

Estado da publicação: Não informado pelo autor submissor

PENSAMENTO COMPUTACIONAL PARA ENSINO DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO EM CURSOS DE TECNOLOGIA EAD

Berenice Milani, Alessandra Dutra, Kheronn Khennedy Machado

<https://doi.org/10.1590/SciELOPreprints.10437>

Submetido em: 2024-11-15

Postado em: 2024-12-04 (versão 1)

(AAAA-MM-DD)

A moderação deste preprint recebeu o endosso de:

Givan José Ferreira dos Santos (ORCID: <https://orcid.org/0000-0003-4104-9313>)

ARTIGO

PENSAMENTO COMPUTACIONAL PARA ENSINO DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO EM CURSOS DE TECNOLOGIA EAD

BERENICE MILANI¹

ORCID: <https://orcid.org/0009-0005-5261-6266>
<beremilani@hotmail.com>

ALESSANDRA DUTRA²

ORCID: <https://orcid.org/0000-0001-5119-3752>
alessandradutra@utfpr.edu.br

KHERONN KHENNEY MACHADO³

ORCID: <https://orcid.org/0000-0003-1382-9099>
kheronn@gmail.com

¹ Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação Stricto Sensu em Ensino de Ciência e Tecnologia. Maringá- PR, Brasil.

² Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação Stricto Sensu em Ensino de Ciência e Tecnologia. Londrina - PR, Brasil.

³ Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação Stricto Sensu em Ensino de Ciência e Tecnologia. Londrina - PR, Brasil.

RESUMO: A Educação a Distância (EaD) tem se estabelecido como forma de democratizar o acesso à educação, especialmente para indivíduos que residem em áreas remotas, enfrentam barreiras geográficas ou necessitam de flexibilidade de horários de estudo. Além disso, no que se refere à aprendizagem de conteúdos, alunos sem base sólida em matemática e/ou tecnologia podem demonstrar dificuldades em resolver problemas de forma lógica e estrutural, o que afeta o aprendizado de conteúdos importantes para a formação desses estudantes. Pensando na formação dos profissionais da área da Tecnologia da Informação, uma das estratégias que pode ser empregada para motivar e auxiliar alunos das séries iniciais de um curso de tecnologia a distância a compreender conteúdos relacionados à programação de dados é o Pensamento Computacional. Com base nessas considerações, este estudo propõe empregar os fundamentos decomposição, reconhecimento de padrões, abstração e algoritmo para auxiliar estudantes do 1º ano do curso à distância “Análise e Desenvolvimento de Sistemas” a compreender conteúdos da disciplina de "Algoritmos e Lógica de Programação". Para isso, realizou-se pesquisa bibliográfica, por meio da base de dados Scopus e Scielo e pesquisa-ação, por meio da qual foi elaborada atividade a ser ofertada a estudantes ingressantes do curso. Esperamos que a proposta possa contribuir efetivamente para o desenvolvimento das habilidades importantes dos futuros programadores para que eles possam atuar de modo efetivo e com propriedade na área de tecnologia da informação.

Palavras-chave: pensamento computacional, educação a distância, ensino de tecnologia.

**COMPUTATIONAL THINKING FOR TEACHING ALGORITHM AND PROGRAMMING LOGIC IN
DISTANCE LEARNING TECHNOLOGY COURSES**

ABSTRACT: Distance Education (DE) has established itself as a means to democratize access to education, especially for individuals residing in remote areas, facing geographical barriers, or requiring flexibility in study schedules. Additionally, regarding content learning, students lacking a solid foundation in mathematics and/or technology may encounter difficulties in solving problems logically and structurally, impacting the learning of essential content for their development. With a focus on training professionals in the field of Information Technology, one strategy that can be employed to motivate and assist early-year students in a distance technology course to understand programming-related content is Computational Thinking (CT). Based on these considerations, this study proposes to employ the fundamentals of decomposition, pattern recognition, abstraction, and algorithm to help first-year students in the distance course "Systems Analysis and Development" understand content in the "Algorithms and Programming Logic" discipline. To this end, bibliographic research was conducted through the Scopus and Scielo databases, along with action research through which an activity was designed for first-year students in the course. We hope the proposal can effectively contribute to developing key skills in future programmers, enabling them to perform effectively and competently in the field of information technology.

Keywords: computational thinking, distance education, technology teaching.

