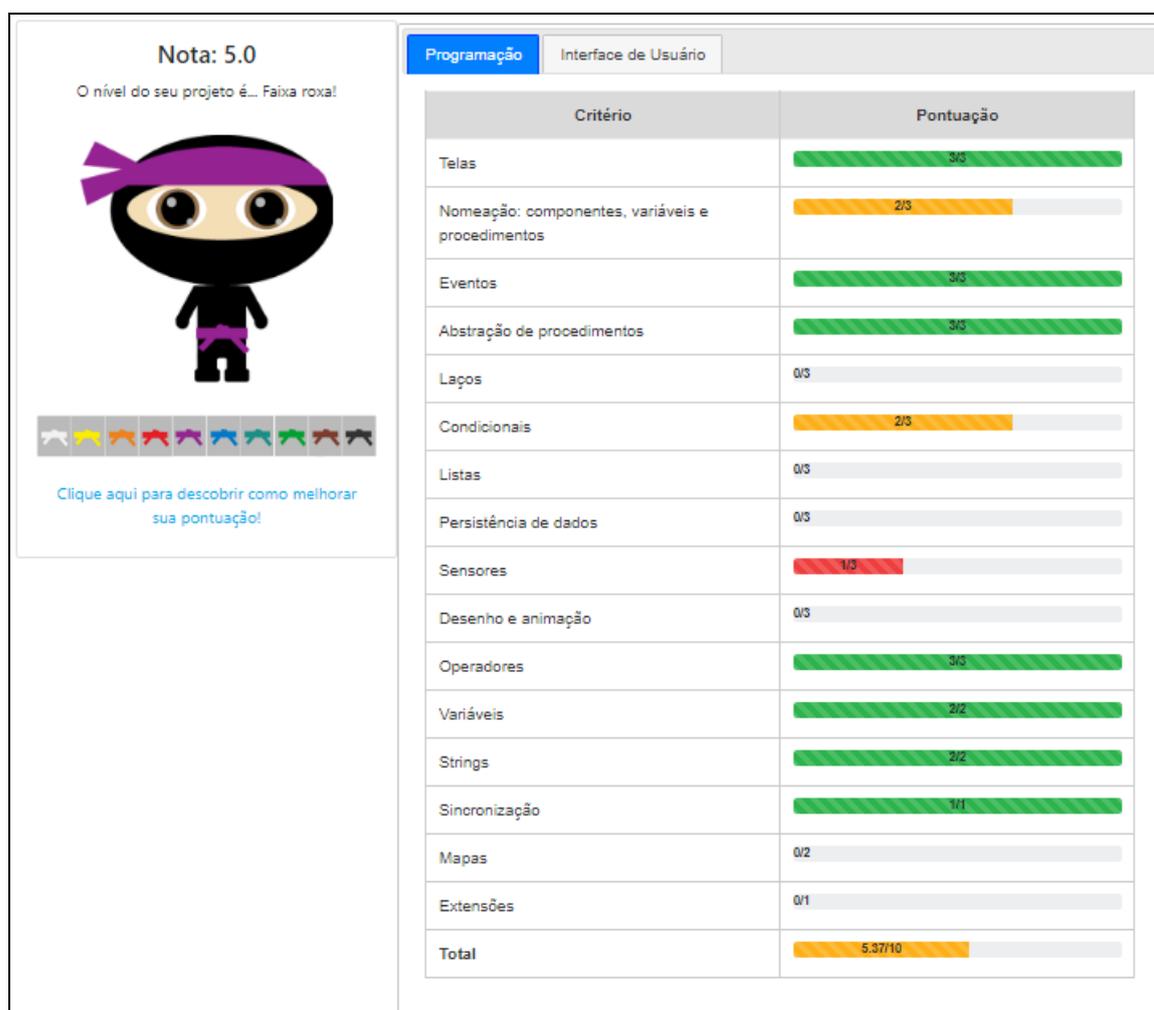
**Quadro 1** – Detalhamento da rubrica CodeMaster para App Inventor

Item	0 pontos	1 ponto	2 pontos	3 pontos
Operadores	Não usa operadores	Usa operadores aritméticos.	Usa operadores relacionais.	Usa operadores booleanos (lógicos).
Variáveis	Sem uso de variáveis.	Modificação ou uso de variáveis predefinidas.	Criação e operação com variáveis	-
Strings	Sem uso de strings.	Uso de string para alterar textos de componentes.	Criação e operação com strings.	-
Nomeação	Menos do que 10% dos nomes são alterados do padrão.	10 a 25% dos nomes são alterados do padrão.	De 26 a 75% dos nomes são alterados do padrão.	Mais de 76% dos nomes são alterados do padrão.
Listas	Não usa listas.	Usa uma lista unidimensional.	Usa mais de uma lista unidimensional.	Usa uma lista de tuplas (map).
Persistência de dados	Não há persistência quando o app é fechado.	Usa persistência em arquivo.	Usa um banco de dados local do App Inventor.	Usa uma base de dados web.
Eventos	Nenhum manipulador de evento é usado (Ex.: On click).	1 tipo de manipulador de eventos é usado	2 tipos de manipuladores de eventos são usados	Mais de 2 tipos de manipuladores de eventos são usados
Laços	Não usa laços	Usa "While" (laço simples)	Usa "For each" (variável simples)	Usa "For each" (item de lista)
Condicionais	Não usa condicionais.	Usa apenas "if"s	Usa apenas "if then else".	Usa um ou mais "if - else if".
Sincronização	Sem uso de temporizador para sincronização.	Uso de temporizador para sincronização.	-	-
Abstração	Não existem procedimentos.	Existe exatamente um procedimento e sua chamada.	Existe mais de um procedimento.	Existem procedimentos tanto para organização quanto para reuso.
Extensões	Sem uso de comandos de extensões.	Uso de comandos de extensões	-	-
Sensores	Sem uso de sensores.	Usa um tipo de sensor.	Usa 2 tipos de sensores.	Usa mais de 2 tipos de sensores
Desenho e Animação	Sem uso de desenho e animação	Uso de área sensível ao toque.	Uso de animação com bolinha predefinida.	Uso de animação com imagem.
Mapas	Sem uso de comandos de mapas.	Uso do comando de mapa.	Uso de comandos de marcadores de mapas.	-
Telas	Apenas uma tela com componentes visuais e que seu estado não se altera com a execução do app.	Apenas uma tela com componentes visuais que se alteram com a execução do app.	Pelo menos duas telas e uma delas altera seu estado com a execução do app.	Duas ou mais telas e pelo menos 2 delas alteram seus estados com a execução do app.

Fonte: ALVES; WANGENHEIM; HAUCK, 2019.

Através do sistema web CodeMaster, é possível validar as habilidades adquiridas em relação ao pensamento computacional por meio da importação do código-fonte da aplicação desenvolvida. Após a importação, a ferramenta analisa e conta a frequência de elementos de código (estruturas de controle, dados, interação, variáveis, etc.) e calcula a pontuação (de 0 a 40) usando conceitos de programação e habilidades do PC. Por fim, é calculada a nota, numa escala de 0 a 10, com base na pontuação total e determina-se a “faixa ninja”<sup>28</sup> como sendo o grau de qualificação do aplicativo levando em consideração essa nota, conforme a Figura 11. A ferramenta apresenta ainda uma pontuação individualizada de cada um dos conceitos avaliados, possibilitando identificar as possíveis habilidades atendidas ou negligenciadas durante a programação (GRESSE VON WANGENHEIM *et al.*, 2018).

**Figura 11** – Resultado da avaliação de um projeto no CodeMaster



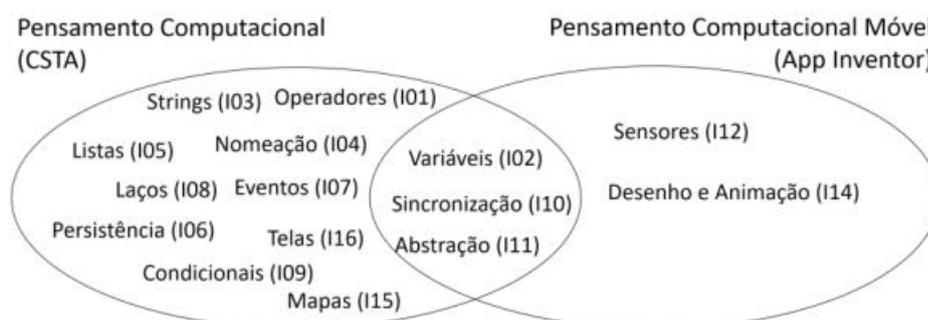
Fonte: CODEMASTER, 2020.

<sup>28</sup> Graduação progressiva composta pela seqüência de cores: branca, amarela, laranja, vermelha, roxa, azul, turquesa, verde, marrom e preta.

Além da avaliação singular, o CodeMaster disponibiliza ainda um “modo professor” onde, através de um cadastro prévio, é possível aferir um conjunto de projetos de uma turma inteira, com a apresentação do resultado individualizado do projeto de cada aluno. Esse processo de avaliação automatizada das atividades práticas de programação atenua possíveis inconsistências ou vieses de uma avaliação manual, além de diminuir consideravelmente o tempo de análise.

Apesar da abordagem da ferramenta se reduzir a verificação do código-fonte do aplicativo para detectar a presença de comandos de programação, inferindo a aprendizagem das competências do pensamento computacional por meio de indicadores previamente definidos, sua aplicação se mostra válida por estar alinhada a proposta de desenvolvimento do PC apresentada em CSTA (2016) e contemplar a rubrica de avaliação do pensamento computacional móvel proposta por Sherman e Martin (2015), conforme as suas subdimensões apresentadas na Figura 12.

**Figura 12** – Subdimensões dos itens da rubrica CodeMaster para App Inventor



Fonte: ALVES; WANGENHEIM; HAUCK, 2019.

O CodeMaster também teve a sua qualidade avaliada e comprovada em termos de corretude, usabilidade, confiabilidade e validade de construto do ponto de vista de pesquisadores no contexto do ensino de computação na Educação Básica, além de ter demonstrado a sua eficácia ao avaliar todos os aplicativos públicos (88.812) disponíveis na Galeria do App Inventor até maio de 2018 (ALVES, 2019).

#### 4. PRODUTO EDUCACIONAL

Neste capítulo é apresentado o artefato elaborado enquanto Produto Educacional do mestrado profissional, podendo ser considerado a principal produção desta pesquisa. Sua organização didática, com conteúdos, conceitos, atividades e materiais complementares, foi validada em condições reais de sala de aula (online) com o respaldo do referencial teórico metodológico apresentado anteriormente. Sua composição é apresentada nas seções seguintes e o conteúdo integral se encontra disponibilizado no Apêndice A.

A elaboração de um Produto Educacional, segundo Silva *et al.* (2019), parte de uma demanda advinda do campo de atuação da instituição de educação profissional e tecnológica que irá produzi-lo. Assim, o ambiente influencia o que será criado, uma vez que a instituição verifica a necessidade de transformar em artefato científico e tecnológico um conhecimento que possa suprir essa demanda. Para isso, o produto deve responder a um problema de pesquisa com um caráter de atender, pelo menos parcialmente, uma necessidade que se apresenta deficitária até o momento. Assim, a questão demandada para a elaboração deste estudo e, conseqüentemente, do Produto Educacional, conforme já discutido nos capítulos anteriores, diz respeito às dificuldades que os alunos ingressantes apresentam com relação ao ensino de programação. Esse problema, por sua vez, é reflexo da falta de processos de ensino e aprendizagem de computação na educação básica brasileira, especificamente do pensamento computacional.

O artefato produzido na tentativa de melhorar esse contexto diz respeito a uma sequência didática pautada nos Três Momentos Pedagógicos visando auxiliar no ensino de programação e no desenvolvimento do pensamento computacional por meio da construção de aplicativos para dispositivos móveis na ferramenta App Inventor. Acredita-se que a partir da compreensão dos conteúdos abordados neste produto, o aluno irá ampliar a sua habilidade de pensamento lógico, a criatividade e

a capacidade de solucionar problemas reais.

A partir da metodologia apresentada no capítulo 3, este produto foi construído com embasamento na literatura científica e com possibilidade de aplicação imediata junto ao público para o qual se destina, que podem ser alunos ingressantes de disciplinas introdutórias de programação em qualquer nível de ensino, obviamente com as devidas adaptações aos diferentes contextos. Além de poder ser empregada na disciplina de algoritmos, por exemplo, para contemplar o conteúdo previsto em sua ementa, a SD também pode ser utilizada por educadores em atividades complementares ou em cursos de formação extracurriculares, não estando limitada a essas alternativas.

Essa flexibilidade se deve, além dos fatores já discutidos, a adoção de um ambiente de desenvolvimento (App Inventor) que implementa uma linguagem de programação visual baseada em blocos (VPL)<sup>29</sup>. Através dos componentes visuais da VPL facilita-se a codificação, pois não é requerida uma estrutura básica prévia para a correta execução do algoritmo, diferente das linguagens textuais tradicionais. A representação do abstrato com o concreto através dos blocos auxilia na compreensão da solução que está sendo construída. Nessa perspectiva, Souza, Batista e Barbosa (2016) afirmam que existe uma tendência de pesquisas em ferramentas de visualização para minimizar as dificuldades dos alunos no ensino e na aprendizagem de programação, visto que facilitam o desenvolvimento de software.

#### **4.1 Website da sequência didática**

Como já expressado anteriormente, a sequência didática é apresentada no formato de um *website*, com o endereço eletrônico: “<https://sites.google.com/view/pceappinventor>”, sob o título de “Sequência didática para o desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor”. Sua elaboração partiu da aplicação do questionário de percepção inicial enviado por e-mail aos alunos. Com o suporte das respostas, iniciou-se o planejamento das atividades a serem desenvolvidas. Já o embasamento teórico se deu a partir do referencial construído nesta pesquisa e, também, na indicação de bibliografia básica e complementar presente no componente curricular

---

<sup>29</sup> Mais informações podem ser encontradas na seção 3.4 deste trabalho.

de algoritmos do projeto pedagógico do curso de Licenciatura em Computação (INSTITUTO FEDERAL FARROUPILHA, 2014).

Cabe destacar ainda que a construção da sequência didática ocorreu anteriormente às aulas, mas sofreu alterações de forma gradativa na medida em que os encontros eram realizados, procurando adaptar o material conforme o ritmo de aprendizagem dos alunos e o seu feedback com relação as práticas desenvolvidas. Também, por se tratarem de turmas diferentes e em distintas fases do componente curricular, foi necessário adequar o conteúdo a realidade de cada Campus, o que é melhor debatido no próximo capítulo desta dissertação.

Assim, o *website* do produto foi desenvolvido no Google Sites por simplificar a criação e edição do conteúdo a partir de um layout de grade que reorganiza automaticamente o design para que tenha uma aparência agradável independentemente do dispositivo utilizado para o acesso, seja computador, tablet ou smartphone (GOOGLE, 2021). Além disso, a ferramenta possibilita o compartilhamento do site com qualquer pessoa, garantindo o acesso público ao conteúdo sem requerer a aquisição de um registro de domínio para tal. Por outro lado, apresenta algumas restrições quanto a personalização do layout e a inclusão de componentes de terceiros, por exemplo, o que não limitou a elaboração do produto, permitindo a incorporação de todo conteúdo necessário para a compreensão da sequência didática.

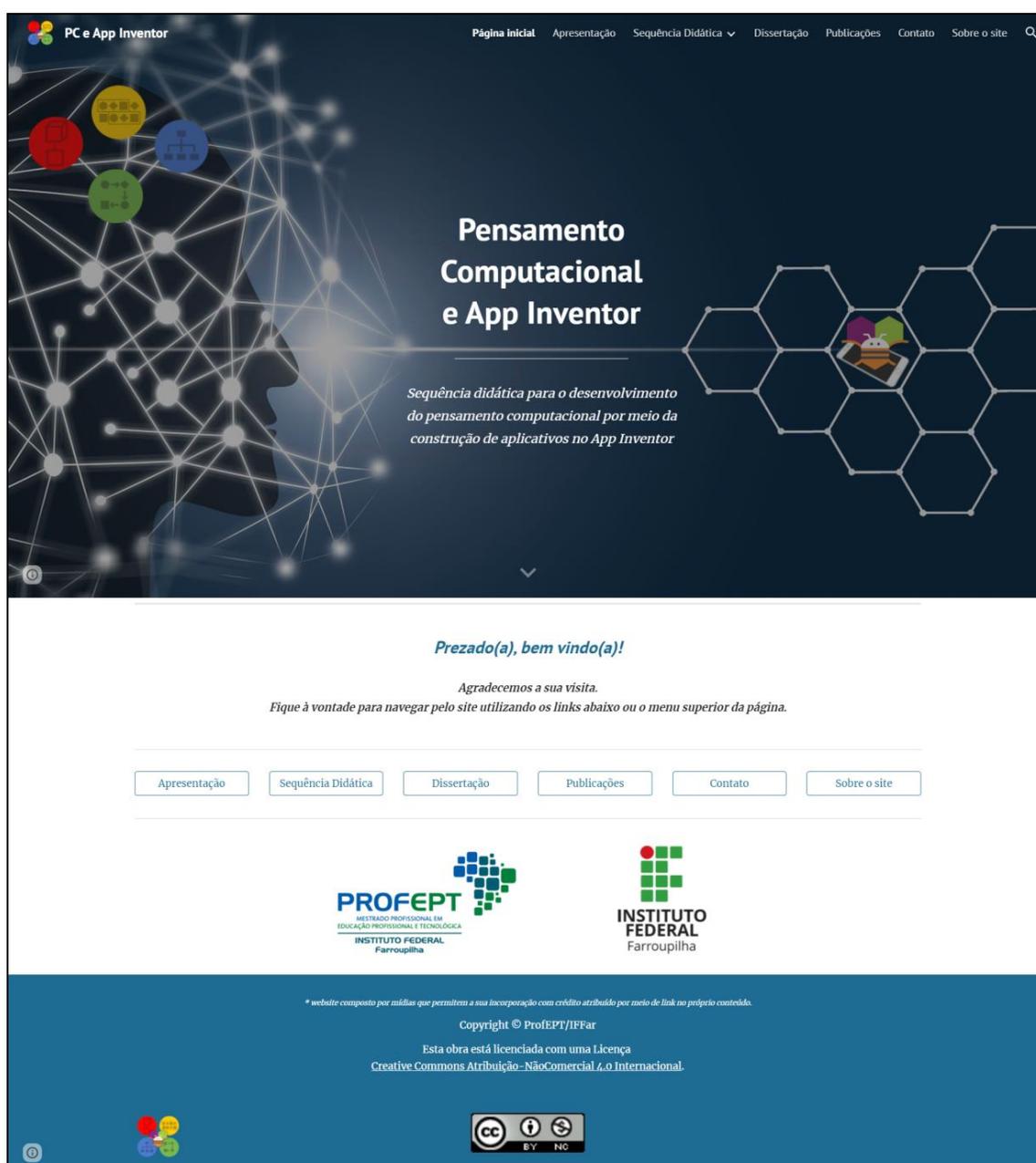
#### **4.1.1 Seção “Página inicial”**

A página inicial do *website* apresenta uma imagem (Figura 13) composta por diferentes elementos que procuram representar, de um lado, o pensamento computacional por meio dos seus quatro pilares (abstração, reconhecimento de padrões, decomposição e algoritmos) e, do outro, mas de forma interligada, uma colmeia com a abelha símbolo do App Inventor. O nome da página é exposto abreviado no topo esquerdo e de forma completa no centro, com o título do produto logo abaixo. Assim, evidencia-se a finalidade central da SD.

A seguir, é apresentada uma mensagem de boas-vindas com os links das seções presentes no *website*, que também podem ser acessadas com um clique a partir do menu de navegação localizado na parte superior direita da página. As seções estão dispostas da seguinte forma: “Página inicial”, “Apresentação”, “Sequência Didática”, “Dissertação”, “Publicações”, “Contato” e “Sobre o site”, tendo

seu conteúdo abordado mais a frente. Abaixo, no rodapé presente em todas as seções, constam as logomarcas do polo do ProfEPT, ao qual este trabalho se vincula, e do IFFar, com os respectivos links para as suas páginas institucionais. Por fim, a licença de uso *Creative Commons*<sup>30</sup> da obra que permite o seu compartilhamento e adaptação, desde que atribuídos os devidos créditos e sem fins comerciais, e o ícone do website, também composto pelos pilares do PC e a logomarca do App Inventor, que aponta para a página inicial.

**Figura 13** – Página inicial do *website* do Produto Educacional



Fonte: Elaborado pelo autor, 2021.

<sup>30</sup> Mais informações em: <https://creativecommons.org/licenses/by-nc/4.0>.

### 4.1.2 Seção “Apresentação”

Na seção seguinte, denominada “Apresentação”, expõe-se a intencionalidade do produto educacional enquanto um espaço pensado e constituído no intuito de compartilhar a sequência didática com professores da Educação Profissional e Tecnológica, bem como demais interessados no tema. Também é listado o programa, a instituição e a linha de pesquisa do estudo, a sua justificativa, o objetivo geral e os específicos, conforme a Figura 14.

Figura 14 – Apresentação do *website* do Produto Educacional



PC e App Inventor

Página inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

## Apresentação

Este espaço foi pensado e constituído no intuito de compartilhar com professores da Educação Profissional e Tecnológica, bem como demais interessados no tema, uma [sequência didática](#) que visa proporcionar o desenvolvimento do pensamento computacional nos educandos, além da introdução dos conceitos básicos de programação, através da construção de aplicativos móveis no App Inventor.

A sequência didática se constitui como Produto Educacional de uma pesquisa de mestrado do Programa de Pós-graduação em Educação Profissional e Tecnológica - ProfEPT, ofertado pelo campus Jaguari do Instituto Federal Farroupilha, enquadrada na linha de pesquisa 2–Organização e Memórias dos Espaços Pedagógicos em Educação Profissional e Tecnológica, a medida que se destina a identificar as dificuldades dos alunos nas disciplinas de programação e a procurar uma alternativa ao método de ensino tradicional.

Aqui é possível encontrar todas as informações relativas a pesquisa realizada junto a um grupo de alunos do curso superior de Licenciatura em Computação, especialmente a sequência didática utilizada durante a disciplina de Algoritmos para proporcionar o desenvolvimento do pensamento computacional e a introdução dos conceitos básicos de programação.

### Justificativa

A resolução de problemas de maneira lógica é uma das habilidades a serem desenvolvidas a partir do ensino de programação. Por meio da programação o aluno amplia a capacidade de organizar e memorizar ideias, desenvolvendo o pensamento computacional na solução de problemas reais. No entanto, é comum discentes enfrentarem dificuldades para desenvolver o pensamento computacional e, conseqüentemente, as habilidades necessárias para programar, o que acaba por afetá-los, levando à desmotivação, reprovação e, em alguns casos, a evasão dos cursos de computação.

No intuito de auxiliar na mitigação desse problema, o que se apresenta como a justificativa deste trabalho, pretende-se compreender como a tecnologia móvel, especificamente a construção de aplicativos, pode auxiliar no ensino de programação na Educação Profissional e Tecnológica e no desenvolvimento do pensamento computacional.

### Objetivos

A pesquisa teve por **objetivo geral** identificar as contribuições e/ou limites da construção de aplicativos móveis com o App Inventor, por meio de uma sequência didática pautada nos 3MP, para a aprendizagem de programação, bem como para o desenvolvimento do pensamento computacional.

Quanto aos **objetivos específicos**, se pretendeu:

- Analisar estudos sobre o desenvolvimento do pensamento computacional e as relações com a formação integrada;
- Identificar as dificuldades apresentadas pelos alunos no ensino de programação;
- Desenvolver uma sequência didática com enfoque na programação para dispositivos móveis pautada nos Três Momentos Pedagógicos e identificar as contribuições e limites no aprendizado de algoritmos;
- Analisar as possíveis contribuições da tecnologia móvel a partir da construção de aplicativos para o aprendizado de programação e o desenvolvimento do pensamento computacional.

Fonte: Elaborado pelo autor, 2021.

### 4.1.3 Seção “Sequência Didática”

A seção intitulada “Sequência Didática” contém o cerne do website, ou seja, constitui o Produto Educacional resultante da pesquisa de mestrado e tem por objetivo encadear e articular as atividades de aprendizagem ao longo da disciplina de Algoritmos. A mesma está dividida em cinco subseções, da aula 1 à aula 5, que podem ser acessadas através dos links presentes na parte inferior da seção ou pelo menu suspenso no topo da página.

Cada uma das aulas se divide em três etapas, conforme preconizam os Três Momentos Pedagógicos, e traz aspectos a respeito do ensino de programação para dispositivos móveis utilizando uma linguagem de programação visual em blocos, o App Inventor, procurando apresentar um caminho possível para o desenvolvimento do pensamento computacional.

Destaca-se que o movimento previsto nos 3MP exigiu planejamento e foi complexo de ser adotado devido ao seu formato multifacetado. Primeiramente, havia a intencionalidade de conduzi-lo a partir da construção de um aplicativo que abordasse algum problema emergido da realidade dos (pelos) alunos. Porém, observou-se que seria difícil contemplar os variados conteúdos presentes na ementa da disciplina de algoritmos com a construção de um único aplicativo. Ou então, se admitida a construção de vários aplicativos, conforme o interesse de cada aluno, dificilmente haveria sincronia entre o andamento das atividades. Além disso, o acompanhamento e a orientação docente (do pesquisador) também seriam comprometidos, tudo isso dentro do curto espaço de tempo das aulas e com a interação limitada em função do formato remoto. Sendo assim, optou-se por realizar as problematizações (primeiro MP) com base na aplicação dos próprios conteúdos em situações do dia a dia dos estudantes. Essa aproximação facilitou a compreensão dos conceitos teóricos previstos no componente curricular e o desenvolvimento dos aplicativos durante as aulas (segundo MP), e possibilitou o exercício de recontextualização (terceiro MP) das situações problematizadoras utilizando o conhecimento científico.

Ressalta-se, ainda, que a SD proposta foi elaborada levando em conta um determinado público, sendo assim, sua utilização com outras turmas exige uma releitura e adaptações. Ainda nessa seção, os 3MP são apresentados do ponto de vista teórico enquanto ferramenta pedagógica para estruturação das aulas, vide

Figura 15.

Figura 15 – Seção “Sequência Didática” do *website*

PC e App Inventor

Página inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

## Sequência Didática

A sequência didática (SD) abaixo constitui o Produto Educacional resultante da pesquisa de mestrado e tem por objetivo encadear e articular as atividades de aprendizagem ao longo da disciplina de Algoritmos.

A mesma está organizada em 5 aulas, cada qual dividida em 3 etapas, conforme preconizam os Três Momentos Pedagógicos, e traz aspectos a respeito do ensino de programação para dispositivos móveis utilizando uma linguagem de programação visual em blocos, o App Inventor.

A sequência didática tem o papel de indicar a função que cada uma das atividades possui, procurando apresentar um caminho possível para o desenvolvimento do pensamento computacional e a aprendizagem de programação a partir da construção de aplicativos móveis, deixando claro a pertinência e a ênfase a ser atribuída a cada conteúdo. Porém, a SD aqui proposta foi elaborada levando em conta um determinado público, sendo assim, sua utilização com outras turmas exige uma releitura e adaptações.

Em seguida, conheça um pouco mais a respeito dos Três Momentos Pedagógicos utilizados enquanto ferramenta pedagógica para estruturação das aulas e acesse a sequência didática dividida em cinco etapas.

### Três Momentos Pedagógicos (3MP)

Originalmente, os Três Momentos Pedagógicos foram pensados enquanto estruturantes do contexto das aulas, ou seja, sendo utilizados para estruturar currículos críticos, discutindo o que é ensinado e não apenas como é ensinado (MUNCHEN, 2010). Em nossa proposta, nos limitamos a utilização dos 3MP à metodologia de trabalho de sala de aula (estruturação das aulas) numa visão mais reduzida e simples de uma ferramenta didático-pedagógica, articulando os momentos com o intuito de ensinar um determinado conteúdo com a perspectiva de diálogo entre os participantes.

Partindo desse pressuposto, Muenchen e Delizoicov (2014) caracterizam os 3MP em: Problematização Inicial, Organização do Conhecimento e Aplicação do Conhecimento. No caso do terceiro momento pedagógico, aqui o referimos como de “Recontextualização do Conhecimento”, por considerarmos que o termo ‘aplicação’ deixa margem ao risco de incorrer na racionalidade técnica, num caráter positivista e simplista.

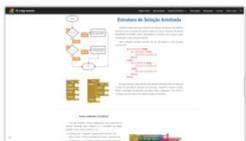
- 1) Problematização inicial (PI):** por meio de situações reais que os alunos experienciam, desafiá-los a expor o que pensam para entendê-los melhor e, assim, problematizar e criar a necessidade de conhecer o que ainda não conhecem. Situações problematizadoras são apresentadas, visando a discussão com base nos conhecimentos prévios dos alunos.
- 2) Organização do conhecimento (OC):** desenvolvimento de atividades, com a orientação docente, sobre os conhecimentos necessários para entender a problematização inicial. O professor aborda o conteúdo para aprofundamento e conhecimento sobre o tema discutido, gerando a percepção de outras explicações a partir do conhecimento científico.
- 3) Recontextualização do conhecimento (RC):** sistematizar o conhecimento elaborado pelo aluno, analisando e interpretando as situações iniciais e outras que possam ser compreendidas pelo mesmo conhecimento. O conhecimento científico é utilizado para reinterpretar as situações problematizadoras, ou seja, o aluno deve ser capaz de empregar o conhecimento em situações reais ou outras não diretamente ligadas ao motivo inicial.



**Aula 1**  
Pensamento Computacional e App Inventor



**Aula 2**  
Algoritmos no App Inventor



**Aula 3**  
Estruturas de Seleção no App Inventor



**Aula 4**  
Estruturas de Repetição no App Inventor



**Aula 5**  
Sensores e Funções no App Inventor

### Referências

MUNCHEN, C. A disseminação dos Três Momentos Pedagógicos: um estudo sobre práticas docentes na região de Santa Maria/RS. Tese de Doutorado (Programa de Pós-Graduação em Educação Científica e Tecnológica). Florianópolis: UFSC/PPGECT, 2010.

MUNCHEN, C.; DELIZOICOV, D. Os três momentos pedagógicos e o contexto de produção do livro “Física”. *Revista Ciência e Educação*, Bauri, v. 20, n. 3, p. 617-638, 2014.

Fonte: Elaborado pelo autor, 2021.

Para a elaboração das aulas, foi adotado um modelo de planejamento didático que contempla: o tema da aula; os conhecimentos e atividades que se espera desenvolver; a sua estruturação em problematização inicial, organização do conhecimento e recontextualização do conhecimento, conforme os 3MP; a revisão do desafio da aula anterior (aulas 2 a 4); o referencial teórico e o conteúdo a ser trabalhado; as atividades práticas de aprendizagem, principalmente no App Inventor; um desafio extraclasse a ser resolvido com o desenvolvimento de um aplicativo; a disponibilização da aula em formato PDF; oficinas, palestras e/ou cursos como material complementar para aprofundar os conhecimentos sobre a temática; e, por fim, as referências. Todas as aulas se baseiam nesse modelo, o que entendemos ser um facilitador para a compreensão dos objetivos de cada encontro e a replicação da experiência em diferentes contextos, mediante reelaboração.

#### *4.1.3.1 Aula 1*

Na primeira aula da SD, se espera conhecer o perfil dos alunos e seus conhecimentos prévios a respeito do tema; realizar uma contextualização a respeito do cenário atual da Computação na Educação Básica e Superior, procurando trazer aspectos da presença do PC nesses níveis de ensino; introduzir a conceituação do PC abordando a sua origem, as modalidades existentes e os pilares que o sustentam; realizar atividades práticas para cada um dos pilares procurando associá-los com atividades cotidianas realizadas pelos educandos; apresentar a ferramenta App Inventor, sua interface, funcionalidades e a comunicação com o dispositivo móvel e criar o primeiro aplicativo móvel e executá-lo no smartphone do aluno.

