

# Conjunto estruturado de atividades didáticas para o ensino introdutório de programação

Struct set of didactic activities for introductory programming teaching

Conjunto estructurado de actividades didácticas para la enseñanza introductoria de la programación

Recebido: 27/06/2022 | Revisado: 09/07/2022 | Aceito: 14/07/2022 | Publicado: 21/07/2022

**Luciana Vescia Lourega**

ORCID: <https://orcid.org/0000-0001-6470-5577>

Universidade Federal de Santa Maria, Brasil

E-mail: [Luciana.lourega@ufsm.br](mailto:Luciana.lourega@ufsm.br)

**Ricardo Andreas Sauerwein**

ORCID: <https://orcid.org/0000-0002-1431-8699>

Universidade Federal de Santa Maria, Brasil

E-mail: [rsauer.ufsm@gmail.com](mailto:rsauer.ufsm@gmail.com)

## Resumo

Este trabalho foi realizado no âmbito da metodologia *Educational Design Research* (EDR), que busca encaminhar soluções para problemas vivenciados na prática docente. A presente pesquisa tem o propósito de criar, desenvolver, implementar e avaliar uma metodologia de ensino baseada em um conjunto de atividades didáticas (ADs) focadas em fornecer ritmo de estudo e *feedback* qualificado aos alunos da disciplina introdutória de programação no curso técnico em informática integrado ao ensino médio. As ADs foram desenvolvidas para realizar um acompanhamento contínuo do desempenho dos alunos, a fim de que o alto índice de evasão e reprovação seja reduzido. Durante esse processo metodológico, o professor consegue identificar as lacunas apresentadas pelos alunos e retomar o conteúdo individualmente e/ou em grupo, não deixando que as dúvidas se acumulem e se tornem obstáculos para o aprendizado. O retorno aos alunos foi realizado por meio de uma devolutiva do aprendizado, levando em consideração a compreensão leitora, as noções matemáticas, o raciocínio lógico e a abstração, aspectos que representam os quatro pilares essenciais a serem avaliados no ensino de programação. Os resultados obtidos indicam que as ADs desenvolvidas são de grande valia na construção do conhecimento dos alunos, observando desempenhos satisfatórios e, conseqüentemente, contribuindo para amenizar o índice de reprovação do curso.

**Palavras-chave:** Ensino; Algoritmo; Ritmo de aprendizagem; Atividades didáticas; *Feedback* qualificado.

## Abstract

This work was carried out within the scope of the Educational Design Research (EDR) methodology, which seeks to provide solutions to problems experienced in teaching practice. The research developed has the purpose of creating, developing, implementing and evaluating a teaching methodology based on a set of didactic activities (DAs) focused on providing students who take the introductory programming subject in the technical course in computer science integrated with high school, pace of study and qualified feedback. The DAs were developed to carry out a continuous monitoring of student performance so that the high rate of dropout and failure are reduced. During this methodological process, the teacher is able to identify the gaps presented by the students and carry out the resumption of the content individually and/or in groups, preventing that the accumulation of doubts being an obstacle to learning. Feedback to students was carried out taking into account reading comprehension, mathematical notions, logical reasoning and abstraction, which represent the four essential pillars to be evaluated in programming teaching. The results obtained indicate that the DAs developed are of great worth in the construction of students' knowledge, observing satisfactory performances and, consequently, contributing to decrease the failure of the Course.

**Keywords:** Teaching; Algorithm; Learning pace; Didactic activities; Qualified feedback.

## Resumen

Este trabajo se llevó a cabo dentro de la metodología de la Investigación del Diseño Educativo (IDE), que busca esbozar soluciones a problemas experimentados en la práctica docente. El objetivo de esta investigación es crear, desarrollar, implementar y evaluar una metodología de enseñanza basada en un conjunto de actividades didácticas (AD) enfocadas a proporcionar un ritmo de estudio y de *feedback* calificado a los estudiantes de la disciplina introductoria de programación en el curso técnico integrado en informática a la enseñanza secundaria. Las AD se desarrollaron para realizar un seguimiento continuo del rendimiento de los estudiantes con el fin de reducir las altas tasas de abandono y

