



**UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ**  
**INSTITUTO DE CIÊNCIAS EXATAS**  
**MESTRADO NACIONAL PROFISSIONAL EM ENSINO DE FÍSICA**

**USO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO COMO**  
**FERRAMENTA PARA O ENSINO DE FÍSICA**

**ROBERTO COSTA NUNES**

Marabá/PA  
Abril de 2019

**ROBERTO COSTA NUNES**

**USO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO COMO  
FERRAMENTA PARA O ENSINO DE FÍSICA**

Dissertação apresentada ao Mestrado Nacional Profissional em Ensino de Física (MNPEF), polo Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA), para obtenção do grau de Mestre em Ensino de Física.

**Orientador:** Prof. Dr. José Elisandro de Andrade

Marabá/PA  
Abril de 2019

**Dados Internacionais de Catalogação-na-Publicação (CIP)**  
**Biblioteca II da UNIFESSPA. CAMAR, Marabá, PA**

---

Nunes, Roberto Costa

Uso de algoritmos e linguagem de programação como ferramenta para o ensino de física / Roberto Costa Nunes ; orientador, José Elisandro de Andrade. — 2019.

Dissertação (Mestrado) – Universidade Federal do Sul e Sudeste do Pará, Instituto de Ciências Exatas - ICE, Mestrado Nacional em Ensino de Física - MNPEF, Marabá, 2019.

1. Física - Estudo e ensino. 2. Algoritmos. 3. Ensino auxiliado por computador. 4. Programação (Computadores). I. Andrade, José Elisandro de, orient. II. Universidade Federal do Sul e Sudeste do Pará. III. Título.

CDD: 22. ed.: 530.07

---

Elaborado por Nádya Lopes Serrão  
Bibliotecária-Documentalista CRB2/575

**SERVIÇO PÚBLICO FEDERAL**  
**UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ**  
**INSTITUTO DE CIÊNCIAS EXATAS-ICE**  
**MESTRADO NACIONAL PROFISSIONAL EM ENSINO DE FÍSICA**

**ATA DA APRESENTAÇÃO E DEFESA DE DISSERTAÇÃO DO MESTRADO  
NACIONAL PROFISSIONAL EM ENSINO DE FÍSICA**

ATA DA APRESENTAÇÃO E DEFESA DE DISSERTAÇÃO DE MESTRADO INTITULADO “USO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO COMO FERRAMENTA PARA O ENSINO DE FÍSICA” PARA CONCESSÃO DO GRAU DE MESTRE EM ENSINO DE FÍSICA, REALIZADA ÀS 14:00 HORAS DO DIA 11 DE ABRIL DE 2019, NO MINIAUDITÓRIO DO BLOCO III, CAMPUS II. A DISSERTAÇÃO FOI APRESENTADA DURANTE 50 MINUTOS PELO CANDIDATO **ROBERTO COSTA NUNES**, DIANTE DA BANCA EXAMINADORA APROVADA PELA SOCIEDADE BRASILEIRA DE FÍSICA, ASSIM CONSTITUÍDA: MEMBROS: PROF. DR. JOSÉ ELISANDRO DE ANDRADE (ORIENTADOR), PROF.<sup>a</sup> DR.<sup>a</sup> MARIA LIDUÍNA DAS CHAGAS (MEMBRO INTERNO), E PROF. DR. FRANCO JEFFERDS DOS SANTOS SILVA (MEMBRO EXTERNO). EM SEGUIDA, O CANDIDATO FOI SUBMETIDO À ARGUIÇÃO, TENDO DEMOSTRADO SUFICIÊNCIA DE CONHECIMENTOS NO TEMA OBJETO DA DISSERTAÇÃO, HAVENDO À BANCA EXAMINADORA DECIDIDO PELA APROVAÇÃO DA MESMA. PARA CONSTAR, FORAM LAVRADOS OS TERMOS DA PRESENTE ATA, QUE LIDA E APROVADA RECEBE A ASSINATURA DOS INTEGRANTES DA BANCA EXAMINADORA E DO CANDIDATO.

CANDIDATO: Roberto Costa Nunes

BANCA: José Elisandro de Andrade

Maria Liduína das Chagas

Franco Jeffers dos Santos Silva

Dedico este trabalho aos meus familiares, em especial à minha esposa que tanto me apoiou nessa empreitada, tornando essa jornada turbulenta bem mais suportável.

## AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Agradeço a Deus por ter me proporcionado desfrutar de grandes momentos, me fornecendo discernimento, saúde e força para iniciar, continuar e terminar este curso.

Agradeço ao meu orientador Prof. Dr. José Elisandro de Andrade, por ter ido além de seu dever como orientador, atuando como amigo e parceiro nessa longa jornada, tendo paciência e sabedoria para indicar os passos a serem seguidos.

Agradeço a UNIFESSPA personalizada na figura dos professores que pôde proporcionar a existência desse curso de mestrado.

Agradeço aos familiares, em especial à minha esposa Sulair, minha irmã Regina e cunhado Evaldo por serem as pessoas mais próximas que me possibilitaram perseverar nesse mestrado.

Agradeço aos meus colegas de curso Cairo, José Gidalto, Magno, Willamy, Leonardo, Mayk, Antônio Augusto, Elaine, Fabio e todos os outros com os quais dividi grandes momentos de alegrias e angústias, aprendendo coisas que antes eram inteligíveis.

Agradeço aos meus queridos professores, dos quais tive a honra de ter assistido as aulas, convivido e aprendido muito mais que fórmulas e equações.

Agradeço à CAPES pelo apoio financeiro por meio da bolsa concedida, o que foi providencial para minha manutenção e essencial para maior dedicação ao mestrado.

Agradeço ainda a cada uma das pessoas que passaram pela vida, mas que deixaram suas marcas em minha personalidade, contribuindo para a formação do meu caráter que tanto me orgulho.

## SUMÁRIO

<b>RESUMO</b> .....	<b>viii</b>
<b>ABSTRACT</b> .....	<b>ix</b>
<b>LISTA DE ILUSTRAÇÕES</b> .....	<b>x</b>
<b>1. INTRODUÇÃO</b> .....	<b>11</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>13</b>
2.1. JUSTIFICATIVA .....	13
2.2. OBJETIVOS .....	14
2.3. METODOLOGIA .....	14
2.4. PESQUISA BIBLIOGRÁFICA .....	17
2.4.1. Ensino de Física.....	20
2.4.2. Teorias de Aprendizagem.....	24
<b>3. APLICAÇÃO DO PROJETO: PROPOSTA DE INTERVENÇÃO</b> .....	<b>28</b>
3.1. SOBRE O VISUALG .....	28
3.2. MINICURSO .....	30
3.2.1. Definição – O que são algoritmos .....	31
3.2.2. Lógica de programação: aprendendo com o VisuAlg .....	32
3.2.2.1. Comando ESCREVA e ESCREVAL – Saída de dados.....	32
3.2.2.2. Variáveis e tipos primitivos.....	33
3.2.2.3. Comando LEIA – Entrada de dados.....	34
3.2.2.4. Operações aritméticas .....	34
3.2.2.5. Operadores relacionais .....	35
3.2.2.6. Estruturas condicionais .....	35
3.2.2.7. Estruturas de repetição .....	36
3.2.2.8. Exercícios .....	37
3.3. TRABALHANDO UM POUCO MAIS .....	40
3.3.1. Um pouco mais de cinemática.....	40
3.3.2. Trabalhando com aceleração .....	42
3.3.3. Trabalhando com escalas termométricas.....	44
3.4. LEVANTAMENTO E ANÁLISE DE DADOS.....	45
3.4.1. Questionário Inicial .....	46
3.4.1.1. Quanto ao tipo de escola que os alunos estudaram .....	46
3.4.1.2. Autoavaliação de conhecimentos em Física.....	47

3.4.1.3.	Autoavaliação de conhecimento em Matemática.....	48
3.4.1.4.	Autoavaliação em programação de computadores.....	49
3.4.1.5.	Já pensou em escrever programas de computadores? .....	50
3.4.1.6.	Uso do laboratório de informática da escola.....	50
3.4.1.7.	Tecnologias que os alunos mais têm acesso.....	51
3.4.1.8.	O minicurso pode ser proveitoso para você? .....	52
3.4.1.9.	Avaliando conhecimentos prévios em Física.....	53
3.4.2.	Questionário Final .....	55
3.4.2.1.	Percepção da relevância do conteúdo do minicurso.....	55
3.4.2.2.	Se o ensino de programação fosse ofertado na grade curricular da sua escola, você faria?.....	56
3.4.2.3.	A construção de algoritmos ajuda na resolução de problemas de Matemática e de Física? .....	56
3.4.2.4.	Verificando se o aluno se acha mais capaz de resolver problemas de Matemática e de Física depois do que foi aprendido.....	57
3.4.2.5.	Opinião dos alunos sobre o conteúdo abordado no minicurso.....	58
3.4.2.6.	Verificando os conhecimentos adquiridos dos alunos .....	59
3.5.	OUTRAS POSSIBILIDADES .....	62
<b>4.</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>69</b>
<b>5.</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>70</b>
	<b>APÊNDICES .....</b>	<b>73</b>
	APÊNDICE A - APOSTILA: MINICURSO DE LÓGICA DE PROGRAMAÇÃO PARA O ENSINO DE FÍSICA.....	73
	APÊNDICE B – LÂMINAS DE APRESENTAÇÃO.....	77
	APÊNDICE C – QUESTIONÁRIO INICIAL .....	82
	APÊNDICE D – QUESTIONÁRIO FINAL .....	83
	<b>PRODUTO EDUCACIONAL.....</b>	<b>84</b>

## RESUMO

A Física promoveu várias evoluções e revoluções na humanidade, possibilitando desde a compreensão do dia a partir do movimento do planeta até viagens espaciais. Assim como a Física está em constante desenvolvimento, o ensino de Física também necessita acompanhar essa evolução, e com o advento de tecnologias digitais as possibilidades de ensino podem ser expandidas e potencializadas. Procurando introduzir os estudantes da educação básica na área da computação, esse estudo se baseia no uso de lógica de programação para construção de algoritmos computacionais voltados para o ensino de Física. O objetivo desse trabalho é fazer com que os alunos busquem a construção de soluções para problemas de Física, elaborando algoritmos de computadores. Para viabilizar esse objetivo, foi utilizado o programa aplicativo VisuAlg para ensinar a lógica de programação na estrutura de um minicurso ofertado aos alunos da educação básica do primeiro ano do ensino médio da escola pública estadual Oneide de Souza Tavares, localizada na cidade de Marabá-PA. Foi necessário utilizar a sala de informática da escola em questão que possibilitou a seleção de 12 alunos. O projeto foi aplicado no mês de agosto de 2018, onde foi possível verificar que os alunos ainda não tinham sido introduzidos na programação e estavam iniciando os estudos de Física, onde apenas 17% dos alunos que formaram o grupo trabalhado conseguiram responder corretamente uma questão que envolve o cálculo de velocidade média e nenhum deles conseguiram resolver uma questão que envolve o cálculo de distância. Após aplicação do minicurso foi possível verificar o ganho que a prática proporcionou para a resolução de problemas de Física, onde ao final 83% dos alunos conseguiram responder corretamente o cálculo de velocidade e 75% conseguiram solucionar o problema de distância.

**PALAVRAS-CHAVE:** Ensino de Física, Algoritmo, Lógica de Programação.

## **ABSTRACT**

The Physics promoted several evolutions and revolutions in the humanity, enabling from the day-a-day understanding from the planet's movement until space travels. Just as Physics is constantly developing, Physics teaching also needs to keep up with this evolution, and with the advent of digital technologies, teaching possibilities can be expanded and enhanced. In order to introduce students of basic education in the area of computing, this study is based on the use of programming logic for the construction of computational algorithms geared towards teaching physics. The objective of this work is to make the students look for the construction of solutions to problems of physics, elaborating algorithms of computers. In order to achieve this objective, the VisuAlg application program was used to teach the programming logic in the structure of a mini-course offered to students of the elementary education of the first-year high school of the state public school Oneide de Souza Tavares, located in the city of Marabá-PA. It was necessary to use the computer room of the school in question that enabled the selection of 12 students. The project was applied in August 2018, where it was possible to verify that the students had not yet been introduced in the program and were beginning their studies of Physics, where only 17% of the students who formed the group worked were able to correctly answer a question that involves the calculation of average speed and none of them have been able to solve an issue that involves calculating distance. After applying the mini-course, it was possible to verify the gain that the practice provided to solve Physics problems, where at the end 83% of the students were able to correctly answer the speed calculation and 75% were able to solve the distance problem.

**KEY WORDS:** Teaching Physics, Algorithm, Programming Logic.

## LISTA DE ILUSTRAÇÕES

Figura 1. Imagem do laboratório de informática da escola .....	15
Figura 2. Tela Inicial do Programa VisuAlg. ....	29
Figura 3. Opção para executar o algoritmo. ....	30
Figura 4. Saída de dados.....	32
Figura 5. Manipulação de variáveis.....	33
Figura 6. Exemplo do comando LEIA .....	34
Figura 7. Exemplo de algoritmo que lê dois números e mostra o resultado da soma .....	35
Figura 8. Exemplo de algoritmo que faça operação relacional entre dois números.....	35
Figura 9. Exemplo de algoritmo com estrutura condicional. ....	36
Figura 10. Exemplo de algoritmo que conte até cinco com a estrutura ENQUANTO. ....	36
Figura 11. Exemplo de algoritmo que conte até cinco com a estrutura REPITA.....	37
Figura 12. Exemplo de algoritmo que conte até cinco com a estrutura PARA DE ATE.....	37
Figura 13. Exercício 1: somar dois números e mostrar resultado na tela.....	38
Figura 14. Exercício 2: exemplo de algoritmo do segundo exercício. ....	38
Figura 15. Exercício 3: exemplo de algoritmo com equação do primeiro grau. ....	39
Figura 16. Exercício 4: exemplo de cálculo da velocidade média. ....	39
Figura 17. Exemplo de como calcular a distância com base na velocidade média e o tempo. ....	41
Figura 18. Exemplo de como encontrar o tempo com base na vel. média e distância .....	41
Figura 19. Exemplo de função horário do MUV.....	42
Figura 20. Exemplo de código para a função horária do espaço no MUV.....	43
Figura 21. Exemplo de algoritmo para converter de Celsius para Fahrenheit. ....	45
Figura 22. Questão 1: quanto ao tipo de escola em que estudou.....	46
Figura 23. Questão 2: Como você avalia o seu aprendizado em Física?.....	47
Figura 24. Questão 3: como você avalia o seu aprendizado em Matemática? .....	48
Figura 25. Questão 4: Como você avalia o seu conhecimento em Programação? .....	49
Figura 26. Questão 5: Você já pensou em escrever programas de computadores?.....	50
Figura 27. Questão 6: É comum utilizar a sala de informática da escola? .....	51
Figura 28. Questão 7: Quais tecnologias você mais tem acesso?.....	52
Figura 29. Questão 8: Você acha que esse minicurso pode ser proveitoso para você?.....	53
Figura 30. Questão 9, letra a: Qual a sua velocidade média? Respostas dos alunos.....	54
Figura 31. Questão 9, letra "a": Qual a sua velocidade média? Índice de acertos. ....	54
Figura 32. Questão 1: percepção do aluno sobre a relevância do conteúdo do minicurso .....	55
Figura 33. Questão 2: se o ensino de programação fosse ofertado na grade curricular .....	56
Figura 34. Questão 3: a construção de algoritmos ajuda na resolução de problemas .....	57
Figura 35. Questão 4: Com o que foi ensinado no minicurso, você se acha mais capaz .....	58
Figura 36. Questão 6a: cálculo de velocidade média. ....	60
Figura 37. Questão 6b: cálculo de distância. ....	61
Figura 38. Linhas de códigos de programa em Java para conversão de temperatura.....	63
Figura 39. Tela do programa criado pelo grupo 1. ....	65
Figura 40. Tela do programa criado pelo grupo 2. ....	66
Figura 41. Tela do programa criado pelo grupo 3. ....	67
Figura 42. Tela do programa criado pelo grupo 4. ....	68

# 1. INTRODUÇÃO

No trabalho realizado por GARLET, BIGOLIN & SILVEIRA (2016) é relatado a importância da iniciação na lógica de programação desde a educação básica, para não apenas colocar os jovens como consumidores de tecnologias, mas entender como estas funcionam. Não distante dessa realidade, as habilidades aprendidas com a programação de computadores, aguçam aspectos formativos nos jovens, além da própria iniciação científica, a habilidade de propor soluções para problemas diversos utilizando o raciocínio lógico e a noção de causa e consequência. Ainda de acordo com os autores, no Reino Unido a educação básica já contempla em seu currículo o ensino de programação e de lógica de programação no ensino básico.

Ao encontro dessa concepção, LEONARDO (2013) diz que já existem projetos que incluem o ensino de programação na educação básica brasileira, concebendo o ensino de informática de uma maneira bem mais ampla e moderna, onde o jovem não aprende apenas a mexer no computador, mas aprende ainda como a tecnologia funciona e como ela pode ser produzida, frisando que “... nas escolas, codificação de computadores deveria estar lado a lado com matérias tradicionais como biologia, química e Física”. Conforme os relatos pontuados pelo autor, o ensino de programação não se dá apenas por necessidades do mercado de trabalho, mas também pelas habilidades desenvolvidas com a aprendizagem da programação, como a linguística, raciocínio matemático, causa e consequência e lógica.

No que tange a construção da Base Nacional Comum Curricular (BNCC), GARLET, BIGOLIN & SILVEIRA (2016) incluem a necessidade da inserção da computação no ensino básico, onde a Sociedade Brasileira de Computação (SBC) lançou a proposta para incluir na BNCC a utilização de programação para construir algoritmos que solucionem problemas do cotidiano dentro da grade curricular do ensino médio, onde os alunos podem se habilitar a criar algoritmos para automatizar soluções de Biologia, Física, Matemática, Português, Química entre outras.

Percebe-se que uma abordagem em que tente conciliar os conceitos físicos com a programação é bastante obscura, uma vez que pode ser levantado diversas barreiras e dificuldades. Justamente diante dessa relação pouco explorada entre programação e ensino de Física, é que este trabalho busca se apoiar, tendo em vista a necessidade de

expor os dados de pesquisa dessa relação, com as dificuldades que podem ser encontradas e ainda os possíveis ganhos para o processo de ensino e aprendizagem de Física e ainda de programação.

A problemática explorada nesse trabalho envolve o uso de algoritmos voltados para o ensino de Física aplicado no ensino básico. A proposta se baseia então no fato dos alunos poderem exercitar os conceitos físicos aprendidos em sala de aula por meio da construção de algoritmos computacionais, adquirindo dessa forma as habilidades esperadas para solucionarem os problemas de Física e ainda adquirem habilidades voltadas para a iniciação na área de construção de códigos de programas de computadores.

A organização desse trabalho busca trazer um texto o mais explicativo possível sobre os assuntos necessários para serem abordados de acordo com a sua proposta, todavia procurou-se construir um texto objetivo e conciso para uma leitura fácil, eficiente e rápida. Vale salientar que esgotar esse tema seria uma tarefa tecnicamente inviável e não estaria dentro dos objetivos desse trabalho, então as explicações aqui contidas buscam contribuir sobre a abordagem de se utilizar a construção de algoritmos na educação básica para apoiar o ensino de Física.

Tendo então essas considerações, o segundo capítulo desse trabalho procura revelar a fundamentação teórica que norteia o seu desenvolvimento, trazendo a justificativa, objetivos e metodologia para o seu desenvolvimento. Ainda nesse mesmo capítulo é descrito sobre as referências bibliográficas voltadas ao ensino de Física e as teorias de aprendizagem que podem ser exploradas, pontuando assim a revisão de literatura mais relevante encontrada que apoiem a finalidade desse trabalho.

Já no terceiro capítulo é dissertado acerca da aplicação da proposta pedagógica, descrevendo a ferramenta utilizada, a didática empregada, o levantamento de dados e a análise dos resultados obtidos. A proposta didática foi fundamentada basicamente no ensino de conceitos introdutórios de lógica de programação que possibilitasse aos alunos desenvolverem algoritmos simples e praticassem por meio de exercícios com soluções coletivas entre o grupo de alunos.

No quarto capítulo são pontuadas as considerações finais, demonstrando assim o que pode ser concluído do trabalho com base na análise dos dados levantados e nos resultados obtidos com a aplicação do projeto.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1. JUSTIFICATIVA**

A utilização da programação como ferramenta para o ensino de Física está baseada na característica da construção de lógica para solucionar problemas de Física em uma linguagem computacional, alterando a forma como o aluno enxerga a tecnologia e a ciência, deixando de ser mero consumidor de tecnologias, mas entender ainda como ela é feita e torná-lo um potencial desenvolvedor de tecnologias futuras. O fato de o aluno propor as soluções de problemas e implementar na forma de programação, faz com que ele construa o seu conhecimento das duas áreas, potencializando a aprendizagem dos conceitos e das equações abordadas na Física da educação básica.

Geralmente os alunos aprendem determinado algoritmo para solução de um dado problema e sentem uma dificuldade ao aplicar esse algoritmo para solução de outros problemas. Quando o aluno constrói o seu próprio algoritmo, ele não está apenas decorando a solução ou memorizando os passos algébricos, está externando o seu entendimento sobre o assunto e compreendendo o problema de fato. Assim, esse trabalho se justifica pelo fato de o aluno poder estudar duas áreas de conhecimento diferente ao mesmo tempo, onde uma está apoiada na outra e exercita simultaneamente os conceitos de programação e de Física, aprimorando ainda a sua compreensão da área de Matemática.

Baseado na inclusão da tecnologia na metodologia de ensino, esse trabalho busca de forma ousada, incluir o uso de algoritmos e programação de computadores para mediar o ensino de Física nas instituições de ensino básico. O intuito é que com essa abordagem os alunos estejam não somente inseridos no uso de recursos informacionais, mas também possam construir por conta própria, programas para computadores ou dispositivos móveis, ou ao menos ter maior compreensão sobre o que está por trás dos programas. Essa perspectiva ainda instiga o uso do raciocínio matemático como uma habilidade que é formada com a construção das estruturas de códigos de computadores, com a tentativa de mudar a visão de que os alunos deixem de

ser meros usuários de tecnologias, transformando a sua perspectiva passiva para ativa quanto a tecnologia da informação.

## **2.2. OBJETIVOS**

O objetivo geral desse trabalho é fazer com que os alunos busquem a construção de soluções para problemas de Física, elaborando algoritmos de computadores e os programem em uma linguagem específica para gerar os resultados. Neste trabalho será utilizado a construção de algoritmos em pseudocódigo por meio do programa VisuAlg, o qual permite a construção e execução de algoritmos construídos em português estruturado, comumente chamado de portugol.

Dessa relação, espera-se alcançar ainda os seguintes resultados:

- Maior fixação do conteúdo de Física;
- Desenvolver a lógica computacional para resolver os problemas de Física;
- Exercitar as relações Matemáticas para montar as fórmulas aprendidas;
- Aprender sobre o funcionamento de variáveis e estruturas da lógica de programação;
- Construção de soluções de problemas;

## **2.3. METODOLOGIA**

Buscando o embasamento teórico para fundamentar este trabalho, inicialmente foram realizadas diversas pesquisas de modo quantitativo, tendo assim uma base de informações que poderiam nortear o mesmo. Com esse estudo foi possível fazer o refino das ideias contidas nas diversas literaturas levantadas, tendo assim a pesquisa qualitativa, sendo possível definir com mais clareza as próximas etapas e visualizar os objetivos e resultados que podem vir a se obter.

Para que os objetivos desse trabalho fossem alcançados, foi pensado na estrutura de oferta do ensino de lógica de programação para alunos da rede pública do

ensino básico, em especial aos alunos de ensino médio, na forma de minicurso. Para dar continuidade ao projeto, a próxima etapa consistiu na busca por uma escola que oferecesse o espaço adequado. Dentro das possibilidades encontradas, foi selecionada a escola pública Professora Oneide de Souza Tavares, localizada na cidade de Marabá, no estado do Pará, bairro Nova Marabá. A referida escola oferece tanto o ensino fundamental II (6º ao 9º ano), como o ensino médio. A escola possui um laboratório de informática com 18 (dezoito) computadores, tendo 12 (doze) deles funcionando e disponíveis para uso. Apesar do espaço do laboratório ser pequeno, aquém da necessidade da própria escola, ele é funcional, sendo útil e utilizado pela escola na medida de sua capacidade. Na Figura 1 é possível observar o espaço com a união de duas fotos na tentativa de mostrar todo o espaço do laboratório de informática.

Figura 1. Imagem do laboratório de informática da escola



Fonte: próprio autor.

Tendo então o espaço escolhido, foi ofertado a uma turma do primeiro ano do ensino médio dessa escola a possibilidade de participar do minicurso, em caráter facultativo, no contraturno do horário normal de aula. No momento em que a sala foi visitada, havia cerca de 30 (trinta) alunos presentes na sala de aula, sendo que desses, 12 (doze) se propuseram a participar, que seria a capacidade do próprio laboratório de informática.

O minicurso se baseou no ensino de estrutura de lógica de programação. A sala de informática foi fundamental para se alcançar os objetivos almejados nesse trabalho, pois o contato com o computador enquanto é ensinado a teoria, proporciona a prática logo em seguida. A codificação se deu em linguagem de algoritmo no que é conhecido

como português estruturado, chamado de portugol, o que torna a codificação de algoritmos em linguagem de fácil abstração e aprendizagem.

Pensando na melhor experiência possível e adequada para o nível dos alunos, foi utilizado o software VisuAlg. Com este programa é possível programar diretamente no computador o código em portugol. Dessa forma, os alunos aprendem a estrutura de programação diretamente no computador, associando as operações matemáticas com as operações que o computador realiza. No final os alunos foram ensinados como construir algoritmos que envolva equações Matemáticas e Físicas e assim desafiados a implementar um algoritmo que envolva equações horárias do movimento.

Uma observação importante que deve ser feita, é que algoritmo e programa são conceitos diferentes. Aquele é uma série de etapas construídas de forma lógica de modo a solucionar um problema, enquanto esse são conjunto de código ou instruções a serem executadas por um computador. Assim, um programa pode ser entendido como um algoritmo escrito em uma linguagem que o computador pode entender (executar). Como foi utilizado o VisuAlg nesse trabalho, o algoritmo é mais estruturado e pode ser executado diretamente por meio do aplicativo utilizando a linguagem portugol. Então, ao falar algoritmo, nesse caso, pode ser compreendido como o programa, e vice-versa, já que o mesmo pode ser executado.

Os assuntos abordados no minicurso englobaram os seguintes tópicos:

- Algoritmo;
- Estrutura e lógica de programação;
- Mostrar informação na tela e ler uma entrada de dados;
- Variáveis e manipulação de dados;
- Operadores aritméticos, lógicos e relacionais;
- Estruturas de repetição e tomadas de decisão;
- Construção de equações e funções;
- Algoritmo para cálculo de velocidade média e distância percorrida.

Por fim, a última etapa do projeto consiste em tabelar e analisar os dados coletados durante a execução do projeto em si. O trabalho é redigido com os resultados e as possíveis conclusões, tendo assim o texto escrito passado pela redação final.

