



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
BACHARELADO EM ENGENHARIA DE SOFTWARE

LEONARDO BRENDON GOMES NASCIMENTO

**UMA PROPOSTA DE METODOLOGIA DE ENSINO DE PROGRAMAÇÃO E
RACIOCÍNIO LÓGICO COMPUTACIONAL PARA A COMUNIDADE NÃO
ACADÊMICA DE COMPUTAÇÃO**

**QUIXADÁ
2016**

LEONARDO BRENDO GOMES NASCIMENTO

**UMA PROPOSTA DE METODOLOGIA DE ENSINO DE PROGRAMAÇÃO E
RACIOCÍNIO LÓGICO COMPUTACIONAL PARA A COMUNIDADE NÃO
ACADÊMICA DE COMPUTAÇÃO**

Monografia apresentada ao Curso de Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Orientadora: Prof.^a. Dra. Ingrid Teixeira Monteiro

**QUIXADÁ
2016**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- N196 Nascimento, Leonardo Brendo Gomes.
Uma proposta de metodologia de ensino de programação e raciocínio lógico computacional para a comunidade não acadêmica de computação. / Leonardo Brendo Gomes Nascimento. – 2016.
51 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,
Curso de Engenharia de Software, Quixadá, 2016.
Orientação: Profa. Dra. Ingrid Teixeira Monteiro.
1. Pensamento Computacional. 2. Processo Ensino-Aprendizagem. 3. Algoritmos. I. Título.

CDD 005.1

LEONARDO BRENDO GOMES NASCIMENTO

**UMA PROPOSTA DE METODOLOGIA DE ENSINO DE PROGRAMAÇÃO E
RACIOCÍNIO LÓGICO COMPUTACIONAL PARA A COMUNIDADE NÃO
ACADÊMICA DE COMPUTAÇÃO**

Monografia apresentada ao Curso de Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel. Área de concentração: Computação.

Aprovado em: _____ / dezembro/ 2016.

BANCA EXAMINADORA

Prof^ª. Dra. Ingrid Teixeira Monteiro (Orientadora)
Universidade Federal do Ceará-UFC

Prof. Me. Valdemir Pereira de Queiroz Neto
Universidade Federal do Ceará-UFC

Prof. Me. Cícero Bandeira L. Filho
Universidade Estadual do Ceará-UECE

A minha mãe, à minha noiva, a meu irmão
e a meus amigos Josué e François.

AGRADECIMENTOS

Porque dEle e por Ele, e para Ele, são todas as coisas; glória, pois, a Ele eternamente. Amém. (Rm 11:36). Agradeço ao criador do universo por ter me ajudado até aqui e me ter feito saber que quando penso que não posso mais, consigo ainda, tirar forças na minha fraqueza e seguir em frente. Sendo totalmente demérito meu algum favor de sua parte.

Agradeço a minha mãe Silvana, por sempre estar comigo me ajudando e sendo um referencial de resiliência e fortaleza, por mostrar que não devemos nos curvar diante das dificuldades da vida.

Agradeço à minha noiva Kamila que sempre está comigo, nas adversidades e nas bonanças, sendo um referencial feminino e que compartilha de perto meus anseios e receios, sendo uma parte precípua em minhas conquistas.

Agradeço ao meu irmão Leandro que desde pequeno me motiva a ir à faculdade e enfrentar seus desafios.

Agradeço à professora Ingrid Monteiro, por ter acreditado em meu trabalho, por sua muita paciência, dedicação, disponibilidade, conselhos, que sem os quais este trabalho não seria possível de se realizar.

Agradeço à professora Tânia Pinheiro, por ter ajudado de perto esse trabalho, por ter esclarecido sobre o conhecimento educacional, conhecimento o qual deve permear todo o ser.

Agradeço aos professores da Banca Examinadora, Valdemir Queiroz e Cícero Bandeira, por suas sugestões de melhorias, críticas, carisma e acréscimos para meu trabalho.

Agradeço aos participantes da aplicação da metodologia proposta e pela salutar aceitação da escola E.E.F. Deputado Flávio Portela Marcílio.

Agradeço aos meus amigos Josué e Franços. Josué por ter desempenhado um papel semelhante ao paterno, me ajudando em todos os momentos, sendo possível por causa deste a minha caminhada na UFC Campus Quixadá. Franços por ter me ajudado no sentido acadêmico, sempre disposto a me ajudar, inclusive nas madrugadas.

“Como é feliz o homem que acha a sabedoria, o homem que obtém o entendimento, pois a sabedoria é muito mais proveitosa do que a prata e rende mais do que o ouro.” (Provérbio de Salomão).

RESUMO

O ensino de programação é um dos principais problemas que está relacionado à evasão de alunos dos cursos de computação, pois nem todos conseguem acompanhar e entender programação sedimentada à primeira vista. É unânime nas instituições de ensino sobre a grande dificuldade em entender programação para novatos da área. Esta é uma das disciplinas em que os alunos mais reprovam em cursos de TI. Pensando nesse problema da computação, que é o ensino de programação, fazem-se necessárias novas maneiras de propagação do conhecimento e aprendizagem do conteúdo, como a proposta metodológica aqui apresentada. Essa proposta metodológica é baseada em níveis incrementais de conteúdo, em que o nível posterior é acrescido de um conhecimento do nível anterior, facilitando e dando uma base para prosseguir em seu entendimento de programação. A metodologia foi aplicada em uma escola de ensino fundamental, através de atividades práticas, questionários e entrevistas. Ao final, foi feito um levantamento dos principais problemas e sugestões de melhoria. Os resultados obtidos demonstraram que a proposta metodológica conseguiu fazer os alunos associarem e reutilizarem conhecimento de um nível anterior para o nível posterior.

Palavras-chave: Pensamento Computacional, Programação Lúdica, Algoritmos.

ABSTRACT

Teaching programming is one of the main problems related to computer science students drop out because not everyone can follow and understand programming at first view. It is unanimous the great difficulty in understanding programming for newcomers to the area. Also, this is one of the subjects that most disapprove students in IT undergraduate courses. Thinking about this problem, which is the programming education, new ways of spreading knowledge and learning content are necessary, thus a new methodological proposal is presented. This methodological proposal is based on incremental levels, wherein the higher level is increased by a knowledge of the previous level, facilitating and providing a basis for further programming in its understanding. The proposed methodology was applied in a middle school, through practical activities, surveys and interviews. At the end we made a list of the main problems and suggestions for improvement. The results showed that the proposed methodology could make students associate and reuse knowledge from a previous level to the subsequent level.

Keywords: Computational Thinking, Recreational Programming, Algorithms.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Tela principal da ferramenta Scratch..... | 18 |
| Figura 2 – Demonstração de um problema matemático no Scratch..... | 19 |
| Figura 3 – Algoritmo simples na linguagem de programação C..... | 20 |

LISTA DE GRÁFICOS

| | |
|---|----|
| Gráfico 1 – Experiência de uso de dispositivo..... | 25 |
| Gráfico 2 – Frequência de uso de dispositivo..... | 26 |
| Gráfico 3 – Utilização de tecnologia para estudar nas disciplinas/matérias..... | 27 |
| Gráfico 4 – Formas de ajuda da tecnologia nos estudos..... | 27 |
| Gráfico 5 – Interesse em tecnologias..... | 28 |
| Gráfico 6 – Conhecimento sobre programação ou códigos de computador..... | 29 |
| Gráfico 7 – Conhecimento sobre linguagens e/ou ferramenta de programação..... | 30 |
| Gráfico 8 – Conhecidos que trabalham com programação..... | 30 |
| Gráfico 9 – Filmes/séries sobre programação..... | 31 |
| Gráfico 10 – Conhecimento de nomes da área de computação..... | 31 |
| Gráfico 11 – Expectativas em relação ao curso..... | 32 |
| Gráfico 12 – Interesse em estudar/seguir na área de tecnologia..... | 33 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Notas dadas às disciplinas..... | 28 |
|--|----|

SUMÁRIO

| | |
|--|----|
| SUMÁRIO..... | 11 |
| 1 INTRODUÇÃO..... | 12 |
| 2 TRABALHOS RELACIONADOS..... | 14 |
| 2.1 Aplicação do pseudocódigo nos ambientes AMBAP e VisualPseudo..... | 14 |
| 2.2 Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica..... | 14 |
| 2.3 Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação..... | 15 |
| 4 FUNDAMENTAÇÃO TEÓRICA..... | 15 |
| 4.1 Pensamento Computacional..... | 16 |
| 4.2 Programação Lúdica..... | 17 |
| 4.3 Algoritmos..... | 19 |
| 5 PROCEDIMENTOS METODOLÓGICOS..... | 21 |
| 5.1 Definição da proposta metodológica..... | 21 |
| 5.2 Planejar a aplicação da metodologia proposta..... | 22 |
| 5.3 Executar a aplicação da metodologia proposta..... | 22 |
| 5.4 Análise descritiva da aplicação da metodologia proposta..... | 22 |
| 6 METODOLOGIA DE ENSINO PCPL&A..... | 22 |
| 6.1 Nível PC da tríade PCPL&A..... | 22 |
| 6.2 Nível PL da tríade PCPL&A..... | 23 |
| 6.3 Nível A da tríade PCPL&A..... | 24 |
| 7 RESULTADOS..... | 25 |
| 6.1 Resultados do questionário de percepção de conhecimento..... | 25 |
| 6.2 Resultados do nível PC da tríade PCPL&A..... | 34 |
| 6.3 Resultados do nível PL da tríade PCPL&A..... | 36 |
| 6.4 Resultados do nível A da tríade PCPL&A..... | 38 |
| 6.5 Levantamento dos principais problemas..... | 40 |
| 6.6 Discussão..... | 41 |
| 8 CONSIDERAÇÕES FINAIS..... | 43 |
| REFERÊNCIAS..... | 45 |
| APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO APRESENTADO AOS PARTICIPANTES DA APLICAÇÃO DA METODOLOGIA..... | 47 |
| APÊNDICE B – QUESTIONÁRIO PERCEPÇÃO DE CONHECIMENTO..... | 48 |
| APÊNDICE C – QUESTIONÁRIO SOBRE O NÍVEL PC DA TRÍADE PCPL&A..... | 50 |
| APÊNDICE D – QUESTIONÁRIO SOBRE O NÍVEL PL DA TRÍADE PCPL&A..... | 50 |
| APÊNDICE E – QUESTIONÁRIO SOBRE O NÍVEL A DA TRÍADE PCPL&A..... | 50 |

1 INTRODUÇÃO

O ensino de programação é um dos grandes desafios da educação em computação (Carpersen e Kölling, 2009). Para quem está tendo o primeiro contato com ela, nem sempre é algo agradável ou simples de se entender, porque demanda boa capacidade de abstração e lógica. O primeiro contato de uma pessoa que não possui conhecimento na área com a linguagem de programação pura pode tornar a experiência frustrante ou triste. Programar é uma atividade mental que exige bastante de quem está pretendendo aprender. Muraret et al (2009) afirmam que, no período entre 2005 e 2009, houve uma redução de 20% de estudantes de ciência da computação na França e apontam o conteúdo de programação como uns dos principais fatores de desistência.

Um levantamento realizado pelo Sindicato das Entidades Mantenedoras de Estabelecimentos de Ensino Superior de São Paulo (SEMESP, 2014) revelou que os cursos da área de tecnologia da informação (TI) são os que possuem a maior taxa de desistência de alunos. Há um elevado nível de frustração e evasão dos estudantes. Segundo o SEMESP (2014), a cada três alunos que entram no curso de Sistemas da Informação, apenas um recebe o diploma. Para cada quatro alunos que entram no curso de Ciência da Computação, apenas um chega ao término do curso.

Vendo o expressivo número de desistências da área da computação, este trabalho busca uma nova maneira de ensinar programação, que envolve o pensamento computacional (WING, 2006) e a programação lúdica (MALAN; LEITNER, 2007). Parte-se do pressuposto de que o uso conjunto desses dois conceitos possibilita um meio mais simples de aprender programação, utilizando apenas conceitos lógicos básicos e inclusivos, a fim de que o estudante desenvolva suas habilidades cognitivas e melhore suas aptidões lógicas. A partir do uso conjunto desses conceitos, propomos a tríade denominada PCPL&A, que significa: PC é o Pensamento Computacional, PL é a Programação Lúdica e A é o Algoritmo.