A organização da aula 1, com base nas premissas dos 3MP, tem como problematização inicial, a partir da introdução dos conceitos iniciais do pensamento computacional, fomentar a discussão a respeito da sua aplicação nas atividades rotineiras. Como o PC se aplica no dia a dia? Quais problemas reais podem ser resolvidos a partir do seu uso? Quais atividades o aluno realiza seguindo os preceitos do PC? Como introduzir o PC e otimizar as suas tarefas?

Também, a seguir da apresentação do App Inventor, demonstram-se exemplos de aplicativos desenvolvidos na ferramenta para que os alunos possam apontar possíveis melhorias e estimular ideias de novos aplicativos para situações problema atuais. Nesse sentido, por meio do diálogo, a problematização procurou

aproximar o PC à realidade dos alunos, fazendo com que compreendessem a sua importância tanto na execução quanto no planejamento de tarefas concretas, e a possibilidade de aplicar esse conhecimento no desenvolvimento de aplicativos móveis.

No momento de organização do conhecimento, são apresentados os quatro pilares do pensamento computacional, de forma teórica e prática, demonstrando a sua aplicabilidade em situações reais. Para isso, utilizam-se os recursos desenvolvidos pelo NAPEAD<sup>31</sup> da UFRGS, realizando uma atividade inicial para introduzir o PC e outras quatro atividades após abordar cada um dos pilares. Ainda, os alunos são orientados no acesso e configuração inicial do App Inventor em seus computadores para, em seguida, desenvolver o primeiro aplicativo na ferramenta e realizar a conexão com o seu smartphone para verificar a execução.

Para recontextualizar o conhecimento (terceiro MP), retoma-se a discussão sobre a utilidade do PC levantada na problematização inicial, agora procurando empregar os conceitos dos quatro pilares. A atividade inicial de introdução do PC (travessia do lobo, da ovelha e da couve<sup>32</sup>) é repetida, com base nos pilares, para verificar o conhecimento adquirido pelo aluno. Também é proposto o desafio de desenvolver uma interface replicando a tela de login de algum sistema utilizado pelo aluno (ex: sistema acadêmico SIGAA utilizado no IFFar) para que possa se familiarizar com o designer e componentes básicos do App Inventor.

Portanto, ao final da aula 1 o discente já possui uma visão geral sobre o pensamento computacional, a sua origem, definição e os pilares que o sustentam. Mesmo que esse conhecimento ainda seja um tanto abstrato, o aluno já conheceu, configurou e criou o primeiro aplicativo no App Inventor, o que lhe possibilita visualizar a importância do PC de forma concreta. Isso também por meio da análise de algumas aplicações desenvolvidas na ferramenta para solução de problemas reais, e da discussão sobre as funcionalidades contempladas e possíveis melhorias que poderiam ser implementadas nestas aplicações.

Na Figura 16 é possível observar parte da aula 1, especificamente o trecho que aborda os pilares do PC. Devido a sua extensão, optou-se por não exibir todo o conteúdo da neste capítulo, assim como das demais aulas, tendo sido disponibilizadas em sua íntegra no Apêndice A.

---

<sup>31</sup> Mais informações em: <https://www.ufrgs.br/napead>.

<sup>32</sup> Disponível em: <https://www.ufrgs.br/napead/projetos/pensamento-computacional/introducao>.

Figura 16 – Segmento da aula 1 sobre os pilares do pensamento computacional

**PC e App Inventor** | Página Inicial | Apresentação | Sequência Didática | Dissertação | Publicações | Contato | Sobre o site

## Sequência Didática Aula 1 Pensamento Computacional e App Inventor

### Pilares do Pensamento Computacional

Ao abordar o pensamento computacional, seja de maneira conceitual ou com intenção prática de solução de problemas, é comum dividi-lo em pilares ou dimensões, ou também chamadas de bases do PC. Porém, assim como na definição do termo, não existe um consenso na literatura quanto aos pilares que compõem o pensamento computacional.

Segundo a SBC (2018), o PC envolve abstração, análise de informações e automação de soluções, sendo essas as três habilidades primordiais para a construção de uma solução algorítmica capaz de resolver um problema. No entanto, a maior parte dos estudos relacionados ao tema apresentam quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos.

**Decomposição**

Processo no qual o problema complexo é dividido em tarefas menores e mais fáceis de serem gerenciadas. A separação em partes menores otimiza a compreensão do problema, o desenvolvimento da solução e a avaliação do resultado e a identificação de erros.

**Reconhecimento de Padrões**

Consiste em identificar similaridades entre os problemas. Possibilita a criação de modelos genéricos que podem ser usados para a construção de soluções mais rápidas e eficientes com base em problemas resolvidos anteriormente.

**Abstração**

Corresponde em simplificar a realidade a partir da criação de uma representação ou modelo do que se está tentando resolver apenas com os elementos relevantes, ou seja, ignorando detalhes desnecessários, tornando os problemas mais fáceis de serem solucionados.

**Algoritmo**

Constitui-se em uma sequência de passos ou regras finita a ser seguida para a resolução do problema. As instruções lógicas contidas em um algoritmo, se seguidas com precisão, levam a solução do problema inicial, bem como podem resolver problemas semelhantes.

**Vamos entender na prática?**

Desafie-se explorando cada um dos pilares do pensamento computacional através das atividades práticas abaixo desenvolvidas pelo [NAPEAD](#) – Produção Multimídia para a Educação da UFRGS.

Assuma o papel de detetive decompondo os fatores determinantes de um caso misterioso para montar a sua teoria.

**Começar**

Enquanto dono de uma locadora, identifique padrões nos clientes para saber exatamente qual a indicação de filme é mais apropriada.

**Começar**

Você é o segurança da portaria responsável por autorizar a entrada do público, examine cada fantasiado abstraindo as características em sua lista.

**Começar**

Auxilie o personagem a atravessar a rua em segurança montando um algoritmo que descreve a sequência de passos até chegar a sua loja.

**Começar**

Fonte: Elaborado pelo autor, 2021.

#### 4.1.3.2 Aula 2

A segunda aula da sequência didática tem por objetivo discutir como é estruturado o processo de resolução de um problema computacional; apresentar as partes que constituem um algoritmo; como se dá a escrita de um algoritmo no App Inventor; introduzir os conceitos de variável e operadores e criar um aplicativo de cálculo de média de notas pelo aluno no App Inventor.

Para isso, na problematização inicial, espera-se discutir com os alunos como se dá o processo de resolução de um problema computacional ou não. Quais os passos necessários desde o entendimento de um problema até a construção e validação de uma solução adequada? Quais problemas reais podem ser resolvidos ou otimizados a partir da construção de um algoritmo? Ainda, apresentam-se as partes que compõem um algoritmo e propõe-se que os alunos analisem o primeiro aplicativo desenvolvido na aula anterior procurando identificar tais partes e a importância de cada uma delas.

Na organização do conhecimento (segundo MP), as partes de um algoritmo são conceituadas e realiza-se uma atividade prática colaborativa no Google Jamboard<sup>33</sup> procurando esclarecer a composição de um algoritmo de cálculo de média entre dois números. Após a construção do quadro pelos alunos, fomenta-se a discussão a respeito de como os pilares do pensamento computacional podem ser aplicados na resolução desse cálculo. Também são apresentados os conceitos de variáveis e operadores e como ocorre a escrita de um algoritmo no App Inventor. Por fim, a partir do conhecimento construído, o aluno desenvolve um aplicativo para cálculo da média de duas notas no App Inventor.

No terceiro momento, de recontextualização do conhecimento, retoma-se a discussão sobre o processo de resolução de problemas computacionais, agora procurando empregar os conceitos apresentados na aula, a exemplo do questionamento: quais variáveis e operadores seriam utilizados para a construção dos algoritmos sugeridos na problematização inicial? Ainda, lança-se o desafio de incrementar o aplicativo de cálculo de média desenvolvido em aula, sendo capaz de obter a média de quatro notas.

Cabe destacar que na aula 2, além da estrutura de um algoritmo no App Inventor, também é apresentada a sua sintaxe em uma linguagem textual tradicional. Isso para auxiliar na compreensão e, principalmente, para que os alunos possam se familiarizar com a escrita adotada nas demais ferramentas e linguagens de programação que naturalmente passarão a utilizar no decorrer do curso de Licenciatura em Computação. Esse movimento também visa proporcionar aos alunos que já possuem conhecimentos prévios em programação, como é o caso da turma do Campus Santo Ângelo que estava na quarta aula da disciplina de

---

<sup>33</sup> Mais informações em: <https://jamboard.google.com>.

Algoritmos quando a pesquisa foi iniciada, a comparação entre a estrutura de uma linguagem textual e uma linguagem visual baseada em blocos (App Inventor). Na Figura 17 é exibido um segmento da aula 2 com o material que contempla esse aprendizado.

Figura 17 – Segmento da aula 2 sobre a escrita de algoritmos

PC e App Inventor

[Página inicial](#)
[Apresentação](#)
[Sequência Didática](#)
[Dissertação](#)
[Publicações](#)
[Contato](#)
[Sobre o site](#)

## Sequência Didática

### Aula 2

Algoritmos no App Inventor

## Escrita de Algoritmos

Independentemente da linguagem de programação utilizada, a escrita de um algoritmo deve sempre considerar:

- Sintaxe:** forma como os comandos devem ser escritos para que sejam interpretados e entendidos pelo dispositivo;
 

```
se (<expressão>) então
  <instrução>
```
- Semântica:** é o significado, conjunto de ações executadas a partir dos comandos;
 

```
se o valor da expressão for verdadeiro,
  a instrução será executada
```

Além disso, um algoritmo computacional costuma seguir uma estrutura básica com a definição de nome, declaração de variáveis e blocos de comandos com início e fim delimitados, a exemplo:

```
algoritmo_nome
  declaração de variáveis globais
  início_bloco
    bloco de comandos
  fim_bloco
fim_algoritmo
```

Podem existir um ou mais blocos de comandos em um algoritmo, cada qual contendo: declaração de variáveis locais, comentários, comandos de E/S e atribuição, estruturas de seleção e repetição e chamadas de funções (MANZANO; OLIVEIRA, 2016).



## Algoritmos no App Inventor

O fato do App Inventor ser uma linguagem de programação visual em blocos acaba por simplificar a construção de aplicativos, uma vez que abstrai parte da estrutura tradicional de um algoritmo.

No App Inventor os trechos de comandos são limitados pelos próprios blocos [1] que, por sua vez, são divididos entre **internos** [2] e de **componentes** [3], tendo a sua funcionalidade categorizada por diferentes cores (MIT, 2020).

### Variáveis

Variáveis são entidades destinadas a guardar uma determinada informação (números, palavras, arquivos, etc.) (MANZANO; OLIVEIRA, 2016). Cada variável possui um nome único e, no App Inventor, a sua declaração ocorre através de um bloco de inicialização [4]. Também existem blocos específicos para alterar [5] ou obter [6] o valor da variável enquanto o aplicativo é executado.

### Operadores

Operadores são utilizados para a construção de algoritmos no App Inventor e se dividem em (MIT, 2020):

- Aritméticos:** conjunto de símbolos que representam as operações básicas da matemática (soma, subtração, multiplicação, divisão, etc) [7];

- 1
- 2
- 3
- 4
- 5
- 6
- 7

#### 4.1.3.3 Aula 3

Na terceira aula da SD, se espera reforçar os conceitos apresentados na aula anterior revisando a construção do aplicativo de cálculo de média a partir do uso de uma estrutura condicional, compreender o que é e como utilizar uma estrutura de seleção e de seleção aninhada no App Inventor e criar um aplicativo para cálculo do índice de massa corporal.

Com base nos Três Momentos Pedagógicos, a organização da terceira aula tem como problematização inicial, a partir da discussão com os alunos, compreender como ocorre a utilização de uma estrutura de seleção no dia a dia. Em quais situações reais utilizamos uma estrutura condicional? Quais problemas podem ser resolvidos ou otimizados com suporte de uma estrutura de seleção? Quais critérios podem ser utilizados para a escolha (seleção)? Ainda, propõe-se aos alunos que analisem o aplicativo desenvolvido na aula anterior com o intuito de identificar como uma estrutura de seleção pode ser implementada para classificar o resultado da média como "aprovado" ou "reprovado", por exemplo.

No momento de organização do conhecimento, espera-se que o aluno compreenda o que é uma estrutura de seleção a partir da apresentação do seu conceito segundo a bibliografia presente na ementa da disciplina de algoritmos. Ademais, demonstra-se a sua construção no App Inventor através do incremento de uma nova funcionalidade no aplicativo de cálculo de média desenvolvido na aula anterior. Da mesma forma, conceitua-se a respeito de uma estrutura de seleção aninhada, demonstrando seu uso no App Inventor.

O movimento de recontextualização do conhecimento (terceiro MP) é realizado através de nova discussão sobre a aplicabilidade de uma estrutura de seleção, agora instigando os alunos a pensarem em problemas que poderiam ser solucionados a partir do desenvolvimento de aplicativos com essa funcionalidade. Como tarefa, lança-se o desafio de desenvolver um aplicativo para o cálculo do índice de massa corporal que apresente a faixa de desnutrição ou obesidade na qual o indivíduo se encontra.

A exemplo da aula anterior, ao final da aula 3, além de compreender a estrutura de um algoritmo condicional no App Inventor, os alunos também conhecem a sua escrita em formato textual, conforme pode ser observado na Figura 18 logo abaixo.

Figura 18 – Segmento da aula 3 sobre estruturas de seleção

PC e App Inventor
Página Inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

## Sequência Didática

### Aula 3

Estruturas de Seleção no App Inventor

### Estrutura de Seleção

Uma estrutura de seleção, também conhecida como estrutura condicional, permite o direcionamento da sequência de execução de um algoritmo de acordo com a aceitação ou não de certas condições (ASCENCIO; CAMPOS, 2012). As condições testadas são do tipo lógico (V ou F). A estrutura de seleção pode ser composta por:

- SE - condição;
- ENTÃO - o que fazer quando verdadeiro;
- SENÃO - o que fazer quando falso;

Se a estrutura de seleção conter somente a condição verdadeira (se/então), é considerada **simples**. Do contrário, havendo um bloco de comandos para a condição falsa (senão), é conhecida como **composta**.

Um algoritmo de seleção costuma ter o seguinte formato:

```

se <condição> então
  bloco de comandos (V)
senão
  bloco de comandos (F)
fim-se

```

No App Inventor, uma estrutura de seleção pode ser utilizada a partir de dois blocos distintos de controle, conforme a figura ao lado.

**Vamos entender na prática?**

Observe as figuras ao lado e reflita a respeito da aplicação de estruturas de seleção no seu dia-a-dia. Como se dá o processo de autoatendimento em um caixa eletrônico ou num caixa de supermercado? Como classificamos e selecionamos os produtos dispostos em uma gôndola? Como ocorre a separação do lixo que produzimos?

Agora, no App Inventor, vamos continuar os passos para a criação do aplicativo de cálculo de média.

- Nos blocos de "Controle", inserir a estrutura "se-então-senão" [1];
- Nos blocos de "Matemática", inserir o operador relacional ">=" e testar se a média é superior ou igual a 7 [2];
- Se verdadeiro, ajustar a propriedade "Texto" incluindo a descrição "Aprovado!". Para isso, inserir o bloco "juntar" do tipo "Texto" [3]. Alterar também a propriedade "CorDeTexto" para azul (bloco "Cores") [4];
- Se falso, ajustar a propriedade "Texto" incluindo a descrição "Reprovado!". Para isso, inserir o bloco "juntar" do tipo "Texto" [5]. Alterar também a propriedade "CorDeTexto" para vermelho (bloco "Cores") [6];
- Remover o bloco anterior que altera a propriedade "Texto" e arrastar o bloco "se-então-senão" para o evento "Clique" do "btnCalcular" [7];

Fonte: Elaborado pelo autor, 2021.

#### 4.1.3.4 Aula 4

Na quarta aula da sequência didática se espera reforçar os conceitos apresentados na aula anterior revisando a construção do aplicativo de índice de

massa corporal, compreender o que são e como utilizar diferentes estruturas de repetição no App Inventor e criar um aplicativo para calcular o fatorial de um número positivo.

A partir do diálogo com os alunos, pretende-se realizar a problematização inicial sobre como ocorre a utilização de uma estrutura de repetição no seu dia a dia. Para fomentar a discussão, são realizados os questionamentos: em quais situações reais utilizamos uma estrutura de repetição? Quais problemas podem ser resolvidos ou otimizados com o suporte de uma estrutura de repetição? Quais artefatos podem ser utilizados durante a sua aplicação para o controle do número de repetições, evitando a execução infinita?

O segundo momento pedagógico da aula 4, de organização do conhecimento, tem por finalidade compreender o que é uma estrutura de repetição nas suas diferentes formas mediante a apresentação do seu conceito e a demonstração da sua construção no App Inventor. Nessa etapa são desenvolvidos exemplos práticos no App Inventor com as estruturas de repetição “para” e “enquanto”. Ainda, apresenta-se a definição e os princípios que envolvem o uso de contadores para auxiliar no controle do número de vezes que uma mesma instrução ocorre.

Para sistematizar o conhecimento elaborado pelo aluno, no terceiro MP é retomada a discussão proposta na problematização instigando os alunos a pensarem em problemas que poderiam ser solucionados a partir do desenvolvimento de aplicativos com o uso de estruturas de repetição, apontando qual tipo de estrutura é mais adequada para cada situação e justificando o porquê do uso de uma ou outra. Para auxiliar na compreensão do conteúdo, lança-se o desafio de desenvolver um aplicativo no App Inventor para o cálculo do fatorial de um número, alertando o aluno de que só é possível realizar o cômputo com números positivos, o que sinaliza a necessidade de utilização de uma estrutura condicional, estudada na aula anterior, para a testagem do algoritmo informado pelo usuário.

Na Figura 19 é possível visualizar um trecho da quarta aula com a introdução às estruturas de repetição, perguntas e imagens que fomentaram a problematização inicial e a conceituação e utilização da estrutura “para” no App Inventor.

Figura 19 – Segmento da aula 4 sobre estruturas de repetição

PC e App Inventor

[Página Inicial](#)
[Apresentação](#)
[Sequência Didática](#)
[Dissertação](#)
[Publicações](#)
[Contato](#)
[Sobre o site](#)

## Sequência Didática

### Aula 4

#### Estruturas de Repetição no App Inventor

**Tabuada do 9**

9 x 1 = 9  
9 x 2 = 18  
9 x 3 = 27  
...  
9 x 10 = 90

### Estruturas de Repetição

São utilizadas em situações onde é necessário repetir uma tarefa mais de uma vez. Dessa forma, possibilitam a repetição de um conjunto de instruções sem que seja necessário escrevê-las várias vezes.

As estruturas de repetição podem ser classificadas em dois tipos:

- **Condicional** - repetição ocorre enquanto a condição lógica testada for verdadeira (ENQUANTO);
- **Incondicional** - tem um número pré-determinado de repetições (PARA);

**Vamos entender na prática?**

Observe as figuras ao lado e reflita a respeito da aplicação de estruturas de repetição no seu dia-a-dia. Como podemos explicar o processo que realizamos para nos alimentarmos? Ou então, para lavar a louça suja após uma refeição? Ainda, como realizamos o cálculo da tabuada de um número?

### Estrutura de Repetição PARA

A estrutura de repetição "Para" é ideal em situações onde o número exato de repetições é conhecido (ASCENCIO; CAMPOS, 2012). Seu funcionamento consiste em repetir a instrução enquanto uma variável de controle (var\_controle) for menor ou igual ao valor final (vlr\_fim). A variável de controle é iniciada através de um comando de atribuição (vlr\_ini) e é incrementada pelo valor da cláusula passo (<inc>), conforme a estrutura abaixo.

```

para <var_controle> = vlr_ini até vlr_fim passo <inc> faça
  bloco de comandos
fim-para

```

No App Inventor, uma estrutura de repetição "Para" pode ser utilizada a partir de um bloco de controle, conforme a figura ao lado, já composto por três blocos numéricos de matemática para inicialização da variável de controle, definição do valor final e do passo.

Fonte: Elaborado pelo autor, 2021.

#### 4.1.3.5 Aula 5

Na quinta e última aula da SD são retomados os conceitos apresentados nas aulas anteriores através de uma revisão geral; revisa-se também a construção do aplicativo de cálculo do fatorial (desafio); visa-se compreender o que é e como utilizar o sensor acelerômetro no App Inventor, como atribuir foco em componentes de interface e criar várias telas num único aplicativo; espera-se que os alunos aprendam como ocorre a escrita de algoritmos através da criação de procedimentos no App Inventor e resolvam diferentes problemas matemáticos desenvolvendo um

aplicativo móvel como desafio final.

Conforme a concepção do primeiro momento pedagógico, pretende-se problematizar com os alunos quais as suas percepções e as mudanças provocadas a partir da aplicação diária dos conhecimentos adquiridos nas aulas anteriores. Ainda, quais sensores do smartphone eles utilizam a partir dos aplicativos instalados em seus dispositivos e como acreditam que podem incorporar esses sensores no desenvolvimento de *apps* pelo App Inventor. Provocar a discussão a respeito de como partes de algoritmos podem ser reaproveitadas para evitar a rescrita das instruções e como podemos resolver problemas e automatizar tarefas repetitivas com tal técnica.

No momento de organização do conhecimento demonstra-se de forma conceitual e prática a utilização do sensor acelerômetro no desenvolvimento de um aplicativo no App Inventor, bem como realizar o controle de foco em diferentes componentes de interface. Apresenta-se o conceito de procedimento ou função e como utilizá-lo na escrita de algoritmos no App Inventor. Ainda, indica-se como se dá a criação e o fluxo de navegação entre diferentes telas em um mesmo aplicativo através do desenvolvimento de um exemplo prático.

No terceiro momento pedagógico é revista a argumentação sobre a repetição de partes de um algoritmo proposta na problematização. Os aplicativos desenvolvidos nas aulas anteriores são analisados procurando identificar blocos de comandos repetitivos que poderiam ser agrupados em uma função, ou conjuntos de instruções que, mesmo não duplicadas, poderiam ser dispostas em formato de procedimento para simplificar o problema, dividindo o algoritmo em partes menores.

Finalizando as atividades, é proposta aos alunos a resolução de diferentes problemas matemáticos através do desenvolvimento de um aplicativo móvel com base nos conhecimentos adquiridos no decorrer das aulas.

Como já abordado anteriormente, a construção da sequência didática foi orientada pelo progresso dos alunos em relação aos conteúdos estudados e as atividades realizadas. Em especial na aula 5, duas temáticas trabalhadas originaram-se em dúvidas e interesses dos alunos com relação a possibilidade de aplicação no App Inventor. Logo, tanto a utilização de múltiplas telas quanto o controle de foco de componentes visuais são temas não contidos no componente curricular de algoritmos, mas que despertaram a curiosidade dos discentes na medida em que foram se apropriando do ambiente do App Inventor, o que nos fez

incorporá-los ao conteúdo da aula. Abaixo (Figura 20) é possível observar um segmento da quinta aula com o material relativo à implementação de várias telas.

Figura 20 – Segmento da aula 5 sobre várias telas no App Inventor

**Várias Telas**

É comum aplicativos móveis se fazerem valer de mais de uma tela em função da limitação de tamanho da interface dos smartphones. O App Inventor permite a criação de várias telas num único aplicativo a partir do botão "Adicionar tela..." localizado na barra superior da janela de Designer.

Ao criar uma nova tela, é necessário lhe atribuir um nome que, posteriormente, será utilizado para a navegação através de blocos de "Controle". Para abrir as telas criadas, deve-se utilizar o bloco "abrir outra tela nomeDaTela", selecionando a tela desejada entre as criadas anteriormente.

Ao abrir uma tela, é recomendado fechá-la ao retornar à tela principal ou anterior através do bloco "fechar tela", afim de liberar memória do sistema.

**Vamos entender na prática?**

No App Inventor, seguindo no aplicativo de cálculo do fatorial, vamos criar uma nova tela denominada "Tela Inicial" que consistirá na tela de abertura do aplicativo com um um botão centralizado que possibilitará o acesso a tela de cálculo do fatorial criada anteriormente.

- Na janela de Designer, clicar no botão "Adicionar Tela..." e informar o nome "TelaInicial";
- Selecionar o componente "TelaInicial" e desmarcar a propriedade "TituloVisível";
- Arrastar para a tela um componente "OrganizaçãoVertical" e alterar:
  - AlinhamentoHorizontal = Centro:3
  - AlinhamentoVertical = Centro:2
  - CorDeFundo = Preto
  - Altura = Preencher principal...
  - Largura = Preencher principal...
- Arrastar para a tela um componente "Botão" e alterar:
  - Renomear = *btInicial*
  - CorDeFundo = Azul
  - FonteNegrito = V
  - TamanhoDaFonte = 18
  - Forma = arredondado
  - Texto = *Cálculo do Fatorial*
  - AlinhamentoDoTexto = centro:1
  - CorDeTexto = Branco
- Na janela de Blocos, nos blocos do "btInicial", inserir o evento "Clique" e, nos blocos de "Controle", inserir o evento "abrir outra tela nomeDaTela Screen1" [1];
- Na janela de Designer, alternar a tela para "Screen1", inserir um componente "Botão" e alterar:
  - Renomear = *btVoltar*
  - FonteNegrito = V
  - Texto = *Voltar*
  - AlinhamentoDoTexto = centro:1
- Na janela de Blocos, nos blocos do "btVoltar", inserir o evento "Clique" e, nos blocos de "Controle", inserir o evento "fechar tela" [2];

quando *btInicial*.Clique  
fazer abrir outra tela nomeDaTela Screen1

quando *btVoltar*.Clique  
fazer fechar tela

Fonte: Elaborado pelo autor, 2021.

#### 4.1.4 Seção "Dissertação"

Na seção seguinte, denominada "Dissertação", trata-se da publicação da pesquisa em formato PDF, passada a defesa junto a banca avaliadora e atendimento

das possíveis alterações sugeridas.

#### 4.1.5 Seção “Publicações”

Este espaço do *website* é dedicado às publicações geradas a partir desta pesquisa. Até o momento, foram publicados três artigos, um no Congresso Internacional de Educação e Tecnologias – CIET:EnPED 2020, sob o título de “Explorando o território da pesquisa-ação: categorização de estudos anteriores com foco no ciberespaço”<sup>34</sup>. O segundo, no I Seminário Regional Sul de Educação Profissional e Tecnológica - SEPT 2021, sob o título de “Formação integrada na Licenciatura em Computação: pensamento computacional e crítico para o ensino de programação”<sup>35</sup>. O terceiro artigo, na Revista Tecnologias Educacionais em Rede (ReTER) - Dossiê Educação Profissional e Tecnologias em Rede - v. 2, n. 4, 2021, sob o título de “Formação integrada na Educação Profissional e Tecnológica: pensamento computacional e crítico por meio do ensino de programação”<sup>36</sup>.

Também foi realizada a publicação do capítulo de livro “Trabalho como princípio educativo na Licenciatura em Computação: formação integrada, pensamento computacional e interação pedagógica”<sup>37</sup> na obra “Trabalho pedagógico na Educação Profissional e Tecnológica em diferentes contextos: desafios e reflexões – volume 2”. Além disso, no dia 17 de dezembro de 2021 ministramos a oficina intitulada “Os Três Momentos Pedagógicos e programação em blocos”<sup>38</sup> num momento formativo da equipe do projeto de iniciação tecnológica “Prorobótica: pensamento computacional na Educação Básica”, conforme é possível observar na Figura 21.

---

<sup>34</sup> Disponível em: <https://cietenped.ufscar.br/submissao/index.php/2020/article/view/1338>.

<sup>35</sup> Disponível em: <https://www.even3.com.br/anais/sept2021/331579-formacao-integrada-na-licenciatura-em-computacao--pensamento-computacional-e-critico-para-o-ensino-de-programacao>.

<sup>36</sup> Disponível em: <https://periodicos.ufsm.br/reter/article/view/67321>.

<sup>37</sup> Disponível em: <https://www.editoracrv.com.br/produtos/detalhes/35917-trabalho-pedagogico-na-educacao-profissional-e-tecnologica-em-diferentes-contextos-brdesafios-e-reflexoes-volume-2>.

<sup>38</sup> Disponível em: <https://sites.google.com/iffarroupilha.edu.br/prorobotica/forma%C3%A7%C3%A3o-da-equipe>.

Figura 21 – Seção “Publicações” do website

PC e App Inventor

Página inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

## Publicações

A presente pesquisa de mestrado nos levou a algumas publicações de artigos em eventos e periódico, além de um capítulo de livro. Também ministramos uma oficina num momento formativo junto a equipe de um projeto de iniciação tecnológica. Os trabalhos estão disponíveis em ordem cronológica nos links abaixo.

**CIET:EnPED**  
Congresso Internacional de Educação e Tecnologias - CIET:EnPED 2020

**Congresso Internacional de Educação e Tecnologias - CIET:EnPED 2020**

EXPLORANDO O TERRITÓRIO DA PESQUISA-AÇÃO: CATEGORIZAÇÃO DE ESTUDOS ANTERIORES COM FOCO NO CIBERESPAÇO

**SEMINÁRIO REGIONAL SUL**  
de Educação Profissional e Tecnológica - SEPT 2021  
MAIO 12, 13 E 14 ONLINE

**I Seminário Regional Sul de Educação Profissional e Tecnológica - SEPT 2021**

FORMAÇÃO INTEGRADA NA LICENCIATURA EM COMPUTAÇÃO: PENSAMENTO COMPUTACIONAL E CRÍTICO PARA O ENSINO DE PROGRAMAÇÃO

**TRABALHO PEDAGÓGICO NA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA EM DIFERENTES CONTEXTOS**

**TRABALHO PEDAGÓGICO NA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA EM DIFERENTES CONTEXTOS: desafios e reflexões – volume 2**

TRABALHO COMO PRINCÍPIO EDUCATIVO NA LICENCIATURA EM COMPUTAÇÃO: formação integrada, pensamento computacional e interação pedagógica

**ReTER**  
revista tecnologias educacionais em rede  
Universidade Federal de Santa Maria

**Revista Tecnologias Educacionais em Rede (ReTER) - Dossiê Educação Profissional e Tecnologias em Rede - v. 2, n. 4, 2021**

FORMAÇÃO INTEGRADA NA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA: PENSAMENTO COMPUTACIONAL E CRÍTICO POR MEIO DO ENSINO DE PROGRAMAÇÃO

**OFICINA: OS TRÊS MOMENTOS PEDAGÓGICOS E PROGRAMAÇÃO EM BLOCOS**  
DIA: 17/02/2021  
14 HORAS

**Oficina: Os Três Momentos Pedagógicos e programação em blocos**

Formação da equipe do projeto de iniciação tecnológica "Prorobótica: pensamento computacional na Educação Básica". Gravação da oficina disponível no site do projeto. Para mais informações, clique aqui.

**PROFEPT**  
MESTRADO PROFISSIONAL EM EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL  
Farroupilha

**INSTITUTO FEDERAL**  
Farroupilha

\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição- NãoComercial 4.0 Internacional.

#### 4.1.6 Seção “Contato”

A próxima seção intitulada “Contato” incorpora um questionário elaborado no Google Formulários para que o visitante possa relatar a sua concepção a respeito do Produto Educacional, contemplando as suas impressões quanto ao conteúdo da sequência didática, a forma de apresentação, a relevância, a possibilidade de utilização em outros contextos e sugestões de alteração ou melhoria. Também é disponibilizado um e-mail de contato caso o internauta deseje saber mais sobre a pesquisa, conforme a Figura 22.

Figura 22 – Seção “Contato” do website

PC e App Inventor

Página inicial Apresentação Sequência Didática Dissertação Publicações **Contato** Sobre o site

## Contato

Desde já agradecemos o contato através do formulário abaixo com o relato da sua concepção a respeito deste Produto Educacional. Fale-nos sobre as suas impressões, o que achou do conteúdo da sequência didática, a sua forma de apresentação em formato de *website*, se acredita que o material é relevante e se pode ser utilizado em outros contextos com diferentes turmas. Ainda, o que poderia ser alterado ou melhorado e as suas sugestões para serem incorporadas ao produto.