PENSAMIENTO COMPUTACIONAL PARA LA ENSEÑANZA DE ALGORITMOS Y LÓGICA DE PROGRAMACIÓN EN CURSOS DE TECNOLOGÍA DE APRENDIZAJE A DISTANCIA

RESUMEN: La Educación a Distancia (EaD) se ha establecido como una forma de democratizar el acceso a la educación, especialmente para individuos que residen en áreas remotas, enfrentan barreras geográficas o necesitan flexibilidad en los horarios de estudio. Además, en lo que se refiere al aprendizaje de contenidos, los estudiantes sin una base sólida en matemáticas y/o tecnología pueden presentar dificultades para resolver problemas de manera lógica y estructurada, lo cual afecta el aprendizaje de contenidos importantes para la formación de estos estudiantes. Pensando en la formación de profesionales en el área de Tecnología de la Información, una de las estrategias que puede emplearse para motivar y ayudar a estudiantes de los primeros años de un curso de tecnología a distancia a comprender contenidos relacionados con la programación de datos es el Pensamiento Computacional. Con base en estas consideraciones, este estudio propone emplear los fundamentos de descomposición, reconocimiento de patrones, abstracción y algoritmos para ayudar a los estudiantes de primer año del curso a distancia de "Análisis y Desarrollo de Sistemas" a comprender contenidos de la asignatura "Algoritmos y Lógica de Programación". Para ello, se realizó una investigación bibliográfica, utilizando las bases de datos Scopus y Scielo, y una investigación-acción, a través de la cual se elaboró una actividad para ser ofrecida a los estudiantes recién ingresados al curso. Esperamos que la propuesta contribuya efectivamente al desarrollo de habilidades importantes de futuros programadores, permitiéndoles actuar de manera eficaz y competente en el área de tecnología de la información.

Palabras clave: pensamiento computacional, educación a distancia, enseñanza de tecnología.

INTRODUÇÃO

A Educação a Distância (EaD) tem se mostrado um meio para democratizar o acesso à educação, particularmente, em um país de proporções continentais como o Brasil.

Indivíduos que residem em áreas remotas ou que enfrentam limitações de recursos encontram no EaD uma oportunidade de continuar os estudos. Além de romper barreiras geográficas,

esta modalidade permite a flexibilização dos horários e a personalização do aprendizado, aspectos especialmente pertinentes em um cenário educacional cada vez mais diversificado e inclusivo.

Este modelo educacional, apoiado pelo avanço tecnológico e a evolução dos métodos pedagógicos, tem mostrado eficácia na oferta de programas educacionais que atendam às necessidades variadas de uma população estudantil heterogênea, muitas vezes, já inseridas no mercado de trabalho e sem formação em sua área de atuação ou sem condições de se mudar para centros urbanos em busca de educação formal. Nesses casos, a educação a distância oferece a oportunidade de obter formação superior.

De acordo com o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira - INEP, o censo da educação superior, realizado em 2022, aponta que o número de concluintes em cursos de graduação em 2012 foi de 1.050.413, 83,40%, de forma presencial, e 16,58%, à distância. Em 2022, o número de concluintes foi de 2.287.456, 62,42%, de forma presencial e 37,58%, à distância (INEP, 2024). Os dados indicam que o ensino em EaD tem se mostrado importante na democratização do acesso à educação superior, particularmente para indivíduos de regiões remotas ou com recursos limitados.

No entanto, muitos estudantes que buscam o ensino à distância enfrentam problemas que prejudicam seu desempenho nesta modalidade, como a dificuldade de lidar com as tecnologias digitais de informação e comunicação, o acesso restrito à internet, a falta de dispositivos adequados (computador, tablet, entre outros) e dificuldades em lidar com plataformas de ensino on-line.

A diversidade de proficiência tecnológica entre os alunos do EaD requer o uso de abordagens pedagógicas adaptativas que contemplem as especificidades e os desafios enfrentados por esses estudantes. Essa preocupação parte da quantidade de disciplinas práticas ofertadas pelos cursos voltados à Tecnologia da Informação (TI), entre elas, as disciplinas que envolvem desenvolvimento de algoritmos. Para o aluno com pouco acesso à internet e com recursos limitados, a aprendizagem pode ser prejudicada, restringindo o desenvolvimento de habilidades e competências técnicas desse estudante ao concluir a graduação.

Diante deste cenário, uma opção que pode ser empregada a fim de auxiliar estudantes ingressantes a resolver problemas e desafios de forma eficiente e compreender conteúdos de disciplinas basilares de cursos de tecnologia, na modalidade de ensino à distância, é o Pensamento Computacional (PC).