O pensamento computacional é um conceito que usa meios lógicos para resolver problemas, não usa apenas códigos e sim conhecimentos diários para dissolver grandes problemas em pequenos. Com essa técnica, o estudante usa conhecimento de algoritmos sem perceber. Para Mor e Noss (2008), o estudante consegue resolver problemas seguindo uma série de passos, deixando-o assim familiarizado com a lógica e a inferência de determinadas regularidades de números, tais como sequência de Fibonacci e números primos. O benefício dessa técnica é que ela alcança principalmente as pessoas leigas no assunto, ajudando-as a

desenvolver sua mentalidade para abstração de dados simples, permitindo certa racionalização, sem necessariamente utilizar uma linguagem de programação tradicional.

Já em relação à programação lúdica, busca-se utilizar alguns ambientes de programação mais simples. O ambiente de programação deste tipo adotados para esse trabalho foi o Scratch¹. A técnica de programação lúdica já usa um pouco de programação, porém de forma bem rudimentar, que não exige do estudante um grande esforço cognitivo, pois as funções são simples de identificar e utilizar. O uso da palavra “lúdica” refere-se ao jeito simples e divertido com que os inventores desses programas os projetaram para torná-los envolventes. Para Setti (2009), é necessário sair do tradicional, em que os programas normalmente são apenas editores de textos e códigos que imprimem resultados em um console ou em um terminal sem expressão ou motivação.

O ensino de programação usando essas técnicas visa, segundo Setti (2009), fomentar um raciocínio lógico para pessoas que não possuem conhecimentos técnicos de programação, pessoas que estejam além das fronteiras da academia da computação. Uma consequência dessa abordagem é que a utilização dessas técnicas possivelmente cativará mais estudantes para à área da computação. Nossa metodologia de ensino propõe oficinas e grupos de estudos, em que, inicialmente, os alunos exploram apenas o pensamento computacional. Assim, os estudantes conseguem perceber os modelos de resolução, e, quando estão mais aptos, passam a ter contato com a programação lúdica. Somente depois, ao unificar o pensamento computacional e a programação lúdica, é que a programação tradicional é apresentada, pois os alunos já têm a necessária base teórica e prática.

A metodologia proposta busca, por meio da tríade PCPL&A, dar um suporte ao ensino de programação para pessoas que não possuem conhecimento na área da computação. Usando a tríade PCPL&A, são fornecidos conhecimentos teóricos e práticos, possibilitando ao aluno conhecimento prévio de computação antes de chegar à programação sedimentada.

A metodologia proposta tem o objetivo geral de sistematizar um novo meio de ensino de programação e raciocínio lógico e fazer a inclusão estudantil de pessoas leigas para a área da computação. Os objetivos específicos da metodologia proposta distribuem-se em definição, planejamento, execução e análise descritiva da aplicação da metodologia proposta.

O nosso trabalho possui a seguinte estrutura: seção 1: introdução, seção 2: trabalhos relacionados, seção 3: Objetivos, seção 4: fundamentação teórica, seção 5: procedimentos metodológicos, seção 6: metodologia de ensino PCPL&A, seção 7: resultados e seção 8: considerações finais.

¹ <https://scratch.mit.edu>

2 TRABALHOS RELACIONADOS

Nesta seção serão apresentados os trabalhos relacionados que auxiliaram na escolha do tema de pesquisa e que servem como base para este estudo. Foram selecionados quatro trabalhos, apresentados nas subseções a seguir.

2.1 Aplicação do pseudocódigo nos ambientes AMBAP e VisualPseudo

Na literatura, estão descritas variadas abordagens para o ensino de programação. Uma das técnicas é o uso do pseudocódigo, uma maneira informal de usar o código computacional sem vinculações à sintaxe de uma linguagem tradicional, objetivando assim a facilidade de entendimento para qualquer pessoa. O uso de pseudocódigo torna o programa mais trivial e menor, além do que o pseudocódigo auxilia na capacidade de abstração e lógica, ficando assim mais fácil de compreender o programa. Nesta categoria, Almeida et al (2004) descrevem o ambiente AMBAP, e Silva e Favero (2005) descrevem o ambiente VisualPseudo. O pseudocódigo possui uma desvantagem muito grave que é a falta de padronização. Essa desvantagem é consequência de o pseudocódigo ser muito genérico.

Os trabalhos de Silva e Favero (2005) e Almeida et al (2004) abordam o uso do pseudocódigo em ambientes integrados para facilitar e motivar o ensino de algoritmos. Assemelha-se a esse estudo por focar no ensino de algoritmos e por buscar novos meios de disseminação do conhecimento de programação. Diferencia-se pelo fato de nosso estudo não se limitar apenas a pseudocódigo, pois nossa metodologia propõe três níveis sequenciais, em que a parte de algoritmos já utiliza linguagem de programação (e não pseudocódigo).

2.2 Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica

Em outra técnica de ensino de programação, Barcelos e Silveira (2012) discutem a identificação de características cognitivas e de um mapeamento de competências nos padrões curriculares brasileiros com atividades usando o pensamento computacional. A análise de Barcelos e Silveira é muito inclinada para relações matemáticas e pensamento computacional. Apesar de ser uma excelente ideia a junção do pensamento computacional e relações matemáticas, os autores concentram-se no pensamento computacional, mencionando vagamente em programação lúdica.

O trabalho de Barcelos e Silveira (2012) usa o pensamento computacional como uma forma de perceber características cognitivas, habilidades relacionadas à abstração lógica

e capacidade de conseguir fazer a decomposição de problemas. Assemelha-se a esse estudo por usar o pensamento computacional, conceito usado em um dos níveis da metodologia proposta e por definir habilidades lógico-cognitivas como um subconjunto de competência para dar uma capacidade de abstração. Diferencia-se pelo fato de nosso estudo não se limitar apenas ao pensamento computacional e educação matemática, já que nossa abordagem possui três níveis, com o uso inicial do pensamento computacional.

2.3 Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação

Aureliano e Tedesco (2012) apresentam um estudo preliminar que tem a finalidade de avaliar o ambiente introdutório de ensino Scratch como uma abordagem alternativa para o processo de ensino-aprendizagem de programação. É um trabalho que faz uma análise de formas de ensinar programação, tanto no ensino tradicional quanto no ensino com programação lúdica. Os autores apenas confirmam que o ensino de programação no estilo tradicional ainda possui resultados nem sempre satisfatórios. Segundo Aureliano e Tedesco (2012), usando-se novos meios de ensino de programação, podem-se ter resultados melhores do que o ensino tradicional. Porém, Aureliano e Tedesco (2012) não detalharam a abordagem aplicada e nem apresentam uma descrição de como fora aplicada a avaliação para coletar os resultados.

O trabalho de Aureliano e Tedesco (2012) assemelha-se a esse estudo por usar a ferramenta Scratch para o ensino de programação, por sua simplicidade de utilização e por uma nova maneira de ensinar programação, saindo um pouco do ensino tradicional. Diferencia-se pelo fato de nosso estudo não se limitar apenas à utilização da ferramenta Scratch e sua avaliação, já que nossa abordagem possui três níveis, com a utilização da ferramenta Scratch no segundo nível.

4 FUNDAMENTAÇÃO TEÓRICA

O uso de novos meios de ensinar programação pode auxiliar no aprendizado e no raciocínio lógico computacional, além de apoiar a obtenção de resultados diferentes daqueles oriundos do ensino tradicional de programação. O ensino tradicional de programação refere-se aos meios adotados pelos cursos superiores para o ensino de programação. Esses novos meios envolvem a adoção de ferramentas simples e autoexplicativas que facilitam a inclusão e a motivação dos participantes.

Nesta seção, serão apresentados os conceitos que serviram de base para a definição e a aplicação da metodologia didática apresentada neste trabalho, a saber: pensamento computacional, programação lúdica e algoritmos.

2.4 Pensamento Computacional

O pensamento computacional é um conceito, para Wing (2006), baseado em várias competências cognitivas e lógicas, sem utilizar algoritmos propriamente ditos em alguma linguagem de programação. O pensamento computacional possui as seguintes características:

- *Conceituar ao invés de programar*: remete-se à utilização do pensamento computacional para dividir um problema grande e difícil em vários menores e mais simples de resolver. Em outras palavras, não é apenas necessário o conhecimento de uma sintaxe de determinada linguagem de programação ou técnicas de algoritmos, é imprescindível uma abstração de alto nível.
- *É uma habilidade fundamental e não utilitária*: é uma habilidade que todos os estudantes deveriam desenvolver, pois não se trata de algo mecânico ou automático, e sim de algo inerente em nossa sociedade, que é a onipresença dos computadores.
- *É a maneira como as pessoas pensam, e não os computadores*: é o processo pelo qual as pessoas irão resolver os problemas e não simplesmente a utilização de uma máquina com inteligência artificial reprogramável capaz de realizar determinada atividade ou função.
- *Complementa e combina a matemática e a engenharia*: segundo Wing (2006), o pensamento computacional pode auxiliar a matemática e a engenharia, no que diz respeito à abstração e lógica.
- *Gera ideias e não artefatos*: permite desenvolver ideias e criar maneiras de soluções de problemas e não simplesmente a construção de um software, hardware ou um produto específico.
- *Para todos em qualquer lugar*: o pensamento computacional não se restringe apenas aos profissionais de computação; todos podem usar sem limitação, pois é algo fundamental para estudantes, que pode auxiliar no seu desenvolvimento lógico-matemático.

O pensamento computacional não se limita apenas aos cursos de computação

especificamente. Essa técnica hodierna pode alcançar áreas afins e auxiliar o professor em sua aula, pois a locomotiva são as ideias e não os artefatos, permitindo uma maior difusão do ensino, que pode, de certa forma, inclinar os estudantes para a computação ou outras áreas das ciências exatas, acurando o conhecimento lógico e abstrato para uma possível graduação futuramente.

Afastando-se um pouco do foco em matemática e raciocínio lógico, o projeto SGD-Br ou Scalable Game Design Brasil vinha desenvolvendo um trabalho voltado para a “autoexpressão via software” que visava obter uma sólida compreensão dos desafios, oportunidades e requisitos tecnológicos especificados para um programa mais amplo de educação para computação em escolas brasileiras. O cerne do SGD-Br era ajudar a propagar a aquisição do pensamento computacional para que as pessoas pudessem entender melhor o significado social da computação, dominar conceitos computacionais rudimentares e aprender a se expressar através de programas de computador (DE SOUZA, GARCIA, et al., 2011) (DE SOUZA, SALGADO, et al., 2014).

2.5 Programação Lúdica

A programação lúdica é uma técnica da computação, em que ferramentas simples e intuitivas são apresentadas ao estudante para tornar mais divertida, interativa e participativa a construção de um programa ou aplicação. Busca assim fomentar a abstração de dados. Com essas ferramentas, é possível ver conceitos de programação tradicional, a partir de uma ótica rudimentar. Usar meios lúdicos para o ensino de programação é algo bem promissor, pois se percebe um retorno mais rápido do que o ensino tradicional de programação, cooperando diretamente em um dos grandes desafios da educação em computação, que é o ensino da programação (CARPERSEN e KÖLLING, 2009). Uma ilustração da programação lúdica pode ser vista na figura 1.

Figura 1 – Tela principal da ferramenta Scratch



Como percebido na figura 1, nas ferramentas lúdicas que auxiliam no ensino de programação, é apresentada uma interface amigável e não um console ou terminal sem expressão/motivação. Desta forma, chama-se a atenção do estudante e se desperta o seu interesse, pois ele não consome seu tempo memorizando palavras-chaves ou palavras reservadas de determinada linguagem. Isso já é um benefício para um iniciante, e mais ainda para aqueles que são integralmente leigos na área. Segundo Malan e Leitner (2007), com essas ferramentas, os usuários não se preocupam com erros sintáticos, focando apenas na criação da lógica de funcionamento de seus projetos.

O Scratch foi desenvolvido com o propósito de ajudar a desenvolver raciocínio lógico para computação sem usar programação avançada, através de uma sintaxe especial. O Scratch é uma ferramenta que possui uma interface amigável que pode ajudar o ensino de programação, já que não é necessário aprender uma sintaxe complexa ou preocupar-se com erros sintáticos. Nele, já existem noções de programação como estrutura condicional e de repetição, operadores e eventos a cliques. O estudante pode utilizar comandos para desenhar figuras geométricas como o mostrado na Figura 2.