Caso queira saber mais sobre a pesquisa ou deseje entrar em contato com os autores, estamos à disposição através do e-mail: [rodrigothomas@gmail.com](mailto:rodrigothomas@gmail.com).

### Concepções - Produto Educacional

Faça login no Google para salvar o que você já preencheu. [Saiba mais](#)

\*Obrigatório

Qual a sua concepção a respeito deste Produto Educacional? Quais as suas impressões com relação ao conteúdo da sequência didática e a apresentação em formato de site? Você acredita que o material é relevante e pode ser utilizado em outros contextos com diferentes turmas? O que poderia ser alterado ou melhorado e quais as suas sugestões para serem incorporadas ao produto? \*

Sua resposta

Enviar Limpar formulário

Nunca envie senhas pelo Formulários Google.

Google Formulários Este conteúdo não foi criado nem aprovado pelo Google.

PROFEPT  
MESTRADO PROFISSIONAL EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL  
Farroupilha

INSTITUTO  
FEDERAL  
Farroupilha

\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença  
Creative Commons Atribuição-NãoComercial 4.0 Internacional.

CC BY NC

Fonte: Elaborado pelo autor, 2021.

#### 4.1.7 Seção “Sobre o site”

Nesta seção, conforme Figura 23, apresenta-se a finalidade do website (divulgação do Produto Educacional), bem como a identificação do programa e a instituição a qual se vincula esta pesquisa. Ademais, o currículo resumido do mestrando e do seu orientador, além de links para os respectivos currículos expandidos na Plataforma Lattes.

Figura 23 – Seção “Sobre o site”

PC e App Inventor

Página inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

## Sobre o site

Site desenvolvido para divulgação do Produto Educacional intitulado "*Sequência didática para o desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor*" apresentado ao Programa de Pós-Graduação em Educação Profissional e Tecnológica – ProfEPT do Instituto Federal Farroupilha - Polo Jaguari/RS - Brasil, pelo mestrando Rodrigo Thomas, sob orientação do Prof. Dr. Adão Caron Cambraia, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Demais informações relativas ao produto e à pesquisa podem ser encontradas na [dissertação](#) "*O Ensino de Programação na Educação Profissional e Tecnológica: desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor*".

---

**Rodrigo Thomas**

Mestrando do Programa de Pós-Graduação em Educação Profissional e Tecnológica (ProfEPT) – IFFAR Polo Campus Jaguari, especialista em Gestão e Governança da Tecnologia da Informação pelo Centro Universitário Senac e graduado em Sistemas de Informação pela Universidade Regional Integrada do Alto Uruguai e das Missões - URI Santo Ângelo. Analista de Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia Farroupilha - Câmpus Santo Ângelo.

[Currículo Lattes](#)




---

**Orientador: Prof. Dr. Adão Caron Cambraia**

Doutor e Mestre em Educação nas Ciências pela Universidade Regional do Noroeste do Estado do Rio Grande do Sul (Unijuí), especialista em Informática aplicada à Educação pela Universidade de Passo Fundo (UPF) e graduado em Informática pela Unijuí. Professor do Mestrado Profissional em Educação Profissional e Tecnológica (ProfEPT) do Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IFFar).

[Currículo Lattes](#)




---

**PROFEPT**  
MESTRADO PROFISSIONAL EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL  
Farroupilha

**INSTITUTO  
FEDERAL**  
Farroupilha

\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença  
Creative Commons Atribuição–NãoComercial 4.0 Internacional.



Fonte: Elaborado pelo autor, 2021.

## 5. RESULTADOS

Este capítulo destina-se a análise dos resultados da pesquisa. Primeiramente, são discutidos aspectos relativos à dinâmica das aulas, em seguida são apresentados os resultados da avaliação individualizada, na ferramenta CodeMaster<sup>39</sup>, dos aplicativos desenvolvidos pelos educandos no App Inventor em cada desafio. Ainda, são analisadas as respostas dos alunos com relação ao questionário de percepção final aplicado após as práticas da sequência didática. Por fim, algumas manifestações exteriorizadas pelos discentes durante os encontros online e as percepções do pesquisador.

### 5.1 Dinâmica das aulas

O planejamento inicial das aulas foi pensado num formato presencial, com a utilização de laboratórios de informática das instituições previamente configurados para o uso do App Inventor. Porém, com o advento da pandemia da Covid-19, as aulas passaram a ocorrer no formato remoto e, sem uma perspectiva de arrefecimento da crise sanitária, foi necessário repensar a proposta das aulas, agora no formato remoto.

Como já exposto anteriormente, a pandemia provocou o cancelamento das aulas no mês de março de 2020, com a retomada ocorrendo em julho, mas com o calendário abreviado, o que impossibilitou a realização das atividades com os alunos ingressantes daquele ano antes da finalização do primeiro semestre letivo. Em vista disso, optou-se por realizar a pesquisa junto as turmas de LC ingressantes em 2021 dos *campi* Santo Augusto e Santo Ângelo.

Aqui cabe uma ressalva quanto ao contexto de cada uma das turmas. No Campus Santo Augusto, a primeira aula da sequência didática ocorreu no dia 21 de junho, segunda-feira, na disciplina de Algoritmos, sendo que os quatro períodos do

---

<sup>39</sup> Mais informações podem ser encontradas na seção 3.5 deste trabalho e no site da ferramenta: [http://apps.computacaonaescola.ufsc.br:8080/i18n\\_pt](http://apps.computacaonaescola.ufsc.br:8080/i18n_pt).

turno da noite foram gentilmente cedidos (e acompanhados) pelo professor titular para a realização das atividades. Esse cenário se repetiu durante os demais encontros com a turma. Até então, os alunos haviam participado somente de uma aula da disciplina, ou seja, estavam no início do semestre. Os encontros seguintes ocorreram nos dias 28 de junho, 12, 19 e 26 de julho. Entre a segunda e a terceira aula, houve uma semana sem o desenvolvimento da sequência didática em função do afastamento do pesquisador por motivo de saúde de familiar.

Já no Campus Santo Ângelo, as práticas se iniciaram no dia 15 de julho, quinta-feira, também na disciplina de Algoritmos, porém apenas os últimos dois períodos do turno da noite foram cedidos pela professora titular. O que também ocorreu no segundo e no quarto encontro com a turma. Já o terceiro e o quinto, foram realizados em dias subsequentes aos encontros anteriores, sextas-feiras, na disciplina de Introdução à Informática. Antes do terceiro encontro, a professora titular realizou uma avaliação da disciplina, sendo também cedidos somente os últimos dois períodos para as práticas, a exemplo da disciplina de Algoritmos. No quinto encontro, a professora liberou os alunos para que pudessem estudar para as avaliações do final de semestre, o que possibilitou a realização do encontro durante todo turno da noite, apesar da baixa adesão dos alunos. Assim, os encontros com a turma de Santo Ângelo ocorreram nos dias 15, 22, 23, 29 e 30 de julho. Antes do primeiro encontro, a turma já havia participado de quatro aulas da disciplina de Algoritmos com atividades de programação realizadas na ferramenta Portugol Studio.

Essa dinâmica pode ser melhor compreendida nas seções seguintes pelos aplicativos desenvolvidos no App Inventor a partir dos desafios de cada aula, pelo número de respostas do questionário final e pela participação dos alunos nos encontros.

## **5.2 Avaliação dos aplicativos no CodeMaster**

Para promover o desenvolvimento do pensamento computacional, além dos conteúdos teóricos e das atividades práticas realizadas durante as aulas, ao final de cada encontro os alunos foram desafiados a construir aplicativos no App Inventor enquanto ação pertencente ao movimento de recontextualização do conhecimento (terceiro momento pedagógico). Aos alunos de ambas as turmas foi solicitada a postagem do código-fonte do aplicativo (arquivo no formato “.aia” exportado do App

Inventor) em uma tarefa criada no Sistema Integrado de Gestão de Atividades Acadêmicas - SIGAA. Porém, como no Campus Santo Ângelo os encontros foram realizados em aulas de disciplinas diferentes e o retorno dos alunos pelo sistema foi baixo, passou-se a solicitar que enviassem os aplicativos ao e-mail do pesquisador.

Apesar do processo de exportação do aplicativo no formato “.aia” ter sido demonstrado várias vezes em aula e estar presente no material disponibilizado aos alunos, alguns poucos acabaram enviando arquivos em formato de imagem (“.jpeg” ou “.png”) com *prints* do App Inventor ou no formato de pacote Android instalável (“.apk”), impossibilitando a análise do código gerado pelos blocos.

Assim, ao final da sequência didática, foram desenvolvidos e encaminhados ao pesquisador 71 aplicativos (“.aia”), seja pelo SIGAA ou e-mail. Desses, 66 por alunos do Campus Santo Augusto e 5 do Campus Santo Ângelo. Vale ressaltar que esse quantitativo é de aplicativos que foram construídos para resolver os desafios extraclasse, se considerados também os aplicativos desenvolvidos em aula, o total é significativamente maior. Quanto ao número bem inferior de aplicativos construídos por alunos do Campus Santo Ângelo, diz respeito a baixa adesão às aulas, o que pode-se atribuir a alguns fatores:

a) Diferentemente do Campus Santo Augusto onde as práticas puderam ocorrer desde o início da noite, em Santo Ângelo os alunos tiveram outras atividades nos dois primeiros períodos, sendo “dispensados”<sup>40</sup> da disciplina após o intervalo, tornando facultada a participação nas práticas da sequência didática;

b) Os encontros foram realizados em disciplinas distintas, dois em dias consecutivos (quintas e sextas-feiras), sendo uma sexta-feira onde os alunos não tinham aula programada;

c) Os aplicativos desenvolvidos pelos alunos não foram utilizados como instrumento avaliativo para a composição da nota da disciplina, ou seja, as tarefas foram compreendidas como não necessárias;

d) Apesar das práticas no Campus Santo Ângelo terem ocorrido mais próximas ao final do semestre com a intencionalidade de se obter dados para a comparação da aprendizagem entre as turmas e também com relação ao que já

---

<sup>40</sup> Não houve controle de frequência por parte das professoras titulares das disciplinas após o intervalo (terceiro e quarto períodos), pois os alunos foram liberados para resolver exercícios propostos na disciplina de Algoritmos ou por terem finalizado a avaliação (prova) na disciplina de Introdução à Informática.

havia estudado, os alunos acabaram relatando a impossibilidade de participar das aulas pelo acúmulo de avaliações e trabalhos exigidos nas demais disciplinas do primeiro semestre.

Essas são hipóteses que, apesar de não comprovadas cientificamente, ajudam a explicar a participação dos sujeitos. Assim, chega-se ao número de participantes e de aplicativos avaliados apresentado na Tabela 2 abaixo.

**Tabela 2** – Número de alunos participantes e de aplicativos desenvolvidos durante as aulas da sequência didática.

Aulas	Alunos participantes			Aplicativos desenvolvidos		
	Santo Augusto	Santo Ângelo	Total	Santo Augusto	Santo Ângelo	Total
1	17	10	27	13	1	14
2	15	9	24	15	-	15
3	16	4	18	16	2	18
4	16	2	18	11	2	13
5	14	4	18	11	-	11
<b>Total</b>				66	5	71

Fonte: Elaborada pelo autor, 2021.

Para verificar a aprendizagem de programação e o desenvolvimento do pensamento computacional, os aplicativos foram avaliados no CodeMaster. Conforme já discutido na metodologia, a ferramenta considera as competências do PC relacionadas à construção de algoritmos em formato de blocos no App Inventor, analisando o código-fonte dos aplicativos e calculando a sua pontuação. A partir do “modo professor”, os arquivos “.aia” encaminhados pelos discentes foram importados no CodeMaster, com o nome de cada arquivo identificando o respectivo aluno, gerando os resultados listados a seguir.

Na aula 1, os alunos foram desafiados a criar um aplicativo que replica a interface de login do sistema acadêmico SIGAA para que pudessem se familiarizar com o designer e componentes básicos do App Inventor. Dos 17 alunos do Campus Santo Augusto presentes na aula, 13 encaminharam o aplicativo desenvolvido. Na Tabela 3, abaixo, é possível observar a nota em programação que cada aplicativo recebeu e o nível em “faixa ninja”, tendo como mais frequente a “faixa vermelha” que representa o nível 4 de progresso (nota final entre 3 e 4). Cabe observar também que a nota final leva em consideração, além da programação, os componentes de interface e sua personalização, a exemplo: formato de botões, tamanho de fontes,

comprimento do texto dos componentes, harmonia de cores, entre outros. Apesar do aspecto visual da interface não ser o foco das práticas realizadas, a nota em interface de usuário demonstra que os alunos conseguiram compreender e utilizar os componentes básicos do App Inventor com certa facilidade, até por se tratar do primeiro contato que tiveram com a linguagem.

**Tabela 3** – Avaliação dos aplicativos desenvolvidos no desafio da aula 1 pelos alunos do Campus Santo Augusto.

Aplicativo	Nota em Programação	Nota em Interface de Usuário	Nota Final	Nível
Aluno1.aia	1.43	7.5	3.25	faixa vermelha
Aluno4.aia	2.14	7.81	3.84	faixa vermelha
Aluno5.aia	2.5	8.44	4.28	faixa roxa
Aluno8.aia	0.36	5.59	1.93	faixa amarela
Aluno10.aia	4.29	6.84	5.05	faixa azul
Aluno11.aia	2.14	7.06	3.62	faixa vermelha
Aluno12.aia	2.86	6.47	3.94	faixa vermelha
Aluno13.aia	1.07	7.81	3.09	faixa vermelha
Aluno14.aia	2.14	7.81	3.84	faixa vermelha
Aluno15.aia	2.14	7.5	3.75	faixa vermelha
Aluno16.aia	1.43	9.44	3.83	faixa vermelha
Aluno17.aia	2.5	5.88	3.51	faixa vermelha
Aluno18.aia	2.5	7.22	3.92	faixa vermelha
<b>Média</b>	<b>2.12</b>	<b>7.34</b>	<b>3.68</b>	

Fonte: Adaptado de CODEMASTER, 2020.

No Campus Santo Ângelo, dos 10 alunos que participaram do encontro, somente o Aluno 4 encaminhou o aplicativo desenvolvido, tendo como resultado a nota final 2.86, recebendo a faixa laranja.

O desafio da aula 2 compreendia o desenvolvimento de um aplicativo para o cálculo da média de quatro números. Para tal, os alunos deveriam utilizar variáveis e operadores aritméticos estudados no segundo encontro, além de componentes de interface para a entrada dos dados e a apresentação do resultado do cálculo. Estiveram presentes 15 alunos do Campus Santo Augusto e todos enviaram o aplicativo desenvolvido, com as notas listadas na Tabela 4. No Campus Santo Ângelo, 9 alunos participaram da aula, porém nenhum encaminhou o aplicativo proposto no desafio.

**Tabela 4** – Avaliação dos aplicativos desenvolvidos no desafio da aula 2 pelos alunos do Campus Santo Augusto.

<b>Aplicativo</b>	<b>Nota em Programação</b>	<b>Nota em Interface de Usuário</b>	<b>Nota Final</b>	<b>Nível</b>
Aluno1.aia	3.21	6.67	4.25	faixa roxa
Aluno3.aia	4.64	8.24	5.72	faixa azul
Aluno4.aia	3.21	8.24	4.72	faixa roxa
Aluno5.aia	3.21	8.24	4.72	faixa roxa
Aluno8.aia	2.86	8.82	4.65	faixa roxa
Aluno10.aia	4.29	7.65	5.30	faixa azul
Aluno11.aia	3.21	9.41	5.07	faixa azul
Aluno12.aia	4.64	7.78	5.58	faixa azul
Aluno13.aia	3.57	7.94	4.88	faixa roxa
Aluno14.aia	2.14	7.81	3.84	faixa vermelha
Aluno15.aia	2.5	8.24	4.22	faixa roxa
Aluno16.aia	3.57	8.24	4.97	faixa roxa
Aluno17.aia	3.21	7.65	4.54	faixa roxa
Aluno18.aia	3.21	7.06	4.37	faixa roxa
Aluno19.aia	3.21	6.47	4.19	faixa roxa
<b>Média</b>	<b>3.38</b>	<b>7.90</b>	<b>4.73</b>	

Fonte: Adaptado de CODEMASTER, 2020.

Ao comparar os resultados da avaliação das duas primeiras aulas, é possível observar uma evolução no desempenho dos alunos do Campus Santo Augusto, principalmente ao analisar o detalhamento da rubrica<sup>41</sup> do CodeMaster que exibe a pontuação para operadores e variáveis separadamente. Nesses quesitos, a maioria dos aplicativos obteve pontuação 1 ou 2, sendo 2 a máxima possível.

Após terem estudado a respeito de estruturas de seleção na aula 3, os discentes foram desafiados a criar um aplicativo de cálculo do índice de massa corporal, classificando a faixa de nutrição em 6 níveis, de desnutrição à obesidade grau três. Participaram da aula 16 alunos do Campus Santo Augusto, mesmo número de aplicativos postados no SIGAA, conforme resultados exibidos na Tabela 5. Mais uma vez, foi possível observar a evolução dos alunos, alguns atingindo a “faixa turquesa” que indica o sétimo nível de classificação.

<sup>41</sup> Apresentada no Quadro 1 presente no seção 3.5 deste trabalho.

**Tabela 5** – Avaliação dos aplicativos desenvolvidos no desafio da aula 3 pelos alunos do Campus Santo Augusto.

Aplicativo	Nota em Programação	Nota em Interface de Usuário	Nota Final	Nível
Aluno1.aia	3.21	7.78	4.58	faixa roxa
Aluno3.aia	3.57	6.76	4.53	faixa roxa
Aluno4.aia	5.71	7.35	6.20	faixa turquesa
Aluno5.aia	5.71	7.35	6.20	faixa turquesa
Aluno6.aia	4.64	8.06	5.67	faixa azul
Aluno7.aia	2.5	6.76	3.78	faixa vermelha
Aluno8.aia	5.36	7.94	6.13	faixa turquesa
Aluno9.aia	5.0	7.35	5.71	faixa azul
Aluno10.aia	5.0	7.63	5.79	faixa azul
Aluno11.aia	4.64	8.53	5.81	faixa azul
Aluno12.aia	5.71	6.39	5.91	faixa azul
Aluno15.aia	5.0	8.06	5.92	faixa azul
Aluno16.aia	4.64	7.65	5.54	faixa azul
Aluno17.aia	5.0	10.0	6.50	faixa turquesa
Aluno18.aia	3.21	7.78	4.58	faixa roxa
Aluno19.aia	2.86	7.35	4.21	faixa roxa
<b>Média</b>	<b>4.48</b>	<b>7.67</b>	<b>5.44</b>	

Fonte: Adaptado de CODEMASTER, 2020.

No Campus Santo Ângelo, 4 alunos participaram do encontro, sendo que 2 enviaram o aplicativo desenvolvido para o e-mail do pesquisador. A avaliação dos mesmos é exposta abaixo na Tabela 6. Apesar da amostra menor no Campus Santo Ângelo, a média da nota final das turmas se assemelha.

**Tabela 6** – Avaliação dos aplicativos desenvolvidos no desafio da aula 3 pelos alunos do Campus Santo Ângelo.

Aplicativo	Nota em Programação	Nota em Interface de Usuário	Nota Final	Nível
Aluno4.aia	7.5	3.25	6.22	faixa turquesa
Aluno15.aia	4.38	7.35	5.27	faixa azul
<b>Média</b>	<b>5.94</b>	<b>5.30</b>	<b>5.75</b>	

Fonte: Adaptado de CODEMASTER, 2020.

O desafio da aula 4, por sua vez, compreendia o desenvolvimento de um aplicativo para o cálculo do fatorial de um número fazendo uso de uma estrutura de repetição. Durante a aula, foram apresentadas duas estruturas distintas, ficando a

critério do aluno optar por uma ou outra. Também foi ressaltada a premissa de que só é possível calcular o fatorial de números positivos, mas sem especificar que seria necessária uma estrutura condicional para a testagem do valor informado pelo usuário. Estiveram presentes 16 alunos do Campus Santo Augusto e 11 deles enviaram o aplicativo desenvolvido, tendo as notas listadas na Tabela 7.

**Tabela 7** – Avaliação dos aplicativos desenvolvidos no desafio da aula 4 pelos alunos do Campus Santo Augusto.

<b>Aplicativo</b>	<b>Nota em Programação</b>	<b>Nota em Interface de Usuário</b>	<b>Nota Final</b>	<b>Nível</b>
Aluno1.aia	3.57	5.83	4.25	faixa roxa
Aluno3.aia	7.5	6.67	7.25	faixa verde
Aluno6.aia	3.93	7.35	4.96	faixa roxa
Aluno8.aia	7.5	6.11	7.08	faixa verde
Aluno9.aia	7.5	6.67	7.25	faixa verde
Aluno10.aia	3.93	8.53	5.31	faixa azul
Aluno12.aia	5.36	7.35	5.96	faixa azul
Aluno16.aia	7.5	7.22	7.42	faixa verde
Aluno17.aia	4.29	6.67	5.00	faixa azul
Aluno18.aia	5.36	7.0	5.85	faixa azul
Aluno19.aia	3.21	7.35	4.45	faixa roxa
<b>Média</b>	<b>5.42</b>	<b>6.98</b>	<b>5.89</b>	

Fonte: Adaptado de CODEMASTER, 2020.

No Campus Santo Ângelo, 2 alunos participaram da aula e ambos submeteram o aplicativo para avaliação, conforme Tabela 8.

**Tabela 8** – Avaliação dos aplicativos desenvolvidos no desafio da aula 4 pelos alunos do Campus Santo Ângelo.

<b>Aplicativo</b>	<b>Nota em Programação</b>	<b>Nota em Interface de Usuário</b>	<b>Nota Final</b>	<b>Nível</b>
Aluno4.aia	5.0	6.11	5.33	faixa azul
Aluno15.aia	7.5	6.11	7.08	faixa verde
<b>Média</b>	<b>6.25</b>	<b>6.11</b>	<b>6.21</b>	

Fonte: Adaptado de CODEMASTER, 2020.

Novamente houve evolução nas notas se comparado a avaliação do desafio da aula anterior, principalmente na nota de programação. Também foi interessante observar que mais da metade dos alunos implementaram uma estrutura condicional para a testagem do número digitado pelo usuário visando garantir a realização do

cálculo apenas com números positivos, o que indica a compreensão do conceito e das situações onde pode ser aplicado.

Na última aula, foi proposto aos alunos que desenvolvessem um aplicativo que contemplasse alguns cálculos matemáticos, dentre eles: calculadora de juros compostos, listagem de números primos, sequência de Fibonacci, cálculo da tabuada de um número e listagem de números pitagóricos. Solicitou-se a implementação de ao menos um desses cálculos fazendo uso dos conhecimentos adquiridos durante as aulas. Para exercitar a aprendizagem colaborativa e a troca de conhecimentos e atendendo a solicitação dos próprios alunos, permitiu-se que o desenvolvimento dos aplicativos ocorresse em grupos de até 3 participantes. Dos 14 alunos do Campus Santo Augusto presentes na aula, 11 encaminharam o aplicativo desenvolvido. Na Tabela 9, abaixo, é possível observar as notas atribuídas a cada aplicativo. Do Campus Santo Ângelo, 4 alunos participaram da aula, porém nenhum deles enviou o aplicativo ao pesquisador.

**Tabela 9** – Avaliação dos aplicativos desenvolvidos no desafio da aula 5 pelos alunos do Campus Santo Augusto.

<b>Aplicativo</b>	<b>Nota em Programação</b>	<b>Nota em Interface de Usuário</b>	<b>Nota Final</b>	<b>Nível</b>
Aluno1.aia	5.36	2.75	4.58	faixa roxa
Aluno3.aia	6.79	7.22	6.92	faixa turquesa
Aluno6.aia	4.64	7.5	5.50	faixa azul
Aluno7.aia	3.21	6.39	4.16	faixa roxa
Aluno8.aia	5.36	7.65	6.05	faixa turquesa
Aluno9.aia	6.07	5.56	5.92	faixa azul
Aluno10.aia	5.71	7.0	6.10	faixa turquesa
Aluno16.aia	5.36	5.0	5.25	faixa azul
Aluno17.aia	6.07	5.59	5.93	faixa azul
Aluno18.aia	3.93	5.28	4.33	faixa roxa
Aluno19.aia	4.29	7.94	5.38	faixa azul
<b>Média</b>	<b>5.16</b>	<b>6.17</b>	<b>5.47</b>	

Fonte: Adaptado de CODEMASTER, 2020.

Para além das notas atribuídas aos aplicativos desenvolvidos no desafio final, foi possível observar que os alunos do Campus Santo Augusto se empenharam na construção de aplicativos realmente funcionais. Assim, nesta tarefa foram desenvolvidos 5 aplicativos para o cálculo de juros compostos, 2 de tabuada, 2 de números primos, 1 para a listagem de termos da sequência Fibonacci e 1 aplicativo

que contemplou três cálculos distintos que não haviam sido sugeridos no enunciado da tarefa: teorema de Pitágoras, conversor de metros em centímetros e de horas em minutos. Apesar da aula 5 ter priorizado a revisão dos conteúdos e o esclarecimento de dúvidas dos discentes, restando pouco tempo para a introdução de novos conceitos, ainda assim, quatro projetos implementaram mais de uma tela e um o uso do sensor acelerômetro, funcionalidades abordadas no último encontro.

Outra questão interessante que pôde ser verificada diz respeito ao cuidado com o aspecto visual dos aplicativos desenvolvidos durante toda sequência didática. A maior parte apresenta uma interface agradável, inclusive implementando componentes não abordados nas aulas, como a inclusão de imagens e funções multimídia para reprodução de áudio, conforme é possível observar na Figura 24. Além de ter impulsionado o interesse e a motivação dos sujeitos, a possibilidade de aprender sobre a utilização de interfaces abrangeu particularidades da interação humano-computador (IHC), uma área da computação primordial para melhorar a experiência de usuários nos uso de sistemas computacionais e que só está presente na grade curricular do quinto semestre da LC, mas através do App Inventor já foi introduzida agora. Essa é apenas mais uma das características, entre outras já discutidas anteriormente, que auxiliam na aprendizagem de programação através de uma linguagem visual baseada em blocos.

**Figura 24** – Aplicativos desenvolvidos pelos alunos no App Inventor durante a realização da sequência didática



Continua

Continuação



Fonte: Elaborado pelo autor, 2021.

Assim, após a avaliação dos aplicativos no sistema web CodeMaster, além dos conceitos de algoritmos e programação, entendemos que as habilidades adquiridas relativas ao pensamento computacional também foram validadas, uma vez que os alunos demonstraram autonomia e capacidade para resolver os desafios propostos aplicando os conhecimentos construídos nas cinco aulas da sequência didática. Chegamos a essa conclusão em virtude do método adotado pela ferramenta, baseado em uma rubrica com indicadores de aprendizagem já discutidos no capítulo 3, e, também, pelo que a literatura consolidada nos apresenta, a começar por Wing (2006) que se refere ao PC como sendo constituído por processos de pensamento envolvidos na criação de soluções algorítmicas. Apesar de difíceis de serem medidos pela sua natureza abstrata, tais processos podem ser representados através de indicadores relacionados aos objetivos de aprendizagem, assumindo que atributos mensuráveis podem ser extraídos do artefato de software desenvolvido pelo aluno, o que se constitui na avaliação realizada pelo CodeMaster.

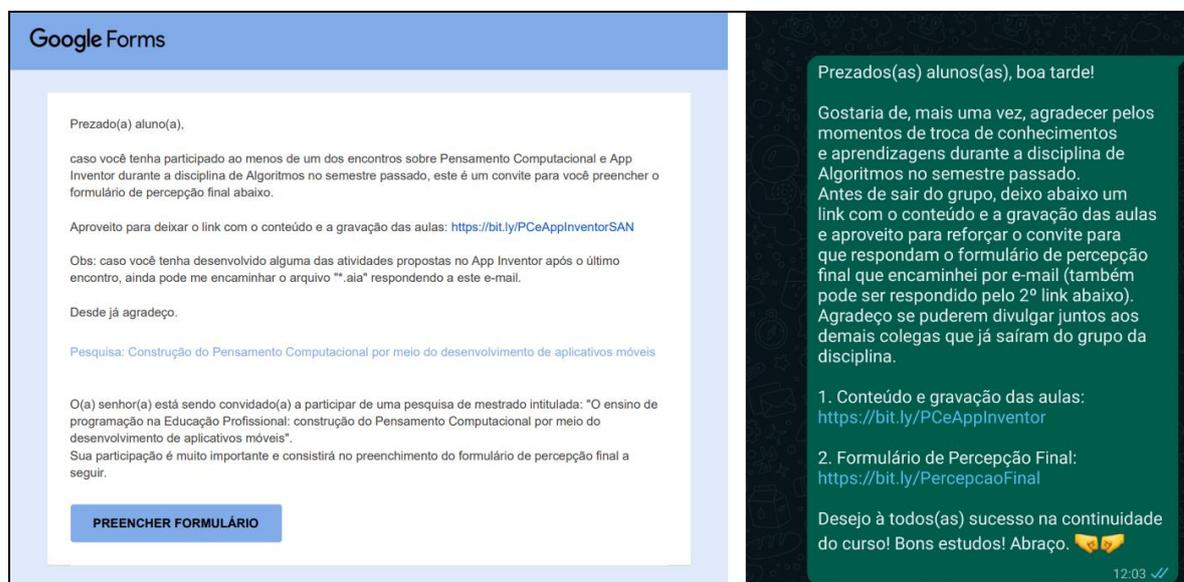
Nesse sentido, Romero, Lepage e Lille (2017) afirmam que a essência do PC é a capacidade de transpor significado abstrato em significado concreto, coisificando um modelo abstrato em um conjunto de instruções concretas como um algoritmo ou programa de computador. Os autores destacam ainda a importância do envolvimento criativo do aluno nas atividades de programação, classificando-o em 5 níveis: 1) exposição passiva à explicações do professor; 2) atividades de programação passo-a-passo; 3) atividades de programação com potencial criativo; 4) programação em equipe e 5) cocriação participativa de conhecimento. Entendemos que durante a sequência didática foram atingidos quatro desses cinco níveis a partir das atividades

de programação propostas, sendo que o último nível só poderia ser alcançado a partir da interação entre diferentes equipes, preferencialmente com características distintas, o que demandaria mais tempo e seria melhor realizado num contexto presencial.

### 5.3 Questionário de percepção final

Conforme o percurso metodológico apresentado no capítulo 3, o questionário de percepção final (Apêndice B) foi enviado para o e-mail (fornecido na aplicação do questionário inicial) dos alunos de Licenciatura em Computação dos *campi* Santo Augusto e Santo Ângelo que participaram das aulas da sequência didática. Como a última atividade da SD ocorreu no final do primeiro semestre letivo, com o prazo de envio da resolução do desafio final postergado por uma semana após a aula 5, e pelo acúmulo de avaliações relatado pelos alunos anteriormente, optou-se pelo envio do questionário no início do segundo semestre. Além do e-mail, o questionário também foi encaminhado em formato de link nos dois grupos de *whatsapp* das turmas, conforme a Figura 25. Junto ao convite, foi disponibilizado o material em PDF utilizado nas aulas e a gravação de todos os encontros.

**Figura 25** – Divulgação do questionário de percepção final



Fonte: Elaborado pelo autor, 2021.

Devido ao baixo número de respostas, o convite foi reencaminhado mais duas vezes, nos dois formatos. Ao final, foram obtidas 5 respostas de alunos do Campus Santo Augusto e apenas uma resposta do Campus Santo Ângelo. Para permitir uma

análise mais assertiva, as respostas do aluno do Campus Santo Ângelo foram incorporadas às do Campus Santo Augusto, uma vez que não haveria sentido analisar estatisticamente uma única opinião que não necessariamente corresponde às percepções dos demais colegas da turma. Assim, as análises a seguir se baseiam em 6 respostas de alunos que participaram das atividades da sequência didática, sendo utilizadas, quando necessário, as descrições: Aluno 1; Aluno 2; Aluno 3; e assim sucessivamente, levando em consideração a identificação atribuída no questionário inicial e apresentada no capítulo 3 e o Campus de origem do educando.