desaprovação escolar. Durante este processo metodológico, el profesor puede identificar las lagunas que presentan los alumnos y retomar el contenido de forma individual y/o en grupo, sin dejar que las dudas se acumulen y se conviertan en obstáculos para el aprendizaje. La devolución a los alumnos se hizo a través de una retroalimentación del aprendizaje, teniendo en cuenta la comprensión lectora, las nociones matemáticas, el razonamiento lógico y la abstracción, aspectos que representan los cuatro pilares esenciales a evaluar en la enseñanza de la programación. Los resultados obtenidos indican que las AD desarrolladas son de gran valor en la construcción de los conocimientos de los alumnos, observando rendimientos satisfactorios y, en consecuencia, contribuyendo a aliviar la tasa de desaprovação del curso.

**Palabras clave:** Enseñanza; Algoritmo; Ritmo de aprendizaje; Actividades didácticas; *Feedback* calificado.

## 1. Introdução

O ensino de programação e suas complexidades têm sido tema de vários trabalhos nacionais e internacionais. Há muito tempo, procura-se encontrar alternativas que auxiliem o aluno a compreender conceitos relacionados à programação de computadores. Ensinar programação não é somente transmitir, aos estudantes, comandos e instruções que o computador deverá executar, trata-se de um processo complexo, que envolve lógica, raciocínio e habilidade por parte dos alunos para conhecer o problema a ser resolvido e elaborar sua solução, o que, muitas vezes, limita-se a uma esfera teórica (Gossmann, 2017). Nos últimos anos, tem sido evidente a migração de métodos tradicionais de ensino para métodos alternativos; com isso, o processo de ensino-aprendizagem — principalmente de algoritmos e de lógica de programação — tem sido uma temática de ampla discussão nas áreas de educação e informática, essencialmente com a criação de novos recursos para apoiar esse procedimento (Zacarias & Mello, 2019). De acordo com Morais *et al.* (2020), não basta explicar o conteúdo, apresentar alguns exemplos e propor exercícios para a turma. Com essa metodologia de ensino, muitas vezes, os alunos ingressantes no curso acabam desmotivados e não conseguem se interessar pela disciplina, o que contribui significativamente para os altos índices de reprovação e taxas de evasão observados atualmente.

Programar é uma tarefa desafiadora e, para amenizar as dificuldades no seu aprendizado, diversas abordagens metodológicas têm sido adotadas para seu ensino em cursos de graduação e de nível médio. Professores preocupados em auxiliar seus alunos buscam alternativas para melhor apoiá-los em sala de aula. Nesse intuito, muito se tem escrito sobre métodos, metodologias, ferramentas, práticas e técnicas adotadas no ensino de programação (Berssanette, 2016; Zacarias & Melo, 2019). No entanto, apesar de todas as estratégias e/ou metodologias criadas, o processo de ensino e aprendizagem da disciplina de programação de computadores ainda tem encontrado muitas dificuldades, pois ainda não há dados precisos sobre qual é a melhor metodologia a ser empregada e qual é o real sucesso desses cursos para os estudantes. De acordo com Berssanette (2016), uma das dificuldades de ensino da disciplina está justamente relacionada à ausência de metodologias adequadas, pois geralmente ela é ministrada de maneira tradicional, apenas com a teorização.

Uma das preocupações atuais na educação é saber como ensinar e como avaliar. Com isso, orienta-se para o uso de metodologias que considerem o ensino-aprendizagem baseado em competências e habilidades (BNCC, 2018). Dentre as competências requeridas para a área de programação, cita-se pensamento computacional<sup>1</sup>, mundo digital<sup>2</sup> e cultura digital<sup>3</sup>. Desse modo, com a implementação da metodologia baseada no desenvolvimento das habilidades dos alunos, deseja-se melhor entender e acompanhar as dificuldades dos alunos de maneira mais próxima, auxiliando-o no processo de aprendizagem e, conseqüentemente, preparando-os para os desafios do mundo atual (BNCC, 2018). As pesquisas nessa área estabeleceram um consenso de que a programação é uma atividade altamente complexa que envolve subtarefas ligadas a diferentes domínios do conhecimento e, a uma variedade de processos cognitivos (Ambrósio, *et al.*, 2011). Assim, valoriza-se um conjunto de

---

<sup>1</sup> Envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos (BNCC, 2018).