Figura 2 – Demonstração de um problema matemático no Scratch.



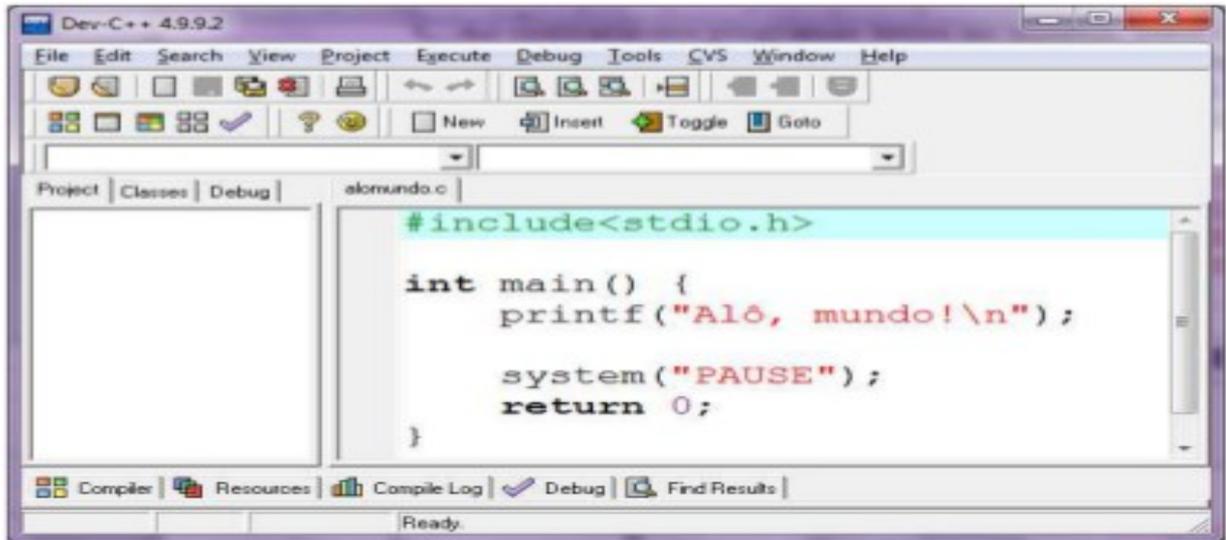
Lewis e Shah (2012) encontraram evidências iniciais do enriquecimento tanto da linguagem algorítmica quanto da linguagem matemática expressadas por alunos, que programaram usando a plataforma Scratch. Os autores perceberam uma correlação positiva entre as notas de alunos da quarta série em testes padronizados de conhecimento de matemática e as notas dos alunos que desenvolveram programas simples usando o Scratch. Assim, com o auxílio da programação lúdica, alguns obstáculos na forma de ensinar podem ser rompidos, melhorando aos poucos o interesse dos alunos em ingressar nessa área.

2.6 Algoritmos

Segundo Cormen (2002), podemos compreender um algoritmo como uma ferramenta para resolver um problema computacional bem específico. O enunciado de um problema ilustra em termos gerais o relacionamento entre a entrada e a saída desejada. O algoritmo descreve um procedimento computacional específico para se alcançar esse relacionamento da entrada com a saída (Cormen, 2002, p. 03). Na PCPL&A, tratamos de algoritmos simples, como um conjunto de linhas de código, sintaxe e semântica, limitados ao paradigma de programação sequencial², para que seja algo fácil de entender e palpável ao estudante que está iniciando a aprendizagem de programação. Segue um exemplo de algoritmo na figura 3.

² Paradigma de programação sequencial é o método em que um passo é realizado logo após outro, como o nome já ilustra, segue uma sequência de passos pré-definidas.

Figura 3 – Algoritmo simples na linguagem de programação C



A Figura 3 apresenta um algoritmo simples, escrito na linguagem de programação C, no ambiente de desenvolvimento integrado (IDE) DevC++ 4.9.9.2, que tem a finalidade de imprimir na tela uma frase clássica do jargão da computação “Alô, mundo!”, além de adicionar uma quebra de linha, codificada como “\n”. Percebe-se que as estruturas do código exigem um pouco de conhecimento de inglês, pois todas as funções implementadas na linguagem C são de origem inglesa, sem mencionar o fato de que o estudante deve saber a estrutura de cabeçalho, neste caso, composto pela palavra “include”, precedida do sinal # e pelo parâmetro informado dentro dos caracteres “< >”, para especificar uma função pré-existente com que se deseja trabalhar, neste caso, a biblioteca padrão “stdio.h”.

O ensino de algoritmos em determinada linguagem de programação não é algo trivial, especialmente para pessoas que nunca viram códigos de computador. Segundo Setúbal (2000), para o ensino de algoritmos, é necessário endereçar algumas questões, são elas:

- *Coerência com os objetivos fundamentais*: Trata-se da clareza e coesão que o professor deve ter no momento de ensinar os algoritmos. Esperam-se os seguintes resultados:
 - O professor deve expressar claramente as ideias, conceitos e as técnicas perante os alunos.
 - O professor deve destacar a importância dos resultados teóricos.
 - O professor deve valorizar as técnicas na resolução de problemas.
- *Ênfase no pensamento crítico*: Trata-se do estímulo da criticidade que os

alunos devem ter, para que possam questionar conhecimentos e técnicas de soluções, para que os eles possam trazer soluções mais otimizadas.

- *A teoria na prática*: Trata-se do uso prático do conhecimento teórico abordado, pois a teoria sem a prática é evasiva e não estimuladora.

O ensino de algoritmos é uma atividade muito importante para o desenvolvimento do raciocínio lógico-matemático. Trata-se de uma atividade intelectual e abstrata que exige bastante do aluno. Algoritmos otimizados podem reduzir os custos computacionais, no que diz respeito a tempo e dinheiro, por isso a importância da difusão e do ensino de algoritmos. Além disso, há o potencial de trazer novos interessados para a área da computação e motivá-los para o desenvolvimento de habilidades cognitivas relacionadas.

Em nossa proposta, o ensino de algoritmo é precedido da aplicação do Pensamento Computacional e da Programação Lúdica. O primeiro permite uma base teórica do assunto, e o segundo uma base prática, porém sem o uso de uma linguagem de programação tradicional. O algoritmo é apresentado aos alunos logo após o pensamento computacional e a programação lúdica, pois o aluno precisa ter uma base de abstração lógico-matemática e de conhecimentos para estruturar sequências, com uma linguagem de programação específica. Para aprender a usar algoritmos na linguagem em questão, é necessário também conhecer as suas palavras-chaves e palavras reservadas. Para Júnior e Rapkiewicz (2004), o domínio de habilidades prévias de matemática e afins são importantes.

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia utilizada para o desenvolvimento deste trabalho divide-se em quatro etapas: Definição da proposta metodológica, planejamento da aplicação da metodologia proposta, execução da aplicação da metodologia proposta e análise descritiva da aplicação da metodologia proposta. Essas etapas são detalhadas a seguir.

3.1 Definição da proposta metodológica

Na primeira etapa, definiu-se a organização da metodologia proposta (PCPL&A), onde usamos conceitos teóricos e práticos de computação para estruturá-la. Foram definidos níveis sequenciais para a metodologia proposta e para cada nível foram definidos instrumentos de avaliação. A metodologia proposta será apresentada em detalhes na seção 5.

3.2 Planejar a aplicação da metodologia proposta

A segunda etapa foi o planejamento da aplicação da metodologia proposta, definindo-se o período de duração da aplicação, a quantidade de encontros, os conteúdos abordados e a forma de avaliação.

3.3 Executar a aplicação da metodologia proposta

A metodologia proposta foi executada na escola municipal Deputado Flávio Portela Marcílio, situada na cidade de Quixadá. Participaram das atividades inicialmente 11 alunos do ensino fundamental II e o horário de realização foi durante a manhã. Os alunos participaram de forma voluntária, tinham idade de 13 a 15 anos e houve permissão dos pais para a participação das aulas por meio da assinatura de um termo de consentimento. Para o recrutamento dos alunos, informamos à escola que tínhamos preferência pelos alunos com maiores notas nas disciplinas de matemática e ciências, então essa foi a forma de selecioná-los. As aulas foram realizadas em quatro semanas durante o mês de junho de 2016. Durante as aulas, foram coletados dados de pesquisa, como respostas dos alunos a questionários, registros e anotações de atividades, material produzido pelos alunos.

3.4 Análise descritiva da aplicação da metodologia proposta

Após o término das aulas, foi feita uma análise descritiva da metodologia utilizada, apresentando-se os resultados obtidos e as lições aprendidas por meio dos dados coletados. Os resultados dessa análise serão apresentados na seção 6.

4 METODOLOGIA DE ENSINO PCPL&A

Nesta seção, será detalhada a proposta metodológica para o ensino de programação para adolescentes não pertencentes à área da computação. Essa proposta de metodologia de ensino foi batizada de PCPL&A, pois possui três níveis em sequência, a saber: PC (Pensamento Computacional), PL (Programação Lúdica) e A (Algoritmos). Os níveis da metodologia serão apresentados dentro do contexto da aplicação na escola.

4.1 Nível PC da tríade PCPL&A

No nível PC (Pensamento Computacional), que é o nível mais simples, foram realizados dois encontros de uma hora e meia, conforme detalhados a seguir.

1º Encontro

No primeiro encontro relativo ao nível PC, foram apresentados os benefícios do aprendizado de computação, por meio de um vídeo³ em que uma criança palestra para adultos sobre a importância de aprender a programar. Em seguida, foi aplicado um questionário de percepção de conhecimento com 11 alunos, para saber o nível dos alunos em relação à área da computação. O questionário está disponível no Apêndice B.

Após esta etapa de sondagem, iniciaram-se as atividades previstas para a tríade PCPL&A, especificamente para o nível PC. No primeiro encontro, foi explicado o que é o pensamento computacional e foram apresentados os conceitos básicos, segundo Wing (2006).

2º Encontro

No segundo encontro, aprofundou-se o conceito de pensamento computacional, mostrando-se alguns conceitos básicos de computação, tais como: estrutura de condições, repetições, variáveis, entre outros. Aprofundamos o conceito de pensamento computacional, finalizando-se assim o nível PC da tríade PCPL&A.

Nesse segundo encontro, foi abordada a prática do pensamento computacional. Na aplicação da metodologia, optamos por usar o problema computacional jantar dos filósofos⁴. Nos momentos finais desse encontro, foi aplicado o questionário sobre o nível PC da tríade PCPL&A com os 7 alunos. O questionário sobre o nível C da tríade PCPL&A está disponível no apêndice no C.

4.2 Nível PL da tríade PCPL&A

No nível PL (Programação Lúdica), que é o nível intermediário, foi realizado três encontros de uma hora e meia, conforme detalhados a seguir:

3º Encontro

No terceiro encontro, iniciou-se uma nova fase da metodologia PCPL&A, que é o nível PL. Nesta etapa, foi usada a ferramenta Scratch, cujas principais funções foram apresentadas por meio de uma aula introdutória. Ao final da familiarização com o Scratch, foram mostrados alguns vídeos⁵ de crianças construindo seus projetos e “joguinhos” com a ferramenta Scratch para motivação dos alunos.

4º Encontro

No quarto encontro, os alunos construíram um pequeno jogo chamado Arkanoid⁶ com a ferramenta Scratch. A partir desse jogo, abordaram-se alguns conceitos de

³ <https://www.youtube.com/watch?v=qW3XT30F0ck>

⁴ https://en.wikipedia.org/wiki/Dining_philosophers_problem

⁵ <https://www.youtube.com/playlist?list=PLPZbINIObmilTgUILaMrfQsiJOg9sfClI>

⁶ <https://pt.wikipedia.org/wiki/Arkanoid>

programação, tais como: estruturas condicionais, estruturas de repetição, movimentação e estados dos atores do Scratch.

5º Encontro

No quinto encontro, continuamos o jogo do Arkanoid e ao final foi aplicado o questionário sobre o nível PL da tríade PCPL&A para 9 alunos. O questionário está disponível no apêndice D.