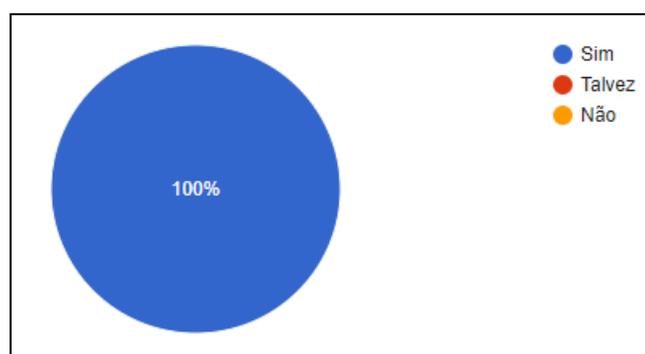
Na primeira pergunta os alunos foram questionados sobre o que entendem por algoritmo. Diferentemente do formulário inicial, onde a maior parte dos alunos afirmou desconhecer o termo, agora as respostas apontam para um entendimento do conceito e das práticas trabalhadas em aula. A exemplo, o Aluno 18 do Campus Santo Augusto respondeu se tratar de “uma sequência ordenada e finita de etapas, cuja execução passo a passo resolve um determinado problema”. Já o Aluno 5, único respondente do Campus Santo Ângelo, apresentou uma definição idêntica e acrescentou: “[...] mas, em um âmbito geral, algoritmos é a base para todo o processo de construção de um código bem estruturado, além, de servir como fundamento para aprender outras linguagens”.

Na questão seguinte os alunos responderam sobre o que entendem por pensamento computacional. Antes do início das aulas da SD, a maioria dos discentes afirmou não conhecer o termo, mas com interesse de aprender a respeito, e aqueles poucos que o conheciam, relacionaram com o conhecimento tecnológico. Passadas as aulas, é possível observar um entendimento por parte dos alunos, porém sem um posicionamento muito claro ou uma definição convergente. O Aluno 8 afirmou que PC “são as maneiras com que percebemos o que está ao nosso redor, de forma que possamos compreender, analisar e solucionar problemas.”. O Aluno 19, por sua vez, respondeu se tratar da “capacidade de saber ler uma sequência de códigos em alguma linguagem e entender o que aquele código faz”, ou seja, limitou o termo a codificação, ao contrário do Aluno 5 de Santo Ângelo que vê o PC trabalhar “conceitos mais teóricos, pode ser offline, ou seja, sem a necessidade de uso do computador. Interage com as dinâmicas propostas nas atividades de programação”. Esses posicionamentos refletem a própria pluralidade de definições existente entre pesquisadores da área, conforme já apresentado no capítulo 2.

As duas perguntas seguintes apresentaram respostas unânimes, todos os

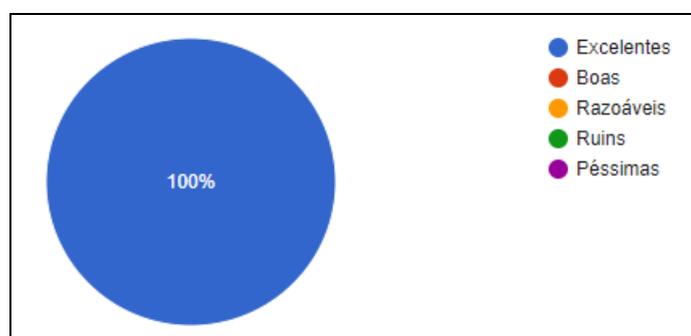
sujeitos acreditaram ser importante aprender sobre o pensamento computacional (Gráfico 11), assim como todos avaliaram como excelentes as atividades propostas (Gráfico 12). Esse retorno nos mostra que, mesmo não apresentando uma definição clara sobre o PC, os alunos compreenderam a importância de desenvolvê-lo, seja para a resolução de problemas reais ou para a aprendizagem de programação. Também nos revela que as atividades foram proveitosas e motivaram os alunos.

**Gráfico 11** – Respostas dos alunos à pergunta: “Você acredita que é importante aprender sobre o Pensamento Computacional?”



Fonte: Adaptado de Google Formulários, 2021.

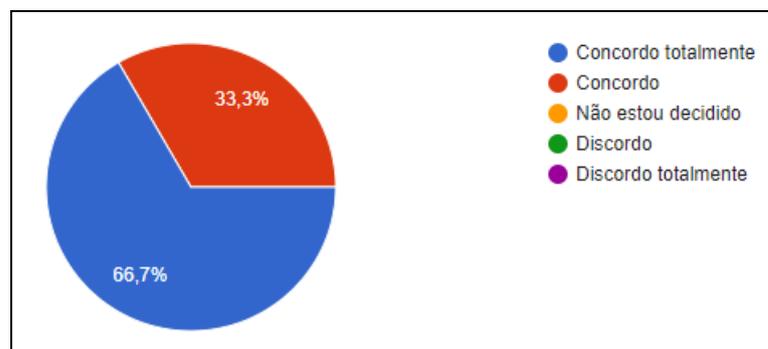
**Gráfico 12** – Respostas dos alunos à pergunta: “De maneira geral, o que você achou das atividades desenvolvidas?”



Fonte: Adaptado de Google Formulários, 2021.

Na próxima pergunta os alunos foram indagados se acreditam que as atividades contribuíram para o seu entendimento dos conceitos abordados na disciplina. A maioria (4 alunos) afirmou concordar totalmente e o restante declarou concordar, conforme Gráfico 13.

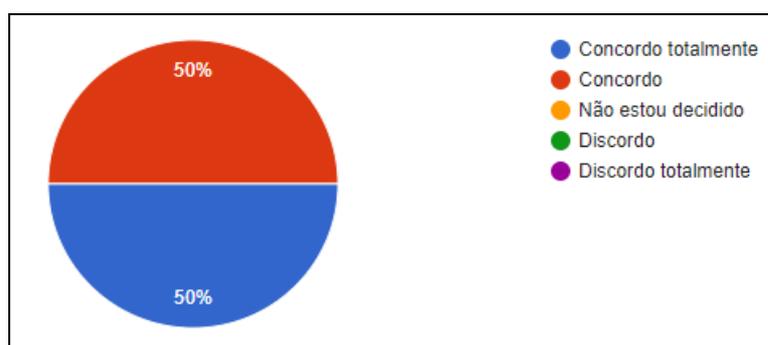
**Gráfico 13** – Respostas dos alunos à pergunta: “Você acredita que as atividades contribuíram para o seu entendimento dos conceitos abordados na disciplina de Algoritmos?”



Fonte: Adaptado de Google Formulários, 2021.

Na questão seguinte os alunos responderam se acreditam que as atividades contribuíram no seu processo de desenvolvimento do Pensamento Computacional. Todos concordaram, sendo que a metade deles concordou totalmente, vide Gráfico 14, abaixo.

**Gráfico 14** – Respostas dos alunos à pergunta: “Você acredita que as atividades contribuíram no seu processo de construção do Pensamento Computacional?”

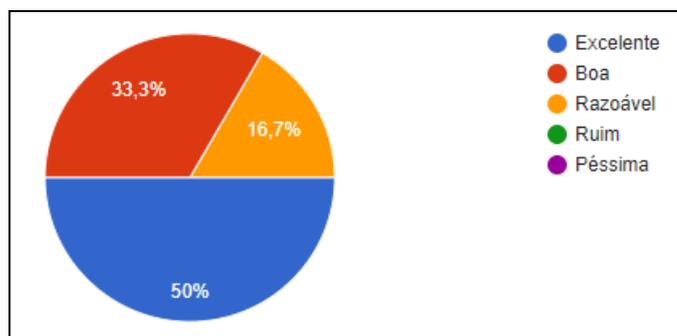


Fonte: Adaptado de Google Formulários, 2021.

A partir das respostas das duas últimas perguntas, entendemos que as atividades foram bem avaliadas pelos alunos, demonstrando que alcançaram o seu objetivo de problematizar situações do dia a dia dos educandos, proporcionar a compreensão dos conceitos de programação e o desenvolvimento do pensamento computacional.

Os discentes também foram perguntados sobre o que acharam da ferramenta de programação em blocos MIT App Inventor. Metade dos alunos classificou a ferramenta como excelente, dois alunos como boa e um como razoável, conforme Gráfico 15.

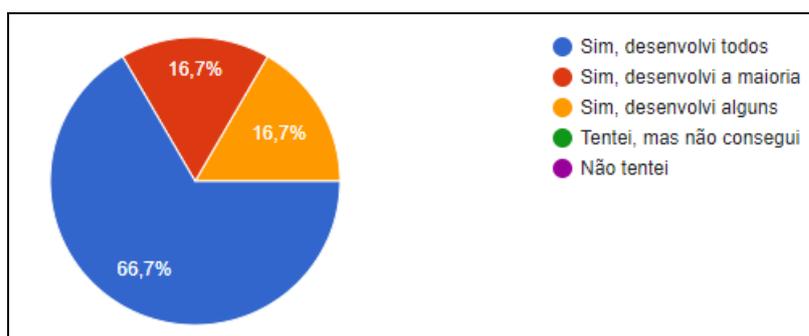
**Gráfico 15** – Respostas dos alunos à pergunta: “O que você achou da ferramenta de programação visual com blocos MIT App Inventor?”



Fonte: Adaptado de Google Formulários, 2021.

A próxima questão diz respeito aos aplicativos propostos ao final de cada atividade. A maioria (4 alunos) afirmou ter conseguido desenvolver todos os aplicativos, um aluno respondeu ter desenvolvido a maioria e o outro ter desenvolvido alguns, conforme pode ser visualizado no Gráfico 16.

**Gráfico 16** – Respostas dos alunos à pergunta: “Você conseguiu desenvolver os aplicativos propostos ao final de cada atividade?”



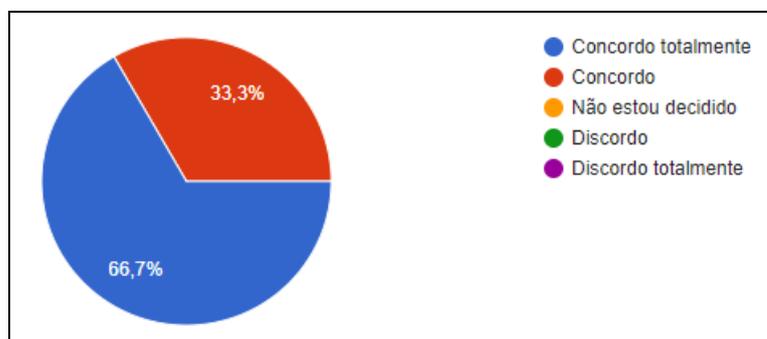
Fonte: Adaptado de Google Formulários, 2021.

Ao observar a impressão que os alunos tiveram do App Inventor e o percentual deles que desenvolveu todos ou a maioria dos aplicativos, chegamos a conclusão de que a escolha pela ferramenta foi adequada e se mostrou eficiente quanto a aprendizagem de programação. Esse entendimento se confirma também a partir da análise da questão seguinte onde os alunos foram questionados quanto às dificuldades na construção dos aplicativos, uma vez que houveram apenas dois relatos, um do Aluno 5 do Campus Santo Augusto que afirmou: “foi minha primeira experiência!”, levando a entender que teve alguma dificuldade inicial, mas sem especificar qual. Já o Aluno 18 respondeu: “Entender a linguagem, pois pra mim foi tudo novidade.”, também indicando ter apresentado alguma dificuldade por ser o primeiro contato com a temática.

Seguindo, os alunos foram indagados se acreditam que os conteúdos se

tornam mais interativos e/ou lúdicos a partir do desenvolvimento de aplicativos móveis. Todos concordaram, sendo que a maioria (4 alunos) declarou concordar totalmente, conforme o Gráfico 17.

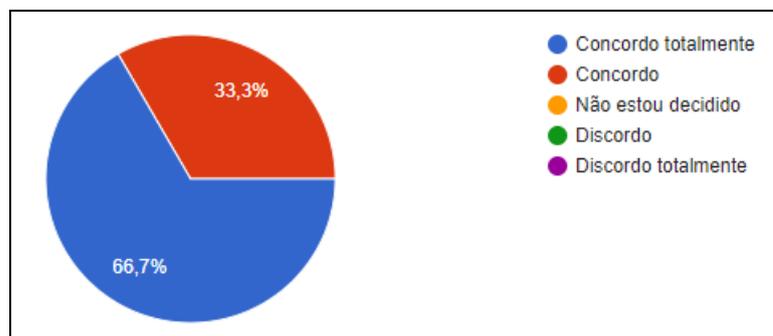
**Gráfico 17** – Respostas dos alunos à pergunta: “Você acredita que os conteúdos se tornam mais interativos e/ou lúdicos a partir do desenvolvimento de aplicativos móveis?”



Fonte: Adaptado de Google Formulários, 2021.

Na pergunta seguinte os alunos foram convidados a opinar quanto a incorporação das atividades à disciplina de Algoritmos de forma permanente para as turmas ingressantes. Assim como na pergunta anterior, a maior parte (4 alunos) concordou totalmente quanto a essa proposição, os outros alunos afirmaram concordar, como é possível observar no Gráfico 18.

**Gráfico 18** – Respostas dos alunos à pergunta: “Você gostaria que essas atividades fossem incorporadas à disciplina de Algoritmos de forma permanente (para as turmas ingressantes)?”



Fonte: Adaptado de Google Formulários, 2021.

As respostas nos levam a percepção de que as ações pedagógicas e instrucionais de programação em blocos com o App Inventor podem ser inseridas no escopo das disciplinas iniciais de algoritmos e programação.

As últimas três perguntas do questionário eram abertas, onde os alunos foram convidados a manifestar o que mais e o que menos gostaram nas atividades desenvolvidas, além de sugestões em caso de repetição da experiência com outras turmas. Com relação ao que mais gostaram, todos os alunos demonstraram satisfação com as atividades desenvolvidas, sendo as respostas apresentadas a

seguir.

“Todas as atividades foram excelentes.” (Aluno 8, Campus Santo Augusto).

“A interação do desenvolvedor com o App.” (Aluno 19, Campus Santo Augusto).

“Todo o aprendizado.” (Aluno 2, Campus Santo Augusto).

“Com o exercício prático é muito mais atrativo.” (Aluno 5, Campus Santo Augusto).

“Gostei de todas as atividades desenvolvidas.” (Aluno 18, Campus Santo Augusto).

“A didática e a forma de explicação, além do contato total com a plataforma e o processo de criação dos Apps.” (Aluno 5, Campus Santo Ângelo).

Quanto ao que menos gostaram, apenas o Aluno 2 respondeu “não tenho relações, tive algumas dificuldades”, mas sem especificar quais. Na última pergunta, o Aluno 5 do Campus Santo Augusto apresentou a seguinte sugestão: “a utilização de aplicativos/tecnologias mais avançadas/atuais”. O outro a responder foi o Aluno 5 do Campus Santo Ângelo, sugerindo: “continuar nesse trabalho e jamais desistir ou desanimar. O processo está muito bom, Parabéns pelo trabalho!”.

Assim, com base no questionário de percepção final, entendemos que as práticas foram bem recebidas e avaliadas pelos alunos, apesar do número de respondentes relativamente menor que o de participantes das aulas. Também foi possível averiguar a aprendizagem dos conceitos de algoritmos e pensamento computacional se comparado ao questionário de percepção inicial.

#### **5.4 Manifestações dos alunos e percepções do pesquisador**

Mesmo com a avaliação dos aplicativos desenvolvidos e da aplicação do questionário final, alguns resultados da realização da sequência didática só podem ser descritos a partir dos momentos de interação entre pesquisador-sujeitos e entre os próprios sujeitos durante os encontros. Ao total, foram mais de 28 horas de aula online através do Google Meet, todas gravadas e disponibilizadas aos alunos como fonte de consulta para esclarecimento de dúvidas, revisão de conceitos ou repetição de práticas que não puderam ser acompanhadas ao vivo. A transcrição de todas as falas destas aulas na íntegra tornar-se-ia um processo demasiadamente moroso e que pouco agregaria aos objetivos da pesquisa. Ao invés disso, são apresentados a seguir alguns trechos de manifestações dos educandos e demais percepções obtidas a partir dos encontros.

Conforme já discutido e justificado no início deste capítulo, o número de alunos que participaram das aulas assiduamente no Campus Santo Ângelo foi muito menor que em Santo Augusto. Pôde-se observar entre os presentes, porém, um grande interesse em aprender os conteúdos através de uma abordagem diferente. Acredita-se que essa motivação esteja relacionada com as possíveis dificuldades que estavam enfrentando para a aprendizagem de programação através da linguagem textual do Portugol Studio, ferramenta adotada pela professora titular da disciplina. Entre os momentos que levam a esta conclusão, está a aula 2, quando o Aluno 8, à 1h31m02s, afirma: "Portugol eu não tô entendendo muito, mas nesse aí já estou mais por dentro!", se referindo a compreensão que estava obtendo com relação ao uso de estruturas de seleção no App Inventor. Logo em seguida outros alunos presentes também manifestaram o mesmo posicionamento, concordando com a fala do Aluno 8. A docente titular da disciplina, a partir dos relatos dos alunos de que estavam entendendo os conceitos aplicados no App Inventor, se manifestou afirmando:

"legal que vocês tenham essa percepção de que usando uma outra linguagem, uma outra plataforma, vocês compreendem alguns aspectos, né, de forma melhor.. porque na verdade existem vários tipos de linguagens de programação e tem aquelas que são só textuais" (Docente titular da disciplina de Algoritmos, Campus Santo Ângelo, 1h32m20s).

Nesse sentido, na aula 3, realizada durante a disciplina de Introdução à Informática, o Aluno 4 externou o comentário da professora de Algoritmos antes do encontro: "vocês não ficam tão animados assim na minha aula", se referindo a motivação dos alunos em participar das práticas da sequência didática. Ainda na aula 3, o Aluno 10, aos 38m52s, espontaneamente se pronunciou afirmando: "muito mais legal que Portugol... deu certo o aplicativo aqui óh, adorei!" ao exibir na câmera do notebook o aplicativo de cálculo da média sendo executado em seu smartphone.

Outro relato interessante de estudante do Campus Santo Ângelo foi registrado ao final da quinta aula, após a fala da docente da disciplina agradecendo a contribuição trazida pelas práticas, o Aluno 3, às 2h41m43s, disse:

"contribuiu muito realmente, já estou aproveitando a oportunidade pra agradecer porque foi muito importante. Eu realmente consegui aprender e é muito importante pra gente que está dentro da área pedagógica porque não adianta ter um conhecimento de algoritmos e não ter um conhecimento de repassar isso pra quem não sabe algoritmos, né, então quem sabe essa questão dos blocos possa vir a ser precedente, né, aos algoritmos em algum momento, e é muito importante realmente. Muito obrigado!"

Por fim, após o final da aula, a professora da disciplina convidou o

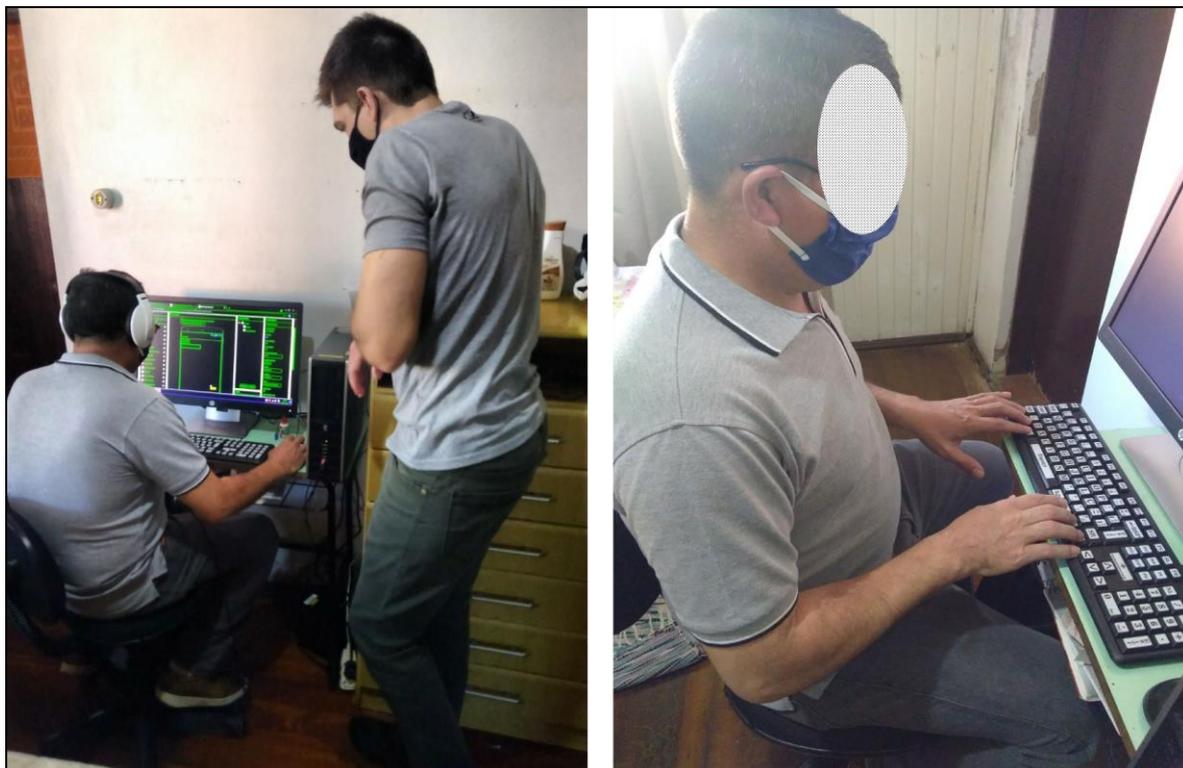
pesquisador para repetir as práticas através de um curso de férias ou como ação de extensão durante o próximo semestre, uma vez que os alunos demonstraram interesse, porém não puderam participar de todas as aulas, principalmente em função do acúmulo de trabalhos e avaliações nas diferentes disciplinas.

Esses relatos corroboram as percepções já expostas anteriormente de que a ferramenta App Inventor, ao utilizar uma linguagem visual baseada em blocos, se mostra uma alternativa muito interessante para o ensino de programação. Não necessariamente em substituição a uma linguagem textual, mas como complementar a ela, possibilitando aos alunos a aprendizagem por meio de distintas abordagens.

Além dos alunos do Campus Santo Ângelo que demonstraram interesse em participar das aulas por estarem com dificuldades de aprendizagem, o Aluno 4 esteve presente em todos os encontros e apresentou grande interesse e extrema facilidade na construção dos aplicativos, trazendo várias sugestões e auxiliando os demais colegas nas aulas. Certamente seria um incentivador das práticas e poderiam ajudar ainda mais os colegas em um cenário de aulas presenciais.

Ainda no Campus Santo Ângelo, um pouco antes do início das aulas da sequência didática, foi comunicado pela professora da disciplina de Algoritmos a existência de um aluno da turma com considerável grau de deficiência visual. O mesmo participou da primeira aula, porém foi possível perceber, a partir de seus questionamentos, que não estava conseguindo acompanhar as atividades propostas. Após o encontro, em nova conversa com a professora, decidiu-se acompanhar a entrega de um teclado adaptado ao aluno, já agendada para a semana seguinte. Portanto, foi acompanhada a entrega na residência do aluno e aproveitou-se para auxiliá-lo nas configurações iniciais do App Inventor, a comunicação com o seu smartphone e o desenvolvimento do primeiro aplicativo previsto na aula 1, conforme é possível observar na Figura 26.

**Figura 26** – Acompanhamento de atividades no App Inventor e entrega de teclado adaptado a aluno com baixa visão



Fonte: Elaborado pelo autor, 2021.

Para a realização das atividades, o aluno dispunha de um computador cedido pelo IFFar, com o leitor de tela NVDA<sup>42</sup> e o sistema operacional configurado em alto contraste, o que limita a escala de cores exibidas. Apesar do auxílio e da orientação presencial, ainda assim o aluno apresentou muitas dificuldades para utilizar o App Inventor, principalmente a interface de blocos, uma vez que a identificação das cores estava prejudicada pelo alto contraste e pelo fato do discente possuir daltonismo. Mesmo o aluno tendo participado de mais duas aulas online, não foi possível observar evolução na sua aprendizagem com o App Inventor, o que nos leva a conclusão de que, nesse cenário específico, uma linguagem visual baseada em blocos não é adequada, sendo mais indicada uma linguagem textual ou ainda outra abordagem que melhor contemple as suas necessidades.

Já no Campus Santo Augusto, foram observadas algumas dificuldades básicas de utilização do computador (falta de letramento digital) que dificultaram a compreensão dos passos a serem executados para a construção de aplicativos no App Inventor. A exemplo do Aluno 8 que relatou na aula 1, à 1h52m42s, “tenho

---

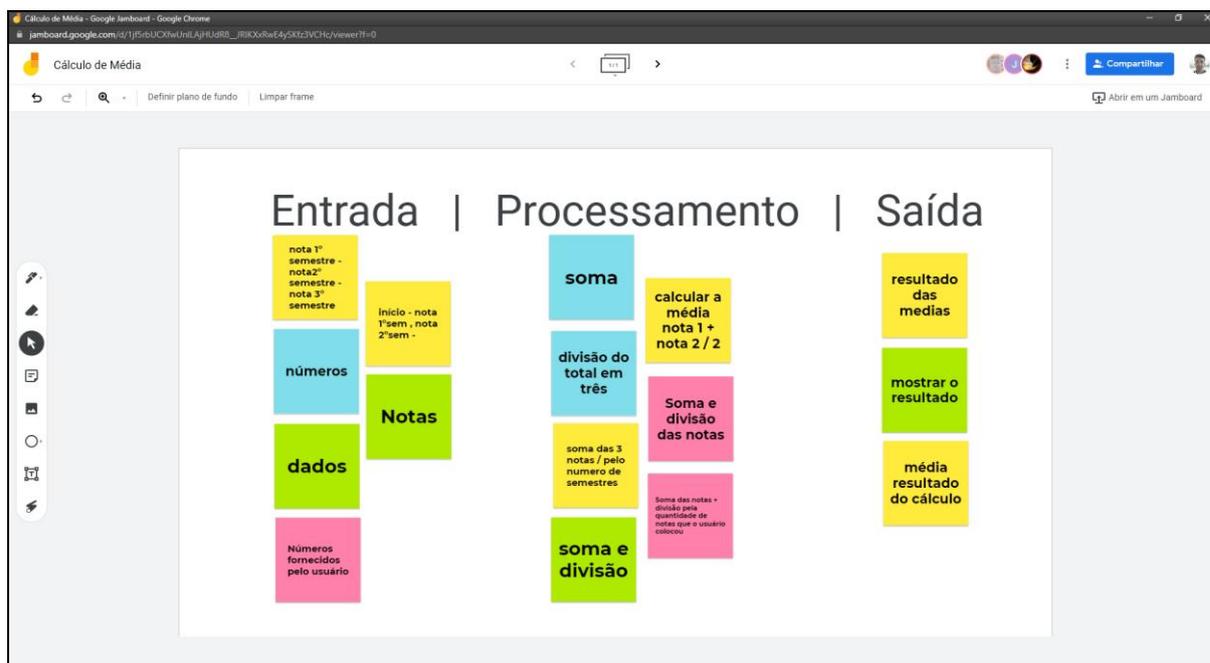
<sup>42</sup> Mais informações em: <https://www.nvaccess.org>.

dificuldade para executar a informática, não a tarefa em si” e também na aula 2, aos 13m12s, “até tentei fazer, mas tive bastante dificuldade pra baixar ali... aquela parte ali eu baixei o programa só que pedia pra salvar no ‘.aia’ e ele salva num diretório só que depois eu não conhecia encontrar no computador”, referindo-se à tarefa de exportação do código-fonte do aplicativo desenvolvido para a postagem no sistema acadêmico. Outras duas falas na segunda aula retratam dificuldades para o desenvolvimento dos desafios extraclasse. O Aluno 6 afirma, aos 12m03s, após ser questionado pelo professor sobre as dúvidas para a resolução do desafio, “a gente trabalha o dia inteiro e se envolve.. não tive tempo de rever”, o que demonstra a realidade da maioria dos discentes que trabalham durante o dia e estudam à noite, sobrando pouco tempo para as atividades extraclasse. Outra dificuldade diz respeito a situação sanitária imposta pela pandemia da Covid-19, quando o Aluno 4 justifica o mesmo questionamento afirmando: “eu não consegui porque estou com as minhas atividades atrasadas porque tive Covid”. Por outro lado, o Aluno 17, diante das dúvidas, pesquisou vídeos e conteúdo extra sobre a construção dos blocos, conforme a sua fala, aos 15m38s:

“tive algumas dúvidas de como selecionar os blocos, né... aí eu complementei com alguns vídeos no Youtube. Até fiz uma atividade bem fácil que nem o Rodrigo passou da soma e divisão... mas assim, tem que ser uma coisa bem minuciosa e detalhada, né, que nem vocês estavam falando na aula passada, quanto mais tu detalhar, mais tu vai conseguir construir”.

Ainda na aula 2, cabe destacar a atividade colaborativa realizada no Google Jamboard para o entendimento da partes de um algoritmo. A partir do compartilhamento de um quadro dividido em “Entrada | Processamento | Saída”, solicitou-se aos alunos que inserissem notas autoadesivas com as respectivas partes de um algoritmo para o cálculo da média entre dois números. A atividade se mostrou muito proveitosa a partir da ampla participação dos discentes na construção do quadro e, em seguida, na discussão a respeito da resolução desse problema usando os pilares do pensamento computacional. O resultado é apresentado abaixo na Figura 27.

**Figura 27** – Resultado da atividade colaborativa de construção das partes de um algoritmo para o cálculo da média entre dois números



Fonte: Elaborado pelo autor, 2021.

Ao final da aula 3, ao serem questionados sobre a sua percepção sobre os encontros até então, os alunos demonstraram contentamento e motivação com relação as atividades desenvolvidas. Em especial, o Aluno 6, a 1h37m04s, afirmou:

"por mais que está sendo bem atípico esse semestre né, eu acho que a gente tá conseguindo, assim, dentro do limite desse pouco tempo [...] a gente já consegue visualizar... eu achei muito interessante a nossa aula de algoritmos ser isso, né... porque eu digo assim, o algoritmo tá presente 24h por dia na vida da gente, [...] tudo são sequencias de passos... estou achando muito proveitoso essa maneira como está sendo aplicado"

"o pensamento computacional tem que estar junto com essa transição da educação... o PC é tudo hoje, é o futuro mesmo. Nós estamos no curso certo porque nós vamos trazer essas mudanças pra educação, para as crianças poderem acompanhar".

Ainda, a exemplo do que ocorreu no Campus Santo Ângelo, o professor da disciplina de algoritmos de Santo Augusto também demonstrou sua satisfação com as aulas da sequência didática, afirmando no início da Aula 4:

"estou feliz com o que estamos produzindo... os alunos estão se empenhando com as oficinas. Estou notando uma motivação da turma em fazer os exercícios e também uma persistência que inclusive eu não observava em outras turmas anteriores. As aulas estão bem elaboradas e ele foi feliz em escolher as aulas com a programação em blocos no App Inventor".

Além das manifestações citadas, os alunos também participaram das aulas sugerindo funcionalidades que gostariam de incorporar aos aplicativos (ex: controle

de foco e arredondamento de decimais) e solicitando revisão dos conteúdos que estavam com maior dificuldade. Assim, as aulas foram se adequando ao ritmo de aprendizagem dos alunos, obviamente de forma distinta para cada uma das turmas, até porque, além da realidade dos educandos, o tempo de duração das aulas também era diferente, conforme já exposto anteriormente.