## 2.4. PESQUISA BIBLIOGRÁFICA

Na pesquisa em ensino, não é muito difícil encontrar textos científicos que abordam as práticas pedagógicas e colocam em xeque aquelas metodologias ditas tradicionais, com profundas críticas às práticas de ensino consideradas antigas e certa glorificação às práticas mais “modernas” com metodologias que elevam o uso de recursos inovadores, principalmente aquelas relacionadas à informática (uso de projetores, sala de informática, internet, modelos computacionais, por exemplo). Apesar dessas críticas geralmente terem uma fundamentação a partir de observações científicas, é necessário ter muita cautela. Uma prática dita tradicional, não necessariamente está fadada ao fracasso, nem tão pouco pode ser considerada ultrapassada. Uma prática apoiada por recursos tecnológicos não necessariamente irá levar ao total aprendizado. Este trabalho busca abordar o uso de tecnologias no ensino, de forma a justificar o seu uso, porém é necessário realizar algumas ponderações sobre as diferentes abordagens de práticas pedagógicas. Existem diversas formas de se ensinar, e cabe ao professor verificar a que melhor se adequa à sua prática docente, aos objetivos desejados, as limitações dos alunos e ainda a realidade da escola e dos alunos.

SINUMA (2017) estabelece uma relação entre algumas formas de aprender e sua efetividade, elencando, em especial, a teoria da Pirâmide de William Glasser e o Cone de Aprendizagem de Edgar Dale. As duas teorias trazem em comum o aspecto de que práticas de ensino e estudo que envolvam maior participação e ação do aluno, como explicar, resumir, estruturar, elaborar ou fazer algo de forma ativa, leva a uma maior retenção do que foi estudado. Dessa forma, atividades que envolvam a construção de algo relacionado ao objeto de estudo, têm maiores chances de sucesso no que diz respeito a retenção na estrutura cognitiva do aluno, em vez dele ser mero receptor passivo da informação. O autor levanta assim a necessidade de constante evolução do espaço escolar para que essa não se torne uma entidade obsoleta.

De acordo com PEREIRA JUNIOR (2017), as estatísticas na educação apontam que a maioria dos estabelecimentos de ensino mantem aulas em sua grande maioria expositivas, ditas tradicionais, deixando os conceitos físicos distantes do cotidiano dos alunos. Para ALMEIDA (2016), o ensino tradicional não está restrito apenas pela prática de aulas expositivas, mas também está marcado com aulas divididas

em turnos, onde é necessário apoio familiar no contra turno para o pleno desenvolvimento cognitivo do aluno. Assim, um professor que está nesse contexto de ensino considerado tradicional, necessita de além de dominar o conhecimento de sua área, uma grande habilidade para passar o seu conhecimento para os alunos.

Para conseguir transpor esses obstáculos, o professor pode se apoiar em metodologias que utilizam recursos diferenciados, conforme o objetivo esperado em sala de aula. Certamente os recursos que possuem grande notoriedade atualmente, são aqueles baseados na Tecnologia da Informação e Comunicação (TIC). Conforme aponta BARBOSA & NUNES (2011), aulas apoiadas em TIC podem ser concebidas de diversas formas, como por exemplo: enciclopédias digitais e sites da internet que possibilitam o acesso, fácil, rápido e interativo a informações; algum fenômeno físico ou teoria pode ter sua abstração facilitada substancialmente com o uso de animações, imagens e vídeos; e aplicativos e *softwares* educativos podem incentivar os alunos nos estudos por meio da ludicidade.

HOHENFELD, LAPA & MARTINS (2007) relatam que o advento das tecnologias da comunicação e informação disponibilizou uma maior quantidade de informação, tanto para o professor, quanto para o aluno, porém indaga se essas informações, ou ainda, as tecnologias estruturam uma nova aula ou permanece na “velha” aula com a presença da tecnologia. Assim, é importante verificar se os professores, em formação ou formados, estão inseridos no contexto das TIC, tendo contemplado as tecnologias no seu processo formativo. Os autores ainda acrescentam que “...de forma alguma, estratégias de ensino através das mídias irão mascarar uma formação específica deficiente ou a falta de condições de trabalho dos professores...”, podendo gerar conflitos no trabalho docente. Para superação de tais conflitos, o professor necessita de uma ação formativa multifacetada nas questões didáticas, metodológicas e tecnológicas de ensino utilizando as tecnologias, assim como nos aspectos do seu conhecimento da disciplina específica.

Enquanto isso, BEHRENS, MASETTO & MORAN (2000) ponderam que o uso da informática para mediar as aulas, vem auxiliar notoriamente o processo de ensino aprendizagem, uma vez que o uso de metodologias de ensino fora do tradicional e ultrapassado instiga o aluno a estudar mais. Apesar disso, o Guia de Tecnologias Educacionais (GTE) do Ministério da Educação (MEC), (BRASIL, p. 17, 2008), pondera que mesmo sendo considerado importante o uso de uma tecnologia, o seu uso

pode se tornar desprovida de sentido se não se atentar para algumas observações. Desta forma, tal abordagem deve estar atrelada a uma visão educacional que compreenda o desenvolvimento humano com: a formação de cidadãos, a gestão democrática, o respeito à profissão do professor, assim como a qualidade social da educação.

Assim, para o uso de recursos tecnológicos em uma perspectiva de ensino, é trivial que essa abordagem tenha alguma orientação pedagógica. O recurso pode facilitar a prática docente, mas seu uso se torna sem sentido quando não traz nenhuma melhoria para o ensino. A utilização de determinado recurso não deve ser vista como a finalidade última da metodologia. O uso de TIC deve estruturar uma prática centrado no aluno, e não no professor, auxiliando na compreensão de conceitos abstratos. Ou seja, o termo facilitador de ensino deve estar atrelado a visão de que possibilita melhor compreensão dos conceitos abordados, melhorando o ensino e a aprendizagem. Veja que isso melhora a finalidade do ensino, mas não torna obrigatoriamente mais fácil o trabalho do professor, podendo, na verdade, torná-lo mais complexo.

MARTINS & GARCIA (2011) utilizam o termo NTIC, se referenciando a Novas Tecnologias da Informação e Comunicação, ponderando que a formação do professor não deve ser de apenas aprender a manusear o computador, utilizando-o como uma ferramenta, mas sim de contemplá-lo em seu cotidiano, em que passe a construir novos conhecimentos, tornando o processo formativo dos alunos na realidade das novas tecnologias da informação e comunicação. Ou seja, o uso das tecnologias deve ser apoiado nas teorias de aprendizagem, buscando uma reflexão sobre o ensino.

A concepção de nascidos na era digital é abordada por LARA *et. al.*(2013), acentuando a característica de que os estudantes atuais já estão inseridos na era digital desde crianças, utilizando, manipulando e operando dispositivos tecnológicos como computadores, vídeo games, câmeras de vídeo, tablets e smartphones com bastante facilidade, como se tudo isso fosse intuitivo aos seus olhos. Os autores defendem que a inserção de TICs para mediar o ensino de Física pode trazer muitos benefícios, e ainda pode contribuir consideravelmente para o processo de ensino e aprendizagem, tornando-se uma estratégia interessante.

A concepção acima é compartilhada por diversos outros autores. Assim é utilizado com certa frequência termos sinônimo a “nativos digitais”, onde as tecnologias digitais já fazem parte do cotidiano dos jovens, e parecem que eles possuem habilidades nativas para manipular certos dispositivos, como computadores e *smartphones*. Tal

visão é sustentada por COSTA, DUQUEVIZ & PEDROZA (2015), onde é colocado que os alunos já vêm inseridos nessa realidade contemporânea, onde a escola não pode se abster de abordar as tecnologias digitais, tanto para ganho didático quanto para o letramento digital.

#### **2.4.1. Ensino de Física**

Pensar em Física como ciência da natureza é pensar em passado, presente e futuro. Passado, pois as descobertas científicas atreladas à Física provocaram uma verdadeira reformulação na vida do homem, dando-lhe certa intimidade com a natureza, descobrindo os seus mais profundos segredos. Basta imaginar a ciência na história do homem: a compreensão de conceitos mais básicos, como a sequência de dias, até os mais complexos, como a nossa relação com os cosmos; conceitos como empuxo e força do vento possibilitou a construção de canoas e embarcações que boiem sobre a água e que possam ser levadas pelo vento que bate nas velas; e ainda as máquinas térmicas que protagonizaram a Primeira Revolução Industrial e possibilitaram realizar trabalhos antes impossíveis. Pensar no presente devido ao ritmo de vida moderno ser propiciado pelas tecnologias possibilitadas diretamente ou indiretamente por estudos de Física: energia, eletricidade, computação e transmissão de informações, por exemplo. Pensar em futuro ao imaginar as quase infinitas possibilidades que os avanços científicos e tecnológicos podem proporcionar: viagem em alta velocidade, exploração espacial, novas formas e fontes de energia, longevidade, etc.

Imaginar a nossa vida sem a Física, seria quase inimaginável. Pode parecer um paradoxo ou algo absurdo, mas compreender a importância da Física é entender que a nossa realidade está formada essencialmente por esta parte da ciência da natureza. Pensar no ensino de Física também é pensar na relação dessa distinta matéria com o cotidiano. Ensinar Física é aproximar os conceitos da realidade do aluno. O aprendizado de Física possibilita uma maior autonomia no seu cotidiano, como, por exemplo, ao fazer uma viagem, baseado na velocidade e na distância, calcular o tempo aproximado para completar o percurso, ou ainda saber que se andar mais rápido, levará menos tempo, e se andar mais devagar, levará mais tempo para terminar a viagem.

A Física não é uma ciência nova, e o ensino de Física é tão antigo quanto a própria ciência, todavia, enquanto atividade de pesquisa em educação é relativamente recente. Para GASPAR (2007) e MOREIRA (2000) seria possível situar o seu início em meados do século XIX com o surgimento dos primeiros livros didáticos de Física, só que sem qualquer fundamentação teórico-pedagógico. No Brasil, o ensino de Física teria pouco mais de meio século, e com esse breve histórico é possível constatar que houve sim alguns acertos, erros, equívocos assim como certas concepções que precisam ser melhoradas. De acordo com GASPAR (2007), e muito bem feita a sua colocação, é necessário repensar o papel do professor no ensino como interação social, alterando a concepção de mero intermediador e recolocando-o no centro do processo educacional, “tornando-o de fato o parceiro mais capaz de quem os alunos jamais vão poder prescindir”.

No ensino básico, a Física está incluída como disciplina para os três anos do ensino médio e nos anos finais do ensino fundamental como parte da disciplina de ciências naturais. A educação tem tentado evoluir para acompanhar as mudanças emergidas com era da tecnologia e comunicação, todavia deixando um ligeiro lastro pela dificuldade em acompanhar as evoluções tecnológicas. Uma breve análise na evolução dos Parâmetros Curriculares Nacionais, percebe-se que a formação dos alunos não é mais voltada unicamente para o mercado de trabalho, evoluindo do parâmetro mecanicista para a independência intelectual e formação de pensamento crítico. Talvez isso ainda seja uma utopia, mas não se pode negar o crescente debate para uma formação de alunos que estejam aptos a viverem no mundo atual, com diversas tecnologias e que possam pensar, pesquisar, decidir e continuar aprendendo em diversas situações, dentro e fora da sala. Esta concepção permeia a prática docente, aliando o conteúdo com as habilidades e competências para a formação de alunos que possam se tornar especialistas competentes.

PIETROCOLA (2005) coloca um debate importante no ensino de Física, destacando a diferença existente entre as pesquisas didáticas e educacionais que avançam muito mais do que é percebido na prática dentro das escolas. O autor atribui de certa forma esse descompasso à formação docente atual, onde o conhecimento específico da área de atuação e o conhecimento pedagógico não têm se mostrado suficiente para preparar o professor para a realidade vivida dentro das escolas. Transformar conceitos complexos em conceitos acessíveis aos alunos do ensino básico,

não é necessariamente um dos objetivos dos cursos de graduação. O universo escolar geralmente é diferente e bem mais complexo do que é trabalhado nas graduações e cursos de formação continuada.

Os termos ensinar e aprender devem ser então relacionados de tal maneira que este depende daquele e aquele implica neste. TRINDADE & TRINDADE (2004) expõem o papel do processo de ensino e aprendizagem na formação do conhecimento e, por consequência, na construção da realidade. O ensino de ciências tem essa relação umbilical com as mudanças de paradigmas que alteram a forma como o homem enxerga o mundo, formulando e testando novas explicações para os fenômenos naturais, observando os acontecimentos com novos olhos. Se o conhecimento evolui e a sociedade se transforma, a escola também deve rumar para essas mudanças e preparar o aluno para esse mundo repleto de transformações.

FREIRE (2002) faz um profundo debate sobre a característica de tornar o aluno independente, focando no termo autonomia, no qual o papel do professor enquanto docente se relaciona profundamente com o ensino e conseqüentemente a aprendizagem. Ensinar instiga a curiosidade, promove a pesquisa, leva a aceitação da constante mudança e da presença do novo. EINSTEIN (1981) expõe uma reflexão filosófica acerca do conhecimento científico construído até então, entoando a percepção humanística da ciência para tornar o ser humano com uma personalidade e não uma mera máquina utilizável capaz de repetir uma sequência de tarefas limitadas. Para ele o conhecimento não deve ser visto como uma obrigação penosa que deva ser meramente aprendida (ou decorada) apenas para fins de avaliação didática, mas como um bem de valor inestimável.

Certamente a valoração do conhecimento é algo bastante relativo, sendo negligenciado por boa parte dos alunos nos diversos níveis de ensino e ainda por uma parcela de professores. POZO & CRESPO (2009) utilizam uma concepção psicológica sobre a motivação ao enfrentar uma tarefa, a qual é resultado da interação entre dois fatores: a expectativa de êxito em uma tarefa e o valor concebido a esse êxito. A motivação no ensino está fortemente relacionada com o empenho do professor e do aluno. Percebe-se que a motivação está fortemente relacionada com o sucesso do ensino de ciências. Enquanto parte dos alunos não vêem significado em aprender certos conteúdos (como equações, funções ou conceitos complexos como eletricidade e magnetismo) tendo dificuldade em encontrar motivação, cresce no professor, de acordo

com os autores, a “sensação de desassossego, de frustração, ao comprovar o limitado sucesso de seus esforços docentes”.

É necessário então realizar uma reflexão dentro da prática docente sobre a função social do ensino. ZABALA (1998), bem como POZO & CRESPO (2009), pontuam que o sistema educacional está permeado pela concepção de que o ensino tem o papel selecionador e propedêutico, funcionando como uma espécie de filtro para quem quer atingir níveis mais avançados de escolarização e instrução. Porém, é necessário ainda considerar a função social formativa do ensino, onde o que é aprendido, em ciências, por exemplo, não é apenas conteúdo que será visto unicamente em avaliações ou exames de seleção, mas também faz parte do mundo em que o aluno está inserido. O conteúdo a ser ensinado pelo professor e o esforço que aluno fará para aprendê-lo, não está restrito às provas, mas expande a própria realidade da natureza e a compreensão do mundo.

ZABALA (1998) relaciona vários conteúdos de aprendizagem que podem ser abordados de acordo com objetivo dado para o ensino, mas destaca em especial três tipos de conteúdo que podem parametrizar bastante a prática docente: conteúdos conceituais, conteúdos procedimentais e conteúdos atitudinais. Aquele relaciona os fatos, conceitos e princípios, já esses os procedimentos, técnicas e métodos, enquanto estes os valores, atitudes e normas. Assim, um ensino que está voltado para a característica seletiva ou propedêutica, irá priorizar os conteúdos conceituais (principalmente) e os procedimentais em detrimento dos atitudinais. Já numa concepção formativa os conteúdos atitudinais ganham mais destaques, só que isso não implica no alijamento dos conteúdos conceituais e procedimentais.

Por fim, não há como se chegar na última palavra da prática docente ou formular alguma metodologia terminada, acabada e que dará certo em todas as situações e circunstâncias. O ensino de ciências está atrelado à própria característica da natureza científica: não se trata de um processo acabado e pronto, mas sim de um processo em construção, que está sempre evoluindo e mudando de acordo com a concepção que temos da atual realidade.

## 2.4.2. Teorias de Aprendizagem

As teorias de aprendizagem surgiram como uma necessidade de se quantificar e qualificar as diversas formas de ensino e concepções de aprendizagem, estruturando assim as mais variadas metodologias de ensino, dando-as embasamentos teóricos e psicológicos (ou não) sobre a construção do conhecimento. Uma metodologia pode estar fundamentada em uma ou mais teoria de aprendizagem. ZABALA (1998) pondera que a prática educativa deve levar em consideração diversos fatores e uma grande variedade de situações, onde o objetivo não pode ser a busca da “fórmula magistral”, mas a melhora da prática.

Uma rápida pesquisa pela literatura atual leva a divisão das teorias de aprendizagem em três vertentes, que são as correntes filosóficas subjacentes: comportamentalista (behaviorismo), cognitivismo (construtivismo) e humanista. MOREIRA (1999) utiliza essa divisão para classificar as diversas teorias de aprendizagem abordadas em seu trabalho, das quais serão destacadas as teorias de Piaget, Vygotsky e Ausubel, que se encaixam na corrente cognitivistas/construtivistas, porém, dependendo da prática docente, pode ser levado características da corrente humanista ou comportamentalista.

Procurando levar a uma melhor compreensão sobre os processos de ensino-aprendizagem, BARROS (2009) sintetizou as ideias de Piaget, Skinner e Gagné, pontuando que o professor, que tem a função de orientador e facilitador, juntamente com o educando são os agentes no processo de aprendizagem, que ocorre mediante o organismo ter um papel ativo na aquisição do conhecimento. Na teoria de Piaget é dada uma atenção especial quanto aos processos que influenciam no desenvolvimento da inteligência. Skinner trabalha a questão da melhora no desempenho, mediante o reforço ou gratificação unido a situações de privação. Gagné se preocupa quanto à estrutura do assunto a ser trabalhado, ou seja, para cada tipo há uma metodologia que melhor se ajusta. Todas as teorias supracitadas apontam que, para haver aprendizado deve-se dar importância às diferenças individuais na assimilação do conhecimento, a relevância das atividades e a necessidade de se promover uma continuidade lógica e psicológica no estudo de qualquer conteúdo.

Para FREITAS (2010) a educação tradicional ou tecnicista ainda possui grandes influências no presente, o modelo de ensino apesar das críticas e fatores negativos, permanece dominando em todos os graus da educação, impondo ao aluno um conhecimento já pronto, ou seja, neutro onde este o recebe sem que haja uma interação entre suas ideias com o assunto trabalhado. O conteúdo das disciplinas deve ser memorizado e depois reproduzido, sendo que através deste procedimento mecânico o educando deixa de refletir e expor seus pensamentos passando a ser apenas um copiador de ideias alheias, perdendo dessa forma, o poder de argumentação e pensamento crítico.

Fundamentado na teoria psicológica de Vygostsky, FREITAS (2010, v. 2, p. 63), afirma que, “o conhecimento se constrói nas relações interpessoais. Portanto, o sujeito do conhecimento não é apenas ativo, mais interativo”. Segundo a autora, tais ideias também são completadas por Vasili Davidov, quando defende que o papel da escola no presente não se configura em dar aos alunos um amontoado de informações, porém instrui-lhes a buscar suas respostas independentemente, colaborando desta forma para a construção do conhecimento.

Buscando promover essa independência do indivíduo na busca pelo saber, JACQUES DELORS (*apud* BEHRENS, 2000, p. 78) defende uma educação continuada, ou seja, uma aprendizagem que se perpetua por toda a vida, o aprender a conhecer. Com isso ele compreende que:

“Este tipo de aprendizagem que visa não tanto à aquisição de um repertório de saberes codificado, mas antes ao domínio dos próprios instrumentos do conhecimento pode ser considerado, simultaneamente, como meio e como finalidade da vida humana. Meio, porque se pretende que cada um aprenda a compreender o mundo que o rodeia, pelo menos na medida em que isso lhe é necessário para viver dignamente, para desenvolver as suas capacidades profissionais, para comunicar. Finalmente, porque seu fundamento é o prazer de compreender, de conhecer, de descobrir”.

Como é possível observar, teorias pedagógicas estão inseridas nos trabalhos de pesquisa da área de ensino. Tal perspectiva, por mais simples que se possa parecer, merece sim de fato atenção para as diferentes concepções de como o aprendizado ocorre. Assim, LARA *et. al.* (2013) diz que é muito comum os trabalhos da área de ensino abordarem teorias pedagógicas como as cognitivas de Piaget, sócio interacionistas de Vygotsky e ainda a aprendizagem significativa de Ausubel e Moreira. Tais abordagens demonstram aparente sucesso para as metodologias que as empregam em sala de aula, muito em fato por priorizarem uma abordagem humanista.

O principal ponto da teoria sócio interacionista de Vygotsky, é a abordagem das zonas de desenvolvimento, onde a aprendizagem ocorre na denominada Zona de Desenvolvimento Proximal (ZDP), e esta se posiciona entre Zona de Desenvolvimento Real, que são as capacidades já adquiridas pelo aluno que ele é capaz de aplicar sozinho o conhecimento, e a Zona de Desenvolvimento Potencial, que são aquelas que o sujeito pode desenvolver com o auxílio de outros. Assim, a ZDP é a distância entre a zona de desenvolvimento real e a potencial, ou seja, a distância entre aquilo que o sujeito já sabe e aquilo que ele tem potencial para desenvolver. (JUNIOR, 2017).

Enquanto na teoria de Piaget o sujeito já sabe os conceitos em seu subconsciente, sendo necessário passar por sequencias de estágios para o seu desenvolvimento, para Vygotsky o aprendizado se dá pelas relações sociais, por meio das interações com outros sujeitos. Talvez a vertente vygotskiana se sobrepõe a piagetiana, todavia não há um encaixe perfeito de nenhuma das duas perspectivas sobre o ensino para o presente trabalho. Diante disso, é relevante ponderar a teoria de aprendizagem significativa por propiciar uma relação do conteúdo com o aspecto prático do indivíduo.

De acordo com MOREIRA (1999), a aprendizagem significativa é uma teoria proposta por David Ausubel, onde o indivíduo é detentor de conceitos prévios, já existentes na sua estrutura cognitiva e internalizados, denominados de subsunçores, que servem de base para a ancoragem de novos conceitos que serão aprendidos. À medida que esse processo ocorre, os subsunçores são ampliados e modificados. A aprendizagem significativa ocorre, portanto, quando novos conhecimentos se relacionam com conceitos já presentes na estrutura cognitiva do aluno. Uma mera memorização arbitrária e literária é considerada mais uma aprendizagem mecânica. O autor diz ainda que quando se aborda uma aprendizagem significativa, porém os alunos não possuem o nivelamento esperado ou não desenvolveram os subsunçores necessários, pode-se utilizar o conceito de organizadores prévios, para servir de ponte entre o que o aluno já sabe e para aquilo que ele deve saber.

Para exemplificar uma aplicação com a abordagem de aprendizagem significativa, é possível citar dentro da Física, por exemplo, que os conceitos de distância e tempo servem de subsunçores para o conceito de velocidade. Quando o conceito de velocidade é internalizado na estrutura cognitiva do aluno, ancorado com os

subsunçores distância e tempo, houve aí aprendizagem significativa. O que é significativamente aprendido não é facilmente esquecido.

CALDAS (2017) coloca que a aprendizagem é significativa quando essa faz parte da realidade do aluno, onde esse realiza simulações do real e desenvolve proposições que sejam capazes de resolver problemas. A autora pondera ainda que o educador não deve se atentar apenas para a transmissão do conteúdo a ser ensinado, mas também de torná-lo acessível ao aluno, em uma linguagem que possibilite a relação com os subsunçores.

Tendo em vista que a teoria de Ausubel é a mais adequada para os objetivos desse trabalho, será abordada uma perspectiva de aprendizagem significativa. Essa visão se dará pelo fato de que ao se utilizar a programação para implementar as soluções de problemas de Física, é codificado as etapas formuladas, não apenas associando aos subsunçores, mas também construindo o conhecimento que eles possuem, tanto de programação quanto de Física.

O maior significado está na característica de que o aluno utiliza a lógica de programação para construir não apenas soluções de problemas de Física, mas desenvolver o raciocínio, melhorando a sua capacidade de desenvolver soluções futuras e diversas. A aprendizagem se torna ainda mais significativa por buscar em seu conhecimento prévio e interligar diversos subsunçores para formular a sua resposta, ampliando e modificando os conceitos subsunçores à medida que ele procura soluções diferentes ou adapta as já existentes para novas situações. ANJOS, FREITAS & ANDRADE NETO (2016) relatam que o processo de construção do algoritmo e conversão para um código, pode ser considerado um processo de transformação do conhecimento adquirido, sendo uma evidência de aprendizagem significativa de Ausubel.

### **3. APLICAÇÃO DO PROJETO: PROPOSTA DE INTERVENÇÃO**

O projeto teve a sua aplicação realizada nos dias 16 e 17 de agosto na Escola Estadual de Ensino Fundamental e Médio Profa. Oneide de Souza Tavares, sendo utilizado a forma de minicurso. Foi abordado o ensino de lógica de programação em português estruturado, comumente chamado de portugal, com auxílio da ferramenta VisuAlg, abordando os conceitos básicos e essenciais da programação para a construção de programas de computadores para iniciantes.

A escola possui um laboratório de informática, sendo possível acomodar 12 (doze) alunos, tendo um aluno por cada computador. O minicurso foi ofertado aos alunos do primeiro ano do Ensino Médio da escola em questão, tendo a inscrição de 12 (doze) alunos.

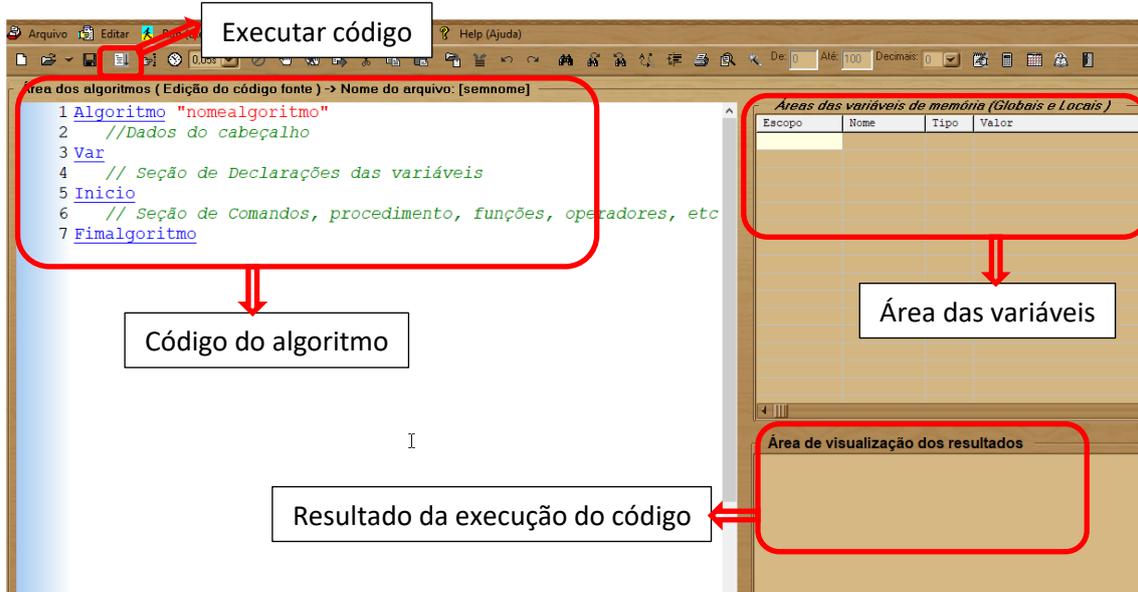
O minicurso foi estruturado para ser ensinado os comandos de entrada e saída de dados, os tipos de variáveis, como utilizar os operadores aritméticos e relacionais, as estruturas condicionais e ainda as estruturas de repetição. Os exemplos e as atividades formuladas visam a aplicação desses conhecimentos com os conteúdos de Física e Matemática, exercitando os conteúdos aprendidos em sala de aula.