4.3 Nível A da tríade PCPL&A

No nível A (Algoritmos), que é o avançado, foram realizados três encontros de uma hora e meia, conforme detalhados a seguir:

6º Encontro

No sexto encontro, iniciamos a última fase da metodologia, o nível A, que é a etapa onde foram abordados os algoritmos, finalmente associados à programação tradicional. Neste encontro, mostrou-se a IDE Eclipse⁷, e apresentamos aos alunos a linguagem de programação Java. Nesta etapa, os alunos foram familiarizados com a linguagem e com a IDE Eclipse de forma bem superficial, em que abordamos da forma mais simples possível conceitos como, por exemplo: variáveis, pacotes, classes, interface, herança, estruturas de repetição e estruturas de condições.

7º Encontro

No sétimo encontro, apresentamos toda a construção de um programa básico, de forma procedural e não orientada a objeto. Construimos com os alunos uma calculadora simples, com as quatro operações (adição, subtração, multiplicação e divisão) em programação na linguagem Java.

8º Encontro

No oitavo encontro, fizemos uma otimização na calculadora e incluímos mais duas operações matemáticas: potenciação e radiciação. Após essa atividade, aplicamos um questionário sobre o nível A da tríade PCPL&A para 4 alunos. O questionário está disponível no apêndice E.

7 RESULTADOS

Nesta seção, relatamos os resultados da aplicação da metodologia PCPL&A. Conforme mencionado anteriormente, a metodologia proposta foi executada com inicialmente

⁷ [https://pt.wikipedia.org/wiki/Eclipse_\(software\)](https://pt.wikipedia.org/wiki/Eclipse_(software))

11 alunos do ensino fundamental II em uma escola municipal de Quixadá. Os resultados estão categorizados em duas modalidades, que são as seguintes:

- Questionários:
 - Questionário de percepção de conhecimento, disponível no apêndice B;
 - Questionário no nível PC da tríade PCPL&A, disponível no apêndice C;
 - Questionário no nível PL da tríade PCPL&A, disponível no apêndice D;
 - Questionário no nível A da tríade PCPL&A, disponível no apêndice E;
- Atividades:
 - Atividade no nível PC (modelagem do problema jantar dos filósofos).
 - Atividade no nível PL (construção do jogo Arkanoid usando a ferramenta Scratch).
 - Atividade no nível A (construção de uma calculadora contendo as operações de adição, subtração, multiplicação e divisão).

Ao final da seção, é apresentada uma síntese dos resultados, levantamento dos principais problemas e uma breve discussão.

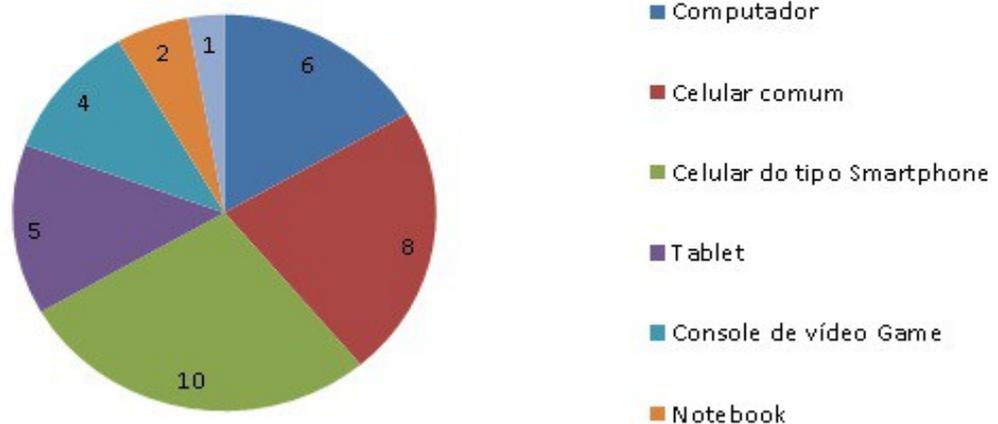
6.1 Resultados do questionário de percepção de conhecimento

No primeiro encontro com os alunos, antes de começar o conteúdo previsto para a tríade, foi aplicado um questionário para traçar perfil dos alunos e saber o que os alunos conhecem de computação e tecnologia da informação. Esse questionário de 14 perguntas (subjetivas, objetivas e algumas perguntas possuíam mais de uma opção para a resposta) foi respondido pelos alunos convidados a participar aulas. Foram respondidas perguntas específicas sobre computação e vida escolar dos alunos, conforme pode ser conferido no Apêndice B. A seguir apresentamos os resultados obtidos com este primeiro questionário.

No gráfico 1, vê-se o resultado da primeira pergunta, sobre a experiência de uso com dispositivos eletrônicos. Entre os dispositivos listados, vê-se a predominância do uso de celulares, pois 10 dos 11 alunos mencionaram ter acesso a *smartphones*, além de que 8 mencionaram o celular comum. O dispositivo menos utilizado é o notebook.

Gráfico 1 – Experiência de uso de dispositivo

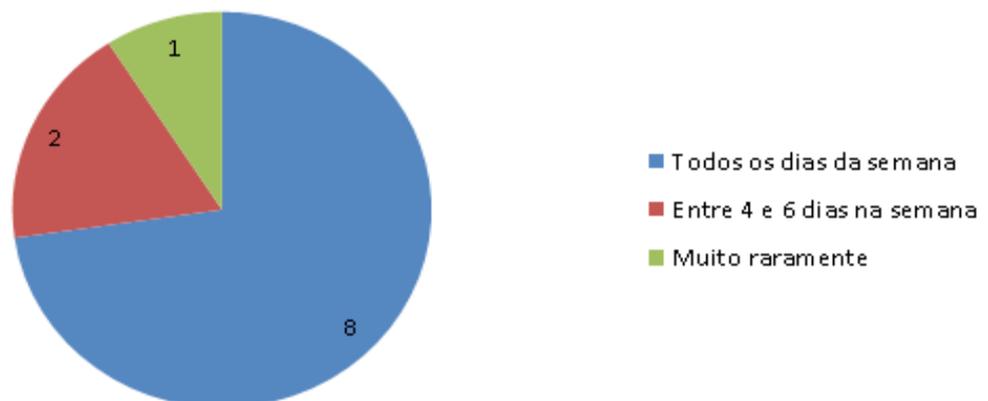
Pergunta 1: Em qual dos seguintes dispositivos você possui experiência?



No gráfico 2, vê-se o resultado sobre a frequência de uso de dispositivos eletrônicos. Pelos dados, podemos verificar que a maior parte dos alunos acessa dispositivos diariamente e que no mínimo 10 alunos acessam entre 4 a 6 dias da semana.

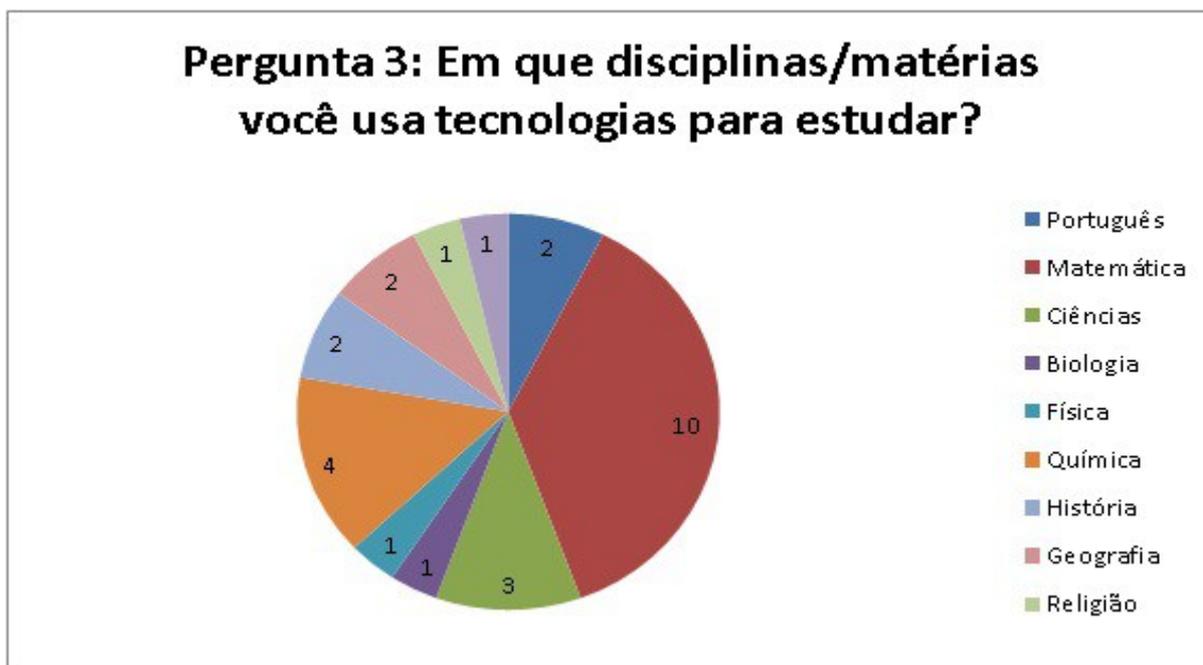
Gráfico 2 – Frequência de uso de dispositivo

Pergunta 2: Com que frequência você utiliza um ou mais dos dispositivos citadas acima?



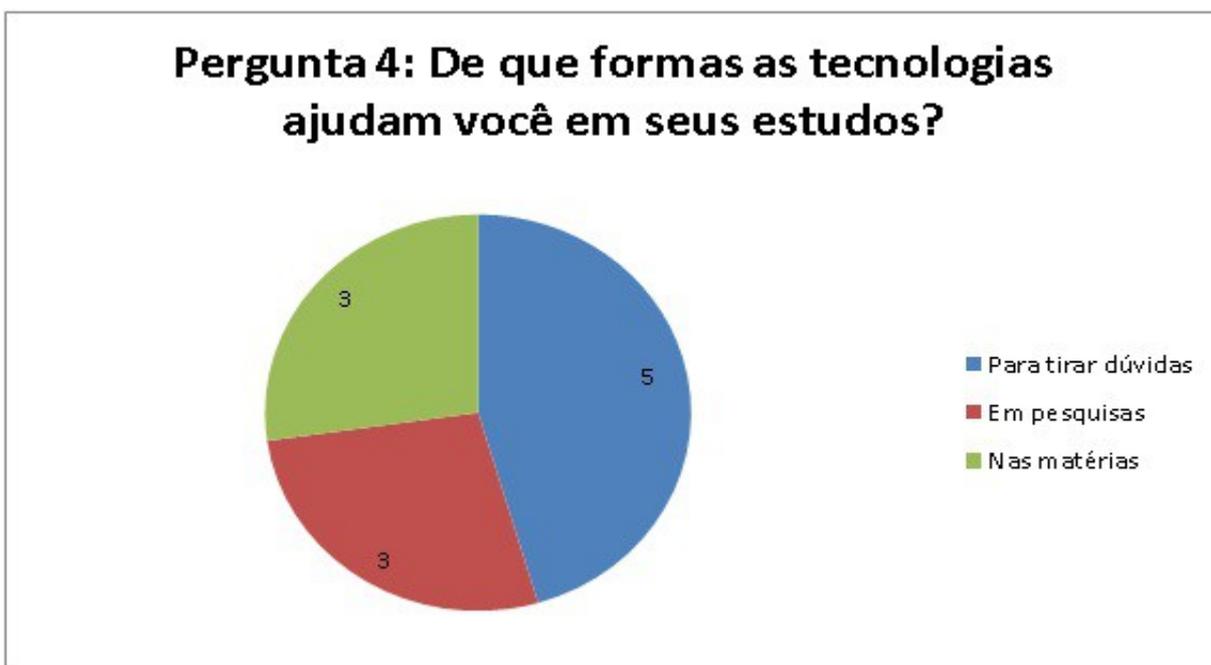
No gráfico 3, vê-se o resultado sobre a utilização de tecnologias para estudar. Pelas disciplinas listadas, podemos perceber a forte evidência do uso de tecnologia para a disciplina de matemática. Dando continuidade à análise de dados, notamos que, depois de matemática, as disciplinas com maiores índices de utilização de tecnologia são química e ciências, o que confirma a nossa forma de recrutamento dos alunos, pois eles estariam mais envolvidos com essa disciplina.