Outra ressalva que precisamos fazer é relativa ao formato das aulas. Num cenário presencial, acredita-se que o desempenho dos alunos seria superior, uma vez que utilizariam os computadores do laboratório de informática previamente configurados e com um hardware padronizado, poderiam acompanhar as ações do pesquisador através do projetor e replicá-las em seus computadores sem a necessidade de transição de tela, e, principalmente, teriam um suporte rápido e acessível do pesquisador e dos próprios colegas para a resolução de dúvidas e o auxílio em caso de erros. Já no formato online e remoto, cada aluno utilizou seu próprio computador, alguns com sistema operacional Windows, outros Linux, e precisou alternar entre as janelas do navegador para acompanhar a aula no Google Meet e replicar as ações no App Inventor, talvez ainda uma terceira janela caso também tenha aberto o PDF com o material da aula disponibilizado previamente. Diante dessas dificuldades, entende-se que a aprendizagem desenvolvida pelos alunos, tanto dos conceitos de programação quanto das habilidades do pensamento computacional, foram extremamente relevantes e superaram o que se esperava diante do cenário imposto.

Cabe ainda uma observação com relação a utilização dos 3MP enquanto estruturantes das aulas. Apesar de não terem sido encontradas experiências anteriores de sua aplicação no ensino de programação, esta pesquisa aponta a viabilidade do seu uso neste contexto, aproximando os conceitos científicos à realidade dos alunos a partir da problematização inicial, a apreensão do conhecimento no segundo MP e sua recontextualização no terceiro momento marcado pela volta ao real.

## 6. CONSIDERAÇÕES FINAIS

Nossa perspectiva de abordar o pensamento computacional como um processo auxiliar na tomada de decisão e formação omnilateral, entende o homem como um ser histórico-social que produz ciência, tecnologia e cultura. O PC é a apropriação do conhecimento da computação para todos, proporcionando construir um novo olhar interpretativo para o mundo, a ciência e a tecnologia. Por isso, compreender como o desenvolvimento de aplicativos para dispositivos móveis facilita o desenvolvimento do pensamento computacional na Educação Profissional e Tecnológica foi um esforço em superar uma formação mecanicista e proporcionar uma formação integrada e integral.

Não se trata, portanto, de somente adquirir competências, a exemplo do que ocorre com a lógica multidisciplinar que fragmenta o saber por meio de disciplinas, mas sim de uma concepção de integralidade que capacita o aluno a desenvolver habilidades do pensamento computacional, como: abstração, análise de informações, reconhecimento de padrões e automação de soluções, além daquelas proporcionadas pelas bases da educação profissional, como: pensamento crítico-reflexivo e autonomia intelectual. Habilidades essas que também ampliam a compreensão de sociedade e qualificam o sujeito a integrar o mundo do trabalho, em que o ser humano é produtor da sua própria realidade, sendo capaz de transformá-la crítica e sistematicamente.

A resolução de problemas de maneira lógica é uma das habilidades a serem desenvolvidas a partir do ensino de programação. O raciocínio computacional não se restringe a programar um computador, mais do que isso, habilita o sujeito a buscar e selecionar informações relevantes, abstrair e decompor os problemas de modo criativo e dinâmico, auxiliando no processo de construção de saberes. Ao abordar o pensamento computacional tivemos por objetivo, dentre outros, desenvolver a criatividade e promover uma aprendizagem onde os sujeitos se utilizem dos

fundamentos e recursos da computação, aliados à sua inteligência e conhecimentos prévios, para inovar e resolver problemas multidisciplinares.

O intuito da pesquisa foi, também, articular a demanda por profissionais qualificados para a produção tecnológica de forma a proporcionar um conhecimento técnico aos discentes e uma formação humana e cidadã, preparando-os para o mundo do trabalho, em que tenham o trabalho como princípio educativo, não como uma redução da formação para o trabalho simples ou especializado com foco na empregabilidade, mas que promove os aspectos intelectuais e o aperfeiçoamento dos educandos. Afinal, caberá aos futuros licenciados em computação promover e difundir as habilidades do pensamento computacional em outros campos da atividade humana, principalmente na Educação Básica, a exemplo do que ocorre em diversos países do mundo.

Nesse sentido, os recursos do App Inventor se mostraram válidos para a aprendizagem de programação e o processo de promoção do PC. A ferramenta facilitou a expressão de soluções computacionais a partir dos conceitos de programação trabalhados durante a sequência didática, gerando como resultado produtos (aplicativos) tangíveis que apresentam o conhecimento apropriado e produzido pelo aluno. Sua utilização nos cursos de Licenciatura em Computação, com um público heterogêneo, com idades variando de 18 a 58 anos, demonstra que a ferramenta também pode ser utilizada no ensino médio e fundamental com demais alunos da Educação Profissional e Tecnológica, seguindo a aplicação dos conceitos abordados na SD. A escolha pela licenciatura, além do que já foi apontado, deveu-se também à viabilidade de realização da pesquisa em período noturno, não gerando conflito com o horário de trabalho do pesquisador. Porém, temos a convicção de que o Produto Educacional pode ser replicado, adotando a mesma metodologia, com qualquer turma de Educação Básica, com a devida adaptação, seja em alguma disciplina de programação ou como um curso de formação extracurricular.

Com base nos resultados deste estudo de caso foi possível perceber ainda que apesar dos alunos interagirem diariamente com dispositivos móveis, em geral, não apresentavam conhecimentos prévios de Computação que pudessem permitir uma facilidade maior na aprendizagem, mesmo assim foram capazes de criar vários aplicativos para resolver os desafios propostos. Destaca-se, também, que as práticas da sequência didática possibilitaram aos participantes a assimilação do conteúdo e a sua aplicação nos projetos desenvolvidos. Inclusive, como relatado no

capítulo de resultados, alguns educandos manifestaram interesse em conhecer mais sobre as funcionalidades do App Inventor não abordadas nos primeiros encontros, o que levou a ajustes na SD para as aulas seguintes.

Além da aprendizagem científico-teórica, observou-se entre os alunos que a construção dos aplicativos no App Inventor fez com que percebessem a importância e a aplicabilidade do conteúdo que aprenderam, desempenhando um papel de protagonismo pela construção de objetos de seu interesse, pessoalmente gratificantes e socialmente úteis. O que se aproxima das ideias de Freire (1996), de que ensinar não é depositar conhecimento, conforme o autor chama de “educação bancária”, mas sim criar possibilidades para a sua própria construção de forma autônoma para resolver problemas do dia a dia. Esse sentido se amplia, uma vez que as práticas contribuíram para a formação de futuros docentes que poderão atuar com o PC, na perspectiva de não apenas aplicar ferramentas de forma instrumental e descontextualizada, mas incorporar, de fato, o construcionismo defendido por Papert (1980) e promover uma educação aproximada do presente em que se encontram os educandos.

Ressalta-se igualmente os desafios impostos devido à pandemia da Covid-19 e, conseqüentemente, ao ensino online que exigiram uma alteração no formato inicialmente planejado para as aulas e na forma de avaliação das práticas. Assim, a metodologia adotada para o ensino e a estruturação da sequência didática foram pensadas e construídas de forma que os envolvidos pudessem participar e realmente aprender, vinculando o conhecimento ao seu cotidiano, fazendo com que se sentissem motivados a desenvolver os aplicativos.

As análises das respostas do questionário de percepção final, a avaliação dos aplicativos no CodeMaster, bem como toda a participação e engajamento dos alunos durante as aulas, nos leva ao entendimento de que o objetivo geral de desenvolver o pensamento computacional e a aprendizagem de programação a partir da construção de aplicativos móveis no App Inventor foi atingido. Em vista disso, é plausível dizer que os Três Momentos Pedagógicos, utilizados como ferramenta didático-pedagógica para a estruturação da sequência didática, se mostraram eficientes e resolutivos à medida que apresentaram um caminho possível para o ensino contextualizado e significativo.

Assim, entendemos que este trabalho resulta em um Produto Educacional relevante que traz importantes contribuições para professores de disciplinas

introdutórias de programação, tanto na Educação Básica quanto no Ensino Superior, pois apresenta um caminho alternativo e possível para o ensino de programação e o desenvolvimento do PC através de uma sequência didática pautada nos 3MP. A fração bibliográfica, a partir de uma revisão sobre o pensamento computacional, enfatiza o seu caráter científico e traz importantes reflexões acerca do tema e sua aproximação com a concepção de formação integral prevista nos Institutos Federais. Já os conceitos de Computação e as atividades propostas possibilitam aos professores iniciantes ou futuros licenciados um aprofundamento teórico e uma melhora significativa em suas práticas docentes. Além disso, faz com que visualizem novas possibilidades educacionais em relação às TICs, de modo a não se limitarem a simplesmente utilizar ferramentas ou reproduzir a metodologia de ensino tradicional.

Cabe salientar que a origem dos problemas associados ao processo de aprendizagem de programação é vasta e envolve diferentes fatores, conforme apontado durante este trabalho. No entanto, ao entender a aprendizagem como um processo de construção do conhecimento, é possível, através de uma abordagem pedagógica, intervir nas diferentes etapas concentrando esforços na aquisição de habilidades iniciais que possam auxiliar todo o processo de ensino, como é o caso das competências relacionadas ao pensamento computacional.

Por fim, acreditamos que esta pesquisa proporcionou subsídios para que os participantes possam analisar e resolver problemas apoiados às habilidades do pensamento computacional e aos conceitos de programação, para criar e inovar em diferentes áreas do conhecimento. Afinal, segundo a SBC (2019, p. 13), “precisamos preparar os jovens para profissões que ainda não existem, usando tecnologias que ainda não foram inventadas e resolvendo problemas que ainda não sabemos”.

## REFERÊNCIAS

AHO, Alfred. V. What is Computation: Computation and Computational Thinking. **Ubiquity Symposium**, 2011. Disponível em:

<https://ubiquity.acm.org/article.cfm?id=1922682>. Acesso em: 10 de jun. de 2020.

ALVES, Fábio J.; CARVALHO, Emerson; VENANCIO, Jessica A. L. O uso das tecnologias digitais de informação e comunicação em sala de aula pelos professores das escolas municipais de uma cidade do sul de minas gerais. **Anais do Workshop de Informática na Escola**, p. 1144, out. 2017. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/7322>. Acesso em: 30 de out. de 2019.

ALVES, Nathalia C. **CodeMaster: Um modelo de avaliação do pensamento computacional na educação básica através da análise de código de linguagem de programação visual**. 2019. 72 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis, SC.

ALVES, Nathalia; WANGENHEIM, Christiane G.; HAUCK, Jean. Um Modelo de Avaliação do Pensamento Computacional na Educação Básica através da Análise de Código de Linguagem de Programação Visual. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], p. 1021, nov. 2019. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/9052>. Acesso em: 10 jun. 2021.

ANTUNES, Ricardo. **Os Sentidos do Trabalho: Ensaio sobre a afirmação e a negação do trabalho**. 2. ed. São Paulo: Boitempo, 2009.

ARANTES, Flávia Linhalis; AMIEL, Tel; FEDEL, Gabriel. Nos rumos da autonomia tecnológica – desafios e lições aprendidas para a formação de jovens. **Anais do Workshop de Informática na Escola**, [S.l.], p. 308, nov. 2014. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/3113/2621>. Acesso em: 03 de set. de 2020.

ARAÚJO, D. C.; RODRIGUES, A. N.; SILVA, C. V. de A.; SOARES, L. S. O Ensino da Computação na Educação Básica Apoiado por Problemas: Práticas de Licenciados em Computação. **Anais do XXIII WEI (Workshop sobre Educação em Computação)** Garanhuns, 2015. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/014.pdf>. Acesso em: 19 de out. de 2019.

BARDIN, Laurence. *Análise de Conteúdo*. São Paulo: Edições 70, 2016.

BATISTA, Esteic Janaina Santos *et al.* Uso do Scratch no ensino de programação em Ponta Porã: das séries iniciais ao ensino superior. **Anais do Workshop de Informática na Escola**, [S.l.], p. 565, nov. 2016. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/6863>. Acesso em: 03 de set. de 2020.

BAUER, Rudieri *et al.* Projeto codIFic@r: Oficinas de Programação em Dispositivos

Móveis no Ensino Fundamental. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], p. 1210, out. 2017. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/7510>. Acesso em: 03 de set. de 2020.

BBC LEARNING. **Introduction to computational thinking**, 2015. Disponível em: <http://www.bbc.co.uk/education/guides/zp92mp3/revision>. Acesso em: 29 de abr. De 2020.

BLIKSTEIN, Paulo. **O pensamento computacional e a reinvenção do computador na educação**. 2008. Disponível em: [http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional.html](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html). Acesso em: 10 de jun. de 2020.

BOMBASAR, James *et al.* Ferramentas para o Ensino-Aprendizagem do Pensamento Computacional: onde está Alan Turing?. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, [S.l.], p. 81, out. 2015. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/5120>. Acesso em: 02 de mai. de 2020.

BOSSE, YORAH; GEROSA, Marco Aurélio. Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo: Um Estudo Preliminar. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 23., 2015, Recife. **Anais do XXIII Workshop sobre Educação em Computação**. Porto Alegre: Sociedade Brasileira de Computação, jul. 2015, p. 426-435. ISSN 2595-6175. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/10259>. Acesso em: 02 de jul. de 2020.

BOUCINHA, R.; BRACKMANN, C.; BARONE, D.; CASALI, A. Construção do Pensamento Computacional através do desenvolvimento de games. **Revista Novas Tecnologias na Educação (RENOTE)**, v. 15, n. 1, 2017.

BRACKMANN, Christian P. **Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica**. 2017. 224 f. Tese (Doutorado em Informática na Educação) - Programa de Pós-Graduação em Informática na Educação, Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação, Universidade Federal do Rio Grande do Sul. Porto Alegre, RS.

BRASIL. Ministério da Educação. **Expansão da Educação Superior, Profissional e Tecnológica**. 2014. Disponível em: [http://portal.mec.gov.br/expansao/images/APRESENTACAO\\_EXPANSAO\\_EDUCACAO\\_SUPERIOR14.pdf](http://portal.mec.gov.br/expansao/images/APRESENTACAO_EXPANSAO_EDUCACAO_SUPERIOR14.pdf). Acesso em: 17 de mai. de 2020.

BRASIL. Ministério da Educação. Secretaria da Educação Básica. **Base nacional comum curricular - BNCC**. Brasília, DF, 2018. Disponível em: <http://basenacionalcomum.mec.gov.br>. Acesso em: 05 de jun. de 2020.

BRASIL. Ministério da Educação. **Instituições da Rede Federal**. 2019. Disponível em: <http://portal.mec.gov.br/rede-federal-inicial/instituicoes>. Acesso em: 17 de mai.

de 2020.

CAMBRAIA, Adão C. Epistemologia da ciência na constituição do professor de Computação. In: UBERTI, H. G.; CONTO, J. M. (Org.) **Formação de professores no IF Farroupilha: novas possibilidades, novos desafios**. São Leopoldo: Oikos, 2016.

CAMBRAIA, Adão C.; KEMP, Adriana T.; ZANON, Lenir B. Ciclo de estudos sobre currículo integrado como movimento transformador de concepções e práticas na educação. In: FERREIRA, L. S.; SIQUEIRA, S.; CALHEIROS, V. C.; MARASCHIN, M. S.; ANDRIGHETTO, M. J. (Org.) **Pesquisas em Educação Profissional e Tecnológica no Rio Grande do Sul: desafios e perspectivas**. Curitiba: CRV, 2021.

CAMBRAIA, Adão Caron; ZANON, Lenir Basso. Desenvolvimento profissional docente numa licenciatura: interlocuções sobre o projeto integrador. **Revista Brasileira de Educação**, v. 23, 2018. Disponível em: <https://doi.org/10.1590/S1413-24782018230043>. Acesso em: 20 de mai. de 2020.

CAS, C. AT S. Developing Computational Thinking. **Teaching London Computing**, 2020. Disponível em: <https://teachinglondoncomputing.org/resources/developing-computational-thinking>. Acesso em: 13 de ago. de 2020.

CASTELLS, Manuel. **A Sociedade em Rede**. 17. ed. São Paulo: Paz e Terra, 2016.

CIAVATTA, Maria. A formação integrada: a escola e o trabalho como lugares de memória e de identidade. **Ensino Médio Integrado: concepção e contradições**. São Paulo: Cortez, 2005.

CIEB. **Currículo de referência em tecnologia e computação**. 2020. Disponível em: <http://curriculo.cieb.net.br/curriculo>. Acesso em: 25 de jun. de 2020.

CODEMASTER. **CodeMaster**. 2020. Disponível em: <http://apps.computacaonaescola.ufsc.br:8080/index.jsp>. Acesso em: 25 de mar. de 2020.

CSIZMADIA, A. *et al.* Computational thinking – a guide for teachers. 2015. Disponível em: <https://community.computingatschool.org.uk/resources/2324/single>. Acesso em: 10 de set. de 2020.

CSTA. **K–12 Computer Science Framework**. 2016. Disponível em: <https://www.k12cs.org>. Acesso em: 20 de jun. de 2020.

DA SILVA, Denilson R.; KURTZ, Fabiana D.; SANTOS, Cristina P. Computational thinking and TPACK in science education: a southern-Brazil experience. **PARADIGMA**, [S. l.], p. 529–549, 2020a. Disponível em: <https://doi.org/10.37618/PARADIGMA.1011-2251.0.p529-549.id912>. Acesso em: 10 de out. de 2021.

DA SILVA, Emanuel O.; FALCÃO, Taciana P. O Pensamento Computacional no Ensino Superior e seu Impacto na Aprendizagem de Programação. *In: WORKSHOP*

SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 28. , 2020b, Cuiabá. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 171-175. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2020.11152>. Acesso em: 27 de jun. de 2021.

DE FRANÇA, Rozelma Soares; DO AMARAL, Haroldo José Costa. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. **Anais do Workshop de Informática na Escola**, [S.l.], p. 179, nov. 2013. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/2646>. Acesso em: 02 de set. de 2020.

DELIZOICOV, Demétrio. **Concepção problematizadora do ensino de ciências na educação formal: relato e análise de uma prática educacional na Guiné Bissau**. 1982. 227 f. Dissertação (Mestrado) – Universidade de São Paulo, São Paulo, SP.

DELIZOICOV, Demétrio. Ensino de Física e a Concepção Freireana de Educação. **Revista de Ensino de Física**, vol. 5, n.2, 1983.

DELIZOICOV, Demétrio; ANGOTTI, José A. **Metodologia do ensino de ciências**. São Paulo: Cortez, 1994.

DELIZOICOV, Demétrio; ANGOTTI, José; PERNAMBUCO, Marta. **Ensino de Ciências: fundamentos e métodos**. 4. ed. São Paulo: Cortez, 2011.

DUNCAN, C.; BELL, T.; ATLAS, J. What do the Teachers Think? Introducing Computational Thinking in the Primary School Curriculum. In: **Proceedings of the Nineteenth Australasian Computing Education Conference (ACE '17)**. Association for Computing Machinery, New York, NY, USA, 2017, p. 65–74.

FRANÇA, Rozelma; TEDESCO, Patrícia. Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], p. 1464, 2015. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/6331>. Acesso em: 02 de set. de 2020.

FREIRE, Paulo. **Pedagogia da Autonomia: saberes necessários à prática educativa**. São Paulo: Paz e Terra, 1996.

FREIRE, Paulo. **Pedagogia do Oprimido**. Rio de Janeiro: Paz e Terra, 2006.

FRIGOTTO, Gaudêncio; CIAVATTA, Maria; RAMOS, Marise. (Org). **Ensino Médio Integrado: concepção e contradições**. São Paulo: Cortez, 2005.

GERALDES, Wendell Bento; MARTINS, Ernane Rosa; AFONSECA, Ulisses Rodrigues. **Uma experiência em ensino de programação de computadores com alunos da rede pública usando a linguagem Scratch**. Anais do Workshop de Informática na Escola, [S.l.], p. 129, nov. 2019. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/8499>. Acesso em: 04 de set. de 2020.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas,

2016.

GIRAFFA, Lucia M. M.; MÜLLER, Luana. Metodologia baseada em sala de Aula Invertida e Resolução de Problemas relacionado ao cotidiano dos estudantes: uma proposta para ensinar programação para iniciantes. **International Journal on Computational Thinking (IJCThink)**, v.1, n. 1, 2017, p. 52-67. ISSN 2594-5602. Disponível em: <http://hdl.handle.net/10923/14541>. Acesso em: 21 de mai. de 2020.

GOOGLE. **Google Workspace - Sites**. 2021. Disponível em: <https://workspace.google.com/intl/pt-BR/products/sites>. Acesso em: 20 de mar. de 2021.

GOMES, Tancicleide C. S.; DE MELO, Jeane C. B.. App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], nov. 2013. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/2725>. Acesso em: 01 de mai. de 2020.

GRESSE VON WANGENHEIM, C. *et al.* CodeMaster - Automatic Assessment and Grading of App Inventor and Snap! **Programs. Informatics in Education**, v. 17, n. 1, 2018. p. 117-150.

GROVER, S.; PEA, R. Computational Thinking in K-12: A Review of the State of the Field. **Educational Researcher**, v. 42, n. 1, p. 38–43, 2013.

HEINTZ, F.; MANNILA, L.; FÄRNQVIST, T. A review of models for introducing computational thinking, computer science and computing in K-12 education, **IEEE Frontiers in Education Conference (FIE)**, Erie, PA, USA, 2016, p. 1-9.

HOED, Rafael M.; LADEIRA, Marcelo. Análise de evasão nos Cursos Superiores de Computação: uma abordagem usando análise de sobrevivência e algoritmo Apriori. In: **Conferência Ibero Americana em Computação Aplicada**, dez. 2016, Lisboa, Portugal. p. 229-236.

IBGE. **PNAD Contínua TIC 2019**. 2021. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/30521-pnad-continua-tic-2019-internet-chega-a-82-7-dos-domicilios-do-pais>. Acesso em: 28 de nov. de 2021.

INSTITUTO FEDERAL FARROUPILHA. **Projetos Pedagógicos dos cursos Técnicos e de Graduação do Campus Santo Ângelo**. [Santo Ângelo: IFFar], 2014. Disponível em: <https://www.iffarroupilha.edu.br/projeto-pedag%3%b3gico-de-curso/campus-santo-%c3%a2ngelo>. Acesso em: 19 de out. de 2019.

INSTITUTO FEDERAL FARROUPILHA. **Sobre o Campus**. [Santo Ângelo: IFFar], 2018. Disponível em: <https://www.iffarroupilha.edu.br/institucional-san>. Acesso em: 19 de out. de 2019.

ISTE, CSTA. **Computational thinking leadership toolkit**. 2011. Disponível em: [www.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf](http://www.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf). Acesso em: 30 de jun. de

2020.

KRZYŻANOWSKI, Lucas *et al.* Ensino de programação: um estudo preliminar nos cursos de licenciatura em Computação no Brasil. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], p. 21, nov. 2019. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/8943>. Acesso em: 16 de jul. de 2020.

KUENZER, Acacia .Z. **Ensino Médio: construindo uma proposta para os que vivem do trabalho**. São Paulo: Cortez, 2007.

KURTZ, F.; DA SILVA, D. Tecnologias de Informação e Comunicação (Tics) como Ferramentas Cognitivas na Formação de Professores. **Revista Contexto & Educação**, v. 33, n. 104, p. 5-33, 2018.

LIUKAS, Linda. **Hello Ruby: adventures in coding**. Feiwel & Friends, 2015.

LISBÔA, Eliana Santana; MONTE-ALTO, Helio; SILVA, Maria Luiza. Clubes de Programação com Scratch nas Escolas e a Interdisciplinaridade. **Anais do Workshop de Informática na Escola**, [S.l.], p. 1174, out. 2017. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/7328>. Acesso em: 03 de set. de 2020.

LISBÔA, Eliana Santana; KARLING, Daniel Antonio. Desenvolvimento do Pensamento Computacional no Ensino Superior: um estudo realizado com a ferramenta App Inventor. **Revista Olhares & Trilhas**, v. 21, n. 1, p. 58-69, 2019.

MINAYO, Maria C. S. **Pesquisa social - Teoria, método e criatividade**. 21. ed. Petrópolis: Vozes, 2002.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 7. ed. São Paulo: Atlas, 2010.

MARTINS, Amilton. R.; MIRANDA, Ghuilhermina L.; ELOY, Adelmo A. Uma Revisão Sistemática de Literatura sobre Autoavaliação do Pensamento Computacional de Jovens. **Revista Novas Tecnologias na Educação (RENTE)**, v. 18, n. 1, 2020.

MARTINS, G. A.; THEÓPHILO, C. R. **Metodologia da investigação científica para Ciências Sociais Aplicadas**. 2. ed. São Paulo: Atlas, 2007.

MEC. Diretrizes Curriculares Nacionais para os cursos de Graduação em Computação (DCN16). **Resolução CNE/CES nº 5, de 16 de novembro de 2016**, [S.l.], p. 9, 2016. Disponível em: [http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=52101-rces005-16-pdf&category\\_slug=novembro-2016-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=52101-rces005-16-pdf&category_slug=novembro-2016-pdf&Itemid=30192). Acesso em: 15 de jul. de 2021.

MIT. **Democratizing the mobile world with MIT App Inventor**. 2013. Disponível em: <http://appinventor.mit.edu/explore/blogs/leo/2013/07/democratizing-mobile-world-mit-app-inventor>. Acesso em: 07 de nov. de 2021.

MIT. **MIT App Inventor**. 2020. Disponível em: <http://appinventor.mit.edu>. Acesso em: 25 de mar. de 2020.

FREEPIK. **Freepik Company**. 2021. Disponível em: <https://br.freepik.com>. Acesso em: 19 de abr. de 2021.

MUENCHEN, C.; DELIZOICOV, D. A construção de um processo didático-pedagógico dialógico: aspectos epistemológicos. **Ensaio: Pesquisa em Educação em Ciências** (Online), Belo Horizonte/MG, v. 14, n. 03, p. 199-215, 2012.

MUENCHEN, C.; DELIZOICOV, D. Os três momentos pedagógicos e o contexto de produção do livro “Física”. **Revista Ciência e Educação**, Bauru, v. 20, n. 3, p. 617-638, 2014.

NUNES, Felipe Becker *et al.* Mundos Virtuais na prática docente: uma visão sobre os desafios e benefícios. **Anais do Workshop de Informática na Escola**, p. 845, out. 2017. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/7303/5101>. Acesso em: 30 de out. de 2019.

ORTIZ, Julia dos S. B.; OLIVEIRA, Carolina Moreira; PEREIRA, Roberto. Aspectos do Contexto Sociocultural dos Alunos estão Presentes nas Pesquisas para Ensinar Pensamento Computacional?. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], p. 520, out. 2018. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/8277>. Acesso em: 11 de jun. de 2020.

PACHECO, Eliezer. **Fundamentos Político-Pedagógicos dos Institutos Federais: diretrizes para uma educação profissional e tecnológica transformadora**. Natal: IFRN, 2015.

PAPERT, Seymour; SOLOMON, Cynthia. **Twenty Things To Do With a Computer**. Massachusetts Institute of Technology, Cambridge, 1971. Disponível em: <https://dspace.mit.edu/handle/1721.1/5836>. Acesso em: 24 de mai. de 2020.

PAPERT, Seymour. **Mindstorms: children, computers, and powerful ideas**. New York: Basic Books, 1980.

PAPERT, Seymour. **LOGO: computadores e educação**. Traduzido por: José Armando Valente, Beatriz Bitelman e Afira Vianna Ripper. São Paulo: Brasiliense, 1986.

PARANÁ. Secretaria da Educação e do Esporte. **Novo Ensino Médio Paranaense**. 2022. Disponível em: [https://professor.escoladigital.pr.gov.br/ensino\\_medio](https://professor.escoladigital.pr.gov.br/ensino_medio). Acesso em: 18 de jan. de 2022.

PIAGET, Jean. **Epistemologia genética**. São Paulo: Martins Fontes, 1990.

RAMOS, Marise. Possibilidades e desafios na organização do currículo integrado. **Ensino Médio Integrado: concepção e contradições**. São Paulo: Cortez, 2005.

RIBEIRO, Leila; FOSS, Luciana; CAVALHEIRO, Simone A. C. Pensamento Computacional: Fundamentos e Integração na Educação Básica. **Anais da Jornada de Atualização em Informática na Educação**, p. 25, 2019. ISSN 2316-7734. Disponível em: <https://www.br-ie.org/pub/index.php/pie/article/view/8699>. Acesso em: 01 de jul. de 2020.

RIBEIRO, Laura; BARBOSA, Glívia; SILVA, Ismael; COUTINHO, Flávio; SANTOS, Natália. Um Panorama da Atuação da Mulher na Computação. In: WOMEN IN INFORMATION TECHNOLOGY (WIT), 13., 2019, Belém. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. 1-10. ISSN 2763-8626. Disponível em: <https://sol.sbc.org.br/index.php/wit/article/view/6707>. Acesso em: 11 de jan. de 2022.

RODRIGUEZ, Carla *et al.* Pensamento Computacional: transformando ideias em jogos digitais usando o Scratch. **Anais do Workshop de Informática na Escola**, [S.l.], p. 62, out. 2015. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/4992>. Acesso em: 03 de set. de 2020.

ROMÁN-GONZÁLEZ, Marcos; PÉREZ-GONZÁLEZ, Juan-Carlos; JIMÉNEZFERNÁNDEZ, Carmen. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. **Computers in Human Behavior**, v. 72, p. 678-691, 2017.

ROMERO, Margarida; LEPAGE, Alexandre; LILLE, Benjamin. Computational thinking development through creative programming in higher education. **International Journal of Educational Technology in Higher Education**, [S. l.], v. 14, n. 1, 2017. Disponível em: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-017-0080-z>. Acesso em: 30 de jan. de 2022.

SANTANA, Sivaldo Joaquim de; OLIVEIRA, Wilk. Desenvolvendo o Pensamento Computacional no Ensino Fundamental com o uso do Scratch. **Anais do Workshop de Informática na Escola**, [S.l.], p. 158, nov. 2019. ISSN 2316-6541. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/8502>. Acesso em: 03 de set. de 2020.

SANTOS, Cristina P.; DA SILVA, Denilson R.; FERREIRA, Giana; DA SILVEIRA, Maria G. F. Explorando o Pensamento Computacional para Despertar Novos Talentos: Relato de uma Experiência. In: WOMEN IN INFORMATION TECHNOLOGY (WIT), 11. , 2017, São Paulo. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2017. ISSN 2763-8626. Disponível em: <https://doi.org/10.5753/wit.2017.3400>. Acesso em: 10 de out. de 2021.

SANTIAGO, Almir D.; KRONBAUER, Artur. Um modelo lúdico para o ensino de conceitos de programação de computadores. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, [S.l.], p. 420, nov. 2016. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/6723/4610>. Acesso em: 9 de dez. de 2022.

SBC. **Diretrizes de ensino de computação na educação básica**. 2018. Disponível em: <http://www.sbc.org.br/educacao/diretoria-de-educacao-basica>. Acesso em: 06 de mai. de 2020.

SBC. **Itinerário Formativo de Computação**. 2019. Disponível em: <https://www.sbc.org.br/documentos-da-sbc/summary/203-educacao-basica/1216-itinerario-formativo-da-computacao>. Acesso em: 06 de mai. de 2020.

SHERMAN, M.; MARTIN, F. The assessment of mobile computational thinking. **Journal of Computing Sciences in Colleges**, v. 30, n. 6, 2015. p. 53–59.

SILVA, Denílson R. **Desenvolvimento do pensamento computacional como dimensão estruturante da atividade do professor de cursos superiores de computação**. 2020. 182 f. Tese (Doutorado em Educação nas Ciências) - Programa de Pós-Graduação em Educação nas Ciências, Universidade Regional do Noroeste do Estado do Rio Grande do Sul. Ijuí, RS.

SILVA, Ronison O.; NASCIMENTO-E-SILVA, Daniel; FERREIRA, Júlia A; SOUZA, Silvestre S. Aspectos relevantes na construção de Produtos Educacionais no contexto da Educação Profissional e Tecnológica. **Revista de Produtos Educacionais e Pesquisas em Ensino (REPPE)**, v. 3, n. 2, 2019. Disponível em: <http://seer.uenp.edu.br/index.php/reppe/article/view/1749>. Acesso em: 8 de jan. de 2022.