#### **3.1. SOBRE O VISUALG**

O VisuAlg tem se mostrado uma importante ferramenta para o ensino de programação. Por meio dele é possível criar, editar, interpretar e executar algoritmos em português estruturado, que é basicamente uma representação de programa em linguagem de algoritmo, ou seja, em alto nível, a ponto que até mesmo pessoas leigas em programação, possam ler e entender o código.

A maior vantagem do VisuAlg é o uso dos comandos todos em português, assim os pré-requisitos para se ensinar lógica de programação são mínimos, sendo necessário apenas algumas habilidades para se manusear o computador. Ele é um programa de livre uso e distribuição, totalmente grátis, desenvolvido com fins didáticos. Na Figura 2 é possível visualizar a tela inicial do VisuAlg.

Figura 2. Tela Inicial do Programa VisuAlg.



Fonte: próprio autor.

Na área “código do algoritmo” é onde o código do programa é inserido, sendo geralmente orientado pelas linhas. Na linha 1 é colocado o nome do programa, podendo ser acrescentado os dados de cabeçalho, como autor, instituição, data, etc, sendo que cada linha deve ser iniciada pela contra barra dupla (//), indicando que se trata apenas de comentário. Na linha 3 é o espaço destinado à declaração de variáveis. Na linha 5 é onde a parte executável do programa é iniciada, ou seja, o código que será executado será colocado nesse espaço, ou seja, entre as marcações “Inicio” e “Fimalgoritmo”. Na área das variáveis é possível observar os tipos e valores assumidos pelas variáveis durante a execução do programa, caso tenha sido declarada alguma variável na parte do código destinada à isso. Na área de visualização dos resultados é possível observar os resultados gerados pela execução do programa, ou seja, as saídas do programa. Quando o código estiver pronto para ser executado ou testado, pode ser utilizado o botão executar indicado na Figura 2, ou a tecla de atalho F9, ou ainda ir na opção “Run(executar)” da barra de menu e em seguida escolher a opção “Rodar o Algoritmo”, conforme indicado na Figura 3.

Figura 3. Opção para executar o algoritmo.



Fonte: próprio autor.

### 3.2. MINICURSO

O minicurso foi elaborado para ser aplicado em apenas dois encontros, onde cada encontro possui uma duração média de 2 (duas) horas. A prática em questão não está voltada unicamente para a programação em si, mas também para introduzir os alunos no ambiente de programação para que os mesmos possam exercitar os seus conhecimentos aprendidos na escola, em especial, claro, os de Física e Matemática.

Com esse objetivo e com o tempo bem limitado, foi então escolhido abordar apenas conceitos introdutórios, que são fundamentais para construção de códigos básicos de qualquer programa, e ao mesmo tempo é suficiente para os objetivos almejados neste trabalho. Assim, o minicurso foi estruturado nos seguintes tópicos:

- Definição: o que são algoritmos;
- Lógica de programação;
- Comando ESCREVA e ESCREVAL – Saída de dados;
- Variáveis e tipos primitivos;
- Comando LEIA – Entrada de dados;
- Operações aritméticas;
- Operadores relacionais;
- Estruturas condicionais;

- Estruturas de repetição.

Foi então elaborado um material impresso no formato de apostila para que os alunos possam acompanhar as explicações, contendo os textos explicativos de cada um dos tópicos citados acima. Essa apostila se encontra no Apêndice A. Há ainda os slides criados para serem utilizados durante o minicurso, que está contido no Apêndice B. Os slides foram utilizados apenas no início para fazer uma breve apresentação, pois a maior parte da explicação do conteúdo se deu meio do uso da tela do programa VisuAlg.

### **3.2.1. Definição – O que são algoritmos**

O termo algoritmo é definido por MANZANO & OLIVEIRA (2014) como um conjunto de passos finitos e organizados de forma lógica que, quando executados, resolvem determinado problema. Este termo está associado a Computação e a Matemática, mas pode-se dizer que até mesmo nas atividades cotidianas algoritmos estão presentes. O conjunto de instruções para se fazer um bolo, por exemplo, seria um exemplo de algoritmo.

Basicamente um algoritmo possui uma rotina e deve ter alguma lógica. Imagine uma tarefa simples, como atravessar a rua. Quando se está diante dessa situação, para executar a tarefa “atravessar a rua”, algumas etapas devem ser realizadas, como: olhar para os lados; verificar se vem veículo; se não vier veículo, atravessar; senão, aguarde e verifique de novo. Esse conjunto de etapas, pode ser chamado de algoritmo, o qual é classificado como não computacional.

Para o exemplo anterior, a sequência de passos pode ser alterada, assim como aumentar ou diminuir a quantidade de etapas, porém, deve-se haver alguma lógica entre os passos. Se o algoritmo fosse construído onde a etapa de olhar os lados estivesse após a ação de atravessar a rua, isso poderia gerar um grande problema. Receitas, instruções de montagem e manuais de uso também podem ser chamados de algoritmos não computacionais. A mesma coisa ocorre com os algoritmos computacionais, pois os mesmos necessitam de lógica para estarem corretos.

### 3.2.2. Lógica de programação: aprendendo com o VisuAlg

#### 3.2.2.1. Comando ESCREVA e ESCREVAL – Saída de dados

Para se escrever um algoritmo em português, é aconselhado utilizar um determinado padrão. Para o VisuAlg, tem-se uma estrutura padrão que deve ser seguida, contendo o escopo detalhado na Tabela 1.

Tabela 1. Estrutura geral de algoritmos no VisuAlg

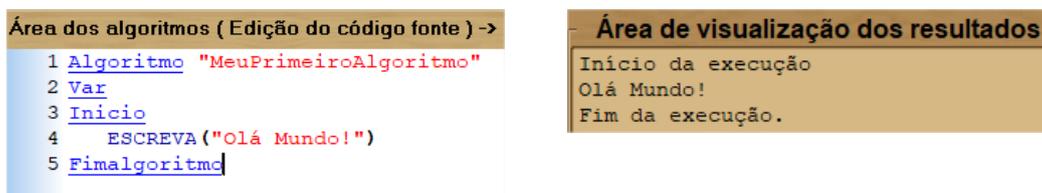
1	ALGORITMO “NomeDoAlgoritmo”
2	VAR
3	INICIO
4	FIMALGORITMO

Fonte: próprio autor.

A primeira linha serve para identificar o algoritmo, sendo que o nome do mesmo fica no espaço entre as aspas duplas. A segunda linha é destinada declaração de variáveis. As variáveis receberão um tópico específico para serem tratadas aqui. A terceira e quarta linha marcam o início e fim, respectivamente, como código que será de fato executado.

Para poder analisar o funcionamento do VisuAlg, pode ser utilizado o seguinte algoritmo da Figura 4, que mostra uma breve saída de dados.

Figura 4. Saída de dados.



Fonte: próprio autor.

Aqui, é utilizado o comando “ESCREVA” para exibir na tela o que for colocado dentro do parêntese, no caso a mensagem “Olá Mundo!”. Pode ser utilizado também o comando “ESCREVAL”, com a diferença de que com este comando é pulado uma linha após a escrita na tela.

### 3.2.2.2. Variáveis e tipos primitivos

As variáveis são espaços na memória do computador que podem receber determinados valores. Elas são utilizadas quando é necessário guardar uma certa informação para o computador poder manipular-la. Para criar uma variável, é preciso declarar ela primeiro, determinando o seu tipo.

Os tipos que serão trabalhados aqui, serão o inteiro, o real, o caractere e o lógico. O tipo inteiro e o real, guardam valores numéricos, tendo os seus nomes associados de forma intuitiva aos conjuntos numéricos aprendidos na Matemática. O tipo caractere serve para guardar valores literários, ou seja, letras e palavras. Por fim, o tipo lógico guarda apenas valor lógico, que pode ser verdadeiro ou falso.

Quando uma variável é declarada, ela pode ser utilizada no corpo do algoritmo, recebendo algum valor diretamente ou resultado de alguma operação. A passagem de um valor para a variável, é chamado de atribuição. Veja o exemplo da Figura 5 do algoritmo “SeuNome” com o uso de uma variável e sua respectiva saída.

Figura 5. Manipulação de variáveis.

The image shows a screenshot of an algorithm editor. On the left, the code is displayed in a window titled "Área dos algoritmos ( Edição do código fonte ) ->". The code consists of seven lines: 1. `Algoritmo "SeuNome"`, 2. `Var`, 3. `nome: caractere`, 4. `Início`, 5. `nome <- "Roberto Costa Nunes"`, 6. `ESCREVA (nome)`, and 7. `Fimalgoritmo`. Red boxes highlight the declaration on line 3 and the assignment on line 5. Red arrows point from these boxes to labels "declaração" and "atribuição" respectively. On the right, a window titled "Área de visualização dos resultados" shows the execution output: "Início da execução", "Roberto Costa Nunes", and "Fim da execução."

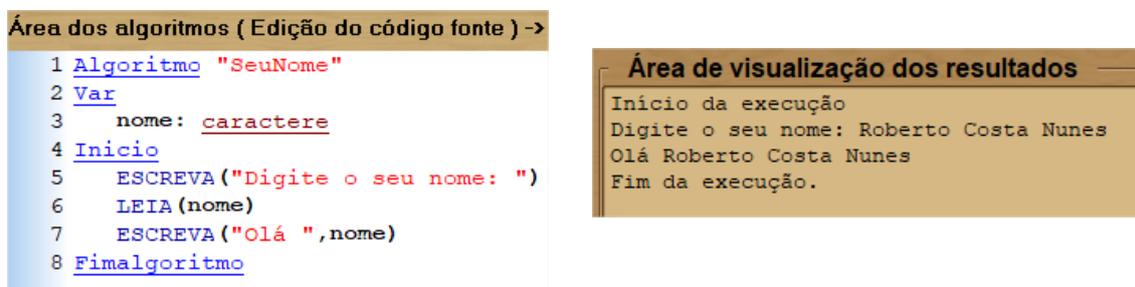
Fonte: próprio autor.

Na 3ª linha, a variável “nome” é criada, tendo apenas a sua declaração com o tipo caractere. Até aí a variável apenas foi criada, mas não possui valor algum. Na linha 5 é feito a atribuição de valor para essa variável, ou seja, a variável “nome”, guarda o valor “Roberto Costa Nunes”, nesse caso. Posteriormente a variável pode ser utilizada para ser acessado ou manipulado o seu valor, o que foi feito na linha 5 com o comando ESCREVA, para mostrar na tela o valor dela na tela de execução.

### 3.2.2.3. Comando LEIA – Entrada de dados

O comando LEIA é utilizado para receber uma entrada de dados. Pode-se dizer que é o contrário do comando ESCREVA, que é para saída. O comando LEIA é muito utilizado para interagir com o usuário, ou seja, por meio dele é possível receber alguma coisa que o usuário digite no teclado, que é a entrada padrão que será utilizada aqui. Observe abaixo na figura 6 como fica o algoritmo “SeuNome” com a utilização do comando LEIA.

Figura 6. Exemplo do comando LEIA



The image shows a screenshot of a programming environment with two main panels. The left panel, titled 'Área dos algoritmos ( Edição do código fonte ) ->', contains the following code:

```
1 Algoritmo "SeuNome"
2 Var
3   nome: caractere
4 Inicio
5   ESCREVA ("Digite o seu nome: ")
6   LEIA (nome)
7   ESCREVA ("Olá ", nome)
8 Fimalgoritmo
```

The right panel, titled 'Área de visualização dos resultados', shows the output of the program:

```
Início da execução
Digite o seu nome: Roberto Costa Nunes
Olá Roberto Costa Nunes
Fim da execução.
```

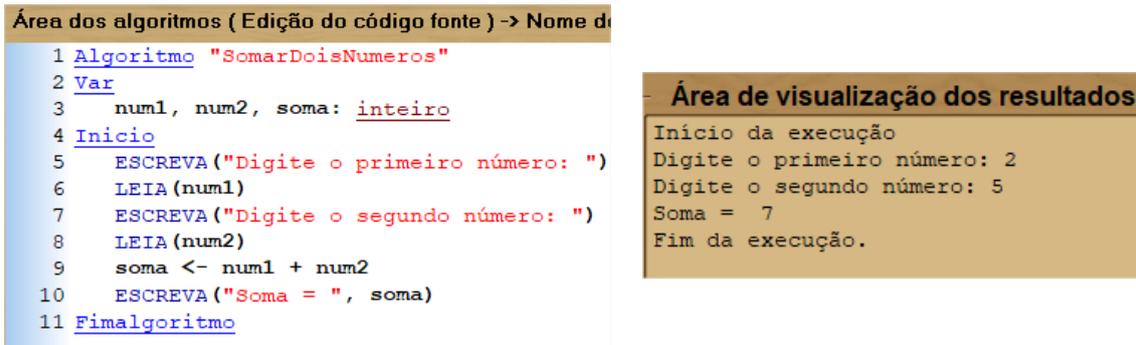
Fonte: próprio autor.

O comando LEIA está na linha 6. Quando o código é executado, o computador espera uma entrada de dados ao chegar no comando LEIA. Após digitar a entrada e apertar a tecla ENTER do teclado, o fluxo de execução continua. Nesse caso, foi digitado o nome “Roberto Costa Nunes”, o qual foi armazenado na variável nome. Na linha 7 foi utilizado o comando ESCREVA para mostrar a saída de dados, porém aqui foi utilizado o recurso de concatenação, para unir a palavra “Olá ” com o que estiver contido na variável nome.

### 3.2.2.4. Operações aritméticas

É possível executar no VisuAlg as operações aritméticas aprendidas na educação básica. As operações e seus respectivos símbolos de operação são: soma (+), subtração (-), multiplicação (\*), divisão (/), divisão inteira (\), exponenciação (^) e resto (%). Para exemplificar as operações aritméticas, veja no exemplo da Figura 7, o algoritmo “SomarDoisNumeros”.

Figura 7. Exemplo de algoritmo que lê dois números e mostra o resultado da soma entre eles.



The image shows a code editor window titled "Área dos algoritmos ( Edição do código fonte ) -> Nome d" and a separate window titled "Área de visualização dos resultados".

```
1 Algoritmo "SomarDoisNumeros"
2 Var
3   num1, num2, soma: inteiro
4 Inicio
5   ESCREVA("Digite o primeiro número: ")
6   LEIA(num1)
7   ESCREVA("Digite o segundo número: ")
8   LEIA(num2)
9   soma <- num1 + num2
10  ESCREVA("Soma = ", soma)
11 Fimalgoritmo
```

The results window shows the following output:

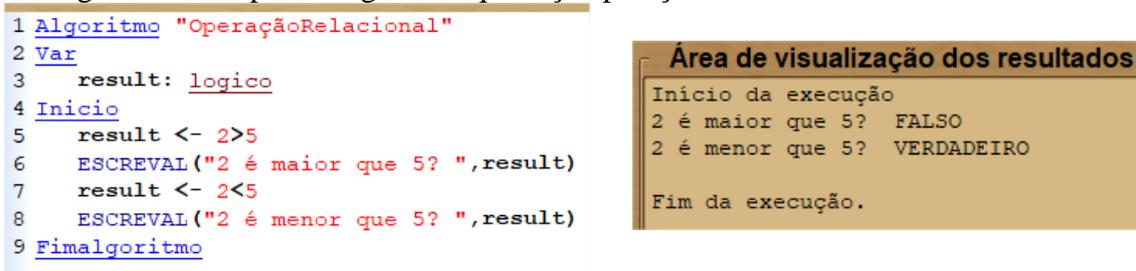
```
Início da execução
Digite o primeiro número: 2
Digite o segundo número: 5
Soma = 7
Fim da execução.
```

Fonte: próprio autor.

### 3.2.2.5. Operadores relacionais

Os operadores relacionais retornam um valor lógico (verdadeiro ou falso). Os operadores relacionais com seus respectivos símbolos de operação são: maior que (>), menor que (<), maior igual (>=), menor igual (<=), igual (=) e diferente (<>). Veja o algoritmo “OperaçãoRelacional” com a exemplificação da operação de maior que e menor que na imagem da Figura 8.

Figura 8. Exemplo de algoritmo que faça operação relacional entre dois números.



The image shows a code editor window titled "Área dos algoritmos ( Edição do código fonte ) -> Nome d" and a separate window titled "Área de visualização dos resultados".

```
1 Algoritmo "OperaçãoRelacional"
2 Var
3   result: logico
4 Inicio
5   result <- 2>5
6   ESCREVAL("2 é maior que 5? ", result)
7   result <- 2<5
8   ESCREVAL("2 é menor que 5? ", result)
9 Fimalgoritmo
```

The results window shows the following output:

```
Início da execução
2 é maior que 5? FALSO
2 é menor que 5? VERDADEIRO
Fim da execução.
```

Fonte: próprio autor.

### 3.2.2.6. Estruturas condicionais

As estruturas condicionais são utilizadas para tomadas de decisões de acordo com uma condição verificada. Assim, pode-se garantir que determinada ação só será executada se certa condição for satisfeita. Para demonstrar essa estrutura, veja o exemplo do algoritmo “EstruturaCondicional” na Figura 9.

Figura 9. Exemplo de algoritmo com estrutura condicional.

```
1 Algoritmo "EstruturaCondicional"
2 Var
3   num1, num2: inteiro
4 Inicio
5   num1 <- 8
6   num2 <- 3
7   SE (num1 > num2) ENTAO
8     ESCREVA (num1, " é maior")
9   SENAO
10    ESCREVA (num2, " é maior")
11   FIMSE
12 Fimalgoritmo
```

**Área de visualização dos resultados**

```
Início da execução
8 é maior
Fim da execução.
```

Fonte: próprio autor.

### 3.2.2.7. Estruturas de repetição

Estruturas de repetição são utilizadas quando se deseja realizar uma determinada operação várias vezes. Pode-se utilizar repetições não controladas e repetições controladas. Na repetição não controlada, uma operação é repetida até que determinada condição seja satisfeita. Nesse caso, é necessário ter cuidado para que a condição seja satisfeita em algum momento, pois corre o risco ficar em execução sem nunca parar. Duas estruturas de repetição do tipo não controlada são possíveis no VisuAlg, o ENQUANTO e o REPITA. Veja o algoritmo “ContarAteCinco” com a estrutura ENQUANTO na Figura 10.

Figura 10. Exemplo de algoritmo que conte até cinco com a estrutura ENQUANTO.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   num, cont: inteiro
4 Inicio
5   num <- 5
6   cont <- 0
7   ENQUANTO (cont <= num) FACA
8     ESCREVAL (cont)
9     cont <- cont + 1
10  FIMENQUANTO
11 Fimalgoritmo
```

**Área de visualização dos resultados**

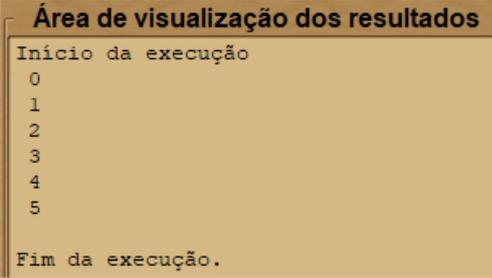
```
Início da execução
0
1
2
3
4
5
Fim da execução.
```

Fonte: próprio autor.

Analise agora a Figura 11 com o algoritmo que faz a mesma coisa, porém com a estrutura REPITA.

Figura 11. Exemplo de algoritmo que conte até cinco com a estrutura REPITA.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   num, cont: inteiro
4 Inicio
5   num <- 5
6   cont <- 0
7   REPITA
8     ESCREVAL(cont)
9     cont <- cont + 1
10  ATE (cont>num)
11 Fimalgoritmo
```

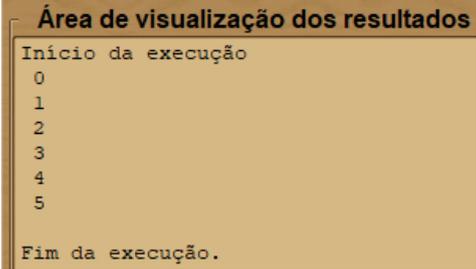


Fonte: próprio autor.

A estrutura de repetição controlada é utilizada quando se quer fazer uma operação em quantidade de passo já conhecida. Essa estrutura é comumente chamada de PARA. Com ela, a operação somente será executada a quantidade de passos determinada. Veja o exemplo do algoritmo “ContarAteCinco” utilizando a estrutura de repetição PARA, na imagem da Figura 12.

Figura 12. Exemplo de algoritmo que conte até cinco com a estrutura PARA DE ATE.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   cont: inteiro
4 Inicio
5   PARA cont DE 0 ATE 5 PASSO 1 FACA
6     ESCREVAL(cont)
7   FIMPARA
8 Fimalgoritmo
```



Fonte: próprio autor.

### 3.2.2.8. Exercícios

Foram feitos quatro exercícios para que os alunos respondessem como atividades durante o curso, com o objetivo de fixar melhor o conteúdo. O primeiro exercício trazia o seguinte comando: faça um algoritmo que receba dois valores inteiros e mostre a soma entre eles.

Os alunos conseguiram construir esse algoritmo sem muita dificuldade. O código que um aluno construiu está na imagem da Figura 13.

Figura 13. Exercício 1: somar dois números e mostrar resultado na tela.

```
1 Algoritmo "exercicio1"
2
3 Var
4   numero1, numero2: inteiro
5 Inicio
6   ESCREVA("Número 1: ")
7   LEIA(numero1)
8   ESCREVA("Número 2: ")
9   LEIA(numero2)
10  ESCREVA("Resultado: ", numero1 + numero2)
11 Fimalgoritmo
```

Fonte: próprio autor.

O segundo exercício estava com o seguinte enunciado: faça um algoritmo que receba 5 números inteiros, mostre como resultado a soma deles e a média aritmética. Com um pequeno incremento de dificuldade, esse exemplo requer um pouco mais de habilidade, todavia é plenamente adequado ao nível dos alunos. Os alunos também fizeram esse exercício, porém alguns tiveram um pouco de dificuldade em compreender o enunciado. A Figura 14 contém exemplo de algoritmo feito por um dos alunos.

Figura 14. Exercício 2: exemplo de algoritmo do segundo exercício.

```
1 Algoritmo "exercicio2"
2
3 Var
4   n1, n2, n3, n4, n5: inteiro
5 Inicio
6   ESCREVA("Número 1: ")
7   LEIA(n1)
8   ESCREVA("Número 2: ")
9   LEIA(n2)
10  ESCREVA("Número 3: ")
11  LEIA(n3)
12  ESCREVA("Número 4: ")
13  LEIA(n4)
14  ESCREVA("Número 5: ")
15  LEIA(n5)
16  ESCREVAL("Soma: ", n1 + n2 + n3 + n4 + n5)
17  ESCREVA("Média: ", (n1 + n2 + n3 + n4 + n5)/5)
18
19 Fimalgoritmo
```

Fonte: próprio autor.

No segundo exercício era esperado que utilizassem alguma das estruturas de repetição, todavia, eles utilizaram estruturas básicas, o qual também resolve o problema, só que utilizando mais linhas de código.

Para o terceiro exercício, foi colocado no comando para considerar a equação do primeiro grau na forma  $ax + b = c$ , e assim os alunos deveriam fazer um algoritmo que receba os valores de  $a$  e de  $b$  e mostre o resultado na tela para o  $x = 0$ ,  $x = 1$ ,  $x =$

2,  $x = 3$ ,  $x = 4$  e  $x = 5$ . Para esse exemplo, os alunos tiveram que interagir entre si para saber como é que se trabalha com equações do primeiro grau. Diferentemente dos exemplos anteriores, onde cada aluno foi capaz de fazer o seu próprio código, nesse eles fizeram pequenos grupos em duplas ou trios para construir o código de forma coletiva, mas cada um teve escrever o código em seu computador. A Figura 15 traz um exemplo de algoritmo que resolve o exercício de número 3.

Figura 15. Exercício 3: exemplo de algoritmo com equação do primeiro grau.

```
1 Algoritmo "exercicio3"  
2 Var  
3   a, b: inteiro  
4 Inicio  
5   ESCREVA ("a: ")  
6   LEIA (a)  
7   ESCREVA ("b: ")  
8   LEIA (b)  
9   ESCREVAL ("x (1)= ", a*1+b)  
10  ESCREVAL ("x (2)= ", a*2+b)  
11  ESCREVAL ("x (3)= ", a*3+b)  
12  ESCREVAL ("x (4)= ", a*4+b)  
13  ESCREVAL ("x (5)= ", a*5+b)  
14 Fimalgoritmo
```

Fonte: próprio autor.

No quarto exercício, foi solicitado aos alunos que fizessem um algoritmo que receba a distância e o tempo de determinado veículo e calcule a velocidade média. Como os alunos já haviam exercitado um pouco nos exemplos anteriores, não houve muita dificuldade em resolver essa questão. A maior parte da demora foi em lembrar ou aprender como se calcula a velocidade média, pois era esperado que eles soubessem efetuar esses cálculos sem dificuldade. Exemplo de um dos algoritmos feito pelos alunos pode ser visto na Figura 16.

Figura 16. Exercício 4: exemplo de cálculo da velocidade média.

```
1 Algoritmo "exercicio4"  
2  
3 Var  
4   distancia, t: inteiro  
5 Inicio  
6   ESCREVA ("Distancia: ")  
7   LEIA (distancia)  
8   ESCREVA ("Tempo: ")  
9   LEIA (t)  
10  ESCREVAL ("Velocidade Média: ", distancia/t)  
11 Fimalgoritmo
```

Fonte: próprio autor.

### 3.3. TRABALHANDO UM POUCO MAIS

O tópico anterior serve como guia para introduzir aos alunos os princípios básicos de programação, sendo plenamente possível ampliar o leque de possibilidades de assuntos que podem ser trabalhados. Esse tipo de codificação possibilita exercitar bastante o funcionamento de fórmulas baseadas em funções e equações, permitindo estressar o problema com uma grande carga de teste, verificando diversas possibilidades. Quando o aluno constrói o seu próprio código, existe maiores chances dele não apenas decorar a fórmula, mas aprender o seu funcionamento, manipulação e reter assim o conhecimento.

#### 3.3.1. Um pouco mais de cinemática

Tradicionalmente os conteúdos que abordam cinemática geralmente são trabalhados na educação básica nos anos finais do ensino fundamental II na disciplina de ciências, e ainda no primeiro ano do ensino médio na disciplina de Física. Porém estes conceitos são requeridos em vários momentos da vida do aluno, tanto em assuntos didáticos, como vestibulares e outros exames de seleção, como em situações práticas cotidianas.

Como visto na Figura 16, no algoritmo para calcular a velocidade média é utilizado duas variáveis (distância e tempo), que basicamente representa a equação

1. (1)

$$velocidade\ média = \frac{distância}{tempo}$$

Nos casos de manipulação da fórmula, é possível encontrar a distância com base na velocidade média e com o tempo. Para isso, é utilizado a equação 2.

$$distância = velocidade\ média \times tempo \quad (2)$$

A adaptação no algoritmo pode ser realizada com alguns ajustes no exemplo mostrado na Figura 16. Para demonstrar esse ajuste, veja a Figura

Figura 17. Exemplo de como calcular a distância com base na velocidade média e o tempo.

