



**IX
CONINFA**
PENSAR E EXISTIR:
Um novo olhar sobre a
importância do ser.

Eixo temático: Governança de Dados

O DESENVOLVIMENTO DO RACIOCÍNIO LÓGICO NO APRENDIZADO DE PROGRAMAÇÃO

Renan Lima Ferraz¹; Victor Matheus Torres Feitoza¹; Franciely Alves de Souza².

INTRODUÇÃO

A programação se torna essencial para ajudar jovens a desenvolverem habilidades importantes, como a capacidade de identificar e resolver problemas, de forma criativa e eficiente. Aprender a tal arte permite que os estudantes pensem de forma lógica e estratégica, além de estimular o raciocínio crítico e o pensamento computacional.

O pensamento computacional é uma habilidade importante que ao ser desenvolvida pode ajudar na idealização e implementação de soluções para problemáticas vivenciadas no dia a dia. De acordo com a visão de Wing (2006), o pensamento computacional é o processo de reconhecer aspectos da computação no mundo ao redor, aplicar ferramentas, técnicas para entender, raciocinar sobre sistemas, processos naturais, sociais e artificiais.

Dentro do contexto escolar, a forma de pensar permite que os jovens programadores resolvam problemas por partes, abstraíam o que não é necessário para a conclusão de uma tarefa, sejam capazes de reconhecer padrões existentes em diferentes problemas e criem algoritmos para resolvê-los.

Segundo Selby e Woollard (2013), o pensamento computacional é um processo cognitivo ou de discernimento que envolve o raciocínio lógico pelo qual os problemas são resolvidos e os artefactos, procedimentos e sistemas são compreendidos. É através dele que aprendendo a

¹ Acadêmico da Faculdade de Sistemas de Informação do UNIRIOS;

² Mestre pela Universidade Federal Rural de Pernambuco – UFRPE. Docente do curso de Sistema de Sistemas UNIRIOS. E-mail: franciely.souza@unirios.edu.br



decompor problemas complexos em partes menores facilita encontrar soluções eficientes por meio da lógica e da criatividade.

Por meio das análises que rodam a programação básica, é possível compreender pensamentos lógicos que se moldam em estruturas condicionais, laços de repetição e operadores lógicos, compreendendo tais conceitos, é possível questionar como o desenvolvimento do raciocínio lógico influencia no aprendizado de programação.

OBJETIVO

Explorar e esclarecer conceitos fundamentais de lógica, partindo da criação de variáveis para o uso das estruturas condicionais (*if-else*), laços de repetição (*loops*) e operadores lógicos;

METODOLOGIA

Segundo Gonsalves (2001), os objetivos precisam ser analisados com base em fatos, tendo em vista a importância da verificação e o caminho mais preciso para que o objetivo seja alcançado. A revisão de literatura consiste na leitura e no estudo das fontes mais importantes do tema que será abordado, buscando um levantamento dos conceitos mais atuais, ou seja, um estudo minucioso em busca do conhecimento aprofundado sobre o assunto a ser estudado.

A pesquisa bibliográfica foi um dos métodos utilizados para o desenvolvimento deste resumo, ela é de acordo com Valente (2003, p. 98) “[...] o que cada indivíduo constrói como produto de processamento, da interpretação e da compreensão de informação”. Outro método utilizado, foi o hipotético-dedutivo que de acordo com Nascimento (2002), é uma modalidade científica que se inicia com um problema ou uma lacuna no conhecimento científico, passando pela formulação de hipóteses e por um processo de inferência dedutiva, o qual testa a predição da ocorrência de fenômenos abrangidos pelas referidas hipóteses:

1. Como seria o ensino de programação sem a base do pensamento computacional;



2. Se não houver variáveis, então algoritmos não podem ser desenvolvidos.

RESULTADOS E DISCUSSÕES

No geral, a criação de variáveis é o ponto de partida para o uso de estruturas condicionais, loops e operadores lógicos. Elas nos fornecem a capacidade de armazenar e manipular dados, permitindo que nossos algoritmos tomem decisões, repitam tarefas e realizem avaliações lógicas. Compreender e aplicar esses conceitos fundamentais de lógica garantirá a eficiência e a correção dos algoritmos que criamos.

Se não houver variáveis, de acordo com Petzold (2000), o desenvolvimento de algoritmos seria bastante limitado. As variáveis são elementos fundamentais na programação, pois permitem armazenar e manipular dados que são utilizados em um algoritmo. Em resumo, as variáveis são componentes essenciais no desenvolvimento de algoritmos, permitindo armazenar e manipular dados para realizar operações, tomar decisões e controlar o fluxo de execução de um programa, sem elas, o desenvolvimento de algoritmos se tornaria praticamente inviável.

Explorar e esclarecer conceitos fundamentais de lógica é essencial para o desenvolvimento de algoritmos eficientes. O uso de variáveis desempenha um papel fundamental nesse processo, pois elas nos permitem armazenar e manipular os dados necessários para aplicar estruturas condicionais, laços de repetição e operadores lógicos.

Nas análises de Metz (2012), podemos entender que as estruturas condicionais, como o "if-else", permitem que nosso algoritmo tome decisões com base em determinadas condições. Por exemplo, podemos utilizar uma estrutura condicional para verificar se uma variável tem um valor específico e, dependendo disso, executar um bloco de código que representa uma ação, ou executar outro bloco de código com uma ação diferente.. As variáveis são essenciais aqui, pois são utilizadas a todo momento para armazenar os valores que serão comparados e tomar as decisões necessárias.