SILVEIRA, Sidnei R. *et al.* Educação a Distância, Sala de Aula Invertida e Aprendizagem Baseada em Problemas: possibilidades para o ensino de programação de computadores. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, p. 1052, out. 2018. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/8333/6008>. Acesso em: 02 de nov. de 2019.

SOUZA, Draylson M.; BATISTA, Marisa H.; BARBOSA, Ellen F. Problemas e Dificuldades no Ensino de Programação: Um Mapeamento Sistemático. **Revista Brasileira de Informática na Educação**, [S.l.], p. 39, abr. 2016. ISSN 2317-6121. Disponível em: <https://www.br-ie.org/pub/index.php/rbie/article/view/3317>. Acesso em: 30 de nov. de 2021.

SOUZA, N. G.; SILVEIRA, S. R.; PARREIRA, F. J. Proposta de uma Metodologia para Apoiar os Processos de Ensino e de Aprendizagem de Lógica de Programação na Modalidade de Educação a Distância. **Revista ECCOM**, v.9, n.18. 2018. Disponível em: <http://fatea.br/seer3/index.php/ECCOM/article/view/851/856>. Acesso em: 26 de out. de 2019.

UNESCO. **O Futuro da Aprendizagem Móvel: implicações para planejadores e gestores de políticas**. Brasília, 2014.

VICARI, Rosa M.; MOREIRA, Álvaro; MENEZES, Paulo B. **Pensamento Computacional: revisão bibliográfica**. UFRGS, v. 02, 2018. Disponível em: <https://lume.ufrgs.br/handle/10183/197566>. Acesso em: 27 de jun. de 2020.

VILLELA, Paulo R. C. **Escassez de mão de obra em ti: uma perspectiva**

**quantitativa.** In: Cadernos temáticos do observatório: mercado de trabalho e formação de mão de obra em TI. SOTEX, 2013. Disponível em: [http://www.softex.br/wpcontent/uploads/2015/10/Cadernos\\_Tematicos\\_Mercado\\_de\\_trabalho.pdf](http://www.softex.br/wpcontent/uploads/2015/10/Cadernos_Tematicos_Mercado_de_trabalho.pdf). Acesso em: 27 de agosto de 2020.

WEINTROP, D.; WILENSKY, U. Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. **Proceedings of International Computing Education Research**, Omaha, Nebraska, USA, 2015.

WING, Jeannette. Computational Thinking. **Communications of the ACM**. New York, v. 49, nº 3, p. 33-35, 2006.

WING, Jeannette. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717–3725, 2008.

WING, Jeannette. **Computational Thinking: What and Why?**. Link Magazine, 2010. Disponível em: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>. Acesso em: 13 de abr. de 2020.

WING, Jeannette. Computational Thinking Benefits Society. **Social Issues in Computing**, New York: Academic Press. 2014. Disponível em: <http://socialissues.cs.toronto.edu/2014/01/computational-thinking>. Acesso em: 25 de ago. de 2020.

WING, Jeannette. **Computational Thinking, 10 years later**. Microsoft Research Blog, 2016. Disponível em: <https://www.microsoft.com/en-us/research/blog/computational-thinking-10-years-later>. Acesso em: 25 de ago. de 2020.

YIN, Roberto. K. **Estudo de caso: planejamento e métodos**. Trad. Daniel Grassi, 2. ed., Porto Alegre: Bookman, 2001.

ZABALA, Antoni. **A prática educativa: como ensinar**. Porto Alegre: Artes Médicas Sul Ltda., 1998.

ZORZO, A. *et al.* **Referenciais de formação para os cursos de graduação em computação**. Sociedade Brasileira de Computação (SBC). 153p, 2018. Disponível em: <https://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia>. Acesso em: 06 de mai. de 2020.

## APÊNDICE A - PRODUTO EDUCACIONAL

**PC e App Inventor** Página Inicial Apresentação Sequência Didática ▾ Dissertação Publicações Contato Sobre o site 🔍

# Pensamento Computacional e App Inventor

*Sequência didática para o desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor*

**Prezado(a), bem vindo(a)!**

*Agradecemos a sua visita.  
Fique à vontade para navegar pelo site utilizando os links abaixo ou o menu superior da página.*

Apresentação
Sequência Didática
Dissertação
Publicações
Contato
Sobre o site



**PROFEPT**  
MESTRADO PROFISSIONAL EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL  
Farroupilha



**INSTITUTO FEDERAL**  
Farroupilha

\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença  
Creative Commons Atribuição–NãoComercial 4.0 Internacional.






Este espaço foi pensado e constituído no intuito de partilhar com professores da Educação Profissional e Tecnológica, bem como demais interessados no tema, uma [sequência didática](#) que visa proporcionar o desenvolvimento do pensamento computacional nos educandos, além da introdução dos conceitos básicos de programação, através da construção de aplicativos móveis no App Inventor.

A sequência didática se constitui como Produto Educacional de uma pesquisa de mestrado do Programa de Pós-graduação em Educação Profissional e Tecnológica - PROFEPT, ofertado pelo campus Jaguari do Instituto Federal Farroupilha, enquadrada na linha de pesquisa 2-Organização e Memórias dos Espaços Pedagógicos em Educação Profissional e Tecnológica, a medida que se destina a identificar as dificuldades dos alunos nas disciplinas de programação e a procurar uma alternativa ao método de ensino tradicional.

Aqui é possível encontrar todas as informações relativas a pesquisa realizada junto a um grupo de alunos do curso superior de Licenciatura em Computação, especialmente a sequência didática utilizada durante a disciplina de Algoritmos para proporcionar o desenvolvimento do pensamento computacional e a introdução dos conceitos básicos de programação.

## Justificativa



A resolução de problemas de maneira lógica é uma das habilidades a serem desenvolvidas a partir do ensino de programação. Por meio da programação o aluno amplia a capacidade de organizar e memorizar ideias, desenvolvendo o pensamento computacional na solução de problemas reais. No entanto, é comum discentes enfrentarem dificuldades para desenvolver o pensamento computacional e, conseqüentemente, as habilidades necessárias para programar, o que acaba por afetá-los, levando à desmotivação, reprovação e, em alguns casos, a evasão dos cursos de computação.

No intuito de auxiliar na mitigação desse problema, o que se apresenta como a justificativa deste trabalho, pretende-se compreender como a tecnologia móvel, especificamente a construção de aplicativos, pode auxiliar no ensino de programação na Educação Profissional e Tecnológica e no desenvolvimento do pensamento computacional.

## Objetivos



A pesquisa teve por **objetivo geral** identificar as contribuições e/ou limites da construção de aplicativos móveis com o App Inventor, por meio de uma sequência didática pautada nos 3MP, para a aprendizagem de programação, bem como para o desenvolvimento do pensamento computacional.

Quanto aos **objetivos específicos**, se pretendeu:

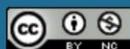
- Analisar estudos sobre o desenvolvimento do pensamento computacional e as relações com a formação integrada;
- Identificar as dificuldades apresentadas pelos alunos no ensino de programação;
- Desenvolver uma sequência didática com enfoque na programação para dispositivos móveis pautada nos Três Momentos Pedagógicos e identificar as contribuições e limites no aprendizado de algoritmos;
- Analisar as possíveis contribuições da tecnologia móvel a partir da construção de aplicativos para o aprendizado de programação e o desenvolvimento do pensamento computacional.



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença  
Creative Commons Atribuição–NãoComercial 4.0 Internacional.





A sequência didática (SD) abaixo constitui o Produto Educacional resultante da pesquisa de mestrado e tem por objetivo encadear e articular as atividades de aprendizagem ao longo da disciplina de Algoritmos.

A mesma está organizada em 5 aulas, cada qual dividida em 3 etapas, conforme preconizam os Três Momentos Pedagógicos, e traz aspectos a respeito do ensino de programação para dispositivos móveis utilizando uma linguagem de programação visual em blocos, o App Inventor.

A sequência didática tem o papel de indicar a função que cada uma das atividades possui, procurando apresentar um caminho possível para o desenvolvimento do pensamento computacional e a aprendizagem de programação a partir da construção de aplicativos móveis, deixando claro a pertinência e a ênfase a ser atribuída a cada conteúdo. Porém, a SD aqui proposta foi elaborada levando em conta um determinado público, sendo assim, sua utilização com outras turmas exige uma releitura e adaptações.

Em seguida, conheça um pouco mais a respeito dos Três Momentos Pedagógicos utilizados enquanto ferramenta pedagógica para estruturação das aulas e acesse a sequência didática dividida em cinco etapas.

## Três Momentos Pedagógicos (3MP)

Originalmente, os Três Momentos Pedagógicos foram pensados enquanto estruturantes do contexto das aulas, ou seja, sendo utilizados para estruturar currículos críticos, discutindo o que é ensinado e não apenas como é ensinado (MUNCHEN, 2010). Em nossa proposta, nos limitamos a utilização dos 3MP à metodologia de trabalho de sala de aula (estruturação das aulas) numa visão mais reduzida e simples de uma ferramenta didático-pedagógica, articulando os momentos com o intuito de ensinar um determinado conteúdo com a perspectiva de diálogo entre os participantes.

Partindo desse pressuposto, Muenchen e Delizoicov (2014) caracterizam os 3MP em: Problematização Inicial, Organização do Conhecimento e Aplicação do Conhecimento. No caso do terceiro momento pedagógico, aqui o referimos como de "Recontextualização do Conhecimento", por considerarmos que o termo 'aplicação' deixa margem ao risco de incorrer na racionalidade técnica, num caráter positivista e simplista.

1) **Problematização inicial (PI)**: por meio de situações reais que os alunos experienciam, desafiá-los a expor o que pensam para entendê-los melhor e, assim, problematizar e criar a necessidade de conhecer o que ainda não conhecem. Situações problematizadoras são apresentadas, visando a discussão com base nos conhecimentos prévios dos alunos.

2) **Organização do conhecimento (OC)**: desenvolvimento de atividades, com a orientação docente, sobre os conhecimentos necessários para entender a problematização inicial. O professor aborda o conteúdo para aprofundamento e conhecimento sobre o tema discutido, gerando a percepção de outras explicações a partir do conhecimento científico.

3) **Recontextualização do conhecimento (RC)**: sistematizar o conhecimento elaborado pelo aluno, analisando e interpretando as situações iniciais e outras que possam ser compreendidas pelo mesmo conhecimento. O conhecimento científico é utilizado para reinterpretar as situações problematizadoras, ou seja, o aluno deve ser capaz de empregar o conhecimento em situações reais ou outras não diretamente ligadas ao motivo inicial.



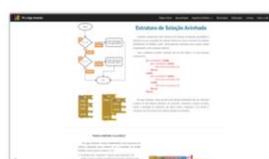
### Aula 1

Pensamento Computacional e App Inventor



### Aula 2

Algoritmos no App Inventor



### Aula 3

Estruturas de Seleção no App Inventor



### Aula 4

Estruturas de Repetição no App Inventor



### Aula 5

Sensores e Funções no App Inventor

## Referências

- MUNCHEN, C. A disseminação dos Três Momentos Pedagógicos: um estudo sobre práticas docentes na região de Santa Maria/RS. Tese de Doutorado (Programa de Pós-Graduação em Educação Científica e Tecnológica). Florianópolis: UFSC/PPGECT, 2010.
- MUNCHEN, C.; DELIZOICOV, D. Os três momentos pedagógicos e o contexto de produção do livro "Física". *Revista Ciência e Educação*, Bauru, v. 20, n. 3, p. 617-638, 2014.



PC e App Inventor

Página inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

## Sequência Didática

### Aula 1

Pensamento Computacional e App Inventor

Na primeira aula se espera conhecer o perfil dos alunos e seus conhecimentos prévios a respeito do tema; realizar uma contextualização a respeito do cenário atual da Computação na Educação Básica e Superior, procurando trazer aspectos da presença do pensamento computacional (PC) nesses níveis de ensino; introduzir a conceitualização do pensamento computacional abordando a sua origem, as modalidades existentes e os pilares que o sustentam; realizar atividades práticas para cada um dos pilares procurando associá-los com atividades cotidianas realizadas pelos educandos; apresentar a ferramenta App Inventor, sua interface, funcionalidades e a comunicação com o dispositivo móvel; criar o primeiro aplicativo móvel e executá-lo no smartphone do aluno.

## Três Momentos Pedagógicos

### 1º MP

A partir da introdução dos conceitos iniciais do pensamento computacional, fomentar a discussão a respeito da sua aplicação nas atividades rotineiras. Como o PC se aplica no dia-a-dia? Quais problemas reais podem ser resolvidos a partir do seu uso? Quais atividades o aluno realiza seguindo os preceitos do PC? Como introduzir o PC e otimizar as suas tarefas?

Também, a seguir da apresentação do App Inventor, demonstrar exemplos de aplicativos desenvolvidos na ferramenta para que os alunos possam apontar possíveis melhorias e estimular ideias de novos aplicativos para situações problema atuais.

Apresentar os quatro pilares do pensamento computacional, de forma teórica e prática, demonstrando a sua aplicabilidade em situações reais. Utilizar os recursos desenvolvidos pelo NAPEAD da UFRGS, realizando uma atividade inicial para introduzir o PC e outras quatro atividades após abordar cada um dos pilares.

Auxiliar os alunos no acesso e configuração inicial do App Inventor. Desenvolver o primeiro aplicativo na ferramenta e realizar a conexão com o seu smartphone para verificar a execução.

### 2º MP

Retomar a discussão sobre a utilidade do PC levantada na problematização inicial, agora procurando empregar os conceitos dos quatro pilares (repetir a atividade inicial de introdução do PC para verificar o conhecimento adquirido pelo aluno).

### 3º MP

Lançar o desafio de desenvolver uma interface replicando a tela de login de algum sistema utilizado pelo aluno (ex: sistema acadêmico) para que possa se familiarizar com o designer e componentes básicos do App Inventor.

## Questionário inicial

**Pensamento Computacional e desenvolvimento de aplicativos**

Você já ouviu ou viu algo a respeito do termo "Pensamento Computacional"?

Sim, tenho domínio a respeito do termo.

Sim, conheço pouco.

Não, mas gostaria de conhecer.

Não, desconheço e não tenho interesse.

No intuito de conhecer um pouco a respeito do perfil dos alunos participantes, bem como coletar subsídios para as atividades a serem propostas durante esta e demais aulas seguintes, optou-se por utilizar um questionário de percepção inicial, desenvolvido na ferramenta Google Formulários, com vistas a obter informações prévias sobre o acesso a internet disponível, quais equipamentos o educando dispõe para participar dos encontros, seu conhecimento prévio sobre algoritmos, pensamento computacional e desenvolvimento de aplicativos móveis, além de possíveis temáticas de interesse.

## Computação na Educação Básica

A introdução da computação na Educação Básica (EB) não tem a intenção de competir com as demais áreas do conhecimento. Pelo contrário, visa capacitar futuros profissionais para diferentes carreiras com capacidade de lidar com a computação e as tecnologias digitais.

Enquanto há anos alguns países da Europa e Estados Unidos já possuem iniciativas de ensino de programação na Educação Básica, no Brasil o debate se iniciou recentemente com a inclusão da computação na BNCC - Base Nacional Comum Curricular (BRASIL, 2018).

Em relação a BNCC, concordamos com a introdução de conhecimentos da Computação na Educação Básica por meio do pensamento computacional como estratégia de produção do conhecimento e não apenas uso das tecnologias como ferramentas. No entanto, discordamos de alguns aspectos, como por exemplo: os itinerários formativos que desintegram o



currículo em nome de uma capacitação para determinadas áreas. Maiores discussões sobre a temática em [Cambráia, Kemp, Zanon \(2021\)](#).

Além da BNCC, existem algumas iniciativas que tem se destacado na busca de introduzir o Pensamento Computacional na matriz curricular da EB. Entre elas:



CIEB

O Centro de Inovação para a Educação Brasileira (CIEB) apresenta um Currículo de Referência em Tecnologia e Computação, organizado em 3 eixos e subdivididos em conceitos, com sugestão de práticas pedagógicas, avaliações e materiais de referência, associados às competências gerais e habilidade da BNCC.

[Saiba mais sobre o CIEB](#)



SBC

A Sociedade Brasileira de Computação (SBC) disponibiliza um documento intitulado "Diretrizes para ensino de Computação na Educação Básica" que trata dos referenciais de formação em Computação, trazendo competências e habilidades divididos em 3 eixos que compõem a Computação na EB.

[Saiba mais sobre a SBC](#)



BNCC

A Base Nacional Comum Curricular (BNCC) estabelece, entre as competências gerais, a necessidade de se compreender, utilizar e criar tecnologias digitais de informação e comunicação como forma de produzir conhecimentos, resolver problemas e exercer autoria na vida pessoal e coletiva.

[Saiba mais sobre a BNCC](#)

## Computação na Educação Superior

A falta de processos de ensino de computação na Educação Básica, acaba gerando dificuldades de aprendizagem de programação nos cursos superiores de Computação. Estudos demonstram que os discentes apresentam problemas para desenvolver o pensamento computacional e, conseqüentemente, as habilidades necessárias para programar, o que acaba por afetá-los, levando à desmotivação, reprovação e, em alguns casos, a evasão dos cursos de computação.

### Bosse e Gerosa (2015)

- 30% de reprovação em disciplinas de introdução à programação, chegando a 50% em alguns casos;
- 25% dos aprovados cursaram duas ou mais vezes a disciplina;

### Hoed e Ladeira (2016)

- Reprovação aumenta a evasão pelo fato da disciplina de algoritmos ser pré-requisito ou fundamental para o sucesso em outra disciplina;

### Krzyzanowski *et al.* (2019)

- 62,5% dos reprovados em algoritmos por nota e 70% dos reprovados por frequência, trancam o curso;

As dificuldades enfrentadas pelos alunos podem ser confirmadas a partir da análise de desempenho da disciplina de Algoritmos do curso de Tecnologia em Sistemas para Internet do IFFAR Campus Santo Ângelo. O curso oferta 35 vagas em turno noturno, sempre no primeiro semestre de cada ano, tendo em vista a periodicidade anual de oferta dos cursos superiores de graduação. Durante quatro anos, 2015 à 2018, o índice médio de reprovação por nota foi superior à 22% dos alunos, enquanto a reprovação por faltas atingiu quase 26%, chegando a um índice médio total de 48% de discentes reprovados durante esse período, conforme é possível visualizar na tabela abaixo.

Tabela 1 – Reprovação de alunos na disciplina de Algoritmos: 2015 à 2018.

Ano	Nº de alunos da disciplina	Nº de reprovados por nota	Índice de reprovação por nota	Nº de reprovados por falta	Índice de reprovação por falta	Índice total de reprovação
2015	24	2	8,3%	4	16,7%	25,0%
2016	33	10	30,3%	8	24,2%	54,5%
2017	35	8	22,9%	9	25,7%	48,6%
2018	27	8	29,6%	10	37,0%	66,7%
<b>Índice médio total de reprovação</b>						<b>48,7%</b>

Fonte: Elaborada pelo autor com dados obtidos junto à Coordenação de Registros Acadêmicos do Campus Santo Ângelo, 2019.

Por outro lado, estudos como o de Da Silva e Falcão (2020), demonstram que a introdução da disciplina de pensamento computacional na matriz curricular dos cursos de Licenciatura em Computação faz com que os alunos apresentem melhora em relação aos níveis de raciocínio lógico e pensamento algorítmico, além de demonstrarem mais confiança e se sentirem mais preparados para aprender programação.

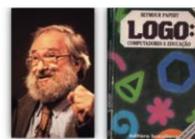
Contudo, não basta a introdução de uma disciplina de pensamento computacional, é necessário uma construção dialógica para contextualizar e envolver os alunos nos estudos, a exemplo da utilização dos 3MP, como ressaltamos nesta sequência didática.

## Pensamento Computacional

A origem do pensamento computacional se deu a partir das ideias introduzidas por Seymour Papert ainda na década de 70, sob influência dos estudos de Jean Piaget, ao desenvolver o construcionismo como uma abordagem que defendia o uso do computador

enquanto uma ferramenta capaz de estimular o desenvolvimento do raciocínio do estudante na solução de problemas e para construir o seu próprio conhecimento.

A primeira citação efetiva do termo pensamento computacional se deu no livro "Mindstorms: children, computers and powerful ideas" (PAPERT, 1980, p. 182), traduzido para português como "LOGO: Computadores e Educação" (PAPERT, 1986), enquanto Papert relata experimentos desenvolvidos em LOGO, uma linguagem interpretativa planejada para garantir acesso à programação de computadores para estudantes sem conhecimentos matemáticos prévios. Porém, há época não foi dada a devida atenção ao termo e, tampouco, Papert ocupou-se em aprofundá-lo. O autor defendia o uso do computador de forma humanizada, enquanto ferramenta para desenvolver habilidades mentais das crianças. Nesse sentido, idealizava o ensino de programação na educação básica, afirmando que a criança "deve programar o computador, e ao fazê-lo, ela adquire um sentimento de domínio [...] e estabelece um contato íntimo com algumas das ideias mais profundas da ciência, da matemática e da arte de construir modelos intelectuais" (PAPERT, 1986, p. 17).



A popularização do termo "pensamento computacional" se deu a partir de um artigo publicado por Wing (2006), conceituando o PC como uma "distinta forma de pensamentos com conceitos básicos da Ciência da Computação para resolver problemas, desenvolver sistemas e para entender o comportamento humano, habilidade fundamental para todos". Traz ainda que o pensamento computacional é "reformular um problema aparentemente difícil em um problema que sabemos como resolver, talvez por redução, incorporação, transformação ou simulação" (WING, 2006, p.33).

Em trabalhos seguintes, a autora refina a conceituação do termo, aproximando e complementando a forma de pensar baseada na Matemática e na Engenharia. Destaca também os processos de pensamento envolvidos na formulação de um problema e na expressão de soluções que possam ser executadas por um humano ou por uma máquina. Ainda complementa como sendo um conjunto de habilidades intelectuais e de raciocínio que indica como as pessoas interagem e aprendem a pensar (WING, 2008; WING 2014).

Além de Wing, diversos pesquisadores apresentam distintas definições para o termo "pensamento computacional", o que nos leva a entender que o mesmo não está totalmente estabelecido, portanto ainda é um conceito em aberto e em constante evolução. Embora se encontrem várias abordagens para o PC, existe uma visão convergente de que sua utilização enquanto ato de pensar como um cientista da computação pode auxiliar na solução de problemas para diferentes situações e áreas. Portanto, o PC não tem a intenção de fazer com que as pessoas pensem como computadores, mas sim que se utilizem dos fundamentos e recursos da computação, aliados à sua inteligência, para abordar (formular e resolver) problemas. Abaixo são apresentadas algumas das diferentes definições de pensamento computacional.

"Distinta forma de pensamentos com conceitos básicos da Ciência da Computação para resolver problemas, desenvolver sistemas e para entender o comportamento humano, habilidade fundamental para todos" (WING, 2006).



"O processo de pensamento envolvido na formulação de problemas, para que suas soluções possam ser representadas como etapas e algoritmos computacionais" (AHO, 2011).



"Saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano - em outras palavras, usar computadores, e redes de computadores, para aumentar nossa produtividade, inventividade e criatividade" (BLIKSTEIN, 2008).



"Habilidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática" (SBC, 2018, p. 2).



"Uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas colaborativamente através de passos claros de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente" (BRACKMANN, 2017, p. 29).

#### Vamos entender na prática?

Para entender melhor o pensamento computacional, desafie-se levando o lobo, a ovelha e a couve de uma margem à outra do rio, mas sem esquecer de considerar os conceitos do pensamento computacional acima apresentados.

Essa atividade prática de introdução foi desenvolvida pelo NAPEAD - Produção Multimídia para a Educação da UFRGS.



## Pilares do Pensamento Computacional

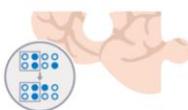
Ao abordar o pensamento computacional, seja de maneira conceitual ou com intenção prática de solução de problemas, é comum dividi-lo em pilares ou dimensões, ou também chamadas de bases do PC. Porém, assim como na definição do termo, não existe um consenso na literatura quanto aos pilares que compõem o pensamento computacional.

Segundo a SBC (2018), o PC envolve abstração, análise de informações e automação de soluções, sendo essas as três habilidades primordiais para a construção de uma solução algorítmica capaz de resolver um problema. No entanto, a maior parte dos estudos relacionados ao tema apresentam quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos.



### Decomposição

Processo no qual o problema complexo é dividido em tarefas menores e mais fáceis de serem gerenciadas. A separação em partes menores otimiza a compreensão do problema, o desenvolvimento da solução e a avaliação do resultado e a identificação de erros.



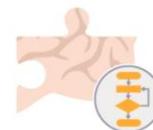
### Reconhecimento de Padrões

Consiste em identificar similaridades entre os problemas. Possibilita a criação de modelos genéricos que podem ser usados para a construção de soluções mais rápidas e eficientes com base em problemas resolvidos anteriormente.



### Abstração

Corresponde em simplificar a realidade a partir da criação de uma representação ou modelo do que se está tentando resolver apenas com os elementos relevantes, ou seja, ignorando detalhes desnecessários, tornando os problemas mais fáceis de serem solucionados.



### Algoritmo

Constitui-se em uma sequência de passos ou regras finita a ser seguida para a resolução do problema. As instruções lógicas contidas em um algoritmo, se seguidas com precisão, levam a solução do problema inicial, bem como podem resolver problemas semelhantes.

#### Vamos entender na prática?

Desafie-se explorando cada um dos pilares do pensamento computacional através das atividades práticas abaixo desenvolvidas pelo [NAPEAD](#) – Produção Multimídia para a Educação da UFRGS.

Assuma o papel de detetive decompondo os fatores determinantes de um caso misterioso para montar a sua teoria.



Enquanto dono de uma locadora, identifique padrões nos clientes para saber exatamente qual a indicação de filme é mais apropriada.



Você é o segurança da portaria responsável por autorizar a entrada do público, examine cada fantasiado abstraindo as características em sua lista.



Auxilie o personagem a atravessar a rua em segurança montando um algoritmo que descreva a sequência de passos até chegar a sua loja.



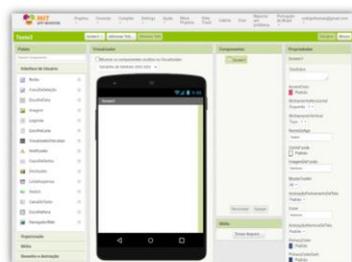
## App Inventor

Diferentes ferramentas, computacionais ou não, podem ser utilizadas para auxiliar o processo de construção de algoritmos. Comumente são empregados softwares que possibilitam a escrita da codificação em linguagem natural ou na forma de pseudocódigo, sendo que cada ambiente possui uma sintaxe específica para o desenvolvimento da lógica de programação. No que tange o ensino do pensamento computacional, é frequente o uso de ambientes de programação visual em blocos, como é o caso do App Inventor.

Desenvolvido pelo Google e mantido atualmente pelo Instituto de Tecnologia de Massachusetts (MIT), o App Inventor é uma ferramenta de código aberto que permite a criação de aplicativos móveis, como jogos, simulações, animações, entre outros, para dispositivos com o sistema Android. Possibilita o desenvolvimento de aplicativos arrastando blocos que se encaixam segundo uma lógica de funcionalidades pré-estabelecidas, levando a uma construção abstrata, pois não requer que o aluno tenha compreensão total do código que será executado pelo aplicativo. O ambiente pode ser acessado gratuitamente, de forma online, por meio de um navegador web e conta com mais de 8 milhões de usuários registrados, de 195 países, e 34 milhões de aplicativos construídos (MIT, 2020).



Saiba mais



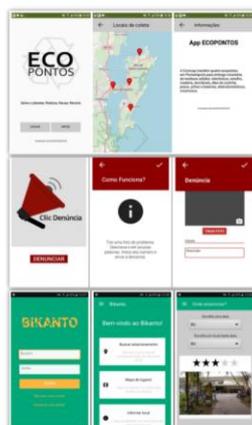
Para definir o comportamento dos componentes inseridos no "Designer", é utilizada a janela de "Blocos", conforme figura ao lado. Nela, são estabelecidas as funcionalidades do aplicativo a partir de blocos conectáveis, que podem ser eventos ou métodos. Cada bloco realiza diferentes ações, desde controles de sequência e continuidade, até operações matemáticas. O aplicativo desenvolvido pode ser executado diretamente em um dispositivo Android conectado ao computador ou através de um emulador que acompanha o

A programação de um aplicativo no App Inventor é baseada em componentes, os quais podem ser manipulados por eventos, métodos e propriedades. O comportamento dos componentes é orientado pelos eventos originados da interação do usuário com o aplicativo.

A interface do App Inventor se divide em duas janelas: "Designer" e "Blocos". A primeira se destina a criar a interface visual do aplicativo, arrastando componentes da "Paleta", como botões, caixas de texto e seleção, figuras, mapas, entre outros, para o "Visualizador", conforme a figura ao lado.



App Inventor.



A partir das funcionalidades existentes no App Inventor, é possível desenvolver diferentes aplicativos para resolver problemas reais. Nas figuras ao lado são exibidos alguns exemplos de aplicativos desenvolvidos pela iniciativa [Computação na Escola](#) do INCod/INE/UFSC com parceria do IFSC. O primeiro aplicativo, [ECOPONTOS](#), permite a visualização de pontos de coleta de material reciclável. O segundo, [ClicDenúncia](#), possibilita ao cidadão realizar denúncias a partir do local onde um problema é observado (ruas esburacadas, falta de iluminação pública, pichações, etc.), sendo a mesma enviada imediatamente ao órgão responsável. O último aplicativo, [Bikanto](#), visa identificar e compartilhar com outras pessoas locais onde é possível e seguro deixar a sua bicicleta (UFSC, 2020).

Ao observar as telas desses aplicativos (clique na figura para ampliar), é possível apontar a implementação de alguma melhoria? Quais funcionalidades os aplicativos poderiam contemplar com base em outros Apps que você utiliza? Que outros problemas poderiam ser resolvidos com o desenvolvimento de Apps semelhantes?

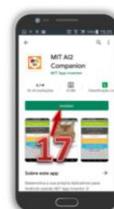
### Vamos entender na prática?

Execute a sequência de passos abaixo para criar o seu primeiro aplicativo "Olá Mundo" no App Inventor.

- Acessar o site: <https://appinventor.mit.edu>;
- Clicar em "Create Apps!" [1];
- Fazer login com a sua conta do Google;
- Aceitar os Termos de Serviço;
- Clicar em "Continue" na janela de apresentação e em "Close" na janela seguinte de boas-vindas;
- Após o login, será exibida a tela inicial onde ficarão salvos todos os projetos;
- Alterar o idioma clicando em "English" na barra superior [2] e selecionar a opção "Português do Brasil" [3];
- Fechar as janelas iniciais, caso sejam exibidas novamente;
- Clicar em "Iniciar novo projeto..." [4] e atribuir o nome "ola\_mundo";
- Observar as colunas: "Paleta", "Visualizador", "Componentes", "Mídia" e "Propriedades";
- Na coluna "Paleta", gaveta "Interface de Usuário", arrastar para a tela do smartphone um componente "Botão" [5] e, também da gaveta "Interface de Usuário", um componente "Notificador" [6];
- Na coluna "Componentes", selecionar o "Botão" [7] e renomeá-lo para "btOlaMundo" [8]. Na coluna "Propriedades", alterar o seu "Texto" digitando "Olá Mundo!" [9];
- Alternar da página de "Designer" para a página de "Blocos" [10];
- Na coluna "Blocos", em "Screen1", clicar no "btOlaMundo" [11] e arrastar o evento "quando btOlaMundo.Clique" para a coluna "Visualizador" [12];
- Na coluna "Blocos", em "Screen1", clicar no "Notificador" [13] e arrastar o procedimento "chamar Notificador1.MostrarAlerta" para a coluna "Visualizador", encaixando no bloco anterior [14];
- Na coluna "Blocos", em "Internos", clicar em "Texto" [15], arrastar o bloco vazio "( )" para a coluna "Visualizador", encaixando no bloco anterior e digitar o texto "Olá Mundo!" [16];

Seu primeiro aplicativo está pronto. Agora, siga os passos abaixo para testá-lo no seu smartphone Android.