<sup>2</sup> Envolve as aprendizagens relativas às formas de processar, transmitir e distribuir a informação de maneira segura e confiável em diferentes artefatos digitais, compreendendo a importância contemporânea de codificar, armazenar e proteger a informação (BNCC, 2018).

<sup>3</sup> Envolve aprendizagens voltadas a uma participação mais consciente e democrática por meio das tecnologias digitais, o que supõe a compreensão dos impactos da revolução digital e dos avanços do mundo digital na sociedade contemporânea (BNCC, 2018).

habilidades que consolidam, aprofundam e ampliam a formação integral (BNCC, 2018). Dentre as diversas habilidades que a área de programação exige, foi identificado um conjunto de quatro habilidades para serem desenvolvidas e avaliadas — a compreensão leitora, a abstração, o raciocínio lógico e as noções matemáticas —, pois algumas delas são citadas em outros trabalhos.

Segundo Coll (1990 *apud* Miranda, 2016), a leitura é o processo em que o leitor, a partir de seus objetivos, realiza um trabalho ativo de compreensão e interpretação do texto, decodificando e analisando as sentenças presentes no corpo textual. A compreensão é o momento no qual você percebe quais são as principais palavras de cada ideia e quais são os dados e as informações possíveis de serem coletadas. Já na abstração, segundo Souza (2015), geralmente o leitor procura isolar os elementos das coisas e de suas relações para poder compreendê-los melhor. Para Menzies (1996 *apud* Bresolin, 2015), o raciocínio lógico é um processo estruturado de pensamento, bem como aprender o conteúdo de um conceito em toda sua extensão, permitindo chegar a uma determinada conclusão ou à resolução de um problema. Por fim, Moura (2007) entende que a noção matemática não é só entender uma linguagem: é também adquirir modos de ação que possibilitem lidar com outros conhecimentos necessários à sua satisfação e às necessidades de natureza integrativa, a fim de construir solução para problemas tanto do indivíduo quanto do coletivo.

Neste trabalho, essas quatro habilidades foram identificadas como pilares que sustentam o ensino de programação. Eles foram identificados por vários autores, como Falkembach *et al.* (2003), que descrevem a dificuldade de interpretação do próprio problema muito antes da dificuldade de interpretação de algum tipo de representação. Koliver, Dorneles e Casa (2004) afirmaram que a apresentação de princípios básicos da lógica é suficiente para resolver a maior parte dos problemas propostos numa disciplina de nível introdutório. Nobre e Menezes (2002 *apud* Schultz, 2003) descrevem as dificuldades em trabalhar a capacidade de abstração do aluno, seja na busca de possíveis soluções, seja na escolha das estruturas de dados. A necessidade de o professor fazer o aluno compreender a abstração envolvida em toda simbologia utilizada é corroborada por Rodrigues (2002). Alguns autores preconizam a necessidade de dominar habilidades matemáticas prévias que sejam ao menos integradas (Gomes & Mendes, 2015). Outros falam que os alunos devem ter um mínimo de conhecimento para poderem compreender esses temas (Koliver; Dorneles; Casa, 2004); no entanto, segundo Baeza-Yates (1995), o nível de conhecimento prévio de lógica matemática não é essencial.

As dificuldades são as mesmas com os alunos do primeiro ano de um curso técnico em informática integrado ao ensino médio de uma Instituição Federal do Estado do Rio Grande do Sul, onde a proposta será desenvolvida. Constata-se uma alta taxa de reprovação que, muitas vezes, leva os alunos a abandonarem o curso. Dentre os diversos assuntos desenvolvidos na disciplina — que se conectam entre si e são abordados em uma sequência lógica — este artigo terá o intuito de desenvolver uma proposta diferenciada e que supra as dificuldades encontradas no dia a dia da sala de aula. O presente estudo propõe a elaboração, implementação e avaliação de uma nova metodologia, baseada num conjunto de atividades didáticas (ADs) que visam desenvolver, no aluno, um ritmo de estudo que o auxilie a sanar as dúvidas no momento em que cada conteúdo está sendo ministrado, não acumulando lacunas em seu aprendizado. Além disso, essas ADs servirão como uma forma avaliativa diferente da tradicional (trabalhos, provas, testes, seminários, etc), pois será baseada nas habilidades alcançadas pelos alunos. . Por fim, para melhor apresentação da pesquisa, este artigo está organizado da seguinte maneira: a Seção 2 apresenta o método de pesquisa proposto; a Seção 3 apresenta os resultados e discussões da pesquisa e a Seção 4 apresenta a conclusão do trabalho.