Já os loops (laços de repetição), baseando-se em Prata (2013), servem para executar um bloco



de código várias vezes enquanto uma condição for verdadeira. Isso nos permite automatizar tarefas que requerem repetição, como processar uma lista de elementos ou executar uma ação até que uma condição seja satisfeita. Neste quesito, variáveis são amplamente utilizadas para controlar o número de iterações e acompanhar o progresso de execução do loop.

Os operadores lógicos, por sua vez, são usados para combinar e avaliar expressões lógicas, existem operadores como *AND* (e), *OR* (ou) e *NOT* (não), que nos permitem criar condições complexas com base em múltiplas variáveis. Esses operadores são vitais para o desenvolvimento de algoritmos que envolvam decisões e controle de fluxo.

Ao adotar metodologias que proporcionem uma experiência mais visual e intuitiva, ao analisar estudos de Pessanha (2019), é possível identificar que o incentivo ao desenvolvimento do raciocínio lógico de forma gradual é acessível. Essas ferramentas auxiliariam os alunos a compreenderem os conceitos fundamentais da programação, como estruturas condicionais, laços de repetição e estrutura de dados, de maneira mais prática e tangível.

A utilização de softwares de programação visual, nos quais eles "arrastam e soltam" blocos de código em uma interface gráfica auxiliaria no desenvolvimento dessas habilidades, dessa forma, eles podem criar programas sem precisar entender completamente a lógica por trás do código. Outra abordagem, baseando-se nas teorias escritas por Backes (2016), uma possível utilização de jogos educativos que ensinem programação de forma divertida e interativa seja conivente. Estes tendem a ensinar os conceitos básicos de programação de uma maneira mais lúdica, como aprender a mover personagens através de comandos simples, podendo assim ajudar os alunos a desenvolver habilidades básicas de programação, mesmo sem ter uma base sólida em pensamento computacional.

Vale ressaltar que, embora seja possível ensinar programação sem a base do pensamento computacional, esses conhecimentos são extremamente importantes para um entendimento mais profundo dos conceitos de programação. Portanto, mesmo adotando essas estratégias alternativas, é recomendado que, em algum momento, os alunos sejam apresentados ao



**IX
CONINFA**
PENSAR E EXISTIR:
Um novo olhar sobre a
importância do ser.

pensamento computacional para que possam ampliar seu conhecimento e habilidades nessa área.

CONSIDERAÇÕES FINAIS

A criação de variáveis é essencial na programação, permitindo armazenar e manipular dados e possibilitando o uso de estruturas como condicionais, loops e operadores lógicos. Sem variáveis, o desenvolvimento de algoritmos seria limitado. As estruturas condicionais permitem tomar decisões com base na análise de dois dados, enquanto os loops executam um bloco de código várias vezes enquanto uma condição for verdadeira. Já os operadores lógicos combinam expressões lógicas para criar condições complexas.

Para ensinar programação, é possível adotar abordagens visuais e intuitivas, como softwares de programação visual ou jogos educativos, que tornam o aprendizado mais prático e divertido. No entanto, é importante reforçar a importância do pensamento computacional como base fundamental para compreender conceitos mais avançados de programação.

A compreensão e aplicação desses conceitos fundamentais de lógica garantirá a eficiência e a correção dos algoritmos criados. Portanto, o desenvolvimento adequado das variáveis, estruturas condicionais, loops e operadores lógicos é essencial para a construção de programas funcionais e eficientes.

Ao ter uma compreensão clara desses princípios, a capacidade de identificar a melhor abordagem para resolver um problema específico será ainda maior. As variáveis ajudam a armazenar e manipular valores, enquanto as estruturas condicionais permitirão tomar decisões com base em condições específicas. Os loops irão permitir a repetição de um bloco de código várias vezes, seja para processar dados ou executar uma tarefa. E os operadores lógicos são usados para combinar condições e fazer comparações. Dominar esses conceitos e aplicá-los corretamente permite que sejam criados programas mais eficientes, evitando erros e mantendo a funcionalidade esperada.



**IX
CONINFA**
PENSAR E EXISTIR:
Um novo olhar sobre a
importância do ser.

PALAVRAS-CHAVE

Pensamento Computacional. Programação. Variáveis. Estruturas. Lógica.

REFERÊNCIAS

GONSALVES, Elisa Pereira. **Iniciação à Pesquisa Científica**. São Paulo. Alinea. 2001.

NASCIMENTO, Dinalva Melo. **Metodologia do trabalho científico: teoria e prática**. Rio de Janeiro. Forense, 2002.

VALENTE, José Armando. **Educação a distância no ensino superior: soluções e flexibilizações**. São Paulo. Interface. 2003.

METZ, Sandi. **Practical Object-Oriented Design in Ruby**. Carolina do Norte: Raleigh. The Pragmatic Bookshelf. 2012.

PRATA, Stephen. **C++ Primer Plus**. Nova Jersey: Upper Saddle River. Addison-Wesley Professional, 6ª Ed. 2013.

PETZOLD, Charles. **Code: The Hidden Language of Computer Hardware and Software**. Washington: Redmond. Microsoft Press. 2000.

BACKES, André. **Algoritmos e Lógica de Programação: Teoria e Prática**. Editora Novatec, 2016.

PESSANHA, Bruno Gomes. **Introdução à Programação**. Editora Brasport, 2019.