```
1 Algoritmo "calcularDistância"
2
3 Var
4   vm, t: inteiro
5 Inicio
6   ESCREVA("Velocidade média: ")
7   LEIA(vm)
8   ESCREVA("Tempo: ")
9   LEIA(t)
10  ESCREVAL("A distância percorrida é de: ", vm*t)
11 Fimalgoritmo
```

Fonte: próprio autor.

Manipulando a fórmula da velocidade média também é possível encontrar o tempo, agora com base na velocidade média e na distância, tendo assim a fórmula apresentada pela equação 3.

$$tempo = \frac{distância}{velocidade\ média} \quad (3)$$

O algoritmo que aborda essa equação pode ser obtido com apenas alguns ajustes do anterior. Na Figura 18 é possível observar um exemplo de solução para esse caso com apenas algumas adaptações dos exemplos anteriores.

Figura 18. Exemplo de como encontrar o tempo com base na velocidade média e a distância

```
1 Algoritmo "calcularTempo"
2
3 Var
4   vm, distancia: inteiro
5 Inicio
6   ESCREVA("Velocidade média: ")
7   LEIA(vm)
8   ESCREVA("Distância: ")
9   LEIA(distancia)
10  ESCREVAL("O tempo é de: ", distancia/vm)
11 Fimalgoritmo
```

Fonte: próprio autor.

Esse conteúdo de velocidade média, geralmente é a parte introdutória dos conceitos de cinemática. As fórmulas e algoritmos podem ser facilmente adaptados com apenas mudanças de nomenclaturas para se trabalhar o Movimento Uniforme (MU), já que nesse tipo de movimento a velocidade instantânea é iguala a velocidade média. Ainda dentro dos assuntos introdutórios de cinemática, pode ser trabalhado as conversões de unidade de medida.

### 3.3.2. Trabalhando com aceleração

Ao trabalhar o Movimento Uniformemente Variado (MUV) é apresentado o conceito de aceleração, que pode ser compreendida como a taxa de variação da velocidade em relação ao tempo. No MUV, a aceleração instantânea é igual a aceleração média, podendo ser representada pela equação 4.

$$a = \frac{\Delta v}{\Delta t} \quad (4)$$

Na equação 4,  $a$  é a aceleração,  $\Delta v$  é a variação da velocidade e  $\Delta t$  é a variação do tempo.

No caso de MUV o cálculo da velocidade média e da velocidade instantânea são diferentes do MU. Para calcular a velocidade instantânea, é utilizado a equação 5.

$$v = v_0 + a * t$$

Na equação 5,  $v$  é a velocidade final (velocidade no instante  $t$ ),  $v_0$  é a velocidade inicial (velocidade no instante  $t = 0$ ),  $a$  é a aceleração e  $t$  é o tempo.

Seguindo a evolução gradativa dos estudos de programação, assim como de Física, é possível escrever o código com apenas algumas adaptações dos exemplos anteriores, ou partindo de outra abordagem. Agora, será necessário trabalhar com mais variáveis. Um Exemplo de como o código pode ser escrito, está detalhado logo abaixo na Figura 19.

Figura 19. Exemplo de função horário do MUV.

```
1 Algoritmo "MRU-Velocidade"
2
3 Var
4   v1, a, t: inteiro
5 Inicio
6   ESCREVA("Velocidade inicial: ")
7   LEIA(v1)
8   ESCREVA("Aceleração: ")
9   LEIA(a)
10  ESCREVA("Tempo: ")
11  LEIA(t)
12  ESCREVAL("A velocidade final é de: ", v1+a*t)
13 Fimalgoritmo
```

Fonte: próprio autor.

Outra fórmula muito útil e bastante requerida nos estudos de MRU, é a função do espaço (ou posição). Geralmente ela é representada pela equação 6 mostrada abaixo.

$$S = S_0 + v_0 * t + \frac{at^2}{2} \quad (6)$$

Na equação 6,  $S$  é a posição final (ou no instante  $t$ ),  $S_0$  é a posição inicial (ou no instante  $t = 0$ ),  $v_0$  é a velocidade inicial (ou no instante  $t = 0$ ),  $a$  é a aceleração e  $t$  é o tempo.

Escrever um algoritmo que trabalhe com essa fórmula é bastante interessante, principalmente pela quantidade de variáveis necessárias. Como sempre, a construção do código é livre, e cada pessoa pode desenvolver linhas de códigos com diferentes abordagens. Seguindo o processo gradativo dos exemplos anteriores, a Figura 20 mostra um exemplo de como essa equação pode ser escrita em algoritmo.

Figura 20. Exemplo de código para a função horária do espaço no MUV.

```
1 Algoritmo "MRU-Espaço"
2
3 Var
4   s1, v1, a, t: inteiro
5 Inicio
6   ESCREVA("Posição inicial: ")
7   LEIA(s1)
8   ESCREVA("Velocidade inicial: ")
9   LEIA(v1)
10  ESCREVA("Aceleração: ")
11  LEIA(a)
12  ESCREVA("Tempo: ")
13  LEIA(t)
14  ESCREVAL("A posição final é de: ", s1+v1*t+(a*t*t)/2)
15 Fimalgoritmo
```

Fonte: próprio autor.

Os exemplos elencados acima demonstram apenas algumas possibilidades de como o assunto pode ser tratado na abordagem de cinemática. Todavia não se resume exclusivamente a isso. A utilização de construção de códigos de algoritmos ou de programas em linguagens específicas de programação é plenamente possível em vários assuntos da Física, como por exemplo, a dinâmica, a estática, energia mecânica e escalas de temperatura.

Uma observação a ser feita nos exemplos apresentados até aqui, é com relação a complexidade dos algoritmos construídos. De acordo com a habilidade de cada aluno,

é possível ser construído códigos mais elaborados, com uso de recursos de laços de repetição ou estruturas condicionais para tomadas de decisão. Optou-se, obviamente, por não utilizar nos exemplos tais recursos, uma vez que poderia tornar o entendimento restrito apenas àqueles que tivessem tal domínio. Os códigos utilizam apenas as estruturas mais básicas possíveis, uma vez que a programação em si é a ferramenta de apoio para trabalhar os conteúdos de Física, e não a finalidade última do trabalho.

### 3.3.3. Trabalhando com escalas termométricas

Ao trabalhar as escalas termométricas dentro do currículo de Física, é mais comum ser feita a relação entre as escalas Celsius, Fahrenheit e Kelvin. As escalas Celsius e Fahrenheit são graduadas, enquanto a Kelvin é absoluta, não tendo assim valores negativos. A relação entre as três escalas pode ser representada por uma única expressão:

$$\frac{T_c}{5} = \frac{T_f - 32}{9} = \frac{T_k - 273}{5} \quad (7)$$

Na equação 7,  $T_c$  é a temperatura em graus Celsius ( $^{\circ}\text{C}$ ),  $T_f$  é a temperatura em graus Fahrenheit e  $T_k$  a temperatura em Kelvin. Assim, a relação entre duas escalas pode ser obtida por meios dessa expressão. Por exemplo, se for necessário fazer uma equivalência de temperatura entre as escalas Celsius e Fahrenheit, têm-se a equação 8.

$$\frac{T_c}{5} = \frac{T_f - 32}{9} \quad (8)$$

Da equação 8, é possível então converter uma temperatura da escala Fahrenheit para a escala Celsius, como mostrado na equação 9.

$$T_c = 5 * \left( \frac{T_f - 32}{9} \right) \quad (9)$$

De forma semelhante é possível converter uma temperatura da escala Celsius para Fahrenheit, de acordo com a equação 10.

$$T_f = \left( \frac{T_c * 9}{5} \right) + 32 \quad (10)$$

Para construir um algoritmo que faça a conversão mostrada na equação 10 é necessário ter atenção na hora de escrever a expressão em linha de código por causa da preferência entre as operações aritméticas. Um exemplo de código que implemente essa equação pode ser observado na Figura 21.

Figura 21. Exemplo de algoritmo para converter de Celsius para Fahrenheit.