## 2. Metodologia

No intuito de desenvolver a proposta metodológica deste trabalho, foram consideradas as habilidades contidas na Base Nacional Comum Curricular (BNCC) referente à área estudada, que conforme Aparício & Costa (2018) suas ausências na maioria das vezes impedem os alunos a desenvolverem seus problemas. Com isso, primeiramente, identificaram-se quatro habilidades

(interpretação de texto, abstração, raciocínio lógico e noções matemáticas) a serem trabalhadas ao longo do ano. Após, elas foram utilizadas para a construção das ADs que nortearam o ensino-aprendizagem na disciplina introdutória de programação. No intuito de construir as ADs, foi necessário pensar uma forma diferenciada de abordar o ensino de programação. Primeiramente, optou-se por apresentar aos discentes o fluxograma como forma inicial para realizar a resolução dos problemas propostos, pois percebe-se que neles o uso de símbolos como fator de abstração é, por vezes, utilizado com o intuito de tornar a assimilação do aluno mais simples (Halley, 2008). Após o aluno já ter um primeiro contato com o ensino de programação e, por meio dos fluxogramas, realizar suas resoluções, foi preciso pensar qual seria a melhor estratégia para que esses fluxogramas se tornassem linhas de código e para isto, utilizou-se uma linguagem de alto nível, a Linguagem C, a qual possui modularidade, recursos de baixo nível e simplicidade.

Depois de elencar quais assuntos são essenciais conter nas atividades didáticas a serem construídas, foi necessário pensar nas tarefas específicas que irão compor cada AD. Dessa forma, elencou-se os conteúdos a serem ministrados durante o ano letivo e uma AD para cada tópico do programa da disciplina foi criado. Fazem parte desse cronograma os conteúdos: conceitos básicos de programação, comandos de seleção (*if...else* e *switch...case*), laço de repetição (*while*, *do...while* e *for*) e estrutura homogênea – vetor. O material a ser desenvolvido para cada AD está dividido em material do professor e material do aluno. A seguir, serão descritos os aspectos utilizados na elaboração de cada material, tendo como base a proposta *Educational Design Research* (EDR), traduzida como Pesquisa em Design Educacional (PDE). Por fim, foi construída uma matriz para analisar e avaliar os alunos de acordo com a metodologia proposta.

Esta metodologia de ensino-aprendizagem se caracteriza por ser qualitativa, pois baseia-se no caráter subjetivo da análise dos alunos (coleta e interpreta as respostas subjetivas individuais dos participantes) e também quantitativa, uma vez que utiliza técnicas e ferramentas baseadas em métricas para determinar sua eficácia.

## 2.1 Construção das ADs

A fim de compor as ADs, levou-se em consideração os seguintes aspectos: uso de fluxogramas; uso da linguagem C; e desenvolvimento de tarefas baseadas nas habilidades a serem alcançadas, levando em consideração o conteúdo programático da disciplina. A fim de verificar, especificamente, o aprendizado do aluno referente a cada estrutura. Essa abordagem permite que o professor monitore as lacunas observadas no decorrer do ano letivo e, se necessário, realize uma retomada dos conteúdos (individualmente ou em grupo), para que nenhum aluno fique com defasagem, evitando que ele se desmotive, reprove ou abandone o curso.

Contudo, é importante ressaltar que cada AD é composta de 2 tarefas, abrangendo as quatro habilidades a serem avaliadas, onde a primeira tarefa de cada AD tem um menor nível de dificuldade. Por último, um questionário foi inserido em cada AD, o qual servirá de auxílio para que o professor construa a atividade didática seguinte e verifique se a percepção dos alunos vai ao encontro das ADs desenvolvidas, levando em consideração a opinião do aluno em relação ao empenho, à aprendizagem, ao foco nos estudos e ao trabalho despendido, conforme a Figura 1.