Gráfico 3 – Utilização de tecnologia para estudar nas disciplinas/matérias



No gráfico 4, vê-se o resultado sobre as formas com que as tecnologias auxiliam os estudos. Pelos dados, verifica-se que a maior parte dos alunos usa as tecnologias para auxiliar nas dúvidas das disciplinas e também nas pesquisas realizadas para os estudos.

Gráfico 4 – Formas de ajuda da tecnologia nos estudos



Na tabela 1, vê-se a ordem de preferência das disciplinas, atribuída por cada aluno, onde “1” indica a disciplina preferida e “9”, a menos preferida.

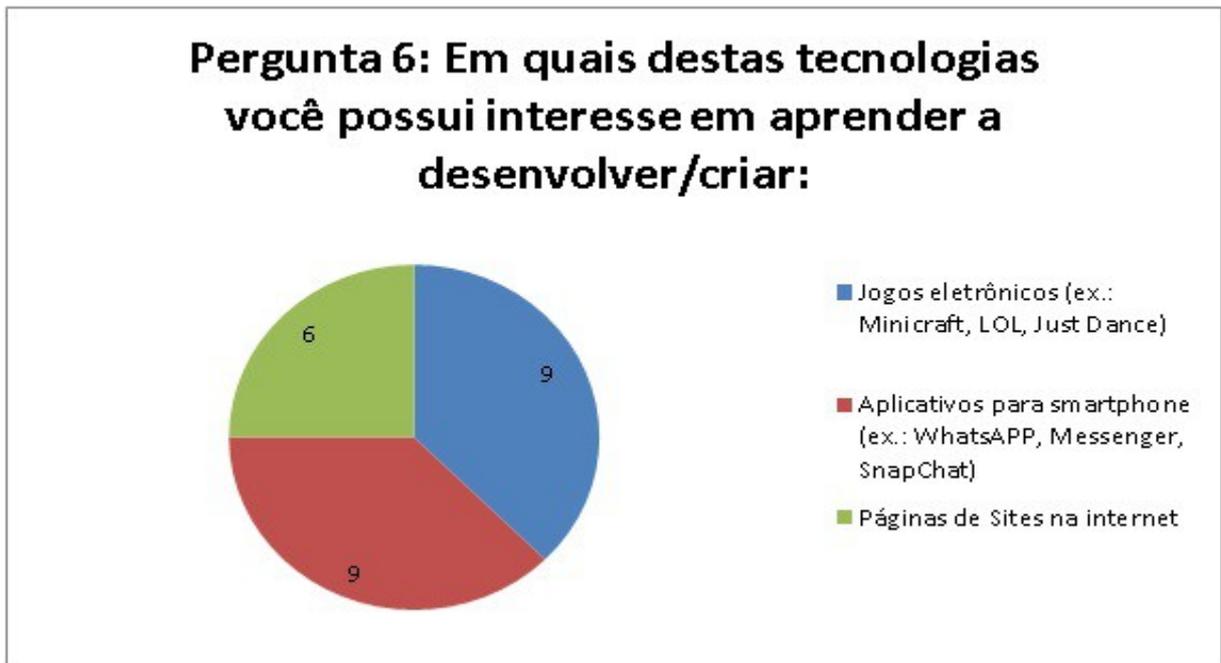
Tabela 1 – Notas dadas às disciplinas.

| | Português | Matemática | Ciências | Biologia | Física | Química | História | Geografia | Outra |
|-----|-----------|------------|----------|----------|--------|---------|----------|-----------|------------------|
| A1 | 5 | 8 | 1 | 3 | 4 | 2 | 7 | 9 | |
| A2 | - | - | - | - | - | - | - | - | |
| A3 | 2 | 3 | 6 | 7 | 5 | 4 | 1 | 9 | |
| A4 | 3 | 9 | 4 | 5 | 6 | 7 | 1 | 2 | |
| A5 | 1 | 1 | 1 | 9 | 9 | 9 | 1 | 1 | |
| A6 | 3 | 2 | 1 | 5 | 5 | 5 | 6 | 9 | |
| A7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| A8 | 2 | 1 | 3 | 6 | 7 | 8 | 4 | 5 | |
| A9 | - | - | - | - | - | - | - | - | Artes e religião |
| A10 | 6 | 1 | 2 | 5 | 3 | 4 | 7 | 8 | |
| A11 | 6 | 2 | 3 | 7 | 8 | 9 | 4 | 5 | Artes |

Com os resultados da tabela, podemos perceber que as duas disciplinas com os dois melhores índices entre todos os alunos são matemática e ciências, o que é um resultado do recrutamento dos alunos para a metodologia proposta. Além disso, dois alunos (aluno 2 e aluno 9) não votaram corretamente, pois o aluno 2 marcou apenas com x não indicando assim cada valor atribuído a cada disciplina e o aluno 9 não marcou nenhuma alternativa, porém na alternativa para outras disciplinas que não estivesse listada, o aluno 9 escreveu artes e religião. O aluno 11 também indicou a disciplina artes na alternativa outras disciplinas.

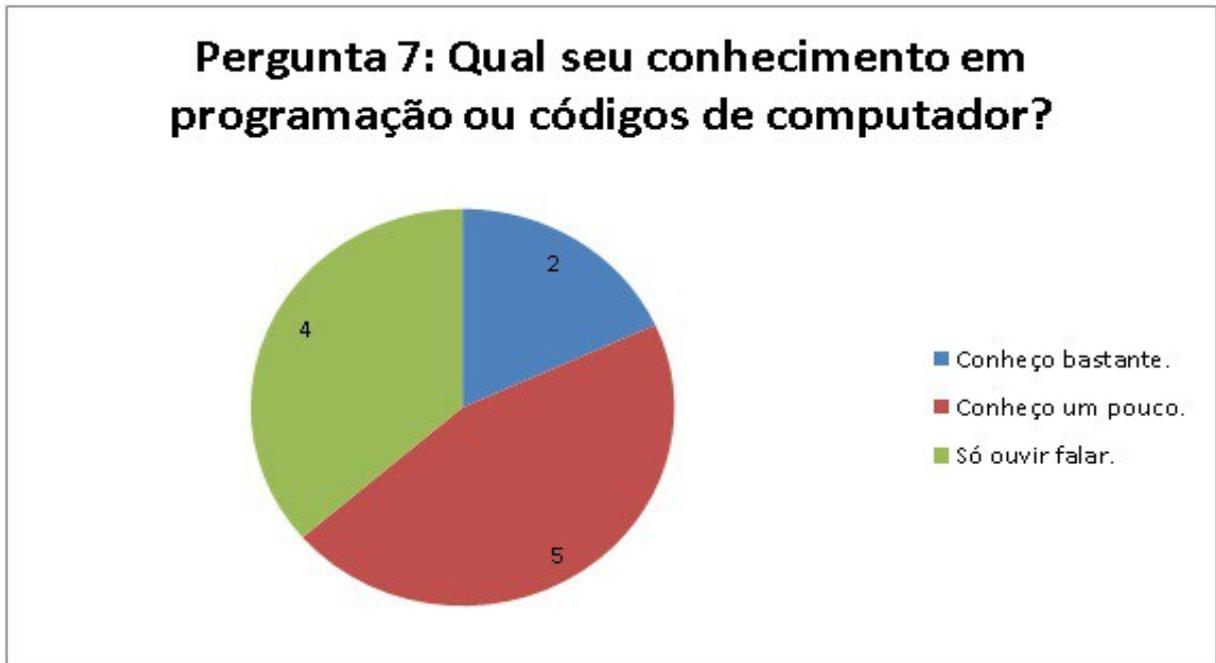
No gráfico 5, vê-se o resultado sobre o interesse em aprender tecnologias. Pelos dados, vê-se que as tecnologias que os alunos mais querem aprender a desenvolver/criar são jogos e softwares para dispositivos móveis, já a tecnologia que os alunos menos escolheram para desenvolver/criar é páginas de sites da internet.

Gráfico 5 – Interesse em tecnologias



No gráfico 6, vê-se o resultado sobre o conhecimento prévio dos alunos em relação à programação. Pelas respostas dos alunos, nota-se que a maior parte dos alunos (9 de 11) não conhece bem sobre o assunto. Com isso, podemos concluir que o mundo da programação ainda é algo desconhecido.

Gráfico 6 – Conhecimento sobre programação ou códigos de computador



No gráfico 7, vê-se o resultado sobre o conhecimento prévio dos alunos em relação a linguagens e ferramentas de programação. Pelas respostas dos alunos, notamos que a maior parte deles desconhece o assunto, porém, 2 dos 3 alunos que mencionam já ter ouvido falar relataram já ter ouvido falar em Java.

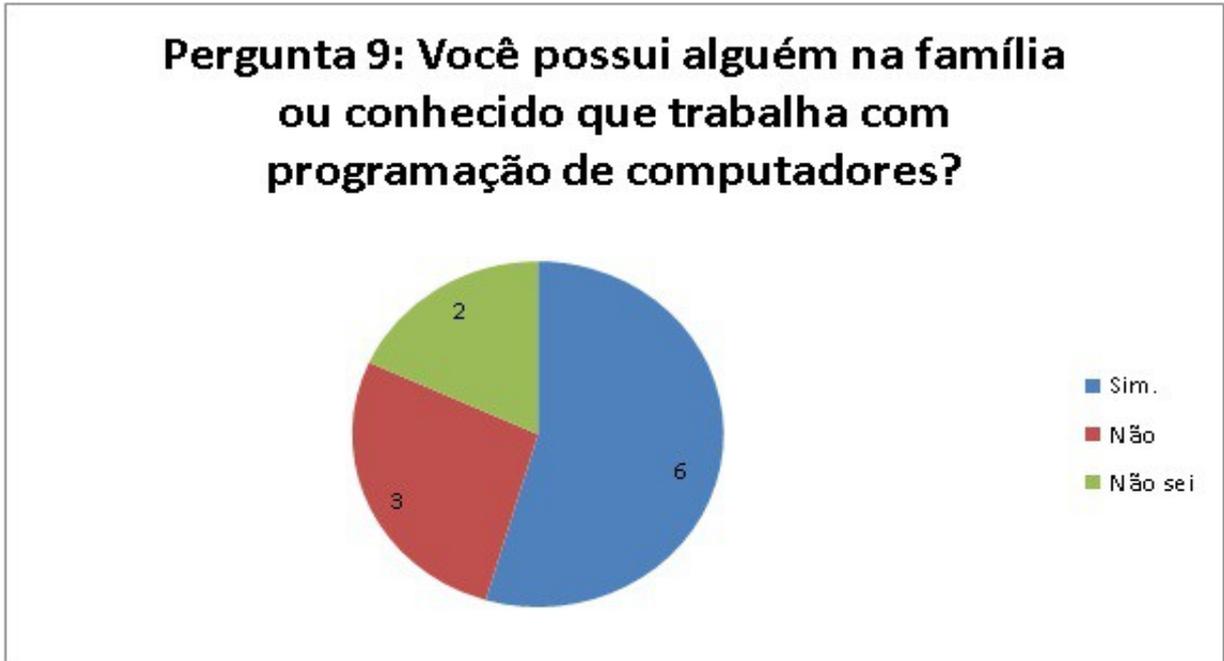
Gráfico 7 – Conhecimento sobre linguagens e/ou ferramenta de programação



No gráfico 8, vê-se o resultado sobre a presença de pessoas na família ou conhecidos que trabalham na área. Pelas respostas dos alunos, notamos que mais da metade

dos alunos conhece ou possui alguém na família que trabalha com programação de computadores.

Gráfico 8 – Conhecidos que trabalham com programação



No gráfico 9, vê-se o resultado sobre o conhecimento dos alunos a respeito de filmes e séries que abordem temas de computação. Pelos dados coletados, podemos notar que 10 dos 11 alunos desconhecem filmes/séries sobre programação, o que apenas confirma a falta de aproximação com os assuntos da proposta metodológica.

Gráfico 9 – Filmes/séries sobre programação.



No gráfico 10, vê-se o resultado sobre o conhecimento dos alunos a respeito de nomes notáveis da área da computação. Pelos dados coletados, podemos observar que os mais conhecidos são: Steve Jobs (fundador da Apple), Mark Zuckerberg (fundador do Facebook) e Bill Gates (fundador da Microsoft).