Pelo fato de uma das autoras deste estudo ministrar a disciplina de Algoritmos e Lógica de Programação a estudantes do curso à distância “Análise e Desenvolvimento de Sistemas” de uma universidade localizada ao norte do Paraná e estar preocupada com as dificuldades que estes alunos enfrentam na aprendizagem de conceitos básicos de algoritmos, estrutura condicional e estrutura de repetição, estrutura de dados homogêneas, estrutura de dados heterogêneas, sub-rotinas e funções, surgiu

o interesse em ofertar uma proposta empregando os fundamentos do PC para auxiliar estes estudantes a compreender estes conteúdos.

Partimos da hipótese de que o PC pode auxiliar os estudantes a desenvolver a habilidade de expressar os algoritmos em forma de descrição narrativa, fluxograma e pseudocódigo, servindo como um passo intermediário entre o pensamento do problema (lógica) e a escrita do código efetivo em uma linguagem de programação.

Conforme descrito por Souza et al. (2019), o português estruturado, também conhecido como “portugol” ou pseudocódigo, é uma forma textual de representação de algoritmos que emprega um subconjunto de palavras da língua portuguesa para demonstrar as estruturas de programação. Esta técnica é essencial para facilitar o entendimento e a aprendizagem inicial da lógica de programação, servindo como um ponto de partida eficaz para iniciantes na área de Tecnologia da Informação.

A lógica de programação, por sua vez, é a habilidade de sequenciar esses algoritmos de uma maneira que o computador possa executar. Manzano e Oliveira (2019) relatam que, no âmbito computacional, o termo "algoritmo" refere-se a um conjunto de regras formais, sequenciais e claramente definidas. Os autores ressaltam que estas regras surgem a partir da compreensão lógica de um problema específico que o programador pretende converter em um programa computacional. Tal programa, então, é capaz de executar operações em que dados de entrada são transformados em dados de saída. Portanto, ao aprender algoritmo, o aluno desenvolve a capacidade de entender e aplicar lógica de programação em qualquer contexto tecnológico.

Como o PC é visto como uma forma de ampliar a capacidade analítica e de resolução de problemas dos indivíduos, vemos a possibilidade de empregá-lo para auxiliar os estudantes de um curso a distância a compreender conteúdos de disciplinas basilares na área de programação.

Seymour Papert, do Massachusetts Institute of Technology, segundo Beecher (2017), foi um dos precursores no desenvolvimento de conceitos que fundamentam o que hoje entendemos por Pensamento Computacional (PC). Em seu livro "Mindstorms: Children, Computers, and Powerful Ideas", Papert introduziu a ideia de "pensamento processual", uma abordagem que apresenta o uso de computadores como ferramentas facilitadoras no ensino e na resolução de problemas, não meramente como executores de tarefas mecânicas (Papert, 1980).

O PC ganhou visibilidade e um novo impulso com a palestra de Jeannette Wing, em 2006, na qual o destacou como uma habilidade essencial, podendo ser integrado efetivamente nos currículos escolares, destacando seu potencial para além dos limites tradicionais da programação e ciência da computação.

Wing (2006) descreve o pensamento computacional como um processo que engloba a resolução de problemas, a projeção de sistemas e a compreensão do comportamento humano, por meio da aplicação de conceitos fundamentais da ciência da computação. Segundo a autora, o pensamento

computacional é constituído por uma variedade de ferramentas mentais que refletem a amplitude do campo da ciência da computação.

Pasqual Junior (2020) ressalta a importância fundamental do pensamento computacional, afirmando que "Pensar computacionalmente consistirá em uma necessidade para estar incluído na sociedade, na escola e no mundo do trabalho." Ele enfatiza que a capacidade de sistematizar soluções através de algoritmos está se tornando cada vez mais essencial. Embora possa parecer uma afirmação ambiciosa, é inegável que a habilidade de resolver problemas utilizando ferramentas computacionais e lógica computacional está cada vez mais presente no cotidiano de diversas profissões. Esta integração destaca a relevância significativa do PC no campo da tecnologia, assim como uma competência essencial em uma ampla gama de atividades profissionais, refletindo sua aplicabilidade e necessidade transdisciplinares.