- No smartphone, abrir o aplicativo "Google Play" para acessar a loja de Apps, pesquisar e instalar o aplicativo "MIT AI2 Companion" [17];



- Voltar ao App Inventor, clicar no menu "Conectar" [18] e em "Assistente AI" [19]. Será gerado um QR Code [20] para acesso ao aplicativo no smartphone.



- No smartphone, abrir o aplicativo "MIT AI2 Companion" recém instalado, pressionar o botão "scan QR code" [21] e apontar a câmera para a tela do App Inventor para ler o código gerado.



Aguarde um pouco... o aplicativo será executado no seu smartphone! Agora pressione o botão "Olá Mundo!" e veja o resultado.

### Desafio

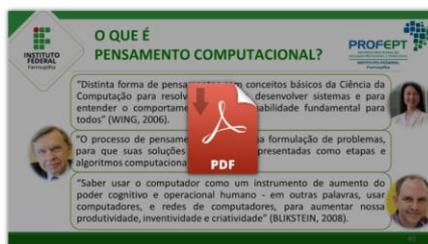


Para se familiarizar com o App Inventor, desenvolva um aplicativo que replica a interface de login do sistema acadêmico SIGAA existente no IFFar.

Procure utilizar diferentes componentes, explorando as gavetas de "Interface de Usuário" e "Organização" para deixar a interface do aplicativo idêntica a figura ao lado.



Clique abaixo para realizar o download da aula 1.



### Vamos aprofundar os conhecimentos?

Para saber mais sobre pensamento computacional e App Inventor, explore o material abaixo que contém vídeos, sites e cursos sobre a temática abordada na aula 1.



Curso de extensão "Formação de Professores em Pensamento Computacional" (UFERN). Mais em: [pensam.indd.ufrn.br](http://pensam.indd.ufrn.br)



Palestra "Pensamento Computacional na Educação Básica: exemplos práticos" Prof. Dr. Christian Brackmann (IFFar)



Evento "Pensamento Computacional como Habilidade do Futuro" (UFPE)



Oficina "Pensamento Computacional na BNCC", Prof. Dr. Humberto José Bortolossi (UFF)



Curso "Introdução ao Pensamento Computacional" (MEC)



Curso "Aplicações do Pensamento Computacional para os Anos Iniciais do Ensino Fundamental" (MEC)



Minicurso "Introdução ao Pensamento Computacional com Foco em Matemática Básica" (UFFB)



Portal da Iniciativa "Computação na Escola" (UFSC/IFSC)



Curso de App Inventor, Prof. Cassio Onodera (IFSP)



Curso de App Inventor, projeto de extensão "Pensando em Códigos" (UNIFAL)



Curso "Algoritmos e Estruturas de Dados com MIT App Inventor", Prof. Dr. Sérgio de Oliveira (UFS)

## Referências

- AHO, Alfred. V. What is Computation: Computation and Computational Thinking. *Ubiquity Symposium*, 2011. Disponível em: <https://ubiquity.acm.org/article.cfm?id=1922682>. Acesso em: 10 de jun. de 2020.
- BLIKSTEIN, Paulo. O pensamento computacional e a reinvenção do computador na educação. 2008. Disponível em: [http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional.html](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html). Acesso em: 10 de jun. de 2020.
- BOSSE, Yoram; GEROSA, Marco Aurélio. Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo: Um Estudo Preliminar. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI)*, 23., 2015, Recife. *Anais do XXIII Workshop sobre Educação em Computação*. Porto Alegre: Sociedade Brasileira de Computação, jul. 2015, p. 426-435. ISSN 2595-6175. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/10259>. Acesso em: 02 de jul. de 2020.
- BRACKMANN, Christian P. Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica. 2017. 224 f. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação, Universidade Federal do Rio Grande do Sul. Porto Alegre, RS.
- BRASIL. Ministério da Educação. Secretaria da Educação Básica. *Base nacional comum curricular - BNCC*. Brasília, DF, 2018. Disponível em: <http://basenacionalcomum.mec.gov.br>. Acesso em: 05 de jun. de 2020.
- CIEB. *Currículo de referência em tecnologia e computação*. 2020. Disponível em: <http://curriculo.cieb.net.br/curriculo>. Acesso em: 25 de jun. de 2020.
- CAMBRAIA, Adão C.; KEMP, Adriana T.; ZANON, Lenir B. Ciclo de estudos sobre currículo integrado como movimento transformador de concepções e práticas na educação. In: FERREIRA, L. S.; SIQUEIRA, S.; CALHEIROS, V. C.; MARASCHIN, M. S.; ANDRIGHETTO, M. J. (Org.) *Pesquisas em Educação Profissional e Tecnológica no Rio Grande do Sul: desafios e perspectivas*. Curitiba: CRV, 2021.
- DA SILVA, Emanuel Oliveira; FALCÃO, Taciara Pontual. O Pensamento Computacional no Ensino Superior e seu Impacto na Aprendizagem de Programação. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI)*, 28., 2020, Cuiabá. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 171-175. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2020.11152>.
- HOED, Rafael M.; LADEIRA, Marcelo. Análise de evasão nos Cursos Superiores de Computação: uma abordagem usando análise de sobrevivência e algoritmo Apriori. In: *Conferência Ibero Americana em Computação Aplicada*, dez. 2016, Lisboa, Portugal. p. 229-236.
- KRZYZANOWSKI, Lucas et al. Ensino de programação: um estudo preliminar nos cursos de licenciatura em Computação no Brasil. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, [S.l.], p. 21, nov. 2019. ISSN 2316-8889. Disponível em: <https://www.br-ie.org/pub/index.php/wcbie/article/view/8943>. Acesso em: 16 de jul. de 2020.
- MIT. *MIT App Inventor*. 2020. Disponível em: <http://appinventor.mit.edu>. Acesso em: 25 de mar. de 2020.
- PAPERT, Seymour. *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books, 1980.
- PAPERT, Seymour. *LOGO: computadores e educação*. Traduzido por: José Armando Valente, Beatriz Bitelman e Afira Vianna Ripper. São Paulo: Brasiliense, 1986.
- SBC. *Diretrizes de ensino de computação na educação básica*. 2018. Disponível em: <http://www.sbc.org.br/educacao/diretoria-de-educacao-basica>. Acesso em: 06 de mai. de 2020.
- UFSC. *Computação na Escola*. 2020. Disponível em: <https://computacaonaescola.ufsc.br>. Acesso em: 25 de mar. de 2020.
- WING, Jeannette. Computational Thinking. *Communications of the ACM*. New York, v. 49, nº 3, p. 33-35, 2006.
- WING, Jeannette. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 366, n. 1881, p. 3717-3725, 2008.
- WING, Jeannette. *Computational Thinking Benefits Society*. *Social Issues in Computing*. New York: Academic Press. 2014. Disponível em: <http://socialissues.cs.toronto.edu/2014/01/computational-thinking>. Acesso em: 25 de ago. de 2020.



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença

Creative Commons Atribuição – NãoComercial 4.0 Internacional.





Na segunda aula se espera discutir como é estruturado o processo de resolução de um problema computacional; apresentar as partes que constituem um algoritmo; como se dá a escrita de um algoritmo no App Inventor; introduzir os conceitos de variável e operadores; criar um aplicativo de cálculo de média de notas.

## Três Momentos Pedagógicos

### 1º MP

Discutir com os alunos como se dá o processo de resolução de um problema computacional ou não. Quais os passos necessários desde o entendimento de um problema até a construção e validação de uma solução adequada? Quais problemas reais podem ser resolvidos ou otimizados a partir da construção de um algoritmo?

Ainda, apresentar as partes que compõem um algoritmo e propor que os alunos analisem o primeiro aplicativo desenvolvido na aula anterior procurando identificar tais partes e a importância de cada uma delas.

Conceituar as partes de um algoritmo e realizar uma atividade prática colaborativa no Google Jamboard procurando esclarecer a composição de um algoritmo de cálculo de média entre dois números. Após a construção do quadro pelos alunos, fomentar a discussão a respeito de como os pilares do pensamento computacional podem ser aplicados na resolução desse cálculo.

Apresentar os conceitos de variável e operadores e como ocorre a escrita de um algoritmo no App Inventor. Desenvolver de um aplicativo para cálculo de média de duas notas.

### 2º MP

Retomar a discussão sobre o processo de resolução de problemas computacionais, agora procurando empregar os conceitos apresentados (ex: quais variáveis e operadores seriam utilizados para a construção dos algoritmos sugeridos na problematização inicial?).

Lançar o desafio de incrementar o aplicativo de cálculo de média desenvolvido em aula, sendo capaz de obter a média de quatro notas.

### 3º MP

#### Revisão do Desafio da Aula 1

Escaneie ou clique no QR Code ao lado para baixar o arquivo do projeto do App Inventor (\*.aia) com a resolução do desafio proposto na aula anterior. Aproveite para revisar e comparar com a interface de login do sistema acadêmico que você desenvolveu.



## Como resolver problemas?

Existem diferentes formas de se pensar a solução de um problema, seja ele computacional ou não. No entanto, guiar-se através de uma sequência lógica pode otimizar o processo de resolução. Com base nos preceitos do pensamento computacional, podemos seguir os seguintes passos para solucionar um problema:

- Entendimento do problema;
- Análise considerando os pilares do pensamento computacional (decomposição, reconhecimento de padrões, abstração);
- Criação de uma sequência de operações para a solução (algoritmo);
- Execução dessa sequência;
- Verificação da adequação da solução;



## Partes de um algoritmo

Os algoritmos normalmente possuem uma estrutura dividida em 3 partes: entrada, processamento e saída (MANZANO; OLIVEIRA, 2016). O entendimento de como começar a construção do algoritmo e como conduzir o seu desenvolvimento passa pela compreensão destas partes.

- **Entrada de dados:** informações necessárias para que o algoritmo possa ser executado (usuário -> computador);
- **Processamento de dados:** avaliadas todas as expressões algébricas, relacionais e lógicas existentes no algoritmo;
- **Saída de dados:** resultados do processamento (computador ->



usuário);

Num algoritmo computacional, a entrada normalmente ocorre a partir do teclado ou de dados armazenados em memória. No processamento esses dados são tratados para que possam resolver o problema ao qual o algoritmo se propõe. Essa é a principal e mais complexa parte do trabalho. Por fim, os resultados são exibidos em um ou mais dispositivos de saída, como: monitor, impressora ou a memória do dispositivo (ASCENCIO; CAMPOS, 2012).

#### Vamos entender na prática?



Entrada | Processamento | Saída

Utilize o [Google Jamboard](#) para realizar uma atividade coletiva colaborativa de construção das partes de um algoritmo para o cálculo da média entre dois números, dividindo o quadro em "Entrada | Processamento | Saída" e solicitando aos participantes que insiram notas autoadesivas em cada uma das partes.

Após, discuta com os participantes como é possível abordar a resolução desse problema utilizando os pilares do pensamento computacional.

## Escrita de Algoritmos

Independentemente da linguagem de programação utilizada, a escrita de um algoritmo deve sempre considerar:

- **Sintaxe:** forma como os comandos devem ser escritos para que sejam interpretados e entendidos pelo dispositivo;

**se (<expressão>) então  
<instrução>**

- **Semântica:** é o significado, conjunto de ações executadas a partir dos comandos;

**se o valor da expressão for verdadeiro,  
a instrução será executada**

Além disso, um algoritmo computacional costuma seguir uma estrutura básica com a definição de nome, declaração de variáveis e blocos de comandos com início e fim delimitados, a exemplo:

```
algoritmo_nome
  declaração de variáveis globais
  início_bloco
    bloco de comandos
  fim_bloco
fim_algoritmo
```

Podem existir um ou mais blocos de comandos em um algoritmo, cada qual contendo: declaração de variáveis locais, comentários, comandos de E/S e atribuição, estruturas de seleção e repetição e chamadas de funções (MANZANO; OLIVEIRA, 2016).



## Algoritmos no App Inventor

O fato do App Inventor ser uma linguagem de programação visual em blocos acaba por simplificar a construção de aplicativos, uma vez que abstrai parte da estrutura tradicional de um algoritmo.

No App Inventor os trechos de comandos são limitados pelos próprios blocos [1] que, por sua vez, são divididos entre **internos** [2] e de **componentes** [3], tendo a sua funcionalidade categorizada por diferentes cores (MIT, 2020).

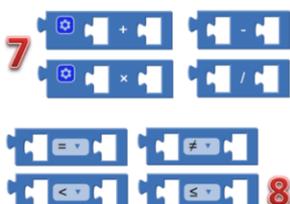
### Variáveis

Variáveis são entidades destinadas a guardar uma determinada informação (números, palavras, arquivos, etc.) (MANZANO; OLIVEIRA, 2016). Cada variável possui um nome único e, no App Inventor, a sua declaração ocorre através de um bloco de inicialização [4]. Também existem blocos específicos para alterar [5] ou obter [6] o valor da variável enquanto o aplicativo é executado.

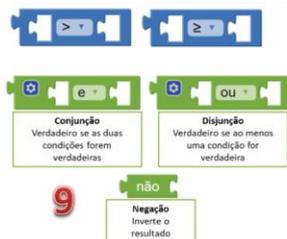
### Operadores

Operadores são utilizados para a construção de algoritmos no App Inventor e se dividem em (MIT, 2020):

- **Aritméticos:** conjunto de símbolos que representam as operações básicas da matemática (soma, subtração, multiplicação, divisão, etc) [7];



- **Relacionais:** utilizados para comparar valores entre variáveis e



expressões do mesmo tipo. O retorno é sempre um valor booleano/lógico (V ou F) [8];

- **Lógicos:** fazem comparações com o objetivo de avaliar expressões em que o resultado pode ser verdadeiro ou falso (lógica booleana) [9];

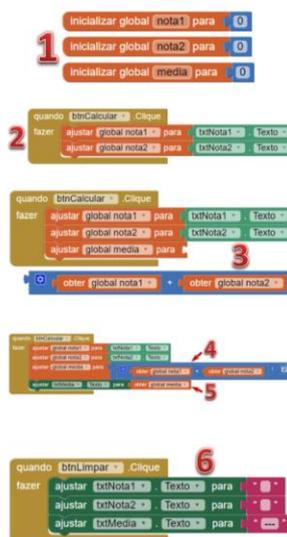
**Vamos entender na prática?**

Execute a sequência de passos abaixo para criar um aplicativo de cálculo de média de dois números.

- Selecionar o componente "Screen1" e alterar as propriedades:
  - Theme = Device Default
  - Título = *Calcula Média*
- Arrastar para a tela um componente "Legenda" e alterar:
  - FonteNegrito = V
  - TamanhoDaFonte = 16
  - Largura = Preencher principal...
  - Texto = *Calcula Média*
  - AlinhamentoDoTexto = centro:1
- Arrastar para a tela um componente "Legenda" e alterar:
  - Texto = *Nota 1*
- Arrastar para a tela um componente "CaixaDeTexto" e alterar:
  - Renomear = *txtNota1*
  - Dica = *Informe a primeira nota*
  - SomenteNúmeros = V
- Arrastar para a tela um componente "Legenda" e alterar:
  - Texto = *Nota 2*
- Arrastar para a tela um componente "CaixaDeTexto" e alterar:
  - Renomear = *txtNota2*
  - Dica = *Informe a segunda nota*
  - SomenteNúmeros = V
- Arrastar para a tela um componente "OrganizaçãoHorizontal" e alterar:
  - Largura = Preencher principal...
- Arrastar um componente "Botão" para dentro do "OrganizaçãoHorizontal" e alterar:
  - Renomear = *btnCalcular*
  - FonteNegrito = V
  - Largura = Preencher principal...
  - Texto = *Calcular*
- Arrastar outro componente "Botão" para dentro do "OrganizaçãoHorizontal" e alterar:
  - Renomear = *btnLimpar*
  - FonteNegrito = V
  - Largura = Preencher principal...
  - Texto = *Limpar*
- Arrastar para a tela um componente "Legenda" e alterar:
  - FonteNegrito = V
  - Largura = Preencher principal...
  - Texto = *A sua média é:*
  - AlinhamentoDoTexto = centro:1
- Arrastar para a tela um componente "Legenda" e alterar:
  - Renomear = *txtMedia*
  - FonteNegrito = V
  - Largura = Preencher principal...
  - Texto = *---*
  - AlinhamentoDoTexto = centro:1



- Na janela de Blocos, inserir variáveis para o cálculo e inicializá-las com "o" (bloco "Matemática") [1];
- Nos blocos do "btnCalcular", inserir o evento "Clique" e definir as variáveis globais conforme as notas informadas pelo usuário [2];
- Nos blocos de "Matemática", inserir o bloco de adição e somar as variáveis de notas [3];
- Nos blocos de "Matemática", inserir o bloco de divisão, incluir o bloco de soma no primeiro parâmetro, informar o número de notas no segundo parâmetro (com outro bloco "Matemática") e atribuir para a variável "media" [4];
- Nos blocos do "txtMedia", inserir o método "ajustar txtMedia.Texto para" e atribuir o valor da variável "media" [5];
- Nos blocos do "btnLimpar", inserir o evento "Clique" e definir a propriedade "Texto" dos campos nota e media [6];



**Desafio**



No aplicativo de cálculo de média construído em aula, implemente a possibilidade de inclusão de mais notas, de modo que seja possível realizar o cálculo da média de ao menos quatro números, conforme a figura ao lado.



Clique abaixo para realizar o download da aula 2.



Vamos aprofundar os conhecimentos?

Para saber mais sobre algoritmos, variáveis e operadores, explore o material abaixo que contém vídeos sobre a temática abordada na aula 2.



Curso "Introdução a Algoritmos", Prof. Gustavo Guanabarra



O que é "o tal do Algoritmo?", Prof. Dionatan Simioni



Aula sobre variáveis, comandos de entrada e saída, atribuição e operadores, Prof. Dr. Junior Freitas (IFC)

## Referências

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. de. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ e Java. 3. ed. São Paulo: Pearson, 2012.

MANZANO, J.; OLIVEIRA, J. Algoritmos - Lógica para desenvolvimento de Programação. 28. ed. São Paulo: Saraiva, 2016.

MIT. MIT App Inventor. 2020. Disponível em: <http://appinventor.mit.edu>. Acesso em: 25 de mar. de 2020.



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição- NãoComercial 4.0 Internacional.





Na terceira aula se espera reforçar os conceitos apresentados na aula anterior revisando a construção do aplicativo de cálculo de média a partir do uso de uma estrutura condicional; compreender o que é e como utilizar uma estrutura de seleção e de seleção aninhada no App Inventor; criar um aplicativo de índice de massa corporal.

## Três Momentos Pedagógicos

### 1º MP

Discutir com os alunos como ocorre a utilização de uma estrutura de seleção no dia-a-dia. Em quais situações reais utilizamos uma estrutura condicional? Quais problemas podem ser resolvidos ou otimizados com suporte de uma estrutura de seleção? Quais critérios podem ser utilizados para a escolha (seleção)?

Ainda, propor que os alunos analisem o aplicativo desenvolvido na aula anterior com o intuito de identificar como uma estrutura de seleção pode ser implementada para classificar o resultado da média como "aprovado" ou "reprovado".

Compreender o que é uma estrutura de seleção, apresentar seu conceito e demonstrar a sua construção no App Inventor através do incremento de uma nova funcionalidade no aplicativo de cálculo de média desenvolvido na aula anterior.

Da mesma forma, conceituar a respeito de uma estrutura de seleção aninhada, demonstrando seu uso no App Inventor.

### 2º MP

Retomar a discussão sobre a aplicabilidade de uma estrutura de seleção, agora instigando os alunos a pensarem em problemas que poderiam ser solucionados a partir do desenvolvimento de aplicativos com essa funcionalidade.

Como tarefa, lançar o desafio de desenvolver um aplicativo para o cálculo do índice de massa corporal que apresente a faixa de desnutrição ou obesidade na qual o indivíduo se encontra.

### 3º MP

#### Revisão do Desafio da Aula 2

Escaneie ou clique no QR Code ao lado para baixar o arquivo do projeto do App Inventor (\*.aia) com a resolução do desafio proposto na aula anterior. Aproveite para revisar e comparar com o aplicativo de cálculo da média de quatro notas que você desenvolveu.



## Estrutura de Seleção

Uma estrutura de seleção, também conhecida como estrutura condicional, permite o direcionamento da sequência de execução de um algoritmo de acordo com a aceitação ou não de certas condições (ASCENCIO; CAMPOS, 2012). As condições testadas são do tipo lógico (V ou F). A estrutura de seleção pode ser composta por:

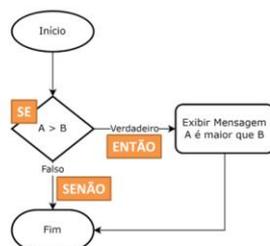
- SE - condição;
- ENTÃO - o que fazer quando verdadeiro;
- SENÃO - o que fazer quando falso;

Se a estrutura de seleção conter somente a condição verdadeira (se/então), é considerada **simples**. Do contrário, havendo um bloco de comandos para a condição falsa (senão), é conhecida como **composta**.

Um algoritmo de seleção costuma ter o seguinte formato:

```
se <condição> então
  bloco de comandos (V)
senão
  bloco de comandos (F)
fim-se
```

No App Inventor, uma estrutura de seleção pode ser utilizada a partir de dois blocos distintos de controle, conforme a figura ao lado.



#### Vamos entender na prática?

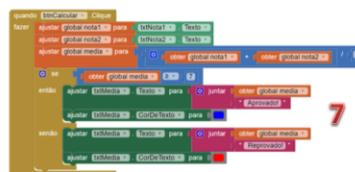
Observe as figuras ao lado e reflita a respeito da aplicação de estruturas de seleção no seu dia-a-dia. Como se dá o processo de autoatendimento em um caixa eletrônico ou num caixa de supermercado? Como classificamos e selecionamos



os produtos dispostos em uma gôndola? Como ocorre a separação do lixo que produzimos?

Agora, no App Inventor, vamos continuar os passos para a criação do aplicativo de cálculo de média.

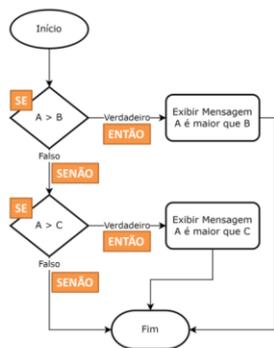
- Nos blocos de “Controle”, inserir a estrutura “se-então-senão” [1];
- Nos blocos de “Matemática”, inserir o operador relacional “>” e testar se a média é superior ou igual a 7 [2];
- Se verdadeiro, ajustar a propriedade “Texto” incluindo a descrição “Aprovado!”. Para isso, inserir o bloco “juntar” do tipo “Texto” [3]. Alterar também a propriedade “CorDeTexto” para azul (bloco “Cores”) [4];
- Se falso, ajustar a propriedade “Texto” incluindo a descrição “Reprovado!”. Para isso, inserir o bloco “juntar” do tipo “Texto” [5]. Alterar também a propriedade “CorDeTexto” para vermelho (bloco “Cores”) [6];



- Remover o bloco anterior que altera a propriedade “Texto” e arrastar o bloco “se-então-senão” para o evento “Clique” do “btnCalcular” [7];

- Na janela de Designer, inserir dois componentes “Som” da gaveta “Mídia” e renomeá-los para “SomAprovado” e “SomReprovado”. Alterar a propriedade “Fonte” enviando os arquivos disponíveis em: [https://drive.google.com/drive/folders/11\\_1T-2T-38\\_R/WmjxkvOhsF2tn5R-bl?usp=sharing](https://drive.google.com/drive/folders/11_1T-2T-38_R/WmjxkvOhsF2tn5R-bl?usp=sharing) (realizar o download antes de inserir no App Inventor) [8];

- Na janela de Blocos, inserir o procedimento “Tocar” dos componentes de som nos respectivos blocos condicionais de média [9];



### Estrutura de Seleção Aninhada

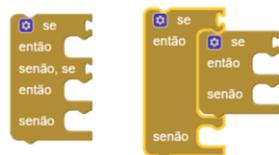
Também conhecida como estrutura de seleção encadeada, possibilita a inclusão de um comando de seleção dentro de outra estrutura de seleção (MANZANO; OLIVEIRA, 2016). Normalmente utilizada para realizar várias comparações com a mesma variável.

Suas condições testadas também são do tipo lógico e o seu formato costuma ser:

```

se <condição1> então
  se <condição2> então
    bloco de comandos (V)
  fim-se
senão
  se <condição3> então
    bloco de comandos (V)
  senão
    bloco de comandos (F)
  fim-se
fim-se
    
```

No App Inventor, uma estrutura de seleção aninhada pode ser utilizada a partir de dois blocos distintos de controle, conforme a figura ao lado, sendo o exemplo da esquerda um bloco único, enquanto o da direita é composto por dois blocos de seleção simples encaixados.



#### Vamos entender na prática?

No App Inventor, vamos implementar uma estrutura de seleção aninhada para verificar se o resultado da média habilita o aluno para o exame (>=6).

- Nos blocos de “Controle”, inserir outra estrutura “se-então-senão” no bloco condicional “senão” existente [1];
- Nos blocos de “Matemática”, inserir o operador relacional “>=” e testar se a média é superior ou igual a 6 [2];
- Se verdadeiro, ajustar a propriedade “Texto” incluindo a descrição “Em Exame!”. Para isso, inserir o bloco “juntar” do tipo “Texto” [3]. Alterar também a propriedade



"CorDeTexto" para amarelo (bloco "Cores") [4];

- Se falso, encaixar os blocos de reprovação que já haviam sido criados na condicional "senão" [5];

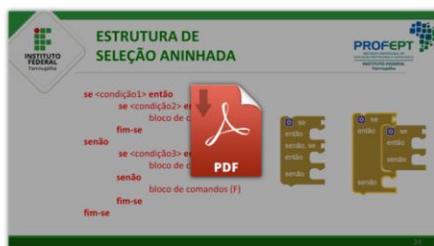
### Desafio



Aproveite os conhecimentos obtidos no exemplo trabalhado em aula e implemente um novo aplicativo para o cálculo do IMC - Índice de Massa Corporal, apresentando a faixa de desnutrição ou obesidade em que o indivíduo se encontra, conforme a classificação sugerida na figura ao lado.



Clique abaixo para realizar o download da aula 3.



### Vamos aprofundar os conhecimentos?

Para saber mais sobre estruturas de seleção, explore o material abaixo que contém vídeos sobre a temática abordada na aula 3.



Aula sobre estruturas de seleção, Prof. Dr. Junior Freitas (IFC)



Aula sobre estruturas condicionais, Prof. Gustavo Guanabara



Aula sobre estruturas de seleção no App Inventor, projeto de extensão "Pensando em Códigos" (UNIFAL)

### Referências

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. de. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ e Java. 3. ed. São Paulo: Pearson, 2012.

MANZANO, J.; OLIVEIRA, J. Algoritmos - Lógica para desenvolvimento de Programação. 28. ed. São Paulo: Saraiva, 2016.



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição - NãoComercial 4.0 Internacional.



PC e App Inventor

Página Inicial Apresentação Sequência Didática Dissertação Publicações Contato Sobre o site

# Sequência Didática

## Aula 4

### Estruturas de Repetição no App Inventor

Na quarta aula se espera reforçar os conceitos apresentados na aula anterior revisando a construção do aplicativo de índice de massa corporal; compreender o que são e como utilizar diferentes estruturas de repetição no App Inventor; criar um aplicativo para calcular o fatorial de um número.

## Três Momentos Pedagógicos

### 1º MP

Discutir com os alunos como ocorre a utilização de uma estrutura de repetição no dia-a-dia. Em quais situações reais utilizamos uma estrutura de repetição? Quais problemas podem ser resolvidos ou otimizados com o suporte de uma estrutura de repetição? Quais artefatos podem ser utilizados durante a sua aplicação para o controle do número de repetições, evitando a execução infinita?

Compreender o que é uma estrutura de repetição nas suas diferentes formas, apresentar seu conceito e demonstrar a sua construção no App Inventor através do desenvolvimento de exemplos práticos.

Da mesma forma, conceituar a respeito do uso de contadores para auxiliar no controle do número de vezes que uma mesma instrução ocorre.

### 2º MP

### 3º MP

Retomar a discussão sobre a aplicabilidade das estruturas de repetição, agora instigando os alunos a pensarem em problemas que poderiam ser solucionados a partir do desenvolvimento de aplicativos com essa funcionalidade e qual tipo de estrutura é mais adequada para cada situação.

Como tarefa, lançar o desafio de desenvolver um aplicativo para o cálculo do fatorial de um número.

### Revisão do Desafio da Aula 3

Escaneie ou clique no QR Code ao lado para baixar o arquivo do projeto do App Inventor (\*.aia) com a resolução do desafio proposto na aula anterior. Aproveite para revisar e comparar com o aplicativo de cálculo do Índice de Massa Corporal que você desenvolveu.



## Estruturas de Repetição

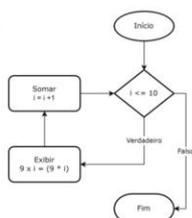
São utilizadas em situações onde é necessário repetir uma tarefa mais de uma vez. Dessa forma, possibilitam a repetição de um conjunto de instruções sem que seja necessário escrevê-las várias vezes.

As estruturas de repetição podem ser classificadas em dois tipos:

- **Condicional** - repetição ocorre enquanto a condição lógica testada for verdadeira (ENQUANTO);
- **Incondicional** - tem um número pré-determinado de repetições (PARA);

### Tabuada do 9

9 x 1 = 9  
9 x 2 = 18  
9 x 3 = 27  
...  
9 x 10 = 90



### Vamos entender na prática?

Observe as figuras ao lado e reflita a respeito da aplicação de estruturas de repetição no seu dia-a-dia. Como podemos explicar o processo que realizamos para nos alimentarmos? Ou então, para lavar a louça suja após uma refeição? Ainda, como realizamos o cálculo da tabuada de um número?



### Estrutura de Repetição PARA

A estrutura de repetição "Para" é ideal em situações onde o número exato de repetições é conhecido (ASCENCIO; CAMPOS, 2012). Seu funcionamento consiste em repetir a instrução enquanto uma variável de controle (var\_controle) for menor ou igual ao valor final (vtr\_fim). A



variável de controle é iniciada através de um comando de atribuição (`vir_ini`) e é incrementada pelo valor da cláusula passo (`<inc>`), conforme a estrutura abaixo.

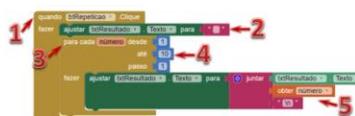
```
para <var_controle> = vir_ini até vir_fim passo <inc> faça
  bloco de comandos
fim-para
```

No App Inventor, uma estrutura de repetição "Para" pode ser utilizada a partir de um bloco de controle, conforme a figura ao lado, já composto por três blocos numéricos de matemática para inicialização da variável de controle, definição do valor final e do passo.

### Vamos entender na prática?

No App Inventor, vamos criar um aplicativo que exiba na tela os números de 1 a 10 utilizando um bloco de repetição do tipo "Para".

- Na janela de Designer, arrastar para a tela um componente "Botão", renomea-lo para "btRepeticao" e alterar a propriedade "Texto" para "Repetição";
- Arrastar para a tela um componente "Legenda", renomea-lo para "txtResultado" e apagar o conteúdo da propriedade "Texto" (em branco);
- Na janela de Blocos, nos blocos do "btRepeticao", inserir o evento "Clique" [1];
- Nos blocos do "txtResultado", inserir o método "ajustar txtResultado.Texto para" e atribuir o valor " " (em branco) a partir de um bloco de "Texto" [2];
- Nos blocos de "Controle", inserir a estrutura "para cada número desde" [3] e alterar o valor final (até) para "10" [4];
- Inserir outro método "ajustar txtResultado.Texto para" e atribuir o valor do próprio "txtResultado.Texto" em conjunto com o valor da variável "número" e um bloco de "Texto" com a descrição "\n" para realizar a quebra de linha a cada repetição. Para isso, inserir o bloco "juntar" do tipo "Texto" [5];



## Estrutura de Repetição ENQUANTO

A estrutura de repetição "Enquanto" é ideal em situações onde o número exato de repetições não é conhecido (ASCENCIO; CAMPOS, 2012). Seu funcionamento consiste em repetir a instrução (o ou "n" vezes) enquanto a condição for verdadeira. A condição é testada no início da instrução (V ou F), sendo a execução finalizada caso a condição se torne falsa. Sua escrita é representada abaixo.

```
enquanto <condição> faça
  bloco de comandos
fim-enquanto
```

No App Inventor, uma estrutura de repetição "Enquanto" pode ser utilizada a partir de um bloco de controle, conforme a figura ao lado.