Gráfico 10 – Conhecimento de nomes da área de computação

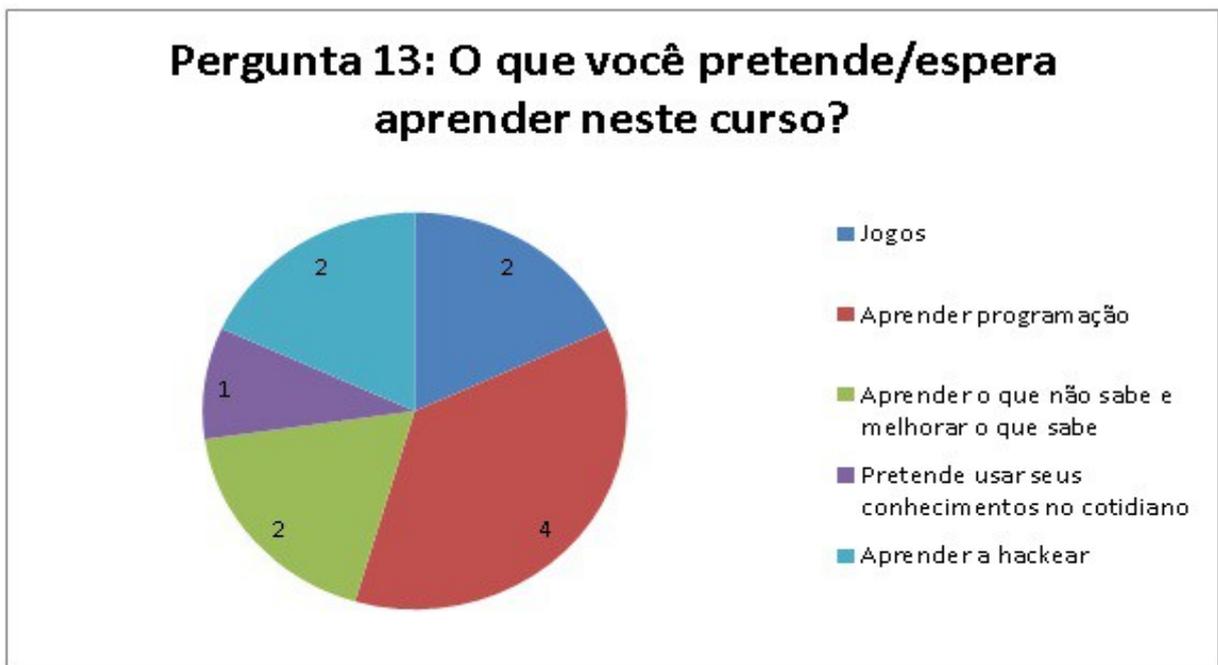


Também questionamos o motivo de os alunos estarem participando das aulas (“Por que você está fazendo este curso”). Mediante esta pergunta, obtivemos o seguinte resultado: 4 dos 11 alunos informaram que foram simplesmente chamados a participar, já os

outros 7 disseram que têm vontade de aprender a programar, de conhecer linguagens de programação e de dominar a área de informática.

No gráfico 11, apresentamos o resultado em relação às expectativas em relação ao curso. Mediante os dados apresentados, oito dos onze alunos indicaram o interesse em tópicos relacionados à computação (programação, jogos e *hacking*), um deles inclusive mencionou que queria aprender a “*hackear* o site da Apple”. Os outros três responderam de forma mais genérica (“aprender o que não sabe e melhorar o que sabe”, “usar meus conhecimentos para o cotidiano”).

Gráfico 11 – Expectativas em relação ao curso



No gráfico 12, apresentamos o resultado em relação ao interesse dos alunos em estudar mais e seguir na área de tecnologia. Como resultado, dos 11 alunos, apenas 3 mostraram-se inseguros em relação a isso, 8 alunos mostraram algum interesse (sim ou talvez). É importante destacar que nenhum aluno respondeu “De jeito nenhum”.

Gráfico 12 – Interesse em estudar/seguir na área de tecnologia



A partir dos resultados obtidos com o questionário, sobre os perfis dos alunos, podemos notar que a maioria gosta de ciências exatas, preferencialmente matemática e ciências, o que facilitou inclusive a aplicação da metodologia proposta, possuem mais experiências com celular do tipo *smartphone* e o usa na maior parte da semana. A maior parte dos alunos possui interesse em seguir na área de tecnologia. Este interesse refletiu-se nas aulas, visto que os alunos conseguiram acompanhar a maior parte dos exemplos dados durante a aplicação.

6.2 Resultados do nível PC da tríade PCPL&A

Nessa seção, serão mostrados os resultados obtidos no nível PC da tríade PCPL&A. Nesta etapa da aplicação da metodologia, sete alunos participaram. Os resultados desse nível da tríade serão apresentados a partir de dois tipos de dados: observação da atividade realizadas durante os 2 encontros e análise do questionário apresentado aos alunos no final do nível PC.

6.2.1 Observação da resolução do problema do jantar dos filósofos usando Pensamento Computacional

As duas primeiras aulas da metodologia proposta foram sobre Pensamento Computacional. Conceitos de condições e repetições foram explicados sem serem chamados assim, para que mais adiante fossem usados com mais facilidade. Foram vistos nessa fase

inicial vários conceitos. Aos alunos, foram mostrados conceitos de computação de forma teórica para que, nos níveis avançados, eles pudessem reutilizar esse conhecimento lógico.

O exercício modelado com pensamento computacional baseou-se no problema de computação conhecido como “jantar dos filósofos”, que é descrito com a seguinte situação: há cinco filósofos e cinco garfos, sendo um garfo para cada filósofo. Segundo o problema, cada filósofo pode estar em apenas um estado, pensando ou comendo, e seu garfo igualmente só pode estar em apenas um estado, livre ou ocupado. Porém, o filósofo em determinado momento vai jantar e, para isso, ele precisa de dois garfos livres e precisa que o seu estado seja “comendo”. Com a apresentação desse problema, conseguimos expor aos alunos alguns conceitos de computação, tais como: estados, condições, repetições e modelagem de problemas.

O problema foi útil para explorarmos conceitos computacionais em situações cotidianas e que, de forma eficiente consegue alcançar boas resoluções. Alguns alunos perceberam que para alguns filósofos comerem outros necessariamente deveriam estar pensando e foi percebido que apenas dois filósofos poderiam comer por vez. No que diz respeito aos garfos, os alunos perceberam que apenas um garfo ficaria livre, pois, para cada filósofo, necessariamente dois garfos devem ter seus estados ocupados, assim apenas dois filósofos poderiam comer por vez, enquanto os outros três ficariam pensando.

6.2.2 Questionário ao final do nível PC da tríade PCPL&A

Ao final do nível PC da tríade PCPL&A, foi aplicado um questionário (Apêndice C) com os sete alunos participantes. Os resultados serão apresentados a seguir.

Pergunta 1: O que você aprendeu nas aulas?

Na pergunta de número 1 sobre PC, percebemos que todos os alunos participantes entenderam o conceito de Pensamento Computacional, por meio de suas respostas por escrito, sendo que alguns ainda foram mais longe, mencionando que era necessária a utilização de lógica/raciocínio para resolver muitas das situações descritas e mencionadas.

Pergunta 2: Qual a importância para a sua vida?

Na pergunta de número 2 sobre PC, notamos a importância que os alunos deram ao projeto e o conteúdo abordado, pois todos eles comentaram que o conteúdo abordado em sala de aula era relevante, indicando os seguintes motivos: Cinco das sete respostas

destacaram a importância da relação com o mercado de trabalho e as outras duas respostas voltaram-se para questões de aprendizagem e conhecimento.

Pergunta 3: Você teve dificuldade? Em quê?

Na pergunta de número 3 sobre PC, os alunos puderam se expressar a respeito de suas dificuldades nas aulas. Quatro dos sete alunos afirmaram ter sentido dificuldades, dois deles afirmaram “ter dificuldades em tudo”, enquanto os outros dois expressaram melhor sua dificuldade: um teve dificuldade em algoritmos e o outro em lógica no PC.

Pergunta 4: Tem interesse em seguir na área?

Na pergunta de número 4 sobre PC, analisamos o interesse dos alunos em seguir na área e obtivemos o seguinte resultado: 4 dos 7 alunos disseram que possuíam interesse, sendo que 1 dos 4 afirmou que o curso valia muito a pena e que iria investir no curso e os 3 restantes responderam com um talvez, indicando dúvida.

Pergunta 5: O que você não sabia sobre Pensamento Computacional?

Na pergunta de número 5 sobre PC, coletamos por escrito o que os alunos não sabiam sobre PC, o resultado foi o seguinte: apenas um dos sete alunos já tinha algum conhecimento e os outros 6 não sabiam o que era PC.

6.3 Resultados do nível PL da tríade PCPL&A

Nessa seção, serão mostrados os resultados obtidos no nível PL da tríade PCPL&A. Assim como no nível PC, os resultados desse nível da tríade serão apresentados a partir da observação das atividades realizadas durante os três encontros e da análise do questionário apresentado aos alunos no final do nível PL.

6.3.1 Observação da construção do jogo Arkanoid usando Programação Lúdica

A segunda fase da metodologia proposta vem logo após a aplicação do Pensamento Computacional. Essa etapa é constituída de conceitos computacionais simples, usando uma ferramenta com uma interface amigável de tal forma que a experiência fosse agradável e divertida. A ferramenta usada fora o Scratch, em que os alunos tiveram facilidades de mexer e aprender sozinhos os comandos de blocos. Foi perceptível o interesse maior por programação lúdica do que por pensamento computacional pelos alunos.

Nessa etapa, os alunos criaram o jogo Arkanoid, houve inclusive alunos que exploraram conceitos e funcionalidades mais avançados do que era ensinado na sala de aula. Dando continuidade ao PC, foram acrescentados alguns componentes lógicos para

complementar o PC, que correspondem aos blocos de comandos que o Scratch fornece. Com esses componentes lógicos, foi possível usar repetição, condições, mudança de fantasia do ator, visão superficial de programação paralela, entre outros conceitos que cooperaria para a última etapa da metodologia proposta.

O programa construído pelos alunos nessa fase foi um jogo simples, em que o jogador não podia deixar uma bolinha passar de uma determinada área, e, com essa bolinha, ele precisa acertar as barras para pontuar. Ao acertar todas as barras com a bolinha ou a bola passando de sua área, o jogo termina. Fora perceptível que os alunos já usavam repetição mesmo sem saber esse conceito em computação, já conseguiam verificar as condições do joguinho, tais quais: se a bola passava da área do jogador e se a bola havia acertado todas as barras, para encerrar o jogo. E ao final desse jogo, os alunos notaram que era necessário raciocínio lógico, por isso que, como será visto adiante, houve várias respostas dos alunos mencionando raciocínio lógico como algo fundamental para essa área. Com isso, trabalhou-se a lógica computacional em conjunto com programação lúdica, para que chegassem a algoritmos (nível A da tríade) com um pouco mais de segurança.

6.3.2 Questionário ao final do nível PL da tríade PCPL&A

Ao final do nível PL da tríade PCPL&A, foi aplicado um questionário (Apêndice D) com os nove alunos participantes. Os resultados serão apresentados a seguir.

Pergunta 1: O que você aprendeu nas aulas?

Pelos dados coletados, podemos notar que a maior parte dos alunos escreveu que aprendeu a usar algo de programação (exceto 1 aluno, que disse que aprendeu a dar importância a cada coisa), com isso podemos concluir que 8 dos 9 alunos conseguiram absorver algum tipo de conhecimento de programação.

Pergunta 2: Qual a importância para sua vida?

Na pergunta de número 2 sobre PL, perguntamos sobre a importância de PL para a vida dos alunos, o resultado foi o seguinte:

- Dois alunos comentaram sobre a questão do aprendizado e seus benefícios
- Dois alunos escreveram sobre a importância da tecnologia em si para as vidas deles.
- Cinco alunos indicaram a possibilidade de mercado de trabalho com ajuda do Scratch.

Com isso, podemos concluir que todos os alunos julgaram importante esse conteúdo e que mais da metade escreveu que pode entrar para o mercado de trabalho usando conceito de programação lúdica na ferramenta Scratch para melhorar seu raciocínio lógico de programação.

Pergunta 3: Você teve dificuldade? Em quê?