```
1 Algoritmo "escalaTermométrica"  
2 // Relação termométrica:  $T_c/5 = (T_f - 32)/9$   
3 Var  
4 Tc, Tf : Real  
5 Inicio  
6   Escreva ("Digite o valor da temperatura em Celsius: ")  
7   Leia (Tc)  
8   Tf <- ((Tc/5)*9)+32  
9   Escreva ("A temperatura em Fahrenheit é: ", Tf)  
10 Fimalgoritmo
```

Fonte: próprio autor.

De modo geral, a escrita de algoritmos pode ser exercitada de diversas maneiras, e o uso de portugol, com essas estruturas mais definidas são exatamente a entrada para a construção de linhas de códigos em uma linguagem de programação mais específica, só que aqui com uma complexidade bem menor e mais inteligível para estudantes da educação básica.

### 3.4. LEVANTAMENTO E ANÁLISE DE DADOS

Para o levantamento de dados foram utilizados dois questionários, um aplicado anteriormente ao minicurso (Apêndice C) e outro ao término (Apêndice D). O primeiro foi para balizar o nível dos alunos e assim ter uma percepção mais acurada do nível dos alunos, sabendo ainda qual a realidade de cada um no ambiente escolar. Já o segundo questionário serviu ainda para se ter o nível de aceitação do curso e o quanto o mesmo pode contribuir para a vida escolar ou se desperta algum interesse do aluno.

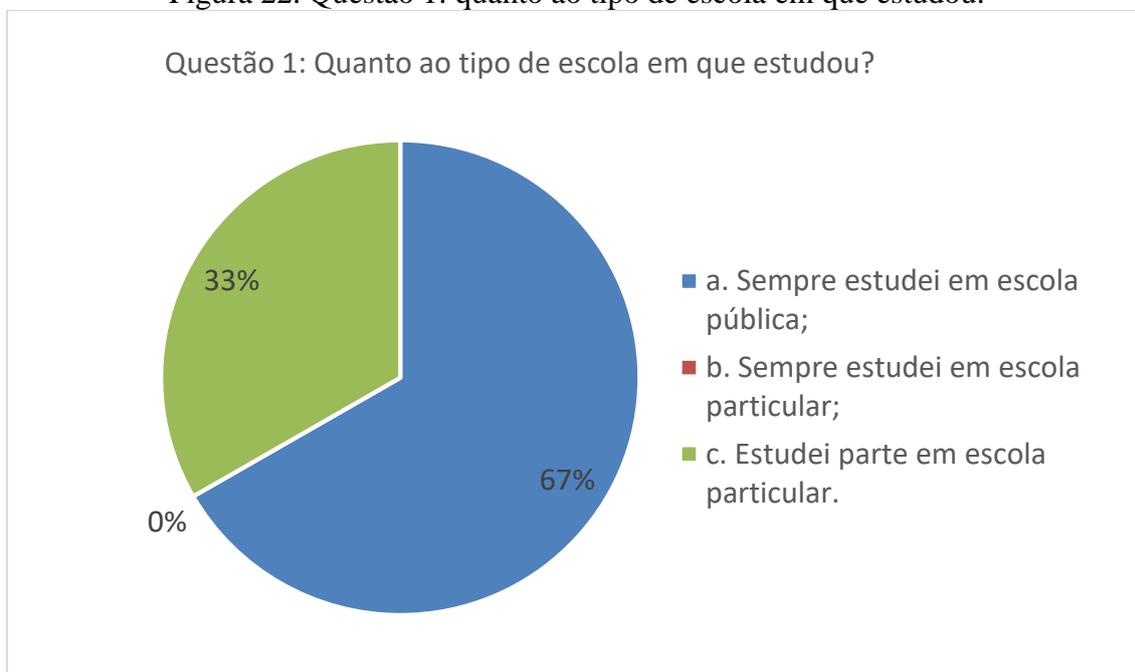
### 3.4.1. Questionário Inicial

O primeiro questionário foi aplicado antes do minicurso com o objetivo de levantar dados referente ao perfil dos alunos, tentando ainda mensurar o quanto o aluno sabe do conteúdo de Física. Foram elaboradas então 9 (nove) questões, em sua maioria questões objetivas e de múltipla escolha. O questionário em questão está contido no Apêndice C desse trabalho.

#### 3.4.1.1. Quanto ao tipo de escola que os alunos estudaram

A primeira pergunta faz referência ao tipo de escola que o aluno estudou. Dos 12 alunos, 8 (oito), cerca de 67%, responderam que sempre estudaram em escola pública, 4 (quatro), cerca de 33%, que estudaram parte em escola particular. As informações referentes a essa questão estão contidas no gráfico da Figura 22, logo abaixo.

Figura 22. Questão 1: quanto ao tipo de escola em que estudou.



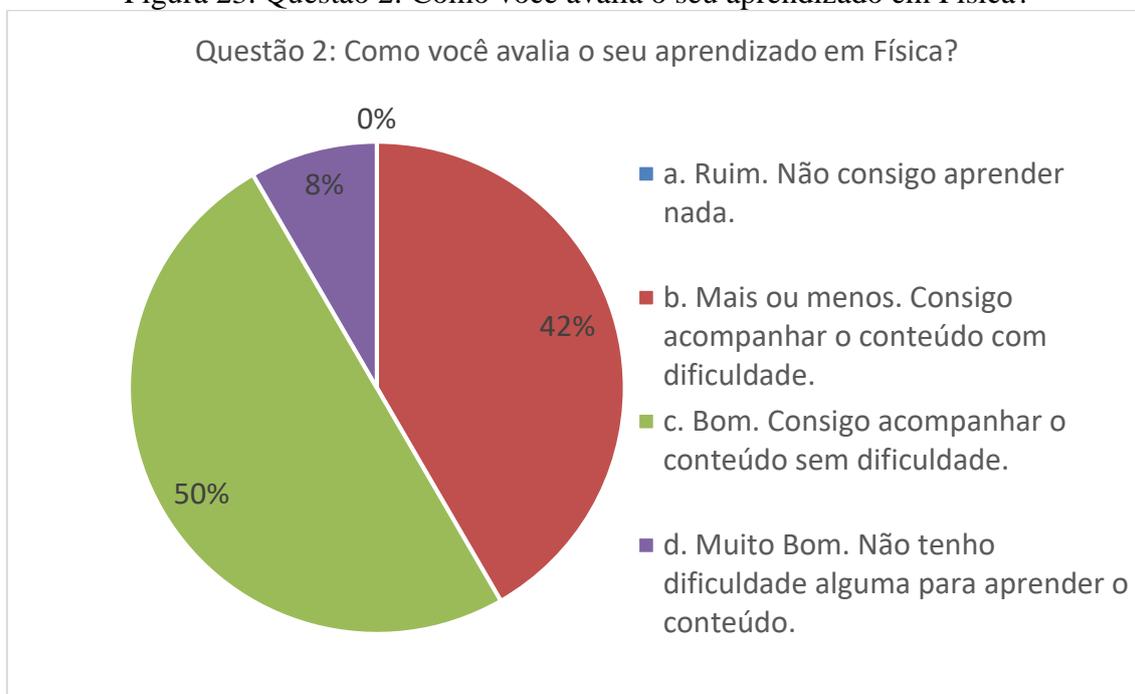
Fonte: próprio autor.

É possível observar que a maioria dos alunos sempre contaram com a rede pública de ensino desde os anos iniciais, e uma pequena parcela dos estudantes puderam estudar em escolas particulares, mesmo sendo apenas nos anos iniciais.

### 3.4.1.2. Autoavaliação de conhecimentos em Física

Já a segunda questão buscava que os alunos fizessem uma autoavaliação de como está o seu aprendizado na matéria de Física. Pode parecer uma questão subjetiva, mas a ideia aqui é que o aluno diga se ele tem facilidade ou não para aprender os conteúdos dessa matéria. Assim, dentre os 12 alunos que responderam ao questionário, 1 (8%) disse que seu aprendizado na matéria é muito bom, não tendo dificuldade alguma para aprender os conteúdos de Física; 6 deles (50%) disseram ter um aprendizado bom, conseguindo acompanhar o conteúdo; 5 dos alunos (42%) disseram que seu aprendizado é mais ou menos, tendo um pouco de dificuldade para acompanhar o conteúdo e nenhum aluno disse ter um aprendizado ruim. Na Figura 23 estão os dados dessa questão dispostos em gráfico.

Figura 23. Questão 2: Como você avalia o seu aprendizado em Física?



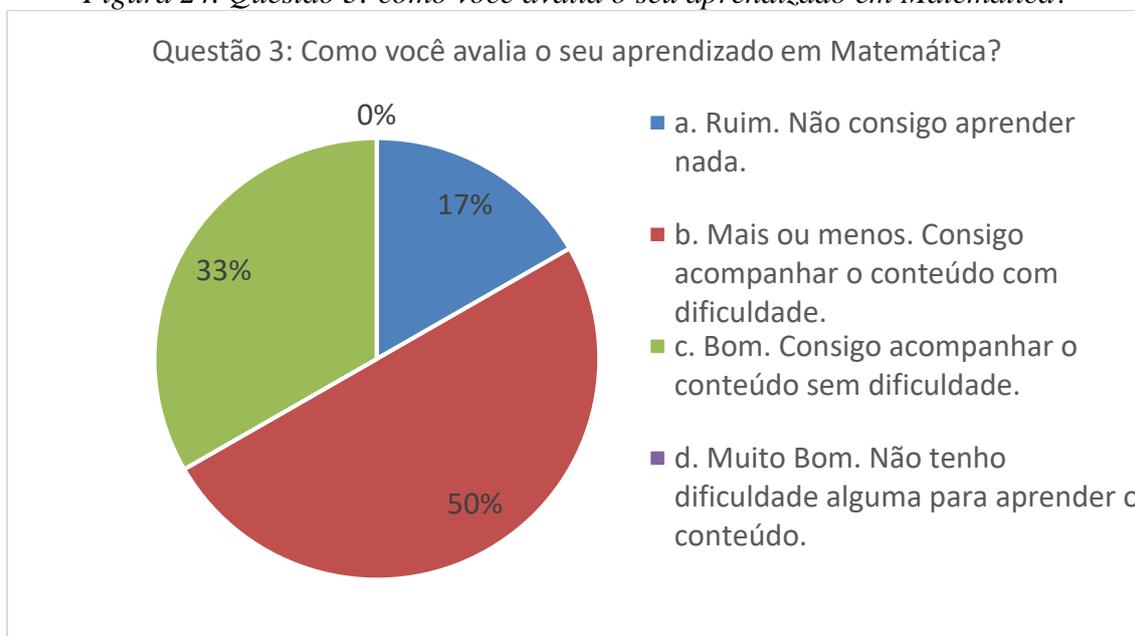
Fonte: próprio autor.

De certa forma, esses dados são estimulantes, pois os alunos, aparentemente, conseguem aprender os conteúdos de Física. Talvez isso implique em uma boa aceitação da matéria dentro desse grupo de alunos, porém isso não significa necessariamente que eles sejam alunos que dominam o conteúdo de forma plena, mas que eles conseguem ao menos acompanhar o conteúdo, nem que seja com algumas dificuldades.

### 3.4.1.3. Autoavaliação de conhecimento em Matemática

De forma similar a segunda questão, na terceira busca saber como está aprendido em Matemática por meio da autoavaliação. Nenhum aluno respondeu que teria aprendido muito bom; 4 (33%) disseram que possuem um aprendizado bom, conseguindo acompanhar o conteúdo sem dificuldade; 6 (50%) revelaram que o seu aprendizado é mediano, acompanhando o conteúdo com alguma dificuldade e 2 (17%) responderam que possuem uma aprendizagem considerada ruim, conforme gráfico da Figura 24.

Figura 24. Questão 3: como você avalia o seu aprendizado em Matemática?



Fonte: próprio autor.

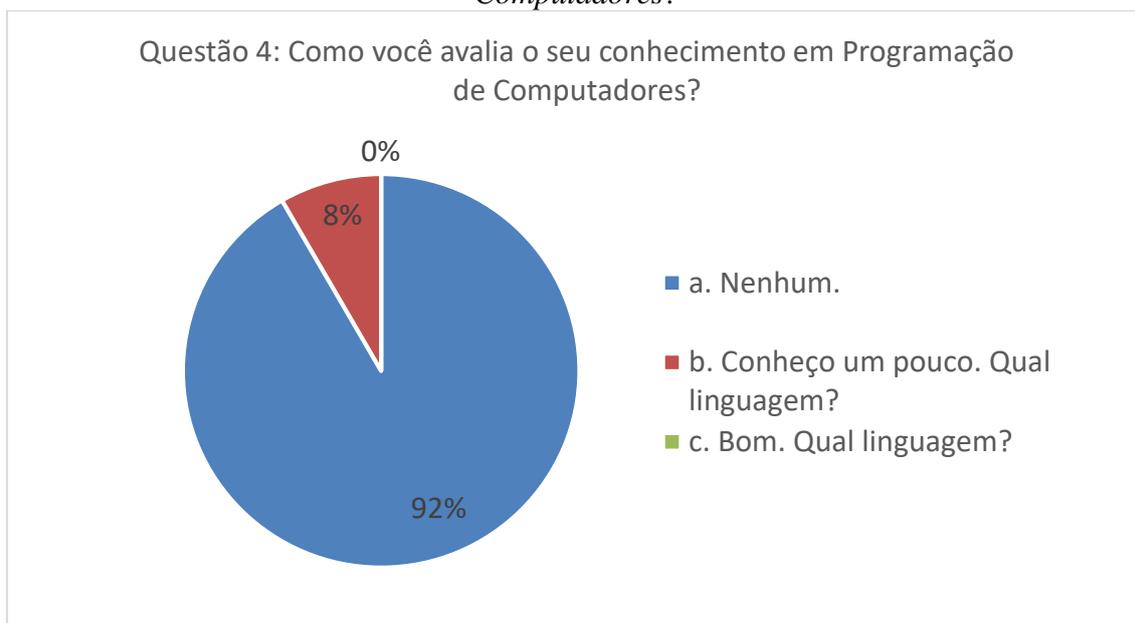
Apesar de Matemática e Física terem uma forte relação entre si, o grupo de alunos se mostraram ter mais dificuldade em Matemática do que em Física. Apesar dos

alunos terem respondido que não possuem muita dificuldade em Física, eles revelaram ter maiores dificuldades em Matemática.

#### 3.4.1.4. Autoavaliação em programação de computadores

A quarta questão foi destinada para saber se os alunos possuem algum conhecimento prévio acerca de programação de computadores. Dos 12 alunos presentes que responderam o questionário, apenas 1 (8%) disse ter algum conhecimento prévio acerca de programação, tendo apenas algumas noções de lógica de programação e formulação de algoritmos. Os outros 11 alunos (92%) disseram não ter nenhum conhecimento em programação de computadores, como pode ser visto no gráfico da Figura 25.

Figura 25. Questão 4: Como você avalia o seu conhecimento em Programação de Computadores?



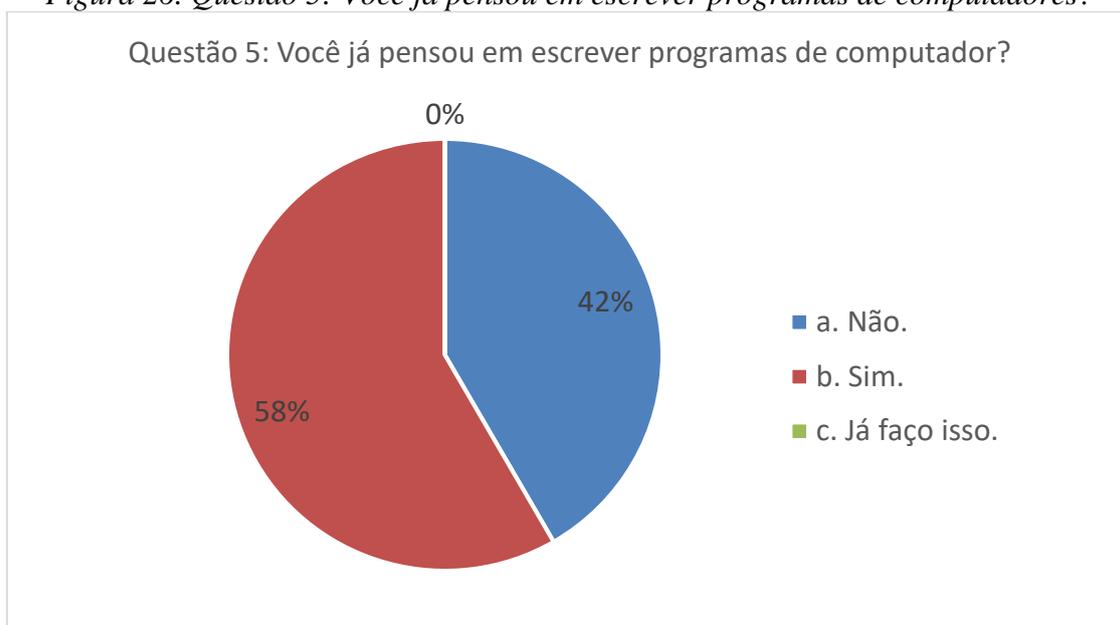
Fonte: próprio autor.

As informações obtidas nessa questão são de grande importância por revelarem que quase todos os alunos desse grupo não possuem conhecimento algum em programação de computadores. Se por um lado isso pode ser considerado um obstáculo, por outro, possibilita analisar melhor os resultados obtidos já que os conhecimentos prévios irão interferir pouco.

### 3.4.1.5. Já pensou em escrever programas de computadores?

Na quinta questão buscou-se verificar se os alunos se imaginavam escrevendo programas de computadores. Apesar de terem pouco conhecimento relacionado a programação de computadores, ao serem questionados se já haviam pensado em escrever programas de computadores, 7 (58%) responderam que sim, e os outros 5 (42%) disseram que não. O gráfico referente à quinta questão está contido na Figura 26.

Figura 26. Questão 5: Você já pensou em escrever programas de computadores?



Fonte: próprio autor.

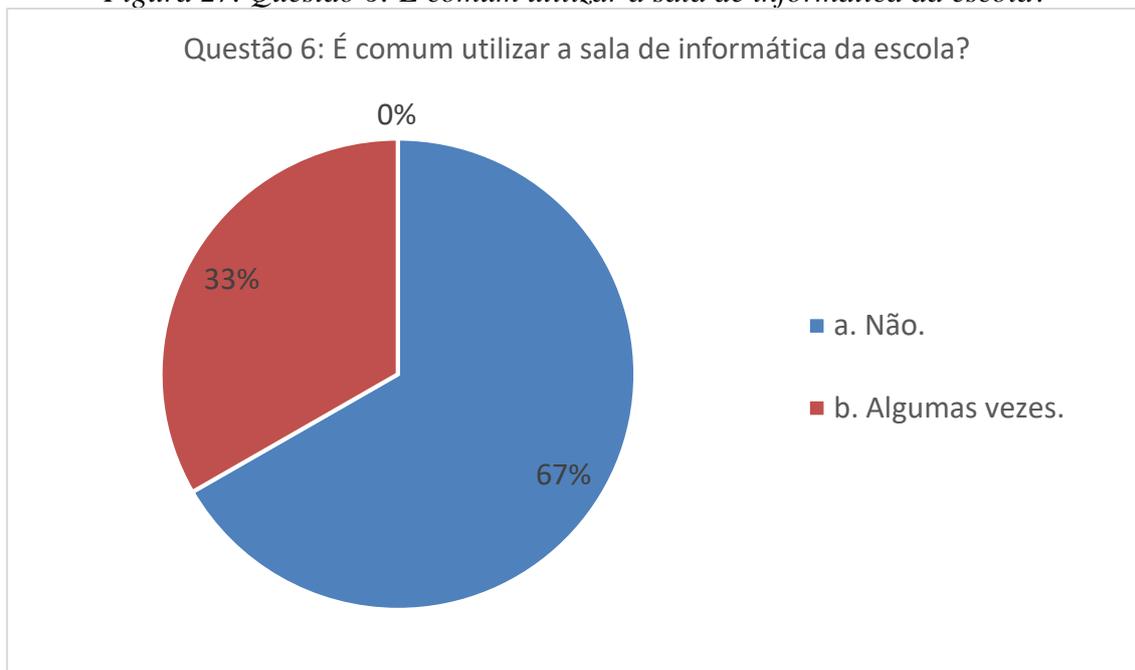
Apesar de uma parte do grupo de alunos terem respondido que nunca pensaram em escrever programas de computadores, isso pode ser devido ao fato de nunca terem tido contato com tipo conteúdo, pois considerando a idade dos alunos e a série escolar, dificilmente eles teriam essa matéria na escola. Mesmo assim, a maior parte dos alunos disseram já ter pensado em escrever programas de computadores, mesmo sem ter tido contato com a matéria.

### 3.4.1.6. Uso do laboratório de informática da escola

Como foi utilizado o laboratório de informática, é interessante se verificar o quanto o mesmo é utilizado, já que espaço que fica a disposição de todos da escola.

Assim, na sexta questão foi perguntado se é comum os mesmos utilizarem o laboratório de informática da escola. Dos alunos que responderam o questionário, 8 deles (67%) disseram não é comum utilizar esse espaço, enquanto os outros 4 restantes (33%) responderam que sim, que é comum utilizar o laboratório de informática. O gráfico com as informações colhidas na questão 6, está contido na imagem da Figura 27.

Figura 27. Questão 6: É comum utilizar a sala de informática da escola?



Fonte: próprio autor.

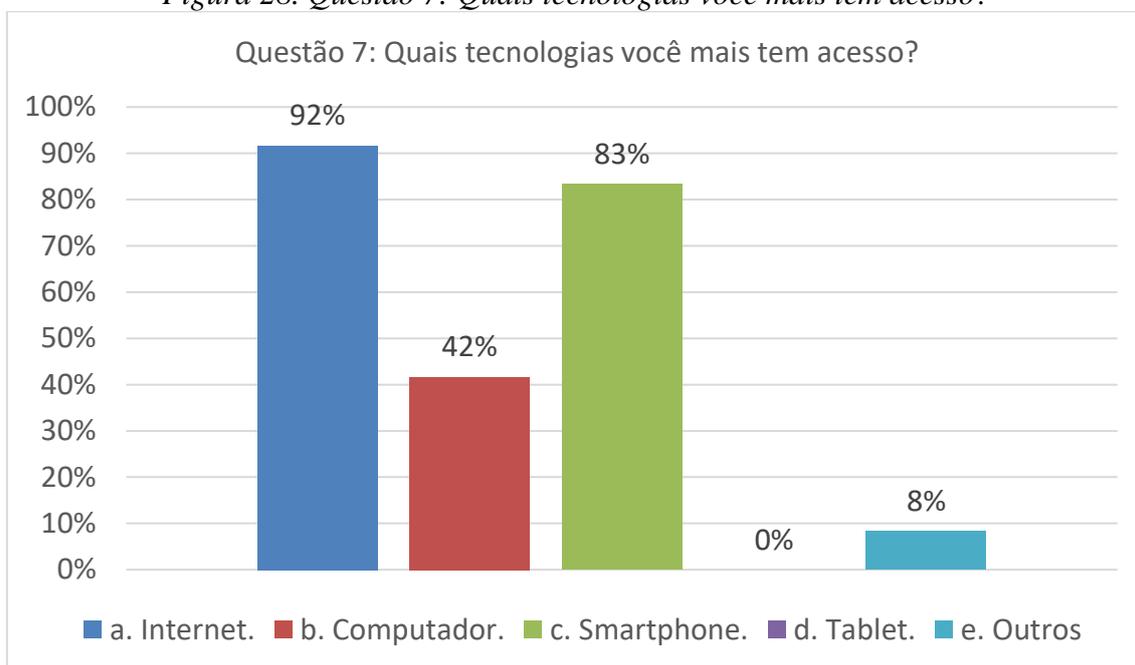
#### 3.4.1.7. Tecnologias que os alunos mais têm acesso

Na sétima pergunta desse questionário, foi procurado saber quais são as tecnologias que os alunos mais têm acesso. Dentro desse grupo de alunos, 11 deles (92%) possuem acesso à internet, 10 (83%) têm contato a smartphone, 5 (42%) possuem acesso a computador, 1 (8%) respondeu ter acesso a outro tipo de tecnologia e nenhum deles a tablet. O gráfico com os dados coletados nessa questão está contido na imagem da Figura 28.

Vale ressaltar no grande acesso à internet que os alunos possuem. Na verdade, era esperado que todos os alunos tivessem acesso, porém um deles respondeu que não tem acesso. Todavia, todos os demais alunos têm acesso a internet com bastante regularidade, seja para interação social ou para informação. Outro ponto de destaque é

no quanto os alunos têm acesso ao smartphone, estando na frente até mesmo do computador.

Figura 28. Questão 7: Quais tecnologias você mais tem acesso?

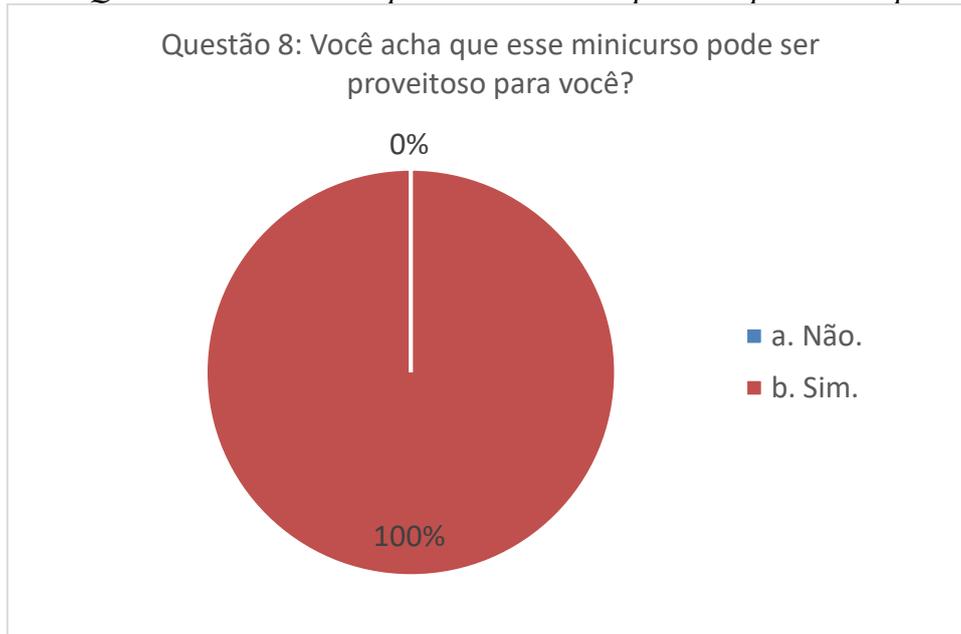


Fonte: próprio autor.

#### 3.4.1.8. O minicurso pode ser proveitoso para você?

Na questão de número 8, os alunos foram perguntados se o minicurso poderia ser útil para eles. Todos os alunos desse grupo responderam que sim. O gráfico referente a questão 8 pode ser visto na Figura 29.

Figura 29. Questão 8: Você acha que esse minicurso pode ser proveitoso para você?



Fonte: próprio autor.

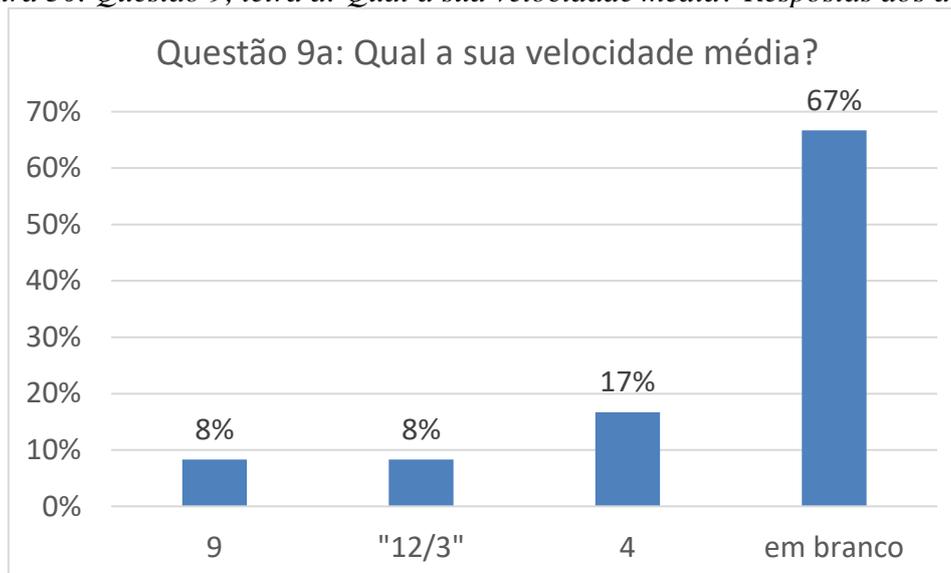
#### 3.4.1.9. Avaliando conhecimentos prévios em Física

A nona e última questão buscava encontrar o patamar do conhecimento dos alunos em Física. Como o conteúdo inicial trabalhado com os alunos do primeiro ano do ensino médio na matéria de Física são os conceitos de cinemática, nessa questão colocou-se uma situação típica que é sempre trabalhada em sala de aula, envolvendo cálculo de velocidade média e função horária da posição.

Nessa questão foi colocado o seguinte texto no enunciado: certo móvel percorre uma distância de 12 metros em 3 segundo.

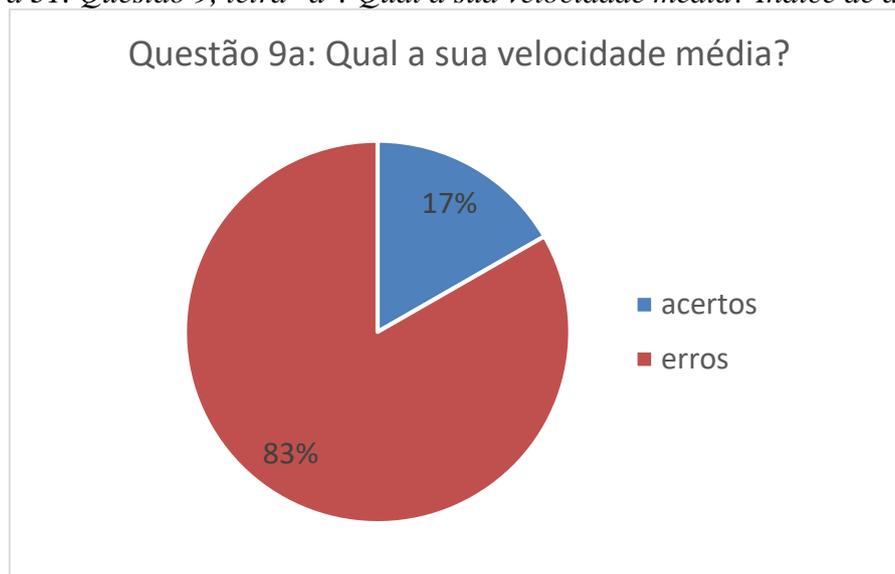
No primeiro comando, letra “a”, perguntou-se qual seria a sua velocidade média. Apenas 2 alunos (17%) conseguiram efetuar os cálculos corretamente, chegando na resposta de 4 m/s. A Figura 30 traz o gráfico com os resultados obtidos nessa questão, enquanto a Figura 31 revela o índice de acerto e erro nessa questão, ou seja, 17% de acerto e 83% de erro.

Figura 30. Questão 9, letra a: Qual a sua velocidade média? Respostas dos alunos.



Fonte: próprio autor.

Figura 31. Questão 9, letra "a": Qual a sua velocidade média? Índice de acertos.



Fonte: próprio autor.

No segundo comando dessa mesma questão, letra “b”, trazia o seguinte comando: mantendo essa velocidade constante, que distância ele teria percorrido em 2 segundos? Em 4 segundos? Em 5 segundos? Em 6 segundos? E em 15 segundos?

Nenhum aluno respondeu essa segunda parte da questão. Nota-se o baixo rendimento dos alunos com essas informações. Ao ser conversado com os alunos, eles revelaram já terem visto realmente esse conteúdo, todavia não se recordavam direito como se trabalha com esse tipo de problema.

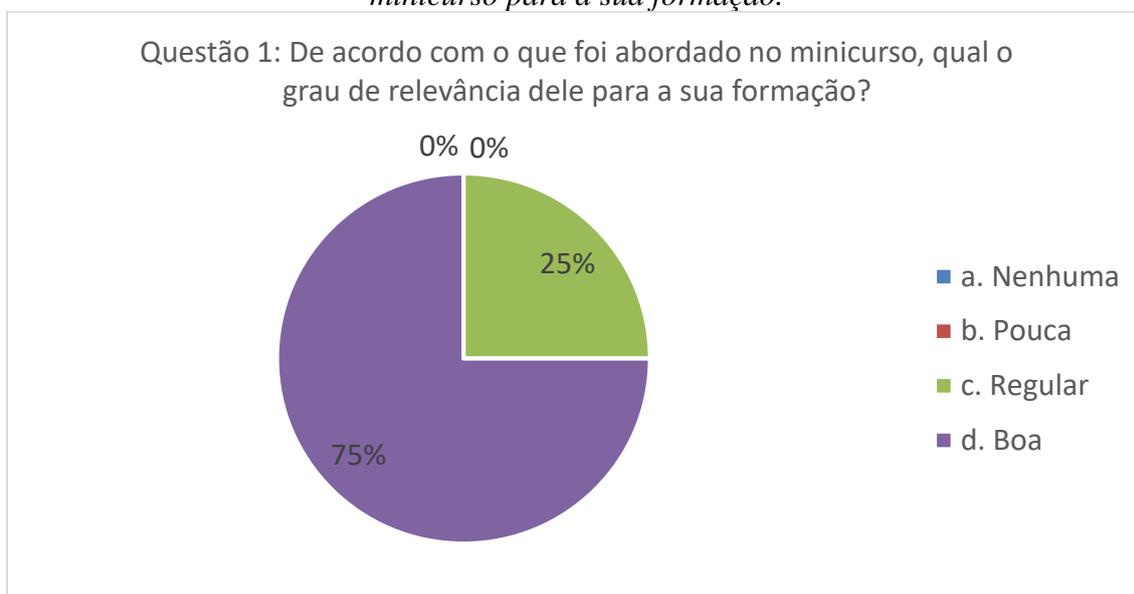
### 3.4.2. Questionário Final

No final do minicurso, no segundo dia, foi aplicado outro questionário para tentar avaliar os resultados dessa prática, verificando a aceitação dos alunos e ainda se esse conteúdo ensinado pode de fato auxiliar o processo de ensino e aprendizagem dos conteúdos de Física. O questionário foi composto por 6 questões e o mesmo pode ser visto no apêndice D.

#### 3.4.2.1. Percepção da relevância do conteúdo do minicurso

A primeira pergunta foi sobre a relevância do conteúdo do minicurso para a formação do aluno de acordo com a sua percepção. Dos 12 alunos que responderam o questionário, 3 (25%) deles escolheram a opção regular e os outros 9 (75%) marcaram a opção de que a percepção é boa. A Figura 32 ilustra o resultado obtido nesse item.

*Figura 32. Questão 1: percepção do aluno sobre a relevância do conteúdo do minicurso para a sua formação.*



Fonte: próprio autor.

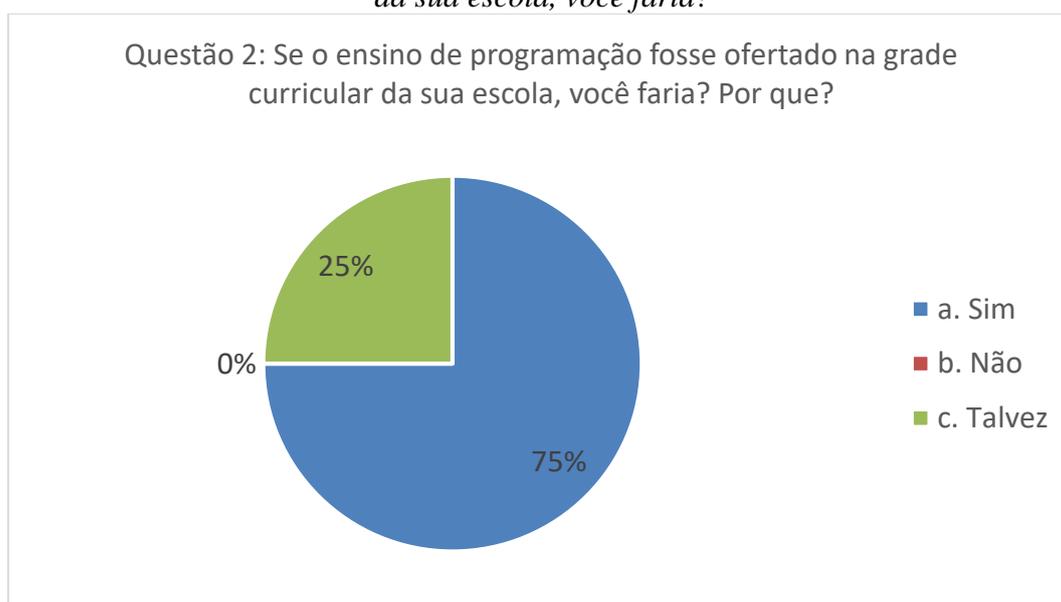
Como nenhum aluno marcou as opções “Nenhum” e “Pouca”, acredita-se que os alunos enxergam alguma relevância do conteúdo de lógica de programação com a sua formação. Dessa forma, pode-se supor que esse conteúdo tem uma aceitação no mínimo regular para esse grupo de alunos.

### 3.4.2.2. Se o ensino de programação fosse ofertado na grade curricular da sua escola, você faria?

Na segunda questão, os alunos foram indagados se eles cursariam alguma disciplina de ensino de programação caso ela fosse ofertada na grade curricular de sua escola. Desse grupo de alunos, 3 deles (25%) responderam que talvez e os outros 9 (75%) marcaram a opção que sim, que fariam. Na Figura 33 pode ser visto o gráfico referente a essa segunda questão.

Com esse resultado, pode-se perceber a inclinação dos alunos desse grupo em aprender programação, pois todos mostraram interesse em cursar uma disciplina voltada para isso. Percebe-se ainda mais a aceitação do conteúdo por esse grupo de alunos.

Figura 33. Questão 2: se o ensino de programação fosse ofertado na grade curricular da sua escola, você faria?



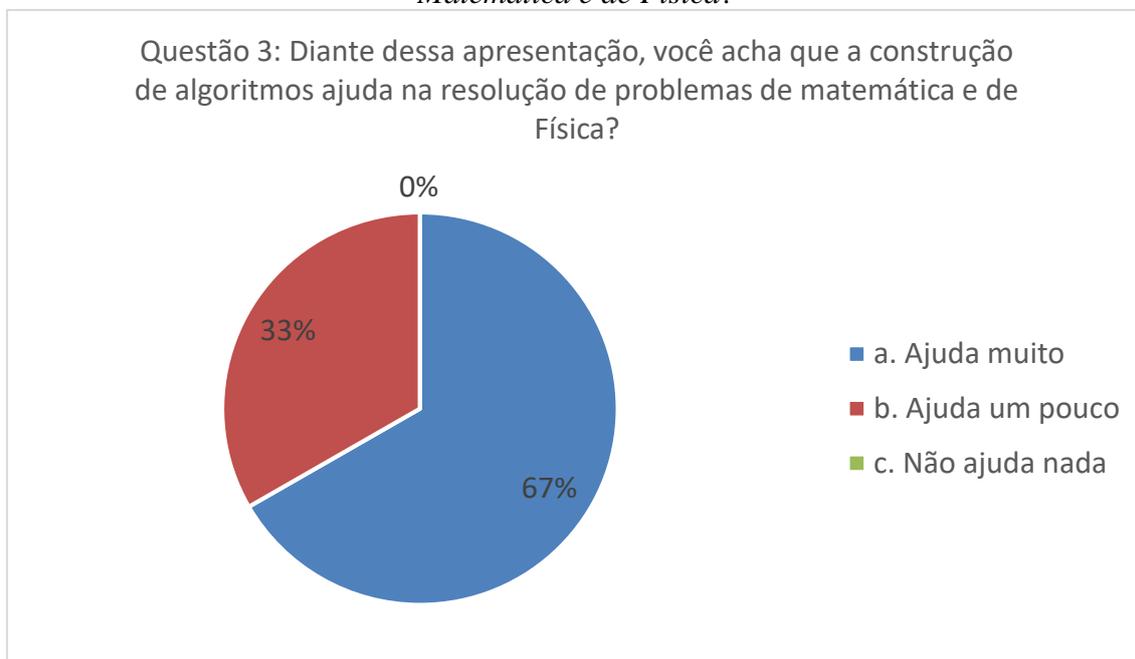
Fonte: próprio autor.

### 3.4.2.3. A construção de algoritmos ajuda na resolução de problemas de Matemática e de Física?

A terceira questão perguntava se os alunos acreditavam que a construção de algoritmos ajuda na resolução de problemas de Matemática e de Física. Do grupo de alunos, 4 deles (33%) responderam que ajuda um pouco, enquanto os outros 8 alunos

(67%) responderam que ajuda muito. Na Figura 34 pode ser visualizado o gráfico com o resultado dessa questão.

*Figura 34. Questão 3: a construção de algoritmos ajuda na resolução de problemas de Matemática e de Física?*



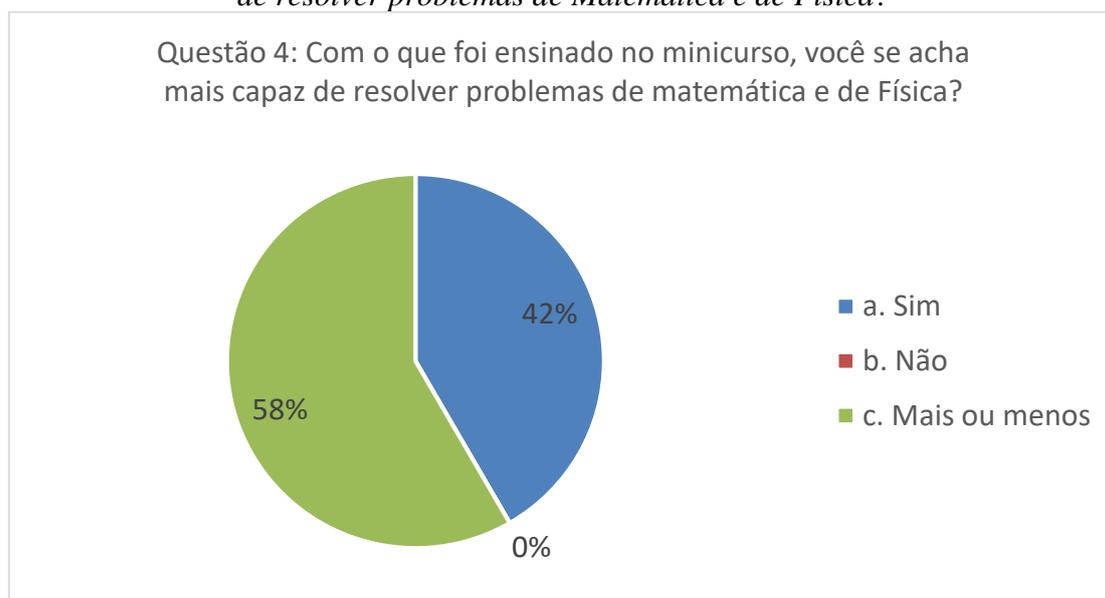
Fonte: próprio autor.

Com esse resultado, todos os alunos desse grupo acreditam que aprender a construir algoritmos ajuda de alguma forma na resolução de problemas de Física e de Matemática.

#### **3.4.2.4. Verificando se o aluno se acha mais capaz de resolver problemas de Matemática e de Física depois do que foi aprendido**

Os alunos foram perguntados na quarta questão se após o que foi ensinado no minicurso eles se sentiam mais capazes de resolver problemas de Matemática e de Física. Desse grupo de alunos, 5 deles (42%) responderam que sim e os 7 restantes (58%) responderam que talvez. A Figura 35 mostra o gráfico com os dados dessa questão.

Figura 35. Questão 4: Com o que foi ensinado no minicurso, você se acha mais capaz de resolver problemas de Matemática e de Física?



Fonte: próprio autor.

Nessa questão é possível notar o que seja talvez um pouco de receio, ou confiança ou a falta dela para responder com convicção. Todavia, vale ressaltar que nenhum aluno do grupo respondeu de forma negativa, ou seja, de alguma forma esses alunos acreditam que o aprendizado de lógica de programação pode de fato auxiliar na resolução de problemas de Física e de Matemática.

#### 3.4.2.5. Opinião dos alunos sobre o conteúdo abordado no minicurso

Na quinta questão foi perguntado aos alunos qual seria a opinião de cada um sobre o minicurso, de modo que eles exprimissem de fato o que cada um achou do conteúdo abordado, em especial para a finalidade do próprio minicurso. Como as respostas para essa questão são de forma discursiva, não foi gerado gráfico com o resultado, mas é possível exibir as respostas de cada um por meio da Tabela 2. Dos alunos presentes, três deles não responderam essa questão, deixando a resposta em branco.

*Tabela 2. Resposta dos alunos à questão 5: opinião dos alunos sobre o conteúdo abordado no minicurso*

Aluno1	Gostei. Muito legal, deveria ter mais vezes. Ajuda muito.
Aluno2	Sobre cada tópico que foi abordado, é presente que esses tipos de programas e sistemas ajudam em problemas matemático e de Física
Aluno3	De muito importância para nossa vida profissional, ajudando na resolução de problemas com rapidez
Aluno4	Bom
Aluno5	Legal
Aluno6	Gostei
Aluno7	Talvez ele ajude
Aluno8	Muito legal
Aluno9	Poderia durar mais, ter mais aula. É bom para praticar.
Aluno10	
Aluno11	
Aluno12	

Fonte: próprio autor.

O fato de ter tido três respostas em branco poderia ter sido melhor explorado, mas não foi imposto aos alunos a obrigatoriedade de responder todas as questões. Esses alunos poderiam não ter ficado a vontade para responder, ou não ter entendido direito a pergunta e assim não terem compreendido como deveriam responder, ou ainda poderiam ter tido uma opinião negativa e ficaram com receio de responder. Todavia, é muito importante observar que todos os demais alunos responderam de forma positiva, emitindo uma resposta que teriam gostado de alguma forma do minicurso, seja para sua própria formação escolar, quanto para um futuro aprendizado profissional.

#### **3.4.2.6. Verificando os conhecimentos adquiridos dos alunos**

Na questão 6 foi colocado o enunciado com a situação de que certo móvel percorre uma distância de 16 metros em 4 segundos. Situação bem parecida com o

problema colocado no questionário para avaliar os conhecimentos prévios do aluno. Dessa forma pode ser possível verificar se os alunos se tornaram mais aptos para resolver esses tipos de problemas de acordo com uma possível evolução ao comparar os resultados.

Essa questão foi dividida em dois comandos. No primeiro, letra a, foi perguntado qual seria a sua velocidade média. Já o segundo comando, letra b, foi perguntado o seguinte: mantendo essa velocidade constante, que distância ele teria percorrido em 2 segundos? Em 3 segundos? Em 5 segundos? Em 6 segundos? E em 15 segundos?

No primeiro comando, 10 alunos (83%) conseguiram responder corretamente, enquanto os outros 2 (17%) erraram, como pode ser visto no gráfico da Figura 36.

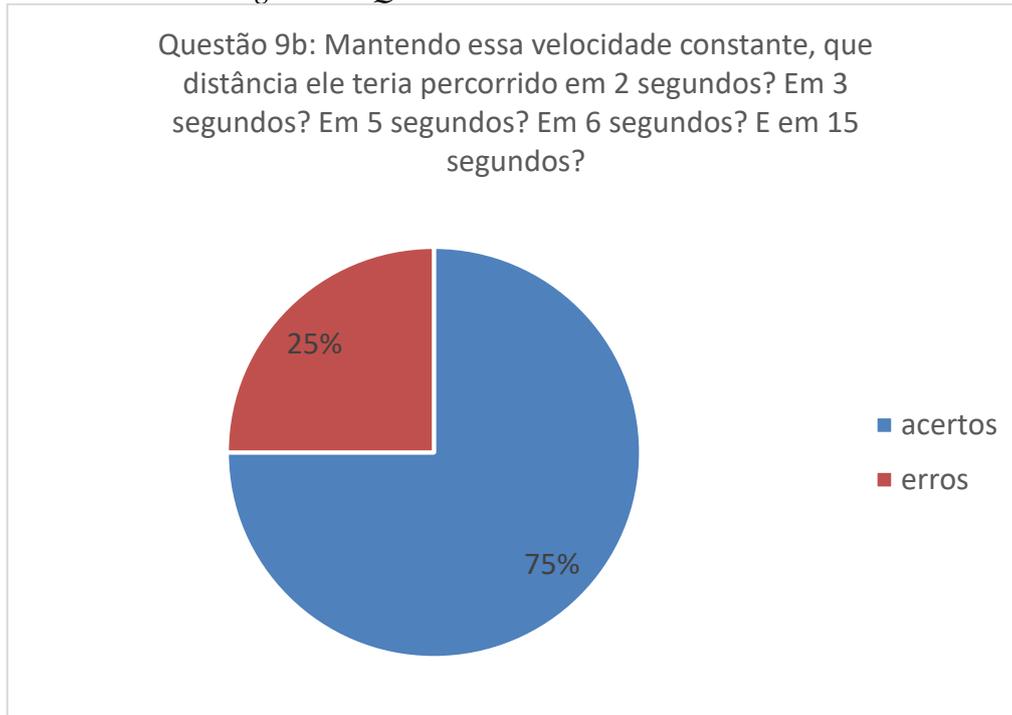
*Figura 36. Questão 6a: cálculo de velocidade média.*



Fonte: próprio autor.

Já no segundo comando, letra b, o número de acertos caiu para 9 (75%) e o número de erros subiu para 3 (25%), conforme mostra o gráfico da Figura 37.

*Figura 37. Questão 6b: cálculo de distância.*



Fonte: próprio autor.

Lembrando que no primeiro questionário apenas dois alunos conseguiram responder à questão que envolvia o cálculo de velocidade média e nenhum deles responderam a questão que envolvia o cálculo da distância, era esperado que eles utilizassem os conhecimentos adquiridos no minicurso para responder essas questões agora. Lembrando ainda que o cálculo de velocidade média foi trabalhado no exercício 3 e o cálculo de distância poderia ser utilizado o que foi aprendido no exercício 2 ao ter trabalhado equação do primeiro grau.

Nota-se um avanço significativo, onde agora quase todos os alunos responderam corretamente esses dois problemas. Pode ser feito duas abordagens sobre esse resultado obtido. Uma é que os alunos já até sabiam resolver esse tipo de problema, mas agora, após o minicurso, eles conseguiram aplicar o conhecimento de forma mais fluída, já que exercitaram muito a fórmula Matemática para criar e testar os códigos dos algoritmos. Outra abordagem é que os alunos acabaram desenvolvendo mais o raciocínio lógico, tendo a capacidade de construir os passos que resolve o problema dado.

### 3.5. OUTRAS POSSIBILIDADES

A proposta apresentada até agora se trata de uma possibilidade, podendo ser alterada de acordo com os objetivos desejados em sala de aula ou ainda de acordo a realidade encontrada pelo professor em sua escola. O que foi apresentado no tópico anterior é exatamente uma metodologia pensada na realidade de escolas públicas da educação básica brasileira, tanto que os resultados expostos são exatamente da aplicação do projeto nessas circunstâncias, mostrando a viabilidade e a possibilidade de ser executado. Porém nada impede que uma metodologia adaptada seja aplicada para uma realidade adversa da exposta nesse trabalho.

Em uma outra situação, totalmente fora do núcleo principal desse trabalho, foi aplicado a uma turma do segundo ano do ensino médio, uma abordagem semelhante. A turma em questão era do curso técnico de manutenção de redes de computadores integrado ao ensino médio do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO), do campus de Araguatins-TO. Como os alunos eram de um curso da área de tecnologia e já haviam sido introduzidos aos conceitos iniciais de programação, foi então abordado o uso de uma linguagem de programação mais específica.

A turma era composta por 20 alunos, e foi proposto aos alunos implementarem uma solução baseada em programação que tratasse do conteúdo de Física previsto. O conteúdo de Física dos alunos era o de termometria, sendo primeiro ensinado a dedução das fórmulas de conversão entre as escalas Celsius, Fahrenheit e Kelvin, e depois a teoria e as respectivas fórmulas sobre a dilatação linear, superficial e volumétrica dos sólidos.

Pelo fato de os alunos já estarem estudando a linguagem de programação Java, então optou-se em utiliza-la para implementação do programa final. Basicamente, após a explicação do conteúdo de Física, os alunos foram levados ao laboratório de informática do campus e foi apresentado a eles como as fórmulas poderiam ser manipuladas em Java.

A Figura 38 ilustra um exemplo de código em Java que trata da conversão de temperatura entre as escalas Celsius e Fahrenheit apresentada aos alunos em Java. Observe que o código pode parecer mais complexo do que os exemplos anteriores construídos em português pelo programa VisuAlg. Acontece que Java é uma linguagem

de programação com aplicações reais para construção de programas diversos, então requer maiores habilidades, e o entendimento do código requer uma maior compreensão de programação.

Figura 38. Linhas de códigos de programa em Java para conversão de temperatura entre as escalas Celsius e Fahrenheit.