### Contador

Diferentemente da estrutura de repetição "Para" que já inicia e incrementa uma variável de controle, a estrutura "Enquanto" requer a implementação dessa variável dentro do bloco de comandos. Essa variável costuma receber o nome de "contador".

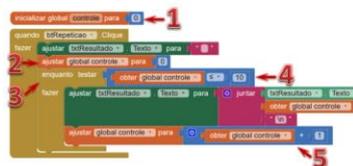
Uma variável "contador" é utilizada para o controle do número de vezes que um bloco de instruções é repetido. A variável recebe ela mesma, mais um determinado valor (normalmente "1") e costuma ser inicializada com "0" (zero), sendo a instrução de incremento exibida abaixo.

```
i = i + 1
```

### Vamos entender na prática?

No App Inventor, vamos alterar o aplicativo que exibe na tela os números de 1 a 10, agora utilizando um bloco de repetição do tipo "Enquanto".

- Na janela de Blocos, inserir uma variável, alterar seu nome para "controle" e inicializá-la com "0" [1];
- Abaixo do bloco "ajustar txtResultado.Texto para", alterar o valor da variável "controle" para "0" inserindo o bloco "ajustar global controle para" [2]. Essa atribuição é essencial para que o algoritmo possa ser executado mais de uma vez;
- Nos blocos de "Controle", inserir a estrutura "enquanto testar" no local da estrutura "para" do exemplo anterior [3];
- Nos blocos de "Matemática", inserir o operador relacional "<=" e testar se a variável "controle" é menor ou igual a 10



[4];

- Manter a atribuição do "txtResultado" do exemplo anterior, inserir outro bloco "ajustar global controle para" e um novo bloco de "Matemática" para atribuir o próprio valor da variável "controle", acrescido de "1" [5];

#### Desafio



Aproveite os conhecimentos obtidos no exemplo trabalhado em aula e implemente um novo aplicativo para o cálculo do fatorial de um número. Não se esqueça que só é possível calcular o fatorial de números positivos.

$$n!$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

Clique abaixo para realizar o download da aula 4.



#### Vamos aprofundar os conhecimentos?

Para saber mais sobre estruturas de repetição, explore o material abaixo que contém vídeos sobre a temática abordada na aula 4.



Aula sobre estruturas de repetição, Prof. Dr. Junior Freitas (IFC)



Aula sobre estruturas de repetição, Prof. Gustavo Guanabara



Aula sobre estruturas de repetição no App Inventor, projeto de extensão "Pensando em Códigos" (UNIFAL)

#### Referências

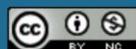
ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. de. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ e Java. 3. ed. São Paulo: Pearson, 2012.



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição-NãoComercial 4.0 Internacional.





Na quinta aula se espera reforçar os conceitos apresentados nas aulas anteriores realizando uma revisão geral; retomar também a construção do aplicativo de cálculo do fatorial; compreender o que é e como utilizar o sensor acelerômetro no App Inventor, como atribuir foco em componentes de interface e criar várias telas num único aplicativo; assimilar como ocorre a escrita de algoritmos através da criação de procedimentos ou funções no App Inventor; resolver diferentes problemas matemáticos desenvolvendo um aplicativo móvel como desafio final.

## Três Momentos Pedagógicos

### 1º MP

Problematizar com os alunos quais as suas percepções e as mudanças provocadas a partir da aplicação dos conhecimentos adquiridos nas aulas anteriores. Ainda, quais sensores do smartphone eles utilizam a partir dos aplicativos instalados em seus dispositivos e como acreditam que podem incorporar esses sensores no desenvolvimento de *apps* pelo App Inventor. Provocar a discussão a respeito de como partes de algoritmos podem ser reaproveitadas para evitar a reescrita das instruções e como podemos resolver problemas e automatizar tarefas repetitivas com tal técnica.

Compreender como utilizar o sensor acelerômetro no desenvolvimento de um aplicativo no App Inventor, bem como realizar o controle de foco em diferentes componentes de interface. Apresentar o conceito de procedimento ou função e demonstrar como utilizá-lo na escrita de algoritmos no App Inventor. Ainda, indicar como se dá a criação e o fluxo de navegação entre diferentes telas em um mesmo aplicativo através do desenvolvimento de um exemplo prático.

### 2º MP

### 3º MP

Retomar a discussão sobre a repetição de partes de um algoritmo a partir do mecanismo conhecido como procedimento ou função. Analisar os aplicativos desenvolvidos nas aulas anteriores procurando identificar blocos de comandos repetitivos que poderiam ser agrupados em uma função, ou conjuntos de instruções que, mesmo não duplicadas, poderiam ser dispostas em formato de procedimento para simplificar o problema, dividindo o algoritmo em partes menores.

Como tarefa, lançar o desafio de resolver diferentes problemas matemáticos desenvolvendo um aplicativo móvel com base nos conhecimentos adquiridos no decorrer das aulas.

### Revisão do Desafio da Aula 4

Escaneie ou clique no QR Code ao lado para baixar o arquivo do projeto do App Inventor (\*.aia) com a resolução do desafio proposto na aula anterior. Aproveite para revisar e comparar com o aplicativo de cálculo do fatorial que você desenvolveu.



quando **SensorAcelerômetro1** Sacudindo  
fazer

## Sensor Acelerômetro

O App Inventor possui diferentes componentes que possibilitam o uso de sensores do smartphone na construção de aplicativos móveis (ex: sensor de luz, de pressão do ar, localização, temperatura, etc). No caso do acelerômetro, trata-se de um componente invisível que mede a aceleração aproximada em três dimensões usando unidades do Sistema Internacional em  $m/s^2$ , detectando se a posição do smartphone está sendo alterada (MIT, 2020) ([clique aqui](#) para saber mais).

O acelerômetro está disponível na paleta "Sensores" e, ao ser inserido, permanece invisível à interface da aplicação, podendo ser manipulado a partir de dois blocos de controle, um que verifica a alteração da aceleração do movimento e outro que identifica se o aparelho está sendo sacudido. Em nosso exemplo, utilizaremos o bloco "Sacudindo", conforme a figura ao lado.

### Vamos entender na prática?

No App Inventor, vamos alterar o aplicativo de cálculo do fatorial (desafio da aula 4), fazendo com que o mesmo comportamento do botão "Limpar" seja realizado ao sacudir o smartphone.

- Na janela de Designer, arrastar para a tela um componente "SensorAcelerômetro";
- Na janela de Blocos, no bloco do "SensorAcelerômetro1",



inserir o evento "Sacudindo" [1];

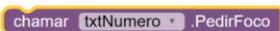
- Ajustar a propriedade "Texto" dos campos número e fatorial conforme realizado anteriormente para o "btLimpar" [2];



## Foco

Embora grande parte da interação de um aplicativo móvel ocorra através de toques na tela realizados pelo usuário sem uma ordem preestabelecida, é possível determinar o foco de alguns componentes, tornando-os ativos, para direcionar a sequência de ações do usuário ou simplesmente melhorar a sua experiência de uso do aplicativo.

No App Inventor, o foco pode ser estabelecido através do evento "PedirFoco" disponível em diferentes blocos de interface de usuário, a exemplo das caixas de texto, conforme a figura ao lado.



### Vamos entender na prática?

No App Inventor, ainda no aplicativo de cálculo do fatorial, vamos ajustar o foco para o componente "txtNumero" após o cálculo do fatorial e após a limpeza dos campos, de modo que o "Número" fique ativo e o usuário já possa digitar um novo valor para executar o cálculo novamente.

- Na janela de Blocos, no bloco do "txtNumero", inserir o procedimento "PedirFoco" após a última alteração da propriedade "Texto" do evento "Sacudindo" do acelerômetro [1]. Repetir o mesmo no evento "Clique" do "btLimpar" e do "btCalcular";

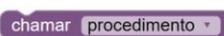


## Procedimento

Como vimos nas aulas anteriores, a escrita de algoritmos pode otimizar a realização de tarefas repetitivas. Da mesma forma, a capacidade de ser executado uma ou mais vezes é uma das premissas de qualquer algoritmo, porém a repetição da escrita de um bloco de comandos que realiza a mesma função não é uma boa prática. Para que a formulação do algoritmo não se torne repetitiva, existe o conceito de procedimento, também conhecido como **função**, **subprograma** ou **sub-rotina** (ASCENCIO; CAMPOS, 2012).

Um procedimento consiste em uma sequência de blocos ou código armazenado sob um nome, sendo utilizado para economizar espaço e tempo de programação, já que pode ser usado várias vezes num mesmo programa (MANZANO; OLIVEIRA, 2016). Portanto, um procedimento possibilita repetir blocos de comandos sem que seja necessário reescrevê-los, bastando chamar o procedimento pelo seu nome sempre que sua sequência de instruções deva ser executada.

No App Inventor, a criação de uma função ocorre a partir de um bloco interno de "Procedimentos" com o formato "para procedimento fazer", conforme a figura ao lado. Já o bloco de chamada do procedimento só é exibido após a sua criação e possui o formato "chamar procedimento".



### Vamos entender na prática?

No App Inventor, seguindo no aplicativo de cálculo do fatorial, vamos criar um procedimento para limpar os campos de número e fatorial e chamá-lo no evento "Clique" do botão limpar e no evento "Sacudindo" do sensor acelerômetro.

- Na janela de Blocos, no bloco interno de "Procedimentos", inserir a função "para procedimento fazer" e alterar seu nome para "limpar" [1];
- No procedimento "limpar", ajustar a propriedade "Texto" dos campos número e fatorial [2];
- No evento "Clique" do "btLimpar", inserir o bloco de chamada do procedimento "limpar" [3]; Repetir o mesmo para o método "Sacudindo" do sensor acelerômetro [4];



## Várias Telas

É comum aplicativos móveis se fazerem valer de mais de uma tela em função da limitação de tamanho da interface dos smartphones. O App Inventor permite a criação de várias telas num único aplicativo a partir do botão "Adicionar tela ..." localizado na barra superior da janela de Designer.



abrir outra tela nomeDaTela Screen1

fechar tela

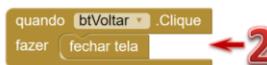
Ao criar uma nova tela, é necessário lhe atribuir um nome que, posteriormente, será utilizado para a navegação através de blocos de "Controle". Para abrir as telas criadas, deve-se utilizar o bloco "abrir outra tela nomeDaTela", selecionando a tela desejada entre as criadas anteriormente.

Ao abrir uma tela, é recomendado fechá-la ao retornar à tela principal ou anterior através do bloco "fechar tela", afim de liberar memória do sistema.

### Vamos entender na prática?

No App Inventor, seguindo no aplicativo de cálculo do fatorial, vamos criar uma nova tela denominada "Tela Inicial" que consistirá na tela de abertura do aplicativo com um um botão centralizado que possibilitará o acesso a tela de cálculo do fatorial criada anteriormente.

- Na janela de Designer, clicar no botão "Adicionar Tela ..." e informar o nome "TelaInicial";
- Selecionar o componente "TelaInicial" e desmarcar a propriedade "TituloVisivel";
- Arrastar para a tela um componente "OrganizaçãoVertical" e alterar:
  - AlinhamentoHorizontal = Centro:3
  - AlinhamentoVertical = Centro:2
  - CorDeFundo = Preto
  - Altura = Preencher principal...
  - Largura = Preencher principal...
- Arrastar para a tela um componente "Botão" e alterar:
  - Renomear = *btInicial*
  - CorDeFundo = Azul
  - FonteNegrito = V
  - TamanhoDaFonte = 18
  - Forma = arredondado
  - Texto = *Cálculo do Fatorial*
  - AlinhamentoDoTexto = centro:1
  - CorDeTexto = Branco
- Na janela de Blocos, nos blocos do "btInicial", inserir o evento "Clique" e, nos blocos de "Controle", inserir o evento "abrir outra tela nomeDaTela Screen1" [1];
- Na janela de Designer, alternar a tela para "Screen1", inserir um componente "Botão" e alterar:
  - Renomear = *btVoltar*
  - FonteNegrito = V
  - Texto = *Voltar*
  - AlinhamentoDoTexto = centro:1
- Na janela de Blocos, nos blocos do "btVoltar", inserir o evento "Clique" e, nos blocos de "Controle", inserir o evento "fechar tela" [2];



### Desafio Final

Aproveite os conhecimentos obtidos nas aulas anteriores e selecione um ou mais dos desafios abaixo para implementar um aplicativo. Procure incluir no aplicativo as funcionalidades abordadas na aula 5.

#### 1. Calculadora de Juros Compostos

Uma aplicação no valor de R\$ "x" é feita por um período de "y" meses a juros de "z"% ao mês. Qual o valor que será resgatado ao final do período?

#### 2. Números Primos

Solicitar que o usuário digite um número, informar se o número é primo e listar todos os números primos até esse número digitado.

#### 3. Sequência de Fibonacci

Solicitar que o usuário informe um número e listar a quantidade de termos da Sequência Fibonacci correspondente. Trata-se de uma sequência de números, começando em zero, onde cada termo subsequente corresponde a soma dos dois anteriores. Ex: 0, 1, 1, 2, 3, 5, 8 ...

#### 4. Tabuada

Calcular e exibir a tabuada completa de um número informado pelo usuário. Ex:

1 x 1 = 1;  
2 x 1 = 2; ...

#### 5. Números Pitagóricos

Solicitar que o usuário informe um número e listar todos os Números Pitagóricos entre 1 e o número digitado. Um número inteiro positivo é dito um Número Pitagórico se existem inteiros positivos a e b tais que  $a^2 + b^2 = c^2$ , por exemplo: 3, 4 e 5. Ex:

a b c  
3 4 5  
5 12 13 ...

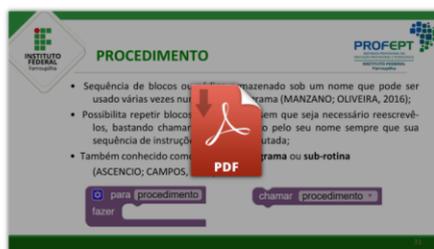


### Questionário final

No intuito possibilitar a comparação de resultados entre a caracterização inicial dos sujeitos até a percepção posterior das práticas propostas, optou-se por utilizar um questionário de percepção final, desenvolvido na ferramenta Google Formulários, com perguntas de avaliação das atividades desenvolvidas.

- RUIB
- PISANNA

Clique abaixo para realizar o download da aula 5.



Vamos aprofundar os conhecimentos?

Para saber mais sobre sensores, procedimentos e várias telas, explore o material abaixo que contém vídeos sobre a temática abordada na aula 5.



Criando o jogo "Mini Golf" com o sensor acelerômetro, Prof. Cassio Onodera (IFSF)



Aula sobre Procedimentos, Prof. Dr. Sérgio de Oliveira (UFSJ)



Múltiplas telas no App Inventor, Laboratório de Mídias Educacionais (UNIFAL)

## Referências

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. de. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ e Java. 3. ed. São Paulo: Pearson, 2012.

MANZANO, J.; OLIVEIRA, J. Algoritmos - Lógica para desenvolvimento de Programação. 28. ed. São Paulo: Saraiva, 2016.

MIT. MIT App Inventor. 2020. Disponível em: <http://appinventor.mit.edu>. Acesso em: 25 de mar. de 2020.



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição-NãoComercial 4.0 Internacional.





## Publicações

A presente pesquisa de mestrado nos levou a algumas publicações de artigos em eventos e periódico, além de um capítulo de livro. Também ministramos uma oficina num momento formativo junto a equipe de um projeto de iniciação tecnológica. Os trabalhos estão disponíveis em ordem cronológica nos links abaixo.



### Congresso Internacional de Educação e Tecnologias - CIET:EnPED 2020

EXPLORANDO O TERRITÓRIO DA PESQUISA-AÇÃO: CATEGORIZAÇÃO DE ESTUDOS ANTERIORES COM FOCO NO CIBERESPAÇO



### I Seminário Regional Sul de Educação Profissional e Tecnológica - SEPT 2021

FORMAÇÃO INTEGRADA NA LICENCIATURA EM COMPUTAÇÃO: PENSAMENTO COMPUTACIONAL E CRÍTICO PARA O ENSINO DE PROGRAMAÇÃO



### TRABALHO PEDAGÓGICO NA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA EM DIFERENTES CONTEXTOS: desafios e reflexões - volume 2

TRABALHO COMO PRINCÍPIO EDUCATIVO NA LICENCIATURA EM COMPUTAÇÃO: formação integrada, pensamento computacional e interação pedagógica



### Revista Tecnologias Educacionais em Rede (ReTER) - Dossiê Educação Profissional e Tecnologias em Rede - v. 2, n. 4, 2021

FORMAÇÃO INTEGRADA NA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA: PENSAMENTO COMPUTACIONAL E CRÍTICO POR MEIO DO ENSINO DE PROGRAMAÇÃO



### Oficina: Os Três Momentos Pedagógicos e programação em blocos

Formação da equipe do projeto de iniciação tecnológica "Prorobótica: pensamento computacional na Educação Básica". Gravação da oficina disponível no [site do projeto](#). Para mais informações, [clique aqui](#).



\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição- NãoComercial 4.0 Internacional.



PC e App Inventor

Página Inicial Apresentação Sequência Didática Dissertação Publicações **Contato** Sobre o site

# Contato

Desde já agradecemos o contato através do formulário abaixo com o relato da sua concepção a respeito deste Produto Educacional. Fale-nos sobre as suas impressões, o que achou do conteúdo da sequência didática, a sua forma de apresentação em formato de *website*, se acredita que o material é relevante e se pode ser utilizado em outros contextos com diferentes turmas. Ainda, o que poderia ser alterado ou melhorado e as suas sugestões para serem incorporadas ao produto.

Caso queira saber mais sobre a pesquisa ou deseja entrar em contato com os autores, estamos à disposição através do e-mail: [rodrigothomas@gmail.com](mailto:rodrigothomas@gmail.com).



## Concepções - Produto Educacional

Faça login no Google para salvar o que você já preencheu. [Saiba mais](#)

**\*Obrigatório**

Qual a sua concepção a respeito deste Produto Educacional? Quais as suas impressões com relação ao conteúdo da sequência didática e a apresentação em formato de site? Você acredita que o material é relevante e pode ser utilizado em outros contextos com diferentes turmas? O que poderia ser alterado ou melhorado e quais as suas sugestões para serem incorporadas ao produto? \*

Sua resposta

**Enviar** [Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Google Formulários Este conteúdo não foi criado nem aprovado pelo Google.

---



**PROFEPT**  
MESTRADO PROFISSIONAL EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL  
Farroupilha



**INSTITUTO  
FEDERAL**  
Farroupilha

\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença  
Creative Commons Atribuição- NãoComercial 4.0 Internacional.



## Sobre o site

Site desenvolvido para divulgação do Produto Educacional intitulado "*Sequência didática para o desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor*" apresentado ao Programa de Pós-Graduação em Educação Profissional e Tecnológica – ProfEPT do Instituto Federal Farroupilha – Polo Jaguari/RS – Brasil, pelo mestrando Rodrigo Thomas, sob orientação do Prof. Dr. Adão Caron Cambraia, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Demais informações relativas ao produto e à pesquisa podem ser encontradas na [dissertação](#) "*O Ensino de Programação na Educação Profissional e Tecnológica: desenvolvimento do pensamento computacional por meio da construção de aplicativos no App Inventor*".

### Rodrigo Thomas

Mestrando do Programa de Pós-Graduação em Educação Profissional e Tecnológica (ProfEPT) – IFFAR Polo Campus Jaguari, especialista em Gestão e Governança da Tecnologia da Informação pelo Centro Universitário Senac e graduado em Sistemas de Informação pela Universidade Regional Integrada do Alto Uruguai e das Missões – URI Santo Ângelo. Analista de Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia Farroupilha – Câmpus Santo Ângelo.

[Currículo Lattes](#)

### Orientador: Prof. Dr. Adão Caron Cambraia

Doutor e Mestre em Educação nas Ciências pela Universidade Regional do Noroeste do Estado do Rio Grande do Sul (Unijuí), especialista em Informática aplicada à Educação pela Universidade de Passo Fundo (UPF) e graduado em Informática pela Unijuí. Professor do Mestrado Profissional em Educação Profissional e Tecnológica (ProfEPT) do Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IFFar).

[Currículo Lattes](#)

\* website composto por mídias que permitem a sua incorporação com crédito atribuído por meio de link no próprio conteúdo.

Copyright © ProfEPT/IFFar

Esta obra está licenciada com uma Licença Creative Commons Atribuição–NãoComercial 4.0 Internacional.

## APÊNDICE B - QUESTIONÁRIOS

### Questionário Inicial

# O ensino de programação na Educação Profissional: construção do Pensamento Computacional por meio do desenvolvimento de aplicativos móveis

\*Obrigatório

1. Endereço de e-mail \*

---

2. Declaro que li o TCLE e concordo em participar da pesquisa: \*

*Marcar apenas uma oval.*

Sim

Não

### Informações Pessoais

3. Qual o seu nome? \*

Essa informação será utilizada apenas para controle de duplicidade de respostas. Não utilizaremos o seu nome para identificá-lo(a).

---

4. Em que ano você nasceu? \*

Responda somente com o ano, no formato de quatro dígitos (Exemplo: 1980).

---

## 5. Qual o seu gênero? \*

*Marcar apenas uma oval.*

- Feminino
- Masculino
- Prefiro não declarar
- Outro: \_\_\_\_\_

## 6. Qual(is) equipamento(s) você possui em casa? \*

É possível selecionar mais de uma alternativa.

*Marque todas que se aplicam.*

- Desktop (computador de mesa)
- Notebook
- Celular/Smartphone
- Tablet

## 7. Você tem acesso à internet em casa? \*

*Marcar apenas uma oval.*

- Sim
- Não

## 8. Se você respondeu "Sim" na pergunta anterior, qual tipo de conexão possui?

Selecione uma ou mais opções.

*Marque todas que se aplicam.*

- Fibra Óptica
- Via Rádio
- ADSL
- Dados móveis (3G, 4G)

Outro:  \_\_\_\_\_

9. Se você possui acesso à internet, qual(is) equipamento(s) utiliza para acessar?

Marcar apenas uma oval por linha.

	Não utilizo	Raramente	Pouco	Frequentemente	Muito
Desktop (computador de mesa)	<input type="radio"/>				
Notebook	<input type="radio"/>				
Celular/Smartphone	<input type="radio"/>				
Tablet	<input type="radio"/>				

### Conhecimentos sobre Algoritmos

10. O que você entende por algoritmo? \*

---



---



---



---



---

11. Quais conceitos você já estudou na disciplina de algoritmos? \*

É possível selecionar mais de uma alternativa.

Marque todas que se aplicam.

- Noções sobre dado, variável, instrução e programa
- Construções básicas (atribuição, leitura e escrita)
- Estruturas de controle (sequência, seleção e iteração)
- Tipos de dados
- Tipos estruturados básicos (vetores e matrizes)
- Subprogramas (funções, procedimentos e recursão)
- Manipulação de arquivos

Outro:  \_\_\_\_\_

12. Quais os passos que você executa para construir um algoritmo desde a apresentação do problema pelo professor(a)? \*

Descreva com suas palavras os passos executados.

---

---

---

---

---

13. Qual(is) a(s) sua(s) principal(is) dificuldade(s) na construção de um algoritmo? \*

---

---

---

---

---

#### Pensamento Computacional e desenvolvimento de aplicativos

14. Você já ouviu ou leu algo a respeito do termo "Pensamento Computacional"? \*

*Marcar apenas uma oval.*

- Sim, tenho domínio a respeito do termo.
- Sim, conheço pouco.
- Não, mas gostaria de conhecer.
- Não, desconheço e não tenho interesse.

15. Se você respondeu "Sim" na pergunta anterior, o que entende por Pensamento Computacional?

---

---

---

---

---

16. Você acredita que é importante aprender sobre o Pensamento Computacional? \*

Justifique sua resposta mais abaixo, na pergunta seguinte.

*Marcar apenas uma oval.*

- Sim  
 Não  
 Talvez

17. Favor justifique a resposta da pergunta anterior quanto a importância do aprendizado do Pensamento Computacional.

---

---

---

---

---

18. Você já utilizou alguma ferramenta de programação visual com blocos? \*

Exemplo: Schatch, MIT App Inventor, Blockly, entre outras.

*Marcar apenas uma oval.*

- Sim  
 Não

19. Se você respondeu "Sim" na pergunta anterior, qual ferramenta utilizou?

Selecione uma ou mais opções.

*Marque todas que se aplicam.*

- Scratch  
 MIT App Inventor  
 Blockly  
 Code.org

Outro:  \_\_\_\_\_

20. Você gostaria de aprender a desenvolver aplicativos móveis para smartphone/celular? \*

*Marcar apenas uma oval.*

Sim

Não

21. Se você respondeu "Sim" na pergunta anterior, qual(is) temática(s) acha interessante abordar ou qual problema gostaria de resolver a partir da construção de um aplicativo móvel?

Cite o(s) tema(s) ou problema(s) de seu interesse.

---

---

---

---

---

---

## Questionário de Percepção Final

# O ensino de programação na Educação Profissional: construção do Pensamento Computacional por meio do desenvolvimento de aplicativos móveis

\*Obrigatório

1. Endereço de e-mail \*

---

2. Declaro que li e concordo em participar da pesquisa: \*

*Marcar apenas uma oval.*

- Sim  
 Não

### Informações Pessoais

3. Qual o seu nome? \*

Essa informação será utilizada apenas para controle de duplicidade de respostas. Não utilizaremos o seu nome para identificá-lo(a).

---

### Avaliação das atividades desenvolvidas

4. O que você entende por Algoritmo? \*

---

---

---

---

---

5. O que você entende por Pensamento Computacional? \*

---

---

---

---

---

6. Você acredita que é importante aprender sobre o Pensamento Computacional? \*

*Marcar apenas uma oval.*

- Sim  
 Não  
 Talvez

7. O que você achou da ferramenta de programação visual com blocos MIT App Inventor? \*

*Marcar apenas uma oval.*

- Excelente  
 Boa  
 Razoável  
 Ruim  
 Péssima

8. De maneira geral, o que você achou das atividades desenvolvidas? \*

*Marcar apenas uma oval.*

- Excelentes
- Boas
- Razoáveis
- Ruins
- Péssimas
- Outro: \_\_\_\_\_

9. Você acredita que as atividades contribuíram para o seu entendimento dos conceitos abordados na disciplina de algoritmos? \*

*Marcar apenas uma oval.*

- Concordo totalmente
- Concordo
- Não estou decidido
- Discordo
- Discordo totalmente

10. Você acredita que as atividades contribuíram no seu processo de construção do Pensamento Computacional? \*

*Marcar apenas uma oval.*

- Concordo totalmente
- Concordo
- Não estou decidido
- Discordo
- Discordo totalmente

11. Você conseguiu desenvolver os aplicativos propostos ao final de cada atividade? \*

*Marcar apenas uma oval.*

- Sim, desenvolvi todos  
 Sim, desenvolvi a maioria  
 Sim, desenvolvi alguns  
 Tentei, mas não consegui  
 Não tentei

12. Qual(is) a(s) sua(s) principal(is) dificuldade(s) na construção dos aplicativos?

---

---

---

---

---

13. Você acredita que os conteúdos se tornam mais atrativos a partir do desenvolvimento de aplicativos móveis? \*

*Marcar apenas uma oval.*

- Concordo totalmente  
 Concordo  
 Não estou decidido  
 Discordo  
 Discordo totalmente

14. Você gostaria que esse tipo de atividade fosse incorporado à disciplina de algoritmos de forma permanente? \*

*Marcar apenas uma oval.*

- Gostaria muito  
 Gostaria  
 Não estou decidido  
 Não gostaria muito  
 Não gostaria

15. O que você mais gostou nas atividades desenvolvidas?

---

---

---

---

---

16. O que você menos gostou nas atividades desenvolvidas?

---

---

---

---

---

17. Você possui alguma sugestão, caso as atividades venham a ser desenvolvidas com outras turmas?

---

---

---

---

---

## APÊNDICE C - TCLE

### Termo de Consentimento Livre e Esclarecido - TCLE

O(a) senhor(a) está sendo convidado(a) a participar de uma pesquisa de mestrado intitulada: *O ensino de programação na Educação Profissional e Tecnológica: construção do Pensamento Computacional por meio do desenvolvimento de aplicativos móveis*. Antes de concordar em participar desta pesquisa e responder este questionário, é muito importante que o(a) Sr(a). compreenda as informações e instruções contidas neste documento. Os participantes do estudo são todos alunos do Curso de Licenciatura em Computação do IF Farroupilha – Campus Santo Ângelo. Todos serão convidados a responder um questionário estruturado a respeito de dados pessoais e de conhecimentos de programação. O questionário será aplicado através de um formulário eletrônico. A pesquisa tem como objetivo principal compreender as possíveis contribuições e/ou limites do desenvolvimento de aplicativos móveis para a aprendizagem de programação na EPT, bem como para a construção do Pensamento Computacional.

O(a) Sr(a)., ao aceitar participar da pesquisa, deverá:

1. Realizar o aceite eletronicamente, o que corresponderá à assinatura do TCLE, o qual poderá ser impresso, se assim o desejar.
2. Responder ao questionário online.

O questionário foi elaborado com o intuito de que o tempo gasto para seu preenchimento seja mínimo, em torno de 10 a 15 minutos. Não é obrigatório responder a todas as perguntas. O(a) Sr(a). não terá despesas e nem será remunerado pela participação.

A pesquisa apresenta riscos mínimos aos participantes, dentre os quais a possibilidade de constrangimento e cansaço ao responder o questionário proposto, bem como um possível desconforto ao informar opiniões pessoais, mesmo com a garantia de confidencialidade assegurada. Caso isso ocorra, o participante poderá desistir da atividade a qualquer momento. O pesquisador responsável se compromete a oferecer assistência integral ao participante da pesquisa que se sentir lesado e, se for o caso, encaminhá-lo ao serviço de atendimento especializado, além de garantir ressarcimento por possíveis danos decorrentes da pesquisa.

A manutenção do sigilo e da privacidade de sua participação e de seus dados está garantida durante todas as fases da pesquisa e posteriormente na divulgação científica. Para isso, a caracterização dos participantes será feita por codificação de sua identidade. Todos os dados obtidos na pesquisa serão utilizados exclusivamente com finalidades científicas.

Não existe benefício ou vantagem direta em participar deste estudo. Os benefícios e vantagens são indiretos, proporcionando retorno social através de possíveis melhorias no método de ensino-aprendizagem de programação e da publicação dos resultados da pesquisa em periódicos científicos.

O(a) Sr(a). pode entrar em contato com o pesquisador responsável a qualquer tempo para informações adicionais através do e-mail e telefones de contato listados abaixo.

**Pesquisador responsável:** Rodrigo Thomas

**Telefone para contato:** (51) 9 9941 9876 / (55) 3931-3922

**E-mail:** rodrigothomas@gmail.com

Ao concordar com este TCLE, uma cópia do mesmo será automaticamente enviada para seu endereço eletrônico informado no questionário online. Ao assinalar a opção “Sim” abaixo, o(a) Sr(a). concorda em participar da pesquisa nos termos deste documento e será direcionado ao questionário. Caso não concorde em participar, apenas feche essa página no seu navegador.

Declaro que li e concordo em participar da pesquisa:

Sim

Não