Na pergunta de número 3 sobre PL, analisamos as dificuldades dos alunos, o resultado foi o seguinte: 4 dos 9 alunos tiveram dificuldades, desses 4 alunos, 2 tiveram dificuldades no início da aplicação da metodologia e os outros 2 afirmaram categoricamente que tiveram dificuldades em tudo e os 5 restantes afirmaram que não enfrentaram dificuldades.

Pergunta 4: Tem interesse em seguir na área?

Na pergunta de número 4 sobre PL, analisamos o interesse dos alunos em seguir na área, o resultado foi o seguinte: 3 dos 9 alunos disseram que tinham interesse, já os outros 6 alunos indicaram ter dúvidas em seguir na área.

Pergunta 5: O que você não sabia sobre programação lúdica?

Na pergunta de número 5 sobre PL, analisamos o que os alunos não sabiam. O resultado foi que todos os nove alunos não conheciam a expressão PL e nem o que ela significava. Entretanto, alguns afirmaram que já conheciam a ferramenta Scratch.

6.4 Resultados do nível A da tríade PCPL&A

Nessa seção, serão mostrados os resultados obtidos no nível A da tríade PCPL&A da metodologia proposta. Os resultados desse nível da tríade também foram organizados a partir de observação e questionário (Apêndice E).

6.4.1 Observação das resoluções de alguns problemas clássicos de computação usando Algoritmos

Na terceira fase da metodologia proposta, foi abordada uma introdução à programação em Java, com aplicação da sintaxe e com a IDE Eclipse. Com o intuito original de aplicar programação, mas de forma gradual, chegamos ao momento em que seria aplicada uma programação mais tradicional, porém bem superficial. Como será visto adiante, todos os alunos responderam “sim” à questão sobre se gostaram ou não de algoritmos.

Nesta etapa, os alunos já conseguiam se lembrar da estrutura de repetição, explorada no nível anterior PL com o Scratch. Para o nível A, foi apresentada a estrutura “for”

que era a equivalente a “repita” no Scratch, que eles estavam acostumados a usar. Mostramos também a estrutura “if” que era a equivalente a “se” no Scratch. Alguns alunos claramente conseguiram assimilar e associar as estruturas nas duas vertentes, e chegavam a dizer surpresos: “É mesmo, professor!”.

Diferente do Scratch, agora os alunos tiveram que aprender a declarar variáveis de início, controle e incrementarão nas repetições, além de aprender que na condição (if) havia uma comparação. A maior parte dos alunos gostou da experiência. Entretanto, percebemos a grande dificuldade dos alunos em alguns momentos, pelos seguintes fatos: as palavras serem em inglês (havia déficit grande nos alunos sobre o idioma) e era preciso conhecer a sintaxe do Java.

Em virtude da grande dificuldade, só foi possível desenvolver um programa, uma calculadora simples com as quatro operações. Foi um exercício feito em linguagem Java, com a IDE eclipse. Nesse exercício, foram abordados os conceitos de repetição, construção de variáveis com seus valores, criação das operações de adição, subtração, multiplicação e divisão. Os alunos acharam tudo até o momento muito interessante, porém mesmo com a junção do pensamento computacional e programação lúdica para chegar a um nível de programação sedimentada, ainda houve dificuldades, porém, eles já conseguiam resolver questões de repetições e condições, mas para dar outros passos sempre precisavam de ajuda.

6.4.2 Questionário ao final do nível A da tríade PCPL&A

Ao final do nível PC da tríade PCPL&A, foi aplicado um questionário (Apêndice E) com os quatro alunos participantes. Os resultados serão apresentados a seguir.

Pergunta 1: O que você aprendeu nas aulas de algoritmos?

Na pergunta de número 1 sobre A, observamos que os alunos aprenderam, o resultado foi o seguinte: 3 dos 4 alunos aprenderam o básico de programação e 1 aluno aprendeu a fazer a calculadora simples.

Pergunta 2: Qual a importância para sua vida?

Na pergunta de número 2 sobre o nível A, analisamos a importância daquele conhecimento para a vida dos alunos, o resultado foi o seguinte: 3 dos 4 alunos associaram o conhecimento ao mercado de trabalho e 1 falou acerca do aprendizado com algoritmos.

Pergunta 3: Você teve dificuldades? Em quê?

Na pergunta de número 3 sobre A, analisamos as dificuldades dos alunos, o resultado foi o seguinte: 3 dos 4 alunos não tiveram dificuldades e apenas 1 afirmou ter dificuldade em tudo.

Pergunta 4: Tem interesse em seguir na área?

Na pergunta de número 4 sobre A, os alunos informaram sobre seu interesse em seguir na área, o resultado foi o seguinte: 3 dos 4 alunos afirmaram que têm vontade de seguir na área e 1 dos 4 alunos tem dúvida.

Pergunta 5: O que você não sabia sobre algoritmos?

Na pergunta de número 5 sobre A, analisamos o que os alunos não sabiam sobre algoritmos, o resultado foi o seguinte:

- 1 dos 4 alunos não sabia de nada.
- 1 dos 4 alunos não sabia muitas coisas.
- 1 dos 4 alunos não sabia muito.
- 1 dos 4 alunos não sabia as palavras em inglês.

Pergunta 6: Você gostou de algoritmos?

Na pergunta de número 6 sobre A, a pergunta foi para saber se os alunos gostaram de ter visto algoritmos, o resultado foi o seguinte: Todos os alunos gostaram de algoritmos.

6.5 Levantamento dos principais problemas

Os principais problemas encontrados durante a aplicação da metodologia proposta serão apresentados e descritos nesta seção. Também serão feitas propostas e sugestões para melhoria do conteúdo, de forma a torná-lo mais atrativo e efetivo.

Escolas e turmas

A dificuldade inicial foi obter uma turma para aplicar a proposta metodológica. A ideia de uma proposta metodológica já estava planejada, mas não sabíamos onde executar, para coletar os dados e assim realizar a pesquisa propriamente dita.

Em nossa busca por uma escola parceira, foram negados os pedidos em duas escolas. Houve o caso de uma escola particular ter confirmado a participação e, devido a um imprevisto de infraestrutura, teve que voltar atrás. Esse problema de em um momento ter uma turma certa para a realização e em instantes depois não ter mais inviabiliza a iniciativa de

oferecer conhecimento de uma forma nova para pessoas que não pertencem a computação, necessariamente adolescentes.

Ambiente

O ambiente é o lugar ocupado para a realização da aplicação da metodologia, sendo ele um dos fatores determinante na continuação dos alunos em sala de aula. Um lugar apropriado para o ensino/aprendizado é bem iluminado, de preferência um lugar que não exponha os alunos ao calor, pois isso gera desconforto.

Para obter êxito, é necessário, antes de começar com os alunos, verificar se o ambiente não oferece desconforto ou contribui para o desestímulo dos participantes, pois se o ambiente for impróprio ou inadequado os alunos não vão se sentir à vontade e poderão se evadir.

Material

O material corresponde aos recursos usados para a boa apresentação e execução da proposta metodológica. Houve problemas comuns como faltar projetor e caixa de som, além do uso de computadores antiquados e sem configuração alguma para suportar os dois últimos níveis da tríade, que precisam da utilização das máquinas. Para minimizar esse problema, chegamos a utilizar recursos próprios, pois a escola não tinha estrutura para arcar com os materiais que seriam cruciais para o bom desempenho da metodologia PCPL&A.

A solução mais viável para esse problema com os materiais é fazer um levantamento de requisitos dos recursos que serão necessários do começo ao final da metodologia proposta, para que haja surpresas com algum déficit de material da escola, pois nem todas as escolas possuem recursos que possam possibilitar uma boa difusão do conhecimento.

Evasão

A evasão foi um problema significativo na realização e execução da tríade PCPL&A. Começamos com 11 alunos e ao final só havia 4 alunos, uma taxa de aproximadamente 63,6% de evasão. Isso provavelmente ocorreu quando os alunos perceberam que era necessário foco e dedicação para o tipo de conhecimento que eles iriam receber.

Há um conjunto de fatores que corroboram para a evasão dos alunos, podemos citar os exemplos anteriormente descritos, mas há também as dificuldades encontradas em

cada aluno. Havia alunos que não sabiam funções iniciais, tais como salvar um arquivo e desligar o computador. Tendo em vista essas dificuldades, seria interessante que, antes de falar de computação, fosse visto a informática básica, pois falar da construção de um objeto é mais difícil do que falar sobre a utilização desse objeto.

6.6 Discussão

Os questionários e atividades evidenciaram dados relevantes e significativos. Os quatro questionários possibilitaram a percepção de problemas, tais como pessoas totalmente leigas ao assunto, dificuldades, dúvidas, pouca experiência com dispositivos tecnológicos. Já as atividades realizadas revelaram algumas dificuldades que os alunos possuíam em raciocínio lógico computacional, construção de programas básicos na ferramenta Scratch e no ambiente de desenvolvimento integrado Eclipse com Java.

Os questionários tinham cinco perguntas padrões (com exceção do questionário do nível A da tríade PCPL&A, que possuía uma pergunta a mais, essa pergunta foi colocada propositalmente, pois o nível de algoritmos é o nível mais avançado por ser já programação sedimentada) que nos ajudou a mapear o entendimento dos alunos em relação ao assunto abordado, ver a importância com que os alunos consideravam ao assunto, evidenciar as dificuldades dos alunos e categorizá-las, notar o interesse dos alunos em seguir na área e o que os alunos sabiam sobre os conteúdos explanados na metodologia.

A maior parte das respostas dos alunos aos questionários, necessariamente nas perguntas de interesse em seguir na área, refere-se de forma positiva para continuar estudando o assunto e manteve sempre um número majoritário para continuar na área, levando em consideração que os alunos que indicaram dúvidas tinham inclinação para seguir na área. Percebemos também que os conceitos abordados na aula eram desconhecidos (Pensamento Computacional. Programação Lúdica e Algoritmos), mas com o passar das aulas os alunos passaram a gostar do assunto.

Apesar dos problemas encontrados, foi possível encontrar pontos positivos na proposta metodológica por meio dos questionários e atividades realizados. Nas atividades, foi possível perceber que os alunos conseguiram fazer a distinção de um nível para outro da tríade PCPL&A e já conseguiram reutilizar o conhecimento do nível anterior no nível atual, que se mostra como o resultado mais importante e o alvo principal da metodologia proposta. Outro ponto importante foi a indicação que os questionários apontaram que a maior parte dos alunos gostou do assunto e possui grande interesse pela área.

O Nível mais difícil e desafiador foi o Nível A (Algoritmos), pelo fato de ser um contato com a programação sedimentada, ou seja, uma linguagem com sintaxe e uma IDE (ambiente de desenvolvimento integrado), palavras em inglês e por esse nível exigir mais raciocínio lógico e comprometimento com o assunto. O nível mais fácil foi o nível PL (Programação Lúdica), apesar de os alunos não conhecerem o termo usado nesse nível, eles já sabiam usar algumas funções (adicionar uma personagem/ator, mudar o palco principal, importar imagem de dentro e fora do Scratch), por isso houve um maior foco nesse nível.

Uma limitação encontrada refere-se à execução da tríade PCPL&A, pois não foi fácil encontrar uma escola que disponibilizasse oito encontros com seus alunos em um ambiente apropriado. Apesar dessa limitação, conseguimos implementar e executar a metodologia proposta. A aplicação de questionários e atividades alcançaram importantes retornos para a pesquisa e assim contribuíram para o levantamento de problemas.

Em relação ao número de evasão (63,3%), não conseguimos concluir se a metodologia proposta ou se apenas o nível A da tríade PCPL&A foi o principal agravante para acarretar esse número de desistência ou se há fatores externos vinculados. Como a metodologia só foi executada uma vez, tivemos dificuldades em mensurar o fator determinante para esse declínio dos alunos para as aulas.

Os quatro alunos que foram até o final da proposta metodológica eram três meninos e uma menina, dentre os três meninos estavam os dois alunos que citaram já ter ouvido falar em Java. Pelos dados e pelo decorrer das aulas, notamos que especificamente estes dois alunos se sobressaíam em relação aos outros.

A linguagem de programação escolhida para o nível A da tríade PCPL&A foi Java, pelo fato de ser uma linguagem popular e com muitos exemplares/ tutoriais à disposição para o mercado de trabalho. Aqui surgem duas críticas à utilização de Java, por ser uma tecnologia difícil de aprender em pouco tempo e por ser pesado para um computador com configurações mais baixas, como eram os computadores da escola onde foi aplicada a metodologia.