```
14 public class JavaApplication1 {
15
16     /**
17      * @param args the command line arguments
18      * Tc = Tk - 273
19      * Tc = 5*((Tf-32)/9)
20      * Tf = ((Tc/5)*9)+32
21      */
22     public static void main(String[] args) {
23         // TODO code application logic here
24         Scanner teclado = new Scanner(System.in);
25         double temperatura;
26
27         //conversão de Celsius para Farenheint
28         System.out.println("##CONVERTENDO TEMPERATURA CELSIUS PARA FARENHEIT##");
29         System.out.println("Insira a temperatura em Celsius: ");
30         temperatura = teclado.nextDouble();
31         temperatura = ((temperatura/5)*9)+32;
32         System.out.println("A temperatura em Farenheit é: "+temperatura);
33
34         //conversão de Farenheint para Celsius
35         System.out.println("");
36         System.out.println("##CONVERTENDO A TEMPERATURA FARENHEIT PARA CELSIUS##");
37         System.out.println("Insira a temperatura em Farenheit: ");
38         temperatura = teclado.nextDouble();
39         temperatura = (5*(temperatura-32))/9;
40         System.out.println("A temperatura em Celsius é: "+temperatura);
41     }
```

Fonte: próprio autor.

Mesmo assim, os alunos aceitaram o desafio e a turma foi dividida em quatro grupos com cinco alunos cada grupo, com um prazo de duas semanas para entregar a versão final do programa, contando com acompanhamento e estudo dirigido nesse meio tempo. Apesar de parecer difícil, os alunos se mostraram bem empenhados na execução do trabalho e ficaram motivados por enxergar a conciliação do conteúdo de Física com a prática de programação e possibilidade de executar um trabalho de certa forma inusitado para eles.

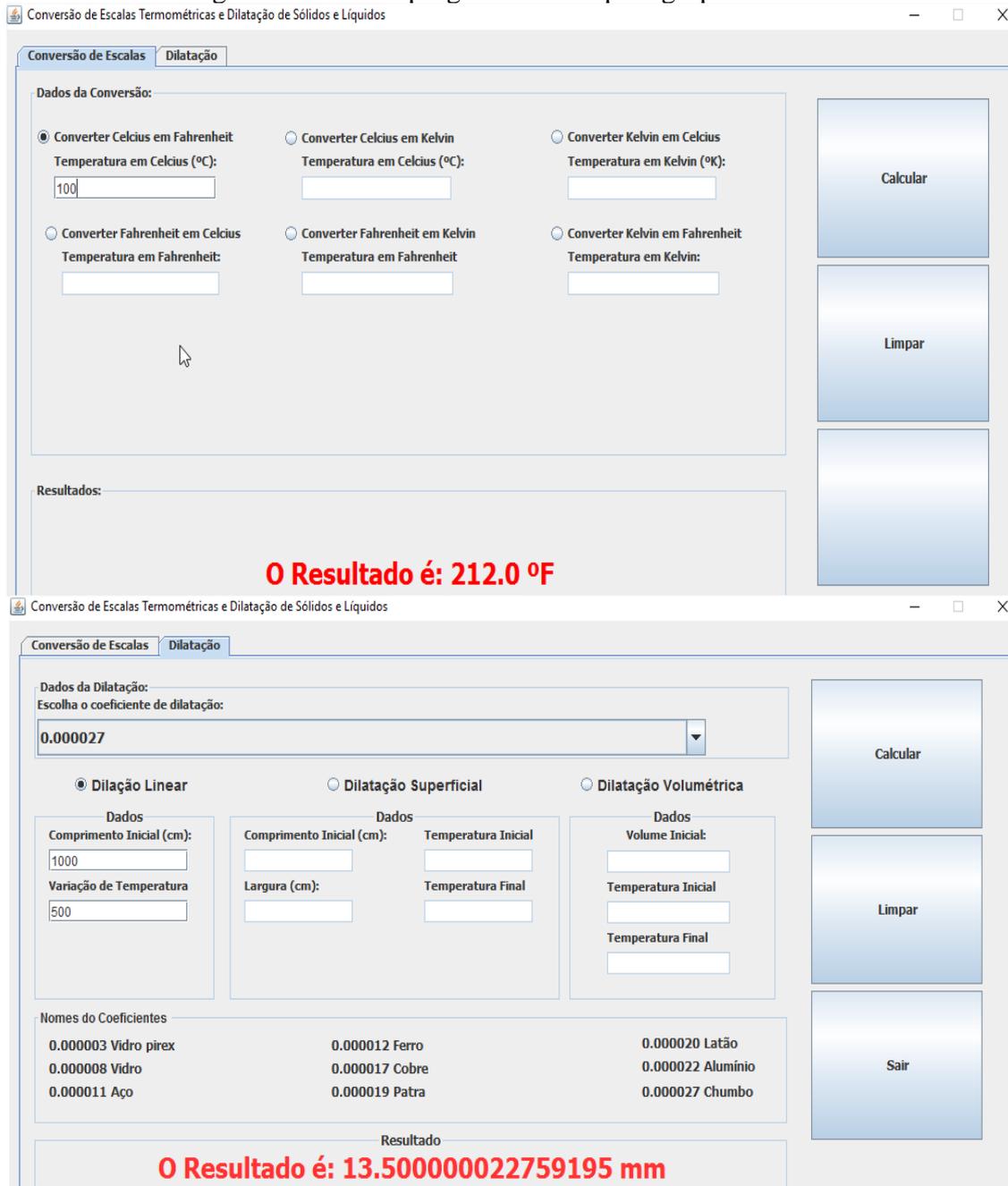
Considerando que se trata de alunos do segundo ano do ensino médio, que ainda não possuem um conhecimento avançado de programação, pode-se dizer que o resultado foi surpreendente positivamente, tanto pelo envolvimento com a matéria de

Física como pelo ganho de experiência bem relevante na programação em Java relatado pelos próprios alunos.

As Figuras 39, 40, 41 e 42 são imagens dos programas finais entregues por cada um dos 4 grupos. Todos eles optaram por implementarem soluções baseadas em janelas, que de certa forma, apesar de ser mais complicado, é mais interessante do que programa que roda em linhas de execução.

Na Figura 39 é apresentado a tela do programa criado pelo primeiro grupo a apresentar. Nesse programa os alunos utilizaram uma única janela, porém separou a parte de conversão de escalas termométricas da dilatação por meio de painéis, como se fossem guias de um navegador.

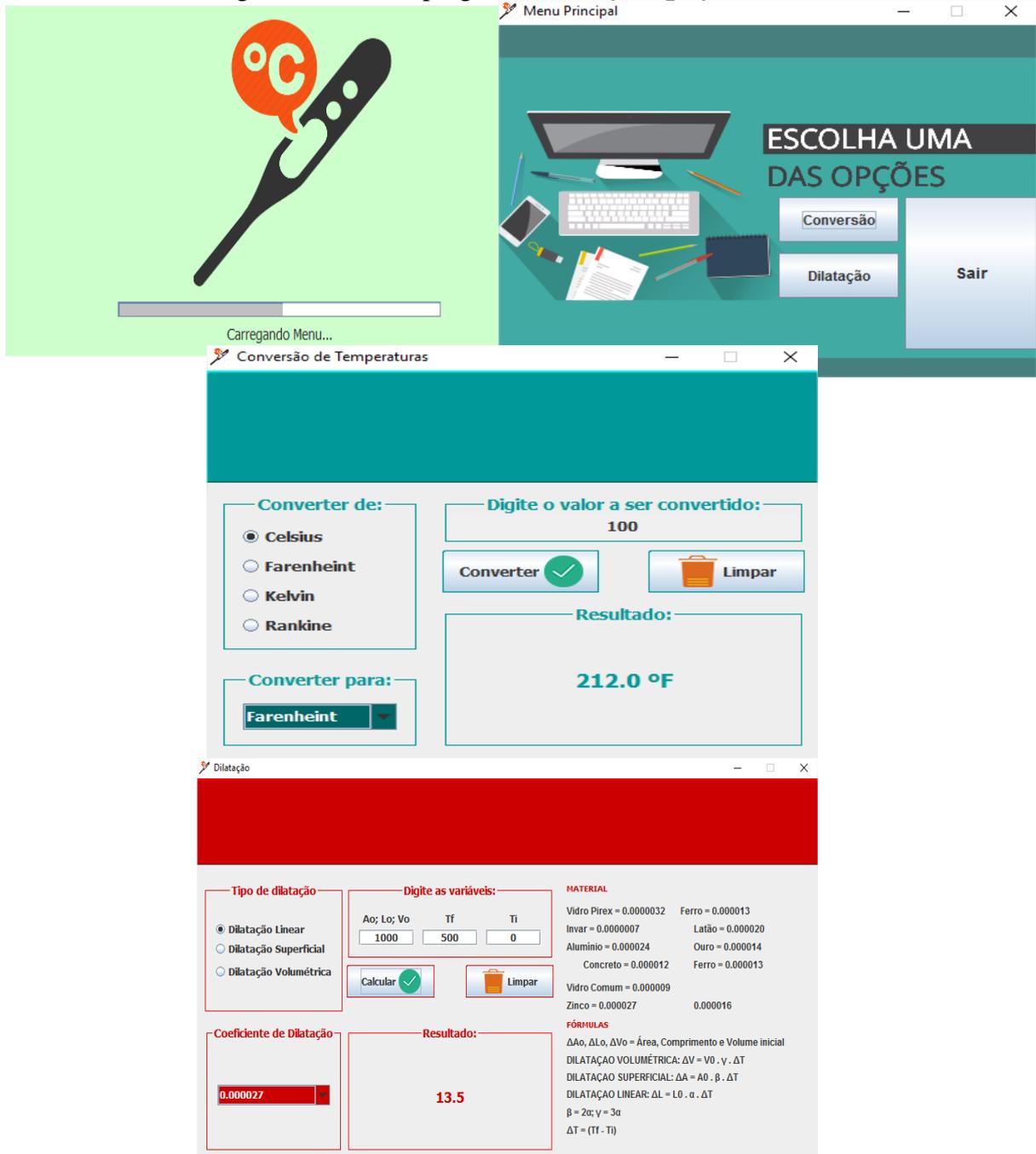
Figura 39. Tela do programa criado pelo grupo 1.



Fonte: próprio autor.

Já a Figura 40 apresenta as telas do programa entregue pelo segundo grupo. Nesse programa os alunos utilizaram janelas independentes, porém com uma janela principal que permitisse escolher entre a opção de ir para a conversão de escalas ou para o cálculo de dilatação. Os alunos criaram, inclusive, uma tela de inicialização. As janelas são bem compactas, porém bastante funcionais e otimizadas, mostrando uma solução bem interessante apresentada pelo grupo.

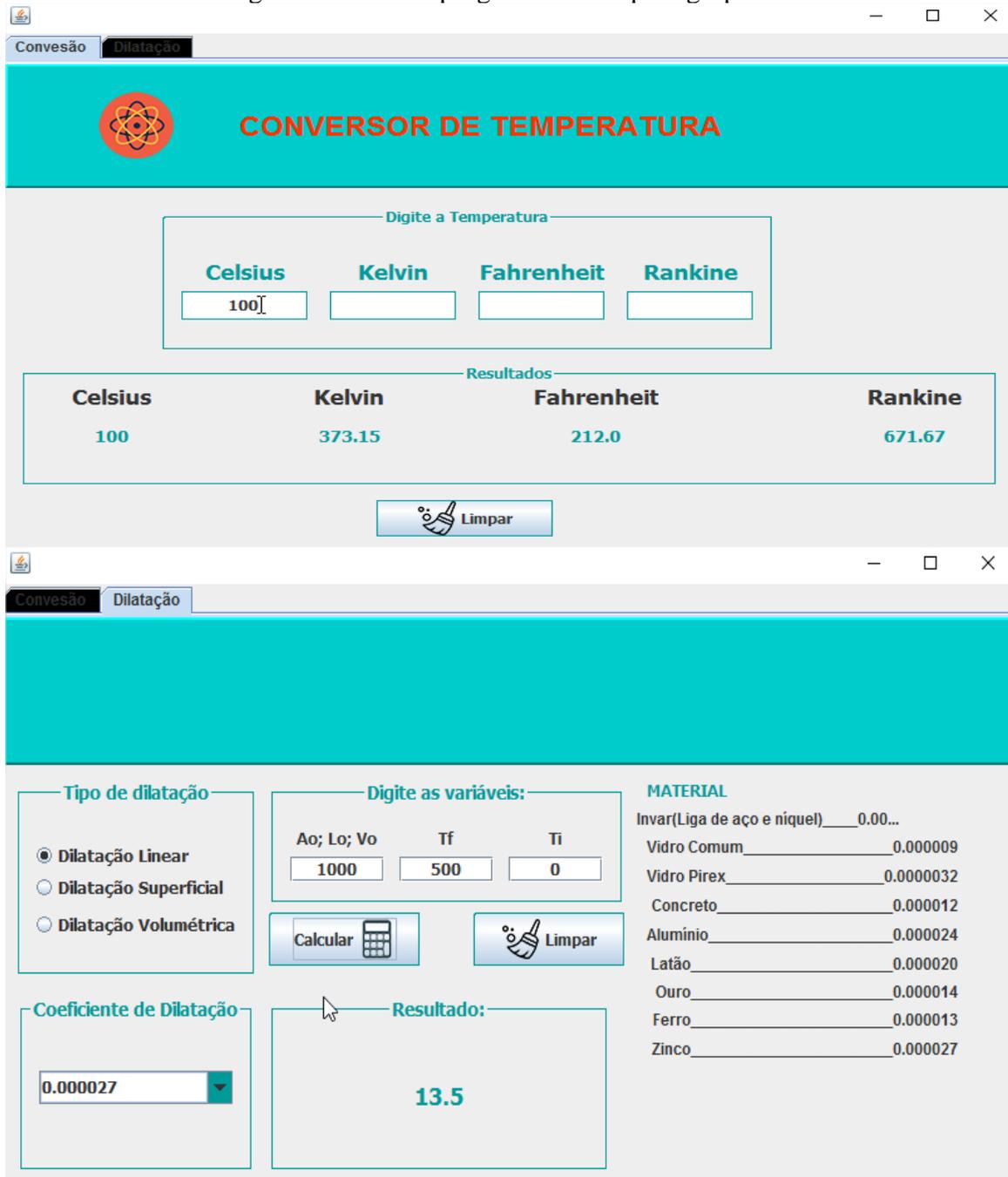
Figura 40. Tela do programa criado pelo grupo 2.



Fonte: próprio autor.

Enquanto isso, a Figura 41 mostra as telas do programa apresentado pelo terceiro grupo. Neste programa o grupo optou por também utilizar uma janela separando em painéis diferentes a parte de conversão de escalas da parte de cálculo de dilatação. Esse grupo optou por considerar além das escalas termométricas mais convencionais (Celsius, Kelvin e Fahrenheit), por também incluir a escala Rankine, pois estava no livro didático de apoio dos alunos.

Figura 41. Tela do programa criado pelo grupo 3.



Fonte: próprio autor.

Por fim, a Figura 42 apresenta a Tela do programa criado pelo quarto grupo. Os alunos desse grupo optaram por seguir a mesma linha de pensamento do primeiro e terceiro grupo para construir o seu programa, com uma única janela separando em painéis a parte de conversão de escalas da parte do cálculo de dilatação. O programa desse grupo também apresenta a escala Rankine implementada além das escalas termométricas mais convencionais.

Figura 42. Tela do programa criado pelo grupo 4.



Fonte: próprio autor.

#### 4. CONSIDERAÇÕES FINAIS

Apesar de praticamente todo o conteúdo do minicurso está focado em ensinar lógica de programação, a finalidade última desse trabalho é tentar encontrar de alguma forma se o fato dos alunos compreenderem a lógica de programação pode propiciar melhora no raciocínio da resolução dos problemas matemáticos e físicos. Ou seja, se a lógica de programação pode ser considerada como uma ferramenta Matemática para o ensino e aprendizagem na Física.

Obviamente que apenas o fato de o aluno saber lógica de programação não o fará um bom aluno nas matérias de exatas, mas é perceptível que eles possuem uma afinidade em compreender os conceitos que envolva a lógica Matemática e assim facilidade em resolver problemas de cálculos. É importante ainda salientar a importância do conteúdo específico das matérias finais para que o aluno aprenda os conceitos, até porque se não tiverem compreendido os conceitos, dificilmente será possível desenvolver o raciocínio para resolver os problemas.

Considerando tudo que foi disposto nesse trabalho e observando os resultados obtidos é possível verificar que os alunos que conseguem construir a lógica de programação podem ter certa facilidade em propor soluções para problemas que envolvem raciocínio matemático, incluindo os cálculos de questões de Física. Enquanto na verificação dos conhecimentos prévios dos alunos apenas 17% conseguiram desenvolver os cálculos de velocidade média e nenhum deles conseguiram calcular a distância percorrida, na verificação dos conhecimentos adquiridos após a intervenção metodológica esse número foi significativamente maior, saltando para 83% dos alunos que conseguiram desenvolver corretamente os cálculos de velocidade média e 75% para os cálculos de distância média.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Otávio Fossa de. *Jogo educacional para o ensino de Relatividade Galileana*. 2016. 205 f. Dissertação (Mestrado em Ensino de Física) – Instituto de Física, Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2016.

ANJOS, J. R. dos; FREITAS, S. dos A.; ANDRADE NETO, A. S. de. *Utilização do software scratch para a aprendizagem de lançamentos de projéteis e conceito de gravidade no ensino fundamental*. ACTIO, Curitiba, v. 1, n. 1, p. 128-144, jul./dez. 2016. Disponível em: <<https://periodicos.utfpr.edu.br/actio>>. Acesso em: 02 jul. 2017.

BARBOSA, Cairo Dias; NUNES, Roberto Costa. *O uso da informática nas aulas de Física das escolas públicas da região metropolitana de Marabá*. Trabalho de Conclusão de Curso, UEPA. 2011.

BARROS, Daniela Melaré Vieira. *Guia didático sobre as tecnologias da comunicação e informação: material para o trabalho educativo na formação*. Rio de Janeiro (RJ): Vieira & Lent, 2009. 157 p.

BEHRENS, Marilda Aparecida; MASETTO, Marcos T.; MORAN, José Manuel. *Novas Tecnologias e mediação pedagógica*. 3 ed. Campinas (SP): Papirus, 2000. 173 p.

BRASIL. *Guia de tecnologias educacionais*. Ministério da Educação. Secretaria de Educação Básica, Brasília, 2008. Disponível em <[http://portal.mec.gov.br/seb/arquivos/pdf/Avalmat/guia\\_de\\_tecnologias\\_educacionais.pdf](http://portal.mec.gov.br/seb/arquivos/pdf/Avalmat/guia_de_tecnologias_educacionais.pdf)>. Acesso em: 30 jun 2017.

CALDAS, Gracilene Garcia. *Atividades experimentais de acústica para o ensino de Física: Uma proposta de inclusão de surdos*. 2017. 132 f. Dissertação (Mestrado em Ensino de Física) – Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará, Belém. 2017.

COSTA, Sandra Regina Santana; DUQUEVIZ, Barbara Cristina; PEDROZA, Regina Lúcia Sucupira. *Tecnologias Digitais como instrumentos mediadores da aprendizagem dos nativos digitais*. Psicol. Esc. Educ., Maringá, v. 19, n. 3, p. 603-610, dez. 2015. Disponível em <<http://www.scielo.br/pdf/pee/v19n3/2175-3539-pee-19-03-00603.pdf>>. acessos em 06 jul. 2017. <http://dx.doi.org/10.1590/2175-3539/2015/0193912>

EINSTEIN, Albert. *Como vejo o mundo*. 11. ed. Rio de Janeiro: Nova Fronteira, 1981.

FREIRE, Paulo. *Pedagogia da autonomia: saberes necessários à prática educacional*. 25. ed. São Paulo: Paz e Terra, 2002.

FREITAS, Maria Teresa de Assunção. *História da Pedagogia: A Perspectiva Vigotskiana e as Tecnologias*. Revista Educação. São Paulo, v. 2, p. 58 – 67, ago. 2010.

GARLET, Daniela; BIGOLIN, Nara Martini; SILVEIRA, Sidnei Renato. *Uma Proposta para o Ensino de Programação de Computadores na Educação Básica*. Universidade Federal de Santa Maria, Frederico Westphalen, Rio Grande do Sul, 2016. Disponível em <[w3.ufsm.br/frederico/images/DanielaGarlet.pdf](http://w3.ufsm.br/frederico/images/DanielaGarlet.pdf)>. Acesso em: 30 jun. 2017.

GASPAR, Alberto. *Cinquenta anos de ensino de Física: muitos equívocos, alguns acertos e a necessidade do resgate do papel do professor*. Artigo apresentado no XV Encontro de Físicos do Norte e Nordeste, Guaratinguetá - SP. Disponível em <[https://edisciplinas.usp.br/pluginfile.php/3360182/mod\\_resource/content/0/CINQ%C3%92CENTA%20ANOS%20DE%20ENSINO%20DE%20F%C3%82SICA.pdf](https://edisciplinas.usp.br/pluginfile.php/3360182/mod_resource/content/0/CINQ%C3%92CENTA%20ANOS%20DE%20ENSINO%20DE%20F%C3%82SICA.pdf)>. Acesso em: 06 abril 2018.

HOHENFELD, Dielson P.; LAPA, Jancarlos M.; MARTINS, Maria Cristina Mesquita. *As Tecnologias da Informação e Comunicação no curso de Física da UEFS*. VI Encontro Nacional de Pesquisa em Educação em Ciências, Florianópolis, SC, v. 6, p. 792-801, dez. 2007. Disponível em <[www.nutes.ufjf.br/abrapec/vienpec/CR2/p792.pdf](http://www.nutes.ufjf.br/abrapec/vienpec/CR2/p792.pdf)>. Acesso em 25 maio 2017.

LARA, Alessandro Luiz de; MANCIA, Letícia Beraldi; SABCHUK, Luiza; PINTO, Angela Emilia Almeida; SAKAGUTI, Paula Mitsuyo Yamasaki. *Ensino de Física Mediado por Tecnologias de Informação e Comunicação: um relato de experiência*. XX Simpósio Nacional de Ensino de Física, São Paulo, SP, jan. 2013. Disponível em <[www2.unifap.br/rsmatos/files/2013/10](http://www2.unifap.br/rsmatos/files/2013/10)>. Acesso em: 30 jun. 2017.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de Oliveira. *Algoritmos - Lógica Para Desenvolvimento de Programação*. 27. Ed. Editora Érica. 2014.

MARTINS, Alisson Antonio; GARCIA, Nilson Marcos Dias. *Ensino de Física e Novas Tecnologias de Informação e Comunicação: Uma Análise da Produção Recente*. VIII Encontro Nacional de Pesquisa em Educação em Ciências, Campinas, São Paulo, v. 8, p. 529-538, dez. 2011. Disponível em

<<http://www.nutes.ufrj.br/abrapec/viiienpec/resumos/R0529-1.pdf>>. Acesso em: 30 jun 2017.

MOREIRA, Marco Antonio. *Ensino de Física no Brasil: Retrospectiva e Perspectivas*. Revista Brasileira de Ensino de Física, vol. 22, p.94 - 99, Março, 2000.

MOREIRA, Marco Antonio. *Teorias de Aprendizagem*. São Paulo: EPU, 1999.

PEREIRA JUNIOR, Miguel da Silva. *Coletânea Virtual de Física para o 9º ano do Ensino Fundamental*. 2017. 79 f. Dissertação (Mestrado em Ensino de Física) – Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará, Belém. 2017.

PEREIRA, Leonardo. (2013) *Escolas Defendem Ensino de Programação a Crianças e Adolescentes*. *Olhar Digital*, 06 Fev. 2013. Disponível em <<http://olhardigital.uol.com.br/noticia/escolas-defendem-ensino-de-programacao-acrianças-e-adolescentes/35075>>. Acesso em: 30 jun. 2017.

PIETROCOLA, Maurício. *Ensino de Física: conteúdo, metodologia e epistemologia em uma concepção integradora*. 2 ed. rev. Florianópolis: Editora da UFSC, 2005.

POZO, Juan Ignacio; CRESPO, Miguel Ángel Gómez. *A aprendizagem e o ensino de ciências: do conhecimento cotidiano ao conhecimento científico*. Tradução Naila Freitas. 5. ed. Porto Alegre: Artmed, 2009.

SIMUNA, Renatho. *Pirâmide de William Glasser ou “Cone da Aprendizagem”*, 02 Ago. 2017. Disponível em <<https://medium.com/@renatho/pir%C3%A2mide-de-william-glasser-ou-cone-da-aprendizagem-49a4670afc9a>>. Acesso em: 08 fev. 2019.

ZABALA, Antoni. *A prática educativa: como ensinar*. Tradução Ernani F. da F. Rosa. Porto Alegre: Artmed, 1998.

# APÊNDICES

## APÊNDICE A - APOSTILA: MINICURSO DE LÓGICA DE PROGRAMAÇÃO PARA O ENSINO DE FÍSICA

### APOSTILA – CURSO DE INTRODUÇÃO À ALGORITMOS

Autor: Roberto Costa Nunes

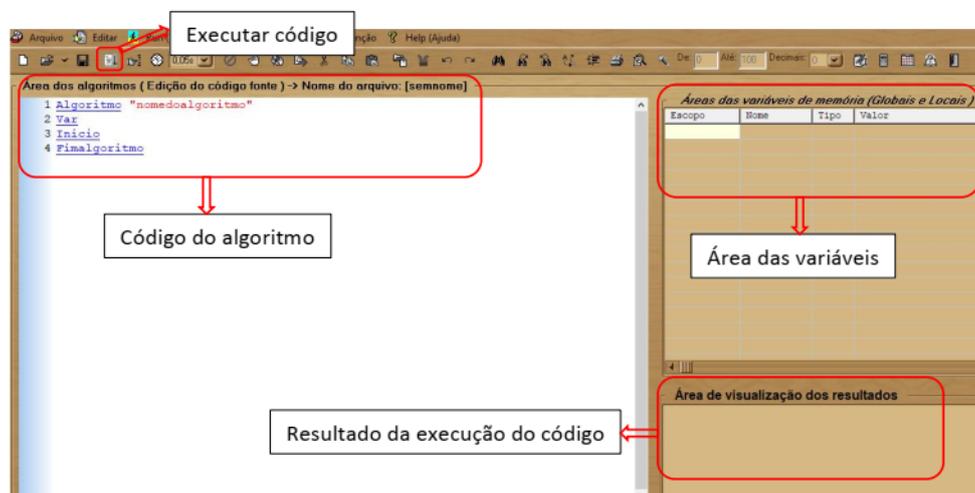
#### 1. DEFINIÇÃO – O QUE SÃO ALGORITMOS?

O termo algoritmo é definido por MANZANO & OLIVEIRA (2014) como um conjunto de passos finitos e organizados de forma lógica que, quando executados, resolvem determinado problema. Este termo está associado a computação e a matemática, mas pode-se dizer que até mesmo nas atividades cotidianas algoritmos estão presentes. O conjunto de instruções para se fazer um bolo, por exemplo, seria um exemplo.

Basicamente um algoritmo possui uma rotina e deve ter alguma lógica. Imagine uma tarefa simples, como atravessar a rua. Quando se está diante dessa situação, para executar a tarefa “atravessar a rua”, algumas etapas devem ser realizadas, como: olhar para os lados; verificar se vem veículo; se não vier veículo, atravessar; senão, aguarde e verifique de novo. Esse conjunto de etapas, pode ser chamado de algoritmo, o qual é classificado como não computacional.

Para o exemplo anterior, a sequência de passos pode ser alterada, assim como aumentar ou diminuir a quantidade de etapas, porém, deve-se haver alguma lógica entre os passos. Se o algoritmo fosse construído onde a etapa de olhar os lados estivesse após a ação de atravessar a rua, isso poderia gerar um grande problema. Receitas, instruções de montagem e manuais de uso também podem ser chamados de algoritmos não computacionais. A mesma coisa ocorre com os algoritmos computacionais, pois os mesmos necessitam de lógica para estarem corretos.

#### 2. LÓGICA DE PROGRAMAÇÃO – VISUALG



O VisualG é um programa desenvolvido com fins didáticos para o ensino da lógica de programação, servindo como um interpretador de algoritmo construído em português estruturado, que é denominado de portugol.

##### 2.1. Comando ESCREVA e ESCREVAL – Saída de dados

Para se escrever um algoritmo em portugol, é aconselhado utilizar um determinado padrão. Para o VisualG, tem-se uma estrutura padrão que deve ser seguida, contendo o seguinte escopo:

1	ALGORITMO “NomeDoAlgoritmo”
2	VAR
3	INICIO
4	FIMALGORITMO

A primeira linha serve para identificar o algoritmo, sendo que o nome do mesmo fica no espaço entre as aspas duplas. A segunda linha é destinada declaração de variáveis. As variáveis receberão um tópico específico para serem tratadas aqui. A terceira e quarta linha marca o início e fim, respectivamente, como código que será de fato executado.

Para poder analisar o funcionamento do VisualG, pode ser utilizado o seguinte algoritmo que mostra uma breve saída de dados.

<b>Área dos algoritmos ( Edição do código fonte ) -&gt;</b> 1 <b>Algoritmo</b> "MeuPrimeiroAlgoritmo" 2 <b>Var</b> 3 <b>Início</b> 4 <b>ESCREVA</b> ("Olá Mundo!") 5 <b>Fimalgoritmo</b>	<b>Área de visualização dos resultados</b> Início da execução Olá Mundo! Fim da execução.
---	--

Aqui, é utilizado o comando "ESCREVA" para exibir na tela o que for colocado dentro do parêntese, no caso a mensagem "Olá Mundo!". Pode ser utilizado também o comando "ESCREVAL", com a diferença de que com este comando é pulado uma linha após a escrita na tela.

### 2.2. Variáveis e tipos primitivos

As variáveis são espaços na memória do computador que podem receber determinados valores. Elas são utilizadas quando é necessário guardar uma certa informação para o computador pode manipula-la. Para criar uma variável, é preciso declarar ela primeiro, determinando o seu tipo.

Os tipos que serão trabalhados aqui, serão o inteiro, o real, o caractere e o lógico. O tipo inteiro e o real, guardam valores numéricos, tendo os seus nomes associados de forma intuitiva aos conjuntos numéricos aprendidos na matemática. O tipo caractere serve para guardar valores literários, ou seja, letras e palavras. Por fim, o tipo lógico guarda apenas valor lógico, que pode ser verdadeiro ou falso.

Quando uma variável é declarada, ela pode ser utilizada no corpo do algoritmo, recebendo algum valor diretamente ou resultado de alguma operação. A passagem de um valor para a variável, é chamado de atribuição. Veja o exemplo abaixo do algoritmo "SeuNome" com o uso de uma variável e sua respectiva saída.

<b>Área dos algoritmos ( Edição do código fonte ) -&gt;</b> 1 <b>Algoritmo</b> "SeuNome" 2 <b>Var</b> 3 <b>nome: caractere</b> 4 <b>Início</b> 5 <b>nome</b> <- "Roberto Costa Nunes" 6 <b>ESCREVA</b> ( <b>nome</b> ) 7 <b>Fimalgoritmo</b>	<b>Área de visualização dos resultados</b> Início da execução Roberto Costa Nunes Fim da execução.
---	---

Diagrama de anotações:  
- Uma caixa vermelha rotulada "declaração" aponta para a linha 3: nome: caractere.  
- Uma caixa vermelha rotulada "atribuição" aponta para a linha 5: nome <- "Roberto Costa Nunes".

### 2.3. Comando LEIA – Entrada de dados

O comando LEIA é utilizado para receber uma entrada de dados. Pode-se dizer que é o contrário do comando ESCREVA, que é para saída. O comando LEIA é muito utilizado para interagir com o usuário, ou seja, por meio dele é possível receber alguma coisa que o usuário digite no teclado, que é a entrada padrão que será utilizada aqui. Observe abaixo como fica o algoritmo "SeuNome" com a utilização do comando LEIA.

<b>Área dos algoritmos ( Edição do código fonte ) -&gt;</b> 1 <b>Algoritmo</b> "SeuNome" 2 <b>Var</b> 3 <b>nome: caractere</b> 4 <b>Início</b> 5 <b>ESCREVA</b> ("Digite o seu nome: ") 6 <b>LEIA</b> ( <b>nome</b> ) 7 <b>ESCREVA</b> ("Olá ", <b>nome</b> ) 8 <b>Fimalgoritmo</b>	<b>Área de visualização dos resultados</b> Início da execução Digite o seu nome: Roberto Costa Nunes Olá Roberto Costa Nunes Fim da execução.
---	---

O comando LEIA está na linha 6. Quando o código é executado, o computador espera uma entrada de dados ao chegar no comando LEIA. Após digitar a entrada e apertar a tecla ENTER do teclado, o fluxo de execução continua. Nesse caso, foi digitado o nome "Roberto Costa Nunes", o qual foi armazenado na variável *nome*. Na linha 7 foi utilizado o comando ESCREVA para mostrar a saída de dados, porém aqui foi utilizado o recurso de concatenação, para unir a palavra "Olá " com o que estiver contido na variável *nome*.

Exercício 1: faça um algoritmo que receba dois valores inteiros e mostre a soma entre eles.

#### 2.4. Operações Aritméticas

É possível executar no VisualG as operações aritméticas aprendidas na educação básica. As operações e seus respectivos símbolos de operação são: soma (+), subtração (-), multiplicação (\*), divisão (/), divisão inteira (\), exponenciação (^) e resto (%). Para exemplificar as operações aritméticas, veja o exemplo do algoritmo "SomarDoisNumeros", que é uma das soluções possíveis para o Exercício 1.

Área dos algoritmos ( Edição do código fonte ) -> Nome d