O PC é essencial para cientistas da computação e tem demonstrado aplicabilidade extensiva em diversos campos educacionais (Kaleem, Hassan e Khurshid, 2024). Segundo os autores, sua relevância crescente para projetar e resolver sistemas complexos tem motivado várias universidades a buscar a integração efetiva do PC e da computação em seus currículos, refletindo sua importância estratégica na formação acadêmica e profissional.

Wu et al. (2024) detalham os quatro componentes essenciais do pensamento computacional. Segundo eles, a decomposição, que representa a primeira fase, consiste em fracionar um problema complexo em partes menores e mais gerenciáveis. A segunda fase, o reconhecimento de padrões, envolve a identificação de semelhanças, repetições ou características únicas dentro do problema. A abstração tem seu foco nas ideias principais e nos elementos essenciais da solução. Por fim, o algoritmo sintetiza as várias partes da solução em uma sequência lógica, formando uma estrutura unificada que aborda o problema principal. Kale et al. (2018) descrevem que, no contexto da resolução de problemas, esta etapa engloba a implementação prática, enquanto, sob a perspectiva do conhecimento, ela é considerada um procedimento.

Conforme destacado por Kaleem, Hassan e Khurshid (2024), o PC além de fortalecer a capacidade de projetar e resolver sistemas complexos, também amplia o alcance educacional do estudante em diversas áreas do conhecimento. Arfé et al. (2019) enfatizam que o pensamento computacional potencializa a capacidade de avaliar problemas, estruturar ações e encontrar soluções, o que é fundamental para estudantes que enfrentam desafios algorítmicos e de programação.

Embora o pensamento computacional (PC) transcenda o âmbito da computação, conforme ressaltado por diversos estudiosos como Jeannette Wing (2006) e Seymour Papert (Papert, 1980; Beecher, 2017), para estudantes na área de tecnologia, adquirir essa habilidade é importante. Pasqual Junior (2020) e Bers et al. (2023) destacam a relevância de incorporar o PC na formação acadêmica como uma

ferramenta pedagógica que gera competências essenciais para a preparação de profissionais aptos a resolver desafios contemporâneos com soluções sistemáticas e inovadoras.

O PC pode ser empregado usando as modalidades plugada, com uso de computadores, tablets e internet e desplugada, sem a necessidade de recursos digitais e conexão à internet. Brackmann et al. (2017) destacam que, muitas atividades podem ser realizadas sem o uso de dispositivos eletrônicos, proporcionando um meio alternativo e acessível para o desenvolvimento das competências associadas ao PC.

Pensando na formação dos profissionais da área da Tecnologia da Informação, a integração do Pensamento Computacional plugado e desplugado como apoio na aprendizagem da construção algoritmos em pseudocódigo da disciplina de "Algoritmos e Lógica de Programação" pode contribuir com o aprendizado desses estudantes, de forma a relacionar a teoria à prática de desenvolvimento de *software*.

MATERIAL E MÉTODOS

Os tipos de pesquisa empregados neste estudo são bibliográfica, pois foram utilizados livros e artigos científicos selecionados da base de dados Scopus e Scielo; e pesquisa-ação, uma vez que foi detectado um problema a ser investigado no âmbito do ensino de Algoritmos da disciplina de Algoritmo e Lógica de Programação de uma universidade no norte do Paraná e elaborada proposta de uso do PC para auxiliar os estudantes a compreender os algoritmos em pseudocódigos.

Muitas instituições de ensino superior à distância adotam um sistema de disciplinas baseado em módulos. Na instituição onde a pesquisa foi realizada, o ano letivo é dividido em quatro módulos, permitindo que novos alunos ingressem em qualquer um deles.

A proposta será estruturada para integrar a carga-horária de um destes módulos e será constituída por duas partes. A primeira delas corresponde a uma aula expositiva gravada onde serão apresentados os fundamentos do PC para aprender o conteúdo da disciplina de Algoritmo e Lógica de Programação e foi estruturada em três etapas: (i) fundamentação teórica sobre o PC, explicação sobre seu surgimento e sua relevância em todas as áreas de atuação; (ii) apresentação dos algoritmos (propriedade, finitude, definição, entrada, processamento, saída), e estruturas de controle (condicional, de repetição, homogênea, heterogênea), funções com introdução sobre a lógica de programação e sua estrutura; (iii) demonstração da aplicação dos fundamentos do Pensamento Computacional para realização de três exercícios de algoritmos em pseudocódigo.