8 CONSIDERAÇÕES FINAIS

Este trabalho procurou aplicar uma proposta de metodologia de ensino de programação e raciocínio lógico computacional para pessoas que não têm conhecimento algum em computação, por meio de três níveis incrementais: pensamento computacional (PC), programação lúdica (PL) e algoritmo (A). Para atingir o objetivo deste trabalho, alguns

passos foram definidos. O primeiro passo foi fazer a definição da metodologia. Em seguida foi fazer o planejamento da execução da metodologia PCPL&A. Com a definição e o planejamento já estabelecidos, pudemos então executar a metodologia proposta. Para cada nível, foram propostos uma atividade e um questionário para coleta de dados. E o último passo foi fazer a análise descritiva da aplicação da metodologia proposta.

O presente trabalho apresentou a definição, aplicação e avaliação de uma proposta metodológica em uma escola pública municipal de Quixadá com o intuito de facilitar a difusão do conhecimento de programação e raciocínio lógico computacional para adolescentes do ensino fundamental II. Essa iniciativa buscou disseminar o conhecimento tecnológico por meio de alguns níveis ou etapas, para que o conhecimento seja passado de maneira mais suave e assim respeitando o entendimento dos alunos, não os submetendo a um conhecimento solidificado de início. Também foram usadas algumas estratégias para obter entendimento e percepção de cada nível da proposta metodológica.

Durante a execução dos passos, foram encontradas algumas dificuldades. Essas dificuldades surgiram desde o início da proposta metodológica, com a dificuldade de encontrar uma escola/ambiente, alunos com disposição para participar, além de desafios em relação ao material e à evasão. Notamos que há dificuldades em manter os participantes motivados e dispostos até o final da proposta metodológica.

Este trabalho mostrou a necessidade e a importância de uma nova maneira de ensinar programação e raciocínio lógico computacional, envolvendo um meio mais lúdico e incremental para atingir determinada lógica nos alunos antes de chegar à programação, já que ela normalmente é limitada a uma sintaxe complexa e pouco envolvente para um novato em computação. Conclui-se também que é necessária uma mudança mais radical no ensino de programação para obter novos resultados e assim deixar o estudante de computação mais motivado.

A maior contribuição deste trabalho foi a percepção dos alunos submetidos a proposta metodológica de conseguir associar e reutilizar conhecimentos de um nível anterior para o seu próximo nível, facilitando e dando base de programação e raciocínio lógico computacional para os alunos, com isso amadurecendo uma abstração para a computação.

Como trabalho futuro, serão realizadas duas mudanças. A primeira mudança é que estamos planejando adicionar um novo nível, que seria o pseudocódigo ou “portugol”. A segunda mudança seria experimentar uma troca na tríade PCPL&A pela tríade PLPC&A.

Assim apresentamos nossa ideia da importância de ensinar programação e raciocínio lógico computacional de uma maneira inovadora, abordando níveis lógicos

computacionais de uma maneira inovadora, abordando níveis graduais/incrementais antes de chegar a um conhecimento sólido de programação.

REFERÊNCIAS

- ALMEIDA, E. S.; HERRERAL, J.D.; FILHO, L.J. S.; ALMEIDA, H. O., COSTA, E.B.; VIEIRA, B.L.; MELO, M.D. **Um Ambiente Integrado para auxílio ao Ensino de Ciência da Computação**, In: Revista Digital da CVA, Vol. 2 – nº8.,2004.
- AURELIANO, V. C. O.; TEDESCO, P. C. A. R. **Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação**. In: 20º Workshop sobre Educação em Computação, Curitiba, Brasil., 2012.
- BARCELOS, T. S.; SILVEIRA, I. F. **Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica** . In: XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC., 2012.
- BENNEDSEN, J.; CASPERSEN, M.E. **Recalling Programming Competence**, Proceedings do 9th Koli Calling International Conference on Computing Education Research, Koli National Park, Finland, p. 86-95., 2009.
- CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., and STEIN, C. (2002). **Algoritmos - Teoria e Prática**. Campus, Rio de Janeiro, 2 edition.
- DE SOUZA, C. S.; GARCIA, A. C. B.; SLAVIEIRO, C.; PINTO, H.; REPENNING, A. **Semiotic traces of computational thinking acquisition**. Third International Symposium on End-User Development, IS-EUD. Torre Canne (BR), Italy: Springer Berlin Heidelberg. 2011. p. 155-170.
- DE SOUZA, C. S.; SALGADO, L. C. D. C.; LEITÃO, C. F.; SERRA, M. M. **Cultural appropriation of computational thinking acquisition research: seeding fields of diversity**. ITiCSE '14 Proceedings of the 2014 conference on Innovation & technology in computer science education. New York, NY, USA: ACM. 2014. p. 117-122.
- JÚNIOR, J. C. R. P.; RAPKIEWICZ, C. E. (2004). **O Processo de Ensino e Aprendizagem de Algoritmos e Programação: Uma Visão Crítica da Literatura**, III Workshop de Educação em Computação e Informática do estado de Minas Gerais (WEIMIG' 2004). Belo Horizonte, MG, Brasil.
- LEWIS, C. M.; SHAH, N. **Building Upon and Enriching Grade Four Mathematics Standards with Programming Curriculum**. SIGCSE '12. Proceedings of SIGCSE 2012. New York: ACM, 2012.
- MALAN, D. J.; LEITNER, H. H. **Scratch for budding computer scientists**. Proceedings do 38th SIGCSE'07, Kentucky, USA, p. 223–227., 2007.
- MURATET, M.; TORGUET, P.; JESSEL, J.-P.; VIALLET, F. Towards a serious game to help students learn computer programming. **International Journal of Computer Games Technology**, v. 2009, p. 3:1–3:12, jan 2009.
- SETTI, M. DE O. G. **O Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos**. Tese (Doutorado em Educação). Curitiba: UFPR, 2009.

SETÚBAL, J. C. (2000). **Uma proposta de Plano Pedagógico para a Matéria de Computação e Algoritmos**, Anais do II Curso: Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI 2000). Editora Universitária Champagnat.

SILVA, M.A.B., FAVERO, E.L.(2005). **Compiladores e Interpretadores Uma Abordagem Prática**.

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

**APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO
APRESENTADO AOS PARTICIPANTES DA APLICAÇÃO DA
METODOLOGIA**

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Somos uma equipe da Universidade Federal do Ceará, Campus Quixadá, que está participando de um projeto de extensão, cujo nome é “Programação nas escolas”, que objetiva ensinar e envolver os participantes com a tecnologia da informação. Nessa etapa do projeto, queremos conhecer as pessoas que irão participar dele, que envolve ensino de programação e jogos lúdicos, como uma ferramenta no auxílio da aprendizagem e conhecer o que esse ensino pode melhorar em suas vidas.

Estamos realizando uma série de pesquisas e solicitamos seu consentimento para a realização e gravação de entrevista/questionários/ aulas com seus filhos. Para decidir sobre o seu consentimento, é importante que você conheça as seguintes informações sobre a pesquisa:

- Os dados coletados durante a entrevista destinam-se estritamente e unicamente a atividades de análise e desenvolvimento do projeto programação nas escolas.
- Nossa equipe tem o compromisso de divulgar os resultados de nossas pesquisas para nos fóruns científicos da área. A divulgação desses resultados pauta-se no respeito à sua privacidade, e o anonimato dos participantes será preservado em quaisquer documentos que elaborarmos.
- O consentimento para a entrevista é uma escolha livre, feita mediante a prestação de todos os esclarecimentos necessários sobre a pesquisa.
- A entrevista pode ser interrompida a qualquer momento, segundo a sua disponibilidade e vontade.
- Nossa equipe encontra-se disponível para contato através do seguinte e-mail: leonardobrendo@hotmail.com

De posse dessas informações e contribuições, gostaríamos que você se pronunciasse acerca da entrevista com seu(s) filho(s) :

() Dou meu consentimento para a sua realização.

() Não consinto com a sua realização.

Nome do participante: _____

Nome do responsável: _____

_____, ____/____/____.
Local **Data**

Assinatura do responsável

Assinatura do bolsista

APÊNDICE B – QUESTIONÁRIO PERCEPÇÃO DE CONHECIMENTO

Aluno: _____

Série: _____

Sexo: _____ **Idade:** _____

01^a) Em qual dos seguintes dispositivos você possui experiência?

- Computador
- Celular comum
- Celular do tipo *smartphone*
- Tablet*
- Console de vídeo game
- Outra(s): _____
- Não possuo experiência com tecnologia.

02^a) Com que frequência você utiliza um ou mais dos dispositivos citadas acima?

- Todos os dias da semana
- Entre 4 e 6 dias na semana
- Entre 2 e 3 dias na semana
- Uma vez por semana
- Entre uma e três vezes por mês
- Muito raramente
- Não uso.

03^a) Em que disciplinas/matérias você usa tecnologias para estudar?

- Português
- Matemática
- Ciências
- Biologia
- Física
- Química
- História
- Geografia

Outra(s): _____

Não uso tecnologias para estudar.

04º) De que formas as tecnologias ajudam você em seus estudos?

05ª) Queremos saber quais disciplinas você mais gosta na escola. Na lista abaixo, coloque números de 1 a 9 em cada disciplina, de acordo com sua preferência. A disciplina que você mais gosta deve ficar com “1” e a que você menos gosta deve ficar com “9”. Caso você queira mencionar uma disciplina que não está listada, preencha a linha “Outra”.

Português

Matemática

Ciências

Biologia

Física

Química

História

Geografia

Outra: _____

06ª) Em quais destas tecnologias você possui interesse em aprender a desenvolver/criar:

Jogos eletrônicos (ex.: Minicraft, LOL, Just Dance)

Aplicativos para *smartphone* (ex.: WhatsApp, Messenger, SnapChat).

Páginas de Sites na internet (ex.: Google, Facebook, Twitter).

Outra(s): _____

07ª) Qual seu conhecimento em programação ou códigos de computador?

Conheço bastante.

Conheço um pouco.

Só ouvir falar.

Nunca nem ouvi falar.

08ª) Você conhece ou já ouviu falar de alguma linguagem e/ou ferramenta de programação?

Sim, já ouvi falar dessas: _____

Não conheço nada sobre isso.

09ª) Você possui alguém na família ou conhecido que trabalha com programação de computadores?

Sim.

- Não.
- Não sei.

10ª) Já assistiu algum filme/série sobre programação de computadores?

- Sim.
- Não, mas já ouvi falar de um filme sobre esse assunto.
- Não e nunca ouvi falar de um filme sobre esse assunto.

11ª) Sobre qual dessas pessoas você já ouviu falar?

- Bill Gates Steve Jobs Mark Zuckerberg Linus Torvalds
- Alan Turing Jack Dorsey Michel Krieger Eduardo Saverim
- Nenhum deles

12ª) Por que você está fazendo este curso?

13ª) O que você pretende/espera aprender neste curso?

14ª) Você pretende estudar/seguir na área de tecnologia?

- Sim, com certeza.
- Talvez.
- Não sei, ainda estou pensando sobre o assunto.
- De jeito nenhum.

APÊNDICE C – QUESTIONÁRIO SOBRE O NÍVEL PC DA TRÍADE PCPL&A

- 1) O que você aprendeu nas aulas?
- 2) Qual a importância para sua vida?
- 3) Você teve dificuldade? Em quê?
- 4) Tem interesse em seguir nessa área?
- 5) O que você não sabia sobre Pensamento Computacional?

APÊNDICE D – QUESTIONÁRIO SOBRE O NÍVEL PL DA TRÍADE PCPL&A

- 1) O que você aprendeu nas aulas?
- 2) Qual a importância para sua vida?
- 3) Você teve dificuldade? Em quê?
- 4) Tem interesse em seguir nessa área?
- 5) O que você não sabia sobre Programação Lúdica?

APÊNDICE E – QUESTIONÁRIO SOBRE O NÍVEL A DA TRÍADE PCPL&A

- 1) O que você aprendeu nas aulas de algoritmos?

- 2) **Qual a importância para sua vida?**
- 3) **Você teve dificuldade? Em quê?**
- 4) **Tem interesse em seguir nessa área?**
- 5) **O que você não sabia sobre Algoritmos?**
- 6) **Você gostou de algoritmos?**