```

1 Algoritmo "SomarDoisNumeros"
2 Var
3   num1, num2, soma: inteiro
4 Inicio
5   ESCREVA("Digite o primeiro número: ")
6   LEIA(num1)
7   ESCREVA("Digite o segundo número: ")
8   LEIA(num2)
9   soma <- num1 + num2
10  ESCREVA("Soma = ", soma)
11 Fimalgoritmo

```

#### Área de visualização dos resultados

```

Início da execução
Digite o primeiro número: 2
Digite o segundo número: 5
Soma = 7
Fim da execução.

```

#### 2.5. Operadores Relacionais

Os operadores relacionais retornam um valor lógico (verdadeiro ou falso). Os operadores relacionais com seus respectivos símbolos de operação são: maior que (>), menor que (<), maior igual (>=), menor igual (<=), igual (=) e diferente (<>). Veja o algoritmo "OperaçãoRelacional" com a exemplificação da operação de maior que e menor que.

```

1 Algoritmo "OperaçãoRelacional"
2 Var
3   result: logico
4 Inicio
5   result <- 2>5
6   ESCREVAL("2 é maior que 5? ",result)
7   result <- 2<5
8   ESCREVAL("2 é menor que 5? ",result)
9 Fimalgoritmo

```

#### Área de visualização dos resultados

```

Início da execução
2 é maior que 5? FALSO
2 é menor que 5? VERDADEIRO
Fim da execução.

```

#### 2.6. Estruturas Condicionais

As estruturas condicionais são utilizadas para tomadas de decisões de acordo com uma condição verificada. Assim, pode-se garantir que determinada ação só será executada se certa condição for satisfeita. Para demonstrar essa estrutura, veja o exemplo do algoritmo "EstruturaCondional".

```

1 Algoritmo "EstruturaCondional"
2 Var
3   num1, num2: inteiro
4 Inicio
5   num1 <- 8
6   num2 <- 3
7   SE (num1 > num2) ENTAO
8     ESCREVA(num1, " é maior")
9   SENAO
10    ESCREVA(num2, " é maior")
11  FIMSE
12 Fimalgoritmo

```

#### Área de visualização dos resultados

```

Início da execução
8 é maior
Fim da execução.

```

## 2.7. Estruturas de Repetição

Estruturas de repetição são utilizadas quando se deseja realizar uma determinada operação várias vezes. Pode-se utilizar repetições não controladas e repetições controladas. Na repetição não controlada, uma operação é repetida até que determinada condição seja satisfeita. Nesse caso, é necessário ter cuidado para que a condição seja satisfeita em algum momento, pois corre o risco ficar em execução sem nunca parar. Duas estruturas de repetição do tipo não controlada são possíveis no VisualG, o ENQUANTO e o REPITA. Veja o algoritmo "ContarAteCinco" com a estrutura ENQUANTO.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   num, cont: inteiro
4 Inicio
5   num <- 5
6   cont <- 0
7   ENQUANTO (cont<=num) FACA
8     ESCREVAL (cont)
9     cont <- cont + 1
10  FIMENQUANTO
11 Fimalgoritmo
```

```
Área de visualização dos resultados
Início da execução
0
1
2
3
4
5
Fim da execução.
```

Analise agora o algoritmo que faz a mesma coisa, porém com a estrutura REPITA.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   num, cont: inteiro
4 Inicio
5   num <- 5
6   cont <- 0
7   REPITA
8     ESCREVAL (cont)
9     cont <- cont + 1
10  ATE (cont>num)
11 Fimalgoritmo
```

```
Área de visualização dos resultados
Início da execução
0
1
2
3
4
5
Fim da execução.
```

A estrutura de repetição controlada é utilizada quando se quer fazer uma operação em quantidade de passo já conhecida. Essa estrutura é comumente chamada de PARA. Com ela, a operação somente será executada a quantidade de passos determinada. Veja o exemplo do algoritmo "ContarAteCinco" utilizando a estrutura de repetição PARA.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   cont: inteiro
4 Inicio
5   PARA cont DE 0 ATE 5 PASSO 1 FACA
6     ESCREVAL (cont)
7   FIMPARA
8 Fimalgoritmo
```

```
Área de visualização dos resultados
Início da execução
0
1
2
3
4
5
Fim da execução.
```

Exercício 2: faça um algoritmo que receba 5 números inteiros, mostre como resultado a soma deles e a média aritmética.

Exercício 3: considere a equação do primeiro grau na forma  $ax + b = 0$ . Faça um algoritmo que receba os valores de  $a$  e de  $b$  e mostre o resultado na tela para  $x=0$ ,  $x=1$ ,  $x=2$ ,  $x=3$ ,  $x=4$  e  $x=5$ .

Exercício 4: faça um algoritmo que receba a distância e o tempo de determinado veículo e calcule a velocidade média.

Exercício 5: faça um algoritmo que receba as 4 notas de um aluno e mostre na tela "APROVADO" se a média for maior ou igual a 6 e "REPROVADO" se for menor que 6.

## APÊNDICE B – LÂMINAS DE APRESENTAÇÃO



**SERVIÇO PÚBLICO FEDERAL**  
**UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ**

**LÓGICA DE PROGRAMAÇÃO COMO FERRAMENTA  
MATEMÁTICA PARA O ENSINO DE FÍSICA.**

Aluno: ROBERTO COSTA NUNES  
Orientador: Dr. José Elissandro de Andrade

Marabá - PA

### **Agenda**

- Apresentação – quem sou
- Introdução – pra que serve algoritmos?
- Lógica de Programação – VisualG

## **Apresentação – quem sou**

### **Formação:**

- Sistema de Informação (UFPA/UNIFESSPA);
- Ciências Naturais com habilitação em Física (UEPA);
- Mestrando em ensino de física (SBF & UNIFESSPA);
- Graduando em Engenharia Elétrica (UNIFESSPA);

### **Atuação:**

- SINOBRAS na área de TI de 2010 à 2015;
- Prof. de ciências na Prefeitura de Marabá desde 2015;
- Prof. Temporário no IFTO/Araguatins de Física (2016);

3

## **Introdução – Para que serve algoritmos?**

- Padrões e rotinas;

Hoje em dia, algoritmos computacionais estão presentes em quase tudo na nossa vida. Além dos tradicionais computadores e notebooks, muitos estão totalmente acostumados com o uso de aplicativos para smartphones e tablets.

O Curso de Algoritmo é a base necessária para quem quer aprender em linguagens famosas do mercado, como C, Java, PHP e muitas outras.

Aprenda as técnicas básicas para a construção de programas para dispositivos eletrônicos.

4

## Introdução – Para que serve algoritmos?

**Definição:** conjunto de passos finitos e organizados de forma lógica que, quando executados, resolvem determinado problema.



Algoritmo AtravessarRua  
Olhar para a direita  
Olhar para a esquerda  
Se estiver vindo carro  
    Não Atravesse  
    senão  
    Atravesse  
Fim-Se  
Fim-Algoritmo



Algoritmo AtravessarRua  
Olhar para a direita  
Olhar para a esquerda  
Se não estiver vindo carro  
    Atravesse  
    senão  
    Não Atravesse  
Fim-Se  
Fim-Algoritmo



Algoritmo AtravessarRua  
Atravesse  
Se estiver vindo carro  
    Olhar para a direita  
    senão  
    Olhar para a esquerda  
Fim-Se  
Não Atravesse  
Fim-Algoritmo

5

## Introdução – Para que serve algoritmos?

- Algoritmos Computacionais:
  - São passos a serem seguidos por um **módulo processador** e seus respectivos **usuários** que, quando executados na ordem correta, conseguem **realizar** determinada **tarefa**.
  - Rotinas executadas por algum processador.

6

## Introdução – Para que serve algoritmos?

Lógica de  
Programação



Linguagem de  
Programação



Sistema  
Completo

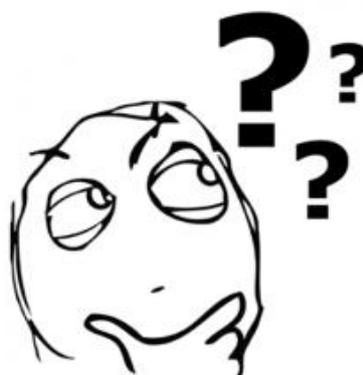


Lógica de programação: fluxograma e português (pseudo-código).

7

## Introdução – Para que serve algoritmos?

Afinal de contas, devo aprender algoritmos ou uma linguagem de programação?



Resposta: AS DUAS COISAS!

8



## APÊNDICE C – QUESTIONÁRIO INICIAL

### QUESTIONÁRIO INICIAL

Nome: \_\_\_\_\_

Turma: \_\_\_\_\_ Idade: \_\_\_\_\_

1. Quanto ao tipo de escola em que estudou:
  - a. Sempre estudei em escola pública;
  - b. Sempre estudei em escola particular;
  - c. Estudei parte em escola particular. \_\_\_\_\_
2. Como você avalia o seu aprendizado em **Física**?
  - a. Ruim. Não consigo aprender nada.
  - b. Mais ou menos. Consigo acompanhar o conteúdo com dificuldade.
  - c. Bom. Consigo acompanhar o conteúdo sem dificuldade alguma.
  - d. Muito Bom. Não tenho dificuldade alguma para aprender o conteúdo.
3. Como você avalia o seu aprendizado em **Matemática**?
  - a. Ruim. Não consigo aprender nada.
  - b. Mais ou menos. Consigo acompanhar o conteúdo com dificuldade.
  - c. Bom. Consigo acompanhar o conteúdo sem dificuldade alguma.
  - d. Muito Bom. Não tenho dificuldade alguma para aprender o conteúdo.
4. Como você avalia o seu conhecimento em **Programação de Computadores**?
  - a. Nenhum.
  - b. Conheço um pouco. Qual linguagem? \_\_\_\_\_
  - c. Bom. Qual linguagem? \_\_\_\_\_
5. Você já pensou em escrever programas de computadores?
  - a. Não.
  - b. Sim.
  - c. Já faço isso.
6. É comum utilizar a sala de informática da escola?
  - a. Não.
  - b. Algumas vezes.
  - c. Bastante. Para quê? \_\_\_\_\_
7. Quais as tecnologias que você mais tem acesso? (pode marcar mais de um)
  - a. Internet.
  - b. Computador.
  - c. Smartphone.
  - d. Tablet.
  - e. Outros: \_\_\_\_\_
8. Você acha que esse minicurso pode ser proveitoso para você?
  - a. Não. \_\_\_\_\_
  - b. Sim. \_\_\_\_\_
9. Certo móvel percorre uma distância de 12 metros em 3 segundo.
  - a. Qual a sua velocidade média?
  - b. Mantendo essa velocidade constante, que distância ele teria percorrido em 2 segundos? Em 4 segundos? Em 5 segundos? Em 6 segundos? E em 15 segundos?





## PRODUTO EDUCACIONAL

### USO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO COMO FERRAMENTA PARA O ENSINO DE FÍSICA

**ROBERTO COSTA NUNES**

Produto educacional aplicado e analisado durante a dissertação a dissertação de mestrado do Programa de Pós-Graduação da Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA) no Curso de Mestrado Nacional Profissional em Ensino de Física (MNPEF), como parte dos requisitos necessários à obtenção do título de Mestre em Ensino de Física.

**Orientador:** Prof. Dr. José Elisandro de Andrade

Marabá/PA  
Março de 2019

## RESUMO

### USO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO COMO FERRAMENTA PARA O ENSINO DE FÍSICA

ROBERTO COSTA NUNES

Orientador: Prof. Dr. JOSÉ ELISANDRO DE ANDRADE

Produto educacional aplicado e analisado durante a dissertação a dissertação de mestrado do Programa de Pós-Graduação da Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA) no Curso de Mestrado Nacional Profissional em Ensino de Física (MNPEF), como parte dos requisitos necessários à obtenção do título de Mestre em Ensino de Física.

Este trabalho apresenta uma metodologia que aborda o uso de lógica de programação no ensino de física, abordando em especial o ensino das fórmulas de cinemática do currículo de física no primeiro ano do ensino médio e demonstrar a possibilidade de trabalhar com outros conteúdos de física, como a termometria. Foi apresentado aos alunos os conceitos introdutórios de como construir algoritmos em português estruturado, e na sequência eles foram instruídos a tentarem construir códigos que implementasse as equações da velocidade e da distância em linhas de código. A proposta foi aplicada à uma turma da educação básica do primeiro ano do ensino médio da escola pública Oneide de Souza Tavares, localizada na cidade de Marabá-PA. Os resultados referentes à aplicação do projeto, com maiores detalhes da metodologia empregada estão contidos na dissertação.