A segunda parte corresponde a realização de uma Coleção de Casos Práticos de Pseudocódigos para os alunos realizarem de forma desplugada.

RESULTADOS E DISCUSSÃO

Nesta seção, será apresentada a proposta da atividade de uso do pensamento computacional a ser disponibilizado aos estudantes ingressantes de um curso de Análise e desenvolvimento de sistemas - modalidade EaD, para auxiliar a aprendizagem de algoritmos em pseudocódigo a estudantes ingressantes do curso de Análise e Desenvolvimento de Sistemas, modalidade EaD, de uma universidade situada ao norte do Paraná.

Etapas da Proposta

Na etapa 1 propõe-se uma discussão sobre o Pensamento Computacional, visando familiarizar os estudantes com o conceito, seus fundamentos e suas amplas aplicações, destacadas por Papert (1980), Wing (2006), Pasqual Junior (2020), Bers et al. (2023), Wu et al. (2024), Kaleem, Hassan e Khurshid (2024) e Beecher (2017). Esta sessão visa introduzir aos alunos os fundamentos do PC - decomposição, reconhecimento de padrões, abstração e algoritmos - utilizando exemplos práticos do cotidiano para ilustrar a relevância do PC fora do contexto da programação. Esta abordagem preparará os estudantes para a prática subsequente, facilitando a transição para aplicações mais técnicas.

A etapa 2 é composta pela explanação do que são algoritmos, abordando suas propriedades essenciais e as estruturas de controle. Essa fundamentação teórica é importante para que os estudantes reconheçam que, apesar da diversidade de linguagens de programação disponíveis, todas seguem uma base lógica comum. Compreender a construção de algoritmos em pseudocódigo nessa fase equipa os alunos com a base lógica e analítica necessária para programar em diversas linguagens.

A terceira parte da etapa 1 envolve práticas projetadas para consolidar o entendimento dos alunos sobre os conceitos discutidos. Propõe-se a elaboração de quatro exercícios de algoritmo, utilizando os fundamentos do Pensamento Computacional. Essas atividades práticas são destinadas a reforçar a compreensão das estruturas de programação e a aplicação efetiva do Pensamento Computacional, proporcionando aos alunos uma experiência mais concreta e aplicável.

Ao realizar esses exercícios, os alunos aplicam o que aprenderam sobre a estrutura e propriedades dos algoritmos e praticam a tradução de soluções narrativas em formatos estruturados e próximos das linguagens de programação. Esta experiência prática é essencial para facilitar a transição dos estudantes para a programação aplicada, preparando-os para enfrentar desafios mais complexos na área de tecnologia e computação.

O exercício 1 envolve a criação de algoritmos em pseudocódigo para somar dois números. Ele reforça a compreensão dos alunos sobre as propriedades e estruturas de controle de um algoritmo, essenciais na disciplina de "Algoritmos e Lógica de Programação".

A atividade inicia-se pelo processo de decomposição, onde se quebra o problema da soma em passos menores e mais gerenciáveis. Em seguida, incentivam-se os estudantes a reconhecerem padrões que surgem frequentemente em tarefas de programação, o que aumenta a compreensão sobre a estruturação lógica dos programas. A abstração é empregada para atender aos aspectos essenciais do problema, descartando informações secundárias. Por fim, a construção de algoritmos é demonstrada através do exercício demonstrado na figura 1, orientando os alunos a ordenar claramente as instruções de captura de entrada, processamento de dados e produção de saída.

Figura 1 - Pseudocódigo – Algoritmo soma

```
INÍCIO  
  // Declaração de variáveis  
  declare num1, num2, resultado como inteiro  
  
  // Obter dados de entrada  
  ESCREVA("Digite o primeiro número:")  
  leia(num1)  
  ESCREVA("Digite o segundo número:")  
  leia(num2)  
  
  // Processamento  
  resultado ← num1 + num2  
  
  // Saída  
  ESCREVA("O resultado da soma é: ", resultado)  
FIM
```

Fonte: os autores (2024)

O exercício 2 propõe a construção de um algoritmo em pseudocódigo com uma estrutura condicional simples para verificar se a soma de dois números é maior ou menor que um limite predefinido.