**Palavras-chave:** Ensino de Física, Lógica de Programação, Cinemática.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## **PRODUTO EDUCACIONAL**

O produto educacional aqui apresentado trata-se de uma proposta de intervenção pedagógica baseado em uma metodologia de ensino com a inserção da linguagem de programação no cotidiano escolar do aluno, assim como sua rotina extraclasse. Como o público alvo desse projeto são os alunos da educação básica de ensino médio da rede pública de ensino, optou-se por utilizar apenas a construção de algoritmos em português estruturado no que é conhecido popularmente como português, por meio do programa aplicativo VisuAlg.

O programa VisuAlg é de livre uso, distribuição grátis e domínio público, usado para o ensino de lógica de programação. Para obter mais informações sobre o programa e baixar a versão mais recente do mesmo, pode ser acessado o site <<http://visualg3.com.br/>>.

A proposta de conteúdo relacionado a Física foi a introdução sobre o estudo do movimento, tradicionalmente trabalhado no primeiro ano do ensino médio. Os passos adiante são exatamente um guia de como o conteúdo foi abordado com os alunos nesse caso. De acordo com a realidade ou a necessidade de quem for trabalhar esse conteúdo, adaptações são totalmente possíveis, como por exemplo, trabalhar os assuntos relacionado à estática ou dinâmica com as suas fórmulas, ou trabalhar ainda equações de conversão de escalas termométricas e dilatação dos materiais, para citar alguns exemplos.

## **DEFINIÇÃO – O QUE SÃO ALGORITMOS?**

O termo algoritmo é definido por MANZANO & OLIVEIRA (2014) como um conjunto de passos finitos e organizados de forma lógica que, quando executados, resolvem determinado problema. Este termo está associado a computação e a Matemática, mas pode-se dizer que até mesmo nas atividades cotidianas algoritmos estão presentes. O conjunto de instruções para se fazer um bolo, por exemplo, seria um exemplo.

Basicamente um algoritmo possui uma rotina e deve ter alguma lógica. Imagine uma tarefa simples, como atravessar a rua. Quando se está diante dessa situação, para executar a tarefa “atravessar a rua”, algumas etapas devem ser realizadas, como: olhar para os lados; verificar se vem veículo; se não vier veículo, atravessar; senão, aguarde e verifique de novo. Esse conjunto de etapas, pode ser chamado de algoritmo, o qual é classificado como não computacional.

Para o exemplo anterior, a sequência de passos pode ser alterada, assim como aumentar ou diminuir a quantidade de etapas, porém, deve-se haver alguma lógica entre os passos. Se o algoritmo fosse construído onde a etapa de olhar os lados estivesse após a ação de atravessar a rua, isso poderia gerar um grande problema. Receitas, instruções de montagem e manuais de uso também podem ser chamados de algoritmos não computacionais. A mesma coisa ocorre com os algoritmos computacionais, pois os mesmos necessitam de lógica para estarem corretos.

## **LÓGICA DE PROGRAMAÇÃO – VISUALG**

O VisuAlg tem se mostrado uma importante ferramenta para o ensino de programação. Por meio dele é possível criar, editar, interpretar e executar algoritmos em português estruturado, que é basicamente uma representação de programa em linguagem de algoritmo, ou seja, em alto nível, a ponto que até mesmo pessoas leigas em programação, possam ler e entender o código.

A maior vantagem do VisuAlg é o uso dos comandos todos em português, assim os pré-requisitos para se ensinar lógica de programação são mínimos, sendo necessário apenas algumas habilidades para se manusear o computador. Ele é um programa de livre uso e distribuição, totalmente grátis, desenvolvido com fins didáticos. Na imagem abaixo é possível visualizar a tela inicial do VisuAlg.

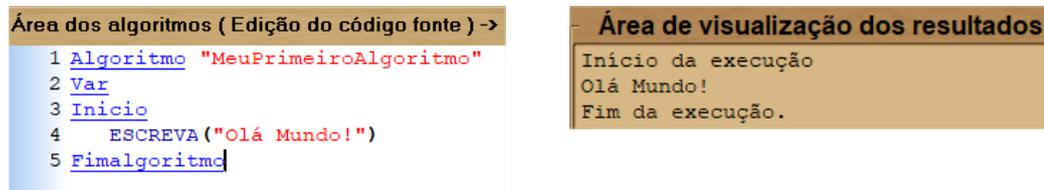
Na área “código do algoritmo” é onde o código do programa é inserido, sendo geralmente orientado pelas linhas. Na linha 1 é colocado o nome do programa, podendo ser acrescentado os dados de cabeçalho, como autor, instituição, data, etc, sendo que cada linha deve ser iniciada pela contra barra dupla (//), indicando que se trata apenas de comentário. Na linha 3 é o espaço destinado à declaração de variáveis. Na linha 5 é onde a parte executável do programa é iniciada, ou seja, o código que será executado será colocado nesse espaço, ou seja, entre as marcações “Início” e “Fimalgoritmo”. Na área das variáveis é possível observar os tipos e valores assumidos pelas variáveis durante a execução do programa, caso tenha sido declarada alguma variável na parte do código destinada à isso. Na área de visualização dos resultados é possível observar os resultados gerados pela execução do programa, ou seja, as saídas do programa.



1	ALGORITMO “NomeDoAlgoritmo”
2	VAR
3	INICIO
4	FIMALGORITMO

A primeira linha serve para identificar o algoritmo, sendo que o nome do mesmo fica no espaço entre as aspas duplas. A segunda linha é destinada declaração de variáveis. As variáveis receberão um tópico específico para serem tratadas aqui. A terceira e quarta linha marca o início e fim, respectivamente, como código que será de fato executado.

Para poder analisar o funcionamento do VisualG, pode ser utilizado o seguinte algoritmo que mostra uma breve saída de dados.



Aqui, é utilizado o comando “ESCREVA” para exibir na tela o que for colocado dentro do parêntese, no caso a mensagem “Olá Mundo!”. Pode ser utilizado também o comando “ESCREVAL”, com a diferença de que com este comando é pulado uma linha após a escrita na tela.

## Variáveis e tipos primitivos

As variáveis são espaços na memória do computador que podem receber determinados valores. Elas são utilizadas quando é necessário guardar uma certa informação para o computador pode manipula-la. Para criar uma variável, é preciso declarar ela primeiro, determinando o seu tipo.

Os tipos que serão trabalhados aqui, serão o inteiro, o real, o caractere e o lógico. O tipo inteiro e o real, guardam valores numéricos, tendo os seus nomes associados de forma intuitiva aos conjuntos numéricos aprendidos na Matemática. O tipo caractere serve para guardar valores literários, ou seja, letras e palavras. Por fim, o tipo lógico guarda apenas valor lógico, que pode ser verdadeiro ou falso.

Quando uma variável é declarada, ela pode ser utilizada no corpo do algoritmo, recebendo algum valor diretamente ou resultado de alguma operação. A passagem de um valor para a variável, é chamado de atribuição. Veja o exemplo abaixo do algoritmo “SeuNome” com o uso de uma variável e sua respectiva saída.

Área dos algoritmos ( Edição do código fonte ) ->

```

1 Algoritmo "SeuNome"
2 Var
3   nome: caractere
4 Início
5   nome <- "Roberto Costa Nunes"
6   ESCREVA (nome)
7 Fimalgoritmo

```

declaração

atribuição

Área de visualização dos resultados

```

Início da execução
Roberto Costa Nunes
Fim da execução.

```

### Comando LEIA – Entrada de dados

O comando LEIA é utilizado para receber uma entrada de dados. Pode-se dizer que é o contrário do comando ESCREVA, que é para saída. O comando LEIA é muito utilizado para interagir com o usuário, ou seja, por meio dele é possível receber alguma coisa que o usuário digite no teclado, que é a entrada padrão que será utilizada aqui. Observe abaixo como fica o algoritmo “SeuNome” com a utilização do comando LEIA.

Área dos algoritmos ( Edição do código fonte ) ->

```

1 Algoritmo "SeuNome"
2 Var
3   nome: caractere
4 Início
5   ESCREVA ("Digite o seu nome: ")
6   LEIA (nome)
7   ESCREVA ("Olá ", nome)
8 Fimalgoritmo

```

Área de visualização dos resultados

```

Início da execução
Digite o seu nome: Roberto Costa Nunes
Olá Roberto Costa Nunes
Fim da execução.

```

O comando LEIA está na linha 6. Quando o código é executado, o computador espera uma entrada de dados ao chegar no comando LEIA. Após digitar a entrada e apertar a tecla ENTER do teclado, o fluxo de execução continua. Nesse caso, foi digitado o nome “Roberto Costa Nunes”, o qual foi armazenado na variável *nome*. Na linha 7 foi utilizado o comando ESCREVA para mostrar a saída de dados, porém aqui foi utilizado o recurso de concatenação, para unir a palavra “Olá ” com o que estiver contido na variável *nome*.

Exercício 1: faça um algoritmo que receba dois valores inteiros e mostre a soma entre eles.

## Operações Aritméticas

É possível executar no VisualG as operações aritméticas aprendidas na educação básica. As operações e seus respectivos símbolos de operação são: soma (+), subtração (-), multiplicação (\*), divisão (/), divisão inteira (\), exponenciação (^) e resto (%). Para exemplificar as operações aritméticas, veja o exemplo do algoritmo “SomarDoisNumeros”, que é uma das soluções possíveis para o Exercício 1.

```
Área dos algoritmos ( Edição do código fonte ) -> Nome d
1 Algoritmo "SomarDoisNumeros"
2 Var
3   num1, num2, soma: inteiro
4 Inicio
5   ESCREVA("Digite o primeiro número: ")
6   LEIA(num1)
7   ESCREVA("Digite o segundo número: ")
8   LEIA(num2)
9   soma <- num1 + num2
10  ESCREVA("Soma = ", soma)
11 Fimalgoritmo
```

```
Área de visualização dos resultados
Início da execução
Digite o primeiro número: 2
Digite o segundo número: 5
Soma = 7
Fim da execução.
```

## Operadores Relacionais

Os operadores relacionais retornam um valor lógico (verdadeiro ou falso). Os operadores relacionais com seus respectivos símbolos de operação são: maior que (>), menor que (<), maior igual (>=), menor igual (<=), igual (=) e diferente (<>). Veja o algoritmo “OperaçãoRelacional” com a exemplificação da operação de maior que e menor que.

```
1 Algoritmo "OperaçãoRelacional"
2 Var
3   result: logico
4 Inicio
5   result <- 2>5
6   ESCREVAL("2 é maior que 5? ",result)
7   result <- 2<5
8   ESCREVAL("2 é menor que 5? ",result)
9 Fimalgoritmo
```

```
Área de visualização dos resultados
Início da execução
2 é maior que 5? FALSO
2 é menor que 5? VERDADEIRO
Fim da execução.
```

## Estruturas Condicionais

As estruturas condicionais são utilizadas para tomadas de decisões de acordo com uma condição verificada. Assim, pode-se garantir que determinada ação só será executada se

certa condição for satisfeita. Para demonstrar essa estrutura, veja o exemplo do algoritmo “EstruturaCondicional”.

```
1 Algoritmo "EstruturaCondicional"
2 Var
3   num1, num2: inteiro
4 Inicio
5   num1 <- 8
6   num2 <- 3
7   SE (num1 > num2) ENTAO
8     ESCREVA (num1, " é maior")
9   SENAO
10    ESCREVA (num2, " é maior")
11   FIMSE
12 Fimalgoritmo
```

Área de visualização dos resultados

```
Início da execução
8 é maior
Fim da execução.
```

## Estruturas de Repetição

Estruturas de repetição são utilizadas quando se deseja realizar uma determinada operação várias vezes. Pode-se utilizar repetições não controladas e repetições controladas. Na repetição não controlada, uma operação é repetida até que determinada condição seja satisfeita. Nesse caso, é necessário ter cuidado para que a condição seja satisfeita em algum momento, pois corre o risco ficar em execução sem nunca parar. Duas estruturas de repetição do tipo não controlada são possíveis no VisualG, o ENQUANTO e o REPITA. Veja o algoritmo “ContarAteCinco” com a estrutura ENQUANTO.

```
1 Algoritmo "ContarAteCinco"
2 Var
3   num, cont: inteiro
4 Inicio
5   num <- 5
6   cont <- 0
7   ENQUANTO (cont<=num) FAÇA
8     ESCREVAL (cont)
9     cont <- cont + 1
10  FIMENQUANTO
11 Fimalgoritmo
```

Área de visualização dos resultados

```
Início da execução
0
1
2
3
4
5
Fim da execução.
```

Analise agora o algoritmo que faz a mesma coisa, porém com a estrutura REPITA.

```

1 Algoritmo "ContarAteCinco"
2 Var
3   num, cont: inteiro
4 Inicio
5   num <- 5
6   cont <- 0
7   REPITA
8     ESCREVAL(cont)
9     cont <- cont + 1
10  ATE (cont>num)
11 Fimalgoritmo

```

```

Área de visualização dos resultados
Início da execução
0
1
2
3
4
5
Fim da execução.

```

A estrutura de repetição controlada é utilizada quando se quer fazer uma operação em quantidade de passo já conhecida. Essa estrutura é comumente chamada de PARA. Com ela, a operação somente será executada a quantidade de passos determinada. Veja o exemplo do algoritmo “ContarAteCinco” utilizando a estrutura de repetição PARA.

```

1 Algoritmo "ContarAteCinco"
2 Var
3   cont: inteiro
4 Inicio
5   PARA cont DE 0 ATE 5 PASSO 1 FAÇA
6     ESCREVAL(cont)
7   FIMPARA
8 Fimalgoritmo

```

```

Área de visualização dos resultados
Início da execução
0
1
2
3
4
5
Fim da execução.

```

Exercício 2: faça um algoritmo que receba 5 números inteiros, mostre como resultado a soma deles e a média aritmética.

Exercício 3: considere a equação do primeiro grau na forma  $ax + b = c$ . Faça um algoritmo que receba os valores de a e de b e mostre o resultado na tela para o  $x=0$ ,  $x=1$ ,  $x=2$ ,  $x=3$ ,  $x=4$  e  $x=5$ .

Exercício 4: faça um algoritmo que receba a distância e o tempo de determinado veículo e calcule a velocidade média.

Exercício 5: faça um algoritmo que receba as 4 notas de um aluno e mostre na tela “APROVADO” se a média for maior ou igual a 6 e “REPROVADO” se for menor que 6.

## SOLUÇÕES PARA OS EXERCÍCIOS

### Exercício 1

```
Algoritmo "exercicio1"  
Var  
    numero1, numero2: inteiro  
Inicio  
    ESCREVA("Número 1: ")  
    LEIA(numero1)  
    ESCREVA("Número 2: ")  
    LEIA(numero2)  
    ESCREVA("Resultado: ", numero1 + numero2)  
Fimalgoritmo
```

### Exercício 2

```
Algoritmo "exercicio2"  
Var  
    num, soma, cont: inteiro  
Inicio  
    soma <- 0  
    PARA cont DE 1 ATÉ 5 FAÇA  
        ESCREVA("Número ", cont, ":")  
        LEIA(num)  
        soma <- soma + num  
    FIMPARA  
    ESCREVAL("Soma: ", soma)  
    ESCREVA("Média: ", soma/cont)  
Fimalgoritmo
```

Outra forma para se resolver o mesmo problema sem utilizar estrutura de repetição pode ser feita da seguinte maneira:

Algoritmo "exercicio2"

Var

n1, n2, n3, n4, n5: inteiro

Inicio

ESCREVA("Número 1: ")

LEIA(n1)

ESCREVA("Número 2: ")

LEIA(n2)

ESCREVA("Número 3: ")

LEIA(n3)

ESCREVA("Número 4: ")

LEIA(n4)

ESCREVA("Número 5: ")

LEIA(n5)

ESCREVAL("Soma: ", n1 + n2 + n3 + n4 + n5)

ESCREVA("Média: ", (n1 + n2 + n3 + n4 + n5)/5)

Fimalgoritmo

### Exercício 3

Algoritmo "exercicio3"

Var

a, b: inteiro

Inicio

ESCREVA("a: ")

LEIA(a)

ESCREVA("b: ")

LEIA(b)

ESCREVAL("x(1)= ", a\*1+b)

ESCREVAL("x(2)= ", a\*2+b)

ESCREVAL("x(3)= ", a\*3+b)

ESCREVAL("x(4)= ", a\*4+b)

ESCREVAL("x(5)= ", a\*5+b)

Fimalgoritmo

#### Exercício 4

Algoritmo "exercicio4"

Var

distancia, t: inteiro

Inicio

ESCREVA("Distancia: ")

LEIA(distancia)

ESCREVA("Tempo: ")

LEIA(t)

ESCREVAL("Velocidade Média: ", distancia/t)

Fimalgoritmo

#### Exercício 5

Algoritmo "semnome"

Var

cont: inteiro

nota, soma, media: real

Inicio

PARA cont DE 1 ATE 4 FAÇA

ESCREVA("Digite a ",cont,"º nota: ")

LEIA(nota)

soma <- soma+nota

FIMPARA

media <- soma/cont

SE (media >= 6) ENTÃO

ESCREVAL("APROVADO!!!")

SENÃO

ESCREVAL("REPROVADO!!!")

FIMSE

Fimalgoritmo

## TRABALHANDO UM POUCO MAIS

O tópico anterior serve como guia para introduzir aos alunos os princípios básicos de programação, sendo plenamente possível ampliar o leque de possibilidades de assuntos que podem ser trabalhados. Esse tipo de codificação possibilita exercitar bastante o funcionamento de fórmulas baseadas em funções e equações, permitindo estressar o problema com uma grande carga de testes, verificando diversas possibilidades. Quando o aluno constrói o seu próprio código, existe maiores chances dele não apenas decorar a fórmula, mas aprender o seu funcionamento, manipulação e reter assim o conhecimento.

### Um pouco mais de cinemática

Tradicionalmente os conteúdos que abordam cinemática geralmente são trabalhados na educação básica nos anos finais do ensino fundamental II na disciplina de ciências, e ainda no primeiro ano do ensino médio na disciplina de física. Porém estes conceitos são requeridos em vários momentos da vida do aluno, tanto em assuntos didáticos, como vestibulares e outros exames de seleção, como em situações práticas cotidianas.

Como visto na Figura 16, no algoritmo para calcular a velocidade média é utilizado duas variáveis (distância e tempo), que basicamente representa a equação 1.

$$velocidade\ média = \frac{distância}{tempo} \quad (1)$$

Nos casos de manipulação da fórmula, é possível encontrar a distância com base na velocidade média e com o tempo. Para isso, é utilizado a equação 2.

$$distância = velocidade\ média \times tempo \quad (2)$$

A adaptação no algoritmo pode ser realizada com alguns ajustes no exemplo mostrado anteriormente para o cálculo da velocidade média. Para demonstrar esse ajuste, veja a figura abaixo.

```

1 Algoritmo "calcularDistância"
2
3 Var
4   vm, t: inteiro
5 Inicio
6   ESCREVA ("Velocidade média: ")
7   LEIA (vm)
8   ESCREVA ("Tempo: ")
9   LEIA (t)
10  ESCREVAL ("A distância percorrida é de: ", vm*t)
11 Fimalgoritmo

```

Manipulando a fórmula da velocidade média também é possível encontrar o tempo, agora com base na velocidade média e na distância, tendo assim a fórmula apresentada pela equação 3.

$$tempo = \frac{distância}{velocidade\ média} \quad (3)$$

O algoritmo que aborda essa equação pode ser obtido com apenas alguns ajustes do anterior. Na figura abaixo é possível observar um exemplo de solução para esse caso com apenas algumas adaptações dos exemplos anteriores.

```

1 Algoritmo "calcularTempo"
2
3 Var
4   vm, distancia: inteiro
5 Inicio
6   ESCREVA ("Velocidade média: ")
7   LEIA (vm)
8   ESCREVA ("Distância: ")
9   LEIA (distancia)
10  ESCREVAL ("O tempo é de: ", distancia/vm)
11 Fimalgoritmo

```

Esse conteúdo de velocidade média, geralmente é a parte introdutória dos conceitos de cinemática. As fórmulas e algoritmos podem ser facilmente adaptados com apenas mudanças de nomenclaturas para se trabalhar o Movimento Uniforme (MU), já que nesse tipo de movimento a velocidade instantânea é iguala a velocidade média. Ainda dentro dos assuntos introdutórios de cinemática, pode ser trabalhado as conversões de unidade de medida.

## Trabalhando com aceleração

Ao trabalhar o Movimento Uniformemente Variado (MUV) é apresentado o conceito de aceleração, que pode ser compreendida como a taxa de variação da velocidade em relação ao tempo. No MUV, a aceleração instantânea é igual a aceleração média, podendo ser representada pela equação 4.

$$a = \frac{\Delta v}{\Delta t} \quad (4)$$

Na equação 4,  $a$  é a aceleração,  $\Delta v$  é a variação da velocidade e  $\Delta t$  é a variação do tempo.

No caso de MUV o cálculo da velocidade média e da velocidade instantânea são diferentes do MU. Para calcular a velocidade instantânea, é utilizado a equação 5.

$$v = v_0 + a * t \quad (5)$$

Na equação 5,  $v$  é a velocidade final (velocidade no instante  $t$ ),  $v_0$  é a velocidade inicial (velocidade no instante  $t = 0$ ),  $a$  é a aceleração e  $t$  é o tempo.

Seguindo a evolução gradativa dos estudos de programação, assim como de física, é possível escrever o código com apenas algumas adaptações dos exemplos anteriores, ou partindo de outra abordagem. Agora, será necessário trabalhar com mais variáveis. Um Exemplo de como o código pode ser escrito, está detalhado na figura abaixo.

```
1 Algoritmo "MRU-Velocidade"
2
3 Var
4   v1, a, t: inteiro
5 Inicio
6   ESCREVA ("Velocidade inicial: ")
7   LEIA (v1)
8   ESCREVA ("Aceleração: ")
9   LEIA (a)
10  ESCREVA ("Tempo: ")
11  LEIA (t)
12  ESCREVAL ("A velocidade final é de: ", v1+a*t)
13 Fimalgoritmo
```

Outra fórmula muito útil e bastante requerida nos estudos de MRU, é a função do espaço (ou posição). Geralmente ela é representada pela equação 6 mostrada abaixo.

$$S = S_0 + v_0 * t + \frac{at^2}{2} \quad (6)$$

Na equação 6,  $S$  é a posição final (ou no instante  $t$ ),  $S_0$  é a posição inicial (ou no instante  $t = 0$ ),  $v_0$  é a velocidade inicial (ou no instante  $t = 0$ ),  $a$  é a aceleração e  $t$  é o tempo.

Escrever um algoritmo que trabalhe com essa fórmula é bastante interessante, principalmente pela quantidade de variáveis necessárias. Como sempre, a construção do código é livre, e cada pessoa pode desenvolver linhas de códigos com diferentes abordagens. Seguindo o processo gradativo dos exemplos anteriores, a próxima imagem mostra um exemplo de como essa equação pode ser escrita em algoritmo.

```

1 Algoritmo "MRU-Espaço"
2
3 Var
4   s1, v1, a, t: inteiro
5 Inicio
6   ESCREVA("Posição inicial: ")
7   LEIA(s1)
8   ESCREVA("Velocidade inicial: ")
9   LEIA(v1)
10  ESCREVA("Aceleração: ")
11  LEIA(a)
12  ESCREVA("Tempo: ")
13  LEIA(t)
14  ESCREVAL("A posição final é de: ", s1+v1*t+(a*t*t)/2)
15 Fimalgoritmo

```

Os exemplos elencados acima demonstram apenas algumas possibilidades de como o assunto pode ser tratado na abordagem de cinemática. Todavia não se resume exclusivamente a isso. A utilização de construção de códigos de algoritmos ou de programas em linguagens específicas de programação é plenamente possível em vários assuntos da física, como por exemplo, a dinâmica, a estática, energia mecânica e escalas de temperatura.

Uma observação a ser feita nos exemplos apresentados até aqui, é com relação a complexidade dos algoritmos construídos. De acordo com a habilidade de cada aluno, é possível ser construído códigos mais elaborados, com uso de recursos de laços de repetição ou estruturas condicionais para tomadas de decisão. Optou-se, obviamente, por não utilizar nos exemplos tais recursos, uma vez que poderia tornar o entendimento restrito apenas àqueles que tivessem tal domínio. Os códigos utilizam apenas as estruturas mais básicas possíveis,

uma vez que a programação em si é a ferramenta de apoio para trabalhar os conteúdos de físicas, e não a finalidade última do trabalho.

### Trabalhando com escalas termométricas

Ao trabalhar as escalas termométricas dentro do currículo de física, é mais comum ser feita a relação entre as escalas Celsius, Fahrenheit e Kelvin. As escalas Celsius e Fahrenheit são graduadas, enquanto a Kelvin é absoluta, não tendo assim valores negativos. A relação entre as três escalas pode ser representada por uma única expressão:

$$\frac{T_c}{5} = \frac{T_f - 32}{9} = \frac{T_k - 273}{5} \quad (7)$$

Na equação 7,  $T_c$  é a temperatura em graus Celsius ( $^{\circ}\text{C}$ ),  $T_f$  é a temperatura em graus Fahrenheit e  $T_k$  a temperatura em Kelvin. Assim, a relação entre duas escalas pode ser obtida por meios dessa expressão. Por exemplo, se for necessário fazer uma equivalência de temperatura entre as escalas Celsius e Fahrenheit, têm-se a equação 8.

$$\frac{T_c}{5} = \frac{T_f - 32}{9} \quad (8)$$

Da equação 8, é possível então converter uma temperatura da escala Fahrenheit para a escala Celsius, como mostrado na equação 9.

$$T_c = 5 * \left( \frac{T_f - 32}{9} \right) \quad (9)$$

De forma semelhante é possível converter uma temperatura da escala Celsius para Fahrenheit, de acordo com a equação 10.

$$T_f = \left( \frac{T_c * 9}{5} \right) + 32 \quad (10)$$

Para construir um algoritmo que faça a conversão mostrada na equação 10 é necessário ter atenção na hora de escrever a expressão em linha de código por causa da preferência entre as operações aritméticas. Um exemplo de código que implemente essa equação pode ser observado na figura abaixo.

```

1 Algoritmo "escalaTermométrica"
2 // Relação temométrica:  $T_c/5 = (T_f - 32)/9$ 
3 Var
4 Tc, Tf : Real
5 Inicio
6   Escreva ("Digite o valor da temperatura em Celsius: ")
7   Leia (Tc)
8   Tf <- ((Tc/5)*9)+32
9   Escreva ("A temperatura em Fahrenheit é: ", Tf)
10 Fimalgoritmo

```

De modo geral, a escrita de algoritmos pode ser exercitada de diversas maneiras, e o uso de português, com essas estruturas mais definidas são exatamente a entrada para a construção de linhas de códigos em uma linguagem de programação mais específica, só que aqui com uma complexidade bem menor e mais inteligível para estudantes da educação básica.