Utilizando os fundamentos do PC na resolução deste exercício, inicia-se pela decomposição do problema em partes menores. O problema é dividido em etapas menores para facilitar sua resolução. O algoritmo começa pedindo que o usuário insira dois números e, em seguida, realize a soma desses números. O próximo passo é verificar se a soma é maior ou menor que um valor limite (neste caso, 10). Ao analisar o código, os alunos podem perceber um padrão comum: a estrutura "Se... então... senão",

que é usada em vários problemas para tomar decisões. Esse padrão pode ser reutilizado em diversas situações em algoritmos que envolvem comparações e tomadas de decisão.

A abstração é utilizada para atender aos aspectos importantes do problema, como a comparação entre o resultado da soma e o limite definido. O algoritmo completo é formado por uma sequência de passos bem definidos: primeiro, recebe dois números do usuário, depois faz a soma, verifica se o resultado é maior ou menor que o limite e, por fim, imprime uma mensagem adequada com base na condição avaliada.

Figura 2 - Pseudocódigo - Exemplo estrutura condicional simples

```
INÍCIO
  // Condicional para verificar se o resultado é maior que 10

  // Declaração de variáveis
  declarar num1, num2, resultado, limite como inteiro

  // Entrada de dados
  ESCREVA("Digite o primeiro número:")
  leia(num1)
  ESCREVA("Digite o segundo número:")
  leia(num2)

  // Definição do limite
  limite ← 10

  // Processamento: soma dos números
  resultado ← num1 + num2

  // Condicional para verificar o resultado
  SE resultado > limite ENTÃO
    ESCREVA("O resultado é maior que ", limite)
  SENÃO
    ESCREVA("O resultado é menor ou igual a ", limite)
  FIM SE
FIM
```

Fonte: os autores (2024)

O exercício 3 envolve a criação de algoritmos em pseudocódigo de laço de repetição.

Inicia-se o processo com a decomposição dos números. O problema de contar números até um valor informado pelo usuário é decomposto em duas etapas principais: Primeiro, o valor limite é capturado. Segundo, um laço de repetição é utilizado para realizar a contagem, incrementando a variável "soma" até atingir o limite. O algoritmo é o laço "ENQUANTO", que repete a operação de incremento da variável "soma" até que a condição "soma <= limite" se torne falsa. O processo de repetição é claro

e ordenado, fornecendo uma sequência de passos bem definidos: captura do limite, inicialização da soma, contagem e exibição dos resultados, até o término do loop.

Figura 3 - Pseudocódigo - Exemplo Laço de repetição

```
INÍCIO
  // Laço enquanto para contar números até um limite

  // Declaração de variáveis
  declarar soma, limite

  // Entrada de dados
  ESCREVA("Digite um número:")
  leia(limite)
  soma ← 0

  // Laço de repetição
  ENQUANTO soma <= limite FAÇA
    soma ← soma + 1
    ESCREVA("Contando:", soma)
  FIM ENQUANTO
FIM
```

Fonte: os autores (2024)

O exercício 4 vamos expandir o primeiro exemplo para introduzir o uso de função. Esse exemplo é factível para introduzir o conceito de funções, mostrando como uma tarefa específica (como somar dois números) pode ser isolada em uma função.

A função recebe dois parâmetros a e b, realiza a soma e retorna o resultado. No programa principal, os valores são lidos e, em seguida, passados para a função soma, que realiza a operação e retorna o resultado.

A lógica de somar dois números está na função soma, tornando o algoritmo mais organizado e reutilizável. O primeiro conceito de PC aplicado está na decomposição, isto é, na leitura dos números e na soma por meio de uma função. A abstração está presente na função soma () que esconde os detalhes do processo de adição, permitindo que o algoritmo principal apenas faça a chamada da função e obtenha o resultado.

Figura 4 - Pseudocódigo - Exemplo Introdução ao uso de função

```
Algoritmo "Soma de Dois Números"
Var
    num1, num2, resultado: inteiro

Funcao soma(a: inteiro, b: inteiro): inteiro
Var
    resultado: inteiro
Início
    resultado <- a + b // A função realiza a soma
    retorne resultado // Retorna o resultado da soma para o programa principal
FimFuncao

Início
    escreva("Digite o primeiro número: ")
    leia(num1)
    escreva("Digite o segundo número: ")
    leia(num2)

    resultado <- soma(num1, num2) // Chama a função para somar os dois números
    escreva("A soma de ", num1, " e ", num2, " é: ", resultado)
Fimalgoritmo
```

Fonte: os autores (2024)

Os exercícios 1, 2, 3 e 4, ao incorporarem os fundamentos do PC, fortalecem a compreensão dos estudantes, trazendo subsídios para que eles apliquem esses conceitos na resolução de problemas, desde os mais simples aos mais complexos, na disciplina de Algoritmos e Lógica de Programação a estudantes do curso à distância Análise e Desenvolvimento de Sistemas.

Coleção de Casos Práticos de Pseudocódigos

A implementação de atividades desplugadas, como a realização de casos práticos de pseudocódigo abrangendo estruturas condicionais, de repetição, além de estruturas de dados homogêneas e heterogêneas, bem como sub-rotinas, é sugerida como uma estratégia pedagógica no contexto da disciplina de Algoritmos e Lógica de Programação, facilitando o acesso e a inclusão de todos os estudantes, independentemente de suas condições tecnológicas. O aluno pode imprimir essa coleção de casos práticos de pseudocódigos e realizá-la usando papel, caneta ou lápis. Essa atividade desplugada permite aos estudantes desenvolverem suas habilidades de pensamento lógico e algorítmico sem a necessidade de computadores.

A prática destes casos facilitará a compreensão dos alunos sobre as estruturas de controle e a lógica de programação, que são essenciais na área de TI.

CONSIDERAÇÕES FINAIS

Este estudo apresentou uma proposta de uso do Pensamento Computacional para auxiliar os estudantes na compreensão dos conteúdos de algoritmos da disciplina de 'Algoritmos e Lógica de Programação' do curso de Análise e Desenvolvimento de Sistemas à distância, utilizando atividades plugadas e desplugadas. A iniciativa visou desenvolver habilidades de PC através de atividades extracurriculares, preparando os estudantes para compreender e elaborar algoritmos.

Por meio dos fundamentos do Pensamento Computacional esta prática poderá auxiliar os estudantes a superarem dificuldades significativas e potencializarem sua aprendizagem em programação, ampliando sua capacidade de atuação no campo da tecnologia da informação.

REFERÊNCIAS

ARFÉ, B. et al. Coding in primary grades boosts children's executive functions. *Frontiers in psychology*, v. 10, 2019. DOI:<https://doi.org/10.3389/fpsyg.2019.02713>

BRACKMANN, C. P. et al. Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*. Anais...New York, NY, USA: ACM, 2017.

BEECHER, K. *Computational thinking: a beginner's guide to problem-solving and programming*. Swindon: BCS, 2017.

BERS, M. U. et al. Coding as another language: Research-based curriculum for early childhood computer science. *Early childhood research quarterly*, v. 64, p. 394–404, 2023. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0885200623000571>. Acesso em: 29 jun. de 2024.

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2022. Disponível em: <http://basenacionalcomum.mec.gov.br/>. Acesso em: 28 jun. 2024.

DENNING, P. Beyond computation thinking. *Communications of the ACM*, v. 52, n. 6, 2009.

DHAWAN, S. Online learning: A panacea in the time of COVID-19 crisis. *Journal of educational technology systems*, v. 49, n. 1, p. 5–22, 2020.

HEMMENDINGER, D. A plea for modesty. *ACM Inroads*, v. 1, n. 2, p. 4, 2010.

HUANG, X. et al. Understanding transactional distance in web-based learning environments: An empirical study. *British journal of educational technology: journal of the Council for Educational Technology*, v. 47, n. 4, p. 734–747, 2016. DOI: <http://dx.doi.org/10.1111/bjet.12263>. Acesso em: 20 jun. de 2024.

KALE, U. et al. Computational what? Relating computational thinking to teaching. *TechTrends: for leaders in education & training*, v. 62, n. 6, p. 574–584, 2018. DOI: <https://doi.org/10.1007/s11528-018-0290-9>

KALEEM, M.; HASSAN, M. A.; KHURSHID, S. K. A machine learning-based adaptive feedback system to enhance programming skill using computational thinking. IEEE access: practical innovations, open solutions, v. 12, p. 59431–59440, 2024. DOI: 10.1109/ACCESS.2024.3391873. Disponível em: <https://ieeexplore.ieee.org/document/3391873>. Acesso em: 21 de jul. de 2024.

INEP. Ensino a distância cresce 474% em uma década. Disponível em: <https://www.gov.br/inep/pt-br/assuntos/noticias/censo-da-educacao-superior/ensino-a-distancia-cresce-474-em-uma-decada>. Acesso em: 20 maio 2024.

INEP. Censo da Educação Superior 2022. Disponível em: <https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-da-educacao-superior/resultados>. Acesso em: 14 jun. de 2024.

MANZANO, J. A. N. G.; OLIVEIRA, J. F.. Algoritmos - Lógica para Desenvolvimento de Programação de Computadores. Rio de Janeiro: Grupo GEN, 2019. E-book. ISBN 9788536531472. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788536531472/>. Acesso em: 03 set. 2024.

PAPERT, S. Mindstorms: children, computers, and powerful ideas. New York: Basic Books Inc, 1980.
PASQUAL JÚNIOR, P. A. Pensamento computacional e tecnologias : reflexões sobre a educação no século XXI. Caxias do Sul, RS : Educs, 2020.

SOUZA, M. A. F.; GOMES, M. M.; SOARES, M. V.; CONCILIO, R. Algoritmos e lógica de programação: um texto introdutório para a engenharia. São Paulo: Cengage Learning Brasil, 2019. E-book. ISBN 9788522128150. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788522128150/>. Acesso em: 03 set. 2024.

WAHYUDIN, W. et al. Learning through computer science unplugged on team assisted individualization on the computational thinking ability. Linguistics and Culture Review, v. 5, n. S3, p. 1442–1452, 2021.

WAZLAWICK, Raul S. Metodologia de Pesquisa para Ciência da Computação. Rio de Janeiro: Grupo GEN, 2020. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788595157712/>. Acesso em: 17 jul. 2024.

WING, J. M. Computational thinking. Communications of the ACM, v. 49, n. 3, p. 33–35, 2006. DOI: <https://dl.acm.org/doi/fullHtml/10.1145/1118178.1118215>. Acesso em: 28 jun. 2024.

WING, J. PENSAMENTO COMPUTACIONAL – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. Revista brasileira de ensino de ciência e tecnologia, v. 9, n. 2, 2016.

WU, T.-T. et al. Identification of problem-solving techniques in Computational Thinking studies: Systematic literature review. SAGE open, v. 14, n. 2, 2024. DOI: 10.1177/21582440241249897. Disponível em: <<https://journals.sagepub.com/home/sgo>>. Acesso em: 21 jul. de 2024.

DECLARAÇÃO DE CONFLITO DE INTERESSE

Os autores declaram que não há conflito de interesse com o presente artigo.

CONTRIBUIÇÃO DAS/DOS AUTORES/AS

Autor1 – Conceituação, metodologia, investigação, análise dos dados e escrita – primeira redação.

Autora2 – Administração do projeto, análise, validação, supervisão e revisão da escrita final.

Autor3– Análise dos dados, discussão dos resultados e revisão.

Este preprint foi submetido sob as seguintes condições:

- Os autores declaram que estão cientes que são os únicos responsáveis pelo conteúdo do preprint e que o depósito no SciELO Preprints não significa nenhum compromisso de parte do SciELO, exceto sua preservação e disseminação.
- Os autores declaram que os necessários Termos de Consentimento Livre e Esclarecido de participantes ou pacientes na pesquisa foram obtidos e estão descritos no manuscrito, quando aplicável.
- Os autores declaram que a elaboração do manuscrito seguiu as normas éticas de comunicação científica.
- Os autores declaram que os dados, aplicativos e outros conteúdos subjacentes ao manuscrito estão referenciados.
- O manuscrito depositado está no formato PDF.
- Os autores declaram que a pesquisa que deu origem ao manuscrito seguiu as boas práticas éticas e que as necessárias aprovações de comitês de ética de pesquisa, quando aplicável, estão descritas no manuscrito.
- Os autores declaram que uma vez que um manuscrito é postado no servidor SciELO Preprints, o mesmo só poderá ser retirado mediante pedido à Secretaria Editorial do SciELO Preprints, que afixará um aviso de retratação no seu lugar.
- Os autores concordam que o manuscrito aprovado será disponibilizado sob licença [Creative Commons CC-BY](#).
- O autor submissor declara que as contribuições de todos os autores e declaração de conflito de interesses estão incluídas de maneira explícita e em seções específicas do manuscrito.
- Os autores declaram que o manuscrito não foi depositado e/ou disponibilizado previamente em outro servidor de preprints ou publicado em um periódico.
- Caso o manuscrito esteja em processo de avaliação ou sendo preparado para publicação mas ainda não publicado por um periódico, os autores declaram que receberam autorização do periódico para realizar este depósito.
- O autor submissor declara que todos os autores do manuscrito concordam com a submissão ao SciELO Preprints.