**Figura 1** – Questionário a ser respondido pelos alunos no final de cada AD.

**Questionário**

Classifique as atividades didáticas segundo os critérios abaixo. Em relação ao/a:

Empenho na disciplina a atividade didática desenvolvida:  
 1 (Não contribuiu)  2 (Contribuiu pouco)  3 (Contribuiu)  4 (Contribuiu muito)  5 (Contribuiu por completo)

Aprendizagem:  
 1 (Não contribuiu)  2 (Contribuiu pouco)  3 (Contribuiu)  4 (Contribuiu muito)  5 (Contribuiu por completo)

Foco nos estudos:  
 1 (Não contribuiu)  2 (Contribuiu pouco)  3 (Contribuiu)  4 (Contribuiu muito)  5 (Contribuiu por completo)

Trabalho despendido para sua resolução:  
 1 (Nada trabalhosa)  2 (Pouco trabalhosa)  3 (Trabalhosa)  4 (Muito trabalhosa)  5 (Excessivamente trabalhosa)

Fonte: Autores.

## 2.2 Material do professor

Inicialmente, para elaborar o material do professor, foi necessário pensar em uma estrutura que pudesse auxiliar o docente a conduzir o desenvolvimento das tarefas específicas em cada AD. A estrutura elaborada é composta por: identificação da turma, nome da atividade a ser desenvolvida, introdução, objetivos didáticos da AD e tarefas. A introdução tem como objetivo descrever um breve resumo do que será ministrado pelo professor e a forma que o mesmo foi desenvolvido em aula. Os objetivos didáticos têm como finalidade mostrar o que se busca atingir em relação ao aluno e ao professor. E as tarefas descrevem os objetivos que terão que ser alcançados pelos alunos e os objetivos a serem verificados pelo professor ao seu final. Vale ressaltar que todas as ADs seguirão a estrutura apresentada na Figura 2, alterando somente as tarefas propostas em cada AD. Dessa forma, cada atividade didática desenvolvida tem como objetivo investigar os conceitos básicos de cada conteúdo ministrado, analisando-o individualmente e de forma integrada aos conteúdos já trabalhados, de modo que, a cada conteúdo ministrado, os alunos consigam desenvolver algoritmos mais elaborados por meio de uma base sólida. A Figura 2 mostra a estrutura desenvolvida que compõe o referido material. Essa estrutura será apresentada por meio de um modelo reduzido referente a AD3, a qual mostra apenas a tarefa 1, pois a tarefa 2 segue essa mesma estrutura.

**Figura 2** – Modelo reduzido, contendo apenas a tarefa 1, do material do professor referente a AD3, comando *switch...case*.

<p><b>TURMA INFO XX</b> <b>ATIVIDADE DIDÁTICA 3 – COMANDO SWITCH...CASE</b></p> <p><b>Introdução</b></p> <p>Esta atividade tem como finalidade verificar a escolha do aluno em utilizar o comando <i>if...else</i> ou <i>switch...case</i> no desenvolvimento da tarefa apresentada a ele. Também será verificada a capacidade do aluno em modificar da estrutura <i>if...else</i> para a <i>switch...case</i> sem alterar o propósito do programa. Antes de desenvolver essa atividade, o aluno estudou o comando <i>switch...case</i>, o qual foi desenvolvido logo após o comando <i>if...else</i>. Ambos os comandos são pertencentes a estrutura de decisão e semelhantes na forma de implementação, sendo importante ao aluno visualizar formas diferenciadas de programar essas estruturas. Para melhor entendimento do discente, o comando foi apresentado com pequenos exemplos desenvolvidos em conjunto com os estudantes na ferramenta DEV-C++. Posteriormente, foram disponibilizados aos alunos problemas para que pudessem desenvolver tanto o raciocínio lógico quanto a escrita de programas na linguagem C. Depois de finalizar as etapas supracitadas, a atividade didática foi desenvolvida.</p> <p><u>Objetivo da atividade didática 3</u></p> <p><u>Aluno</u></p> <ul style="list-style-type: none"><li>- Analisar o uso dos comandos <i>if...else</i> e <i>switch...case</i> em diferentes contextos.</li><li>- Fornecer material para o professor a fim de analisar o desenvolvimento do aluno.</li><li>- Ditar ritmo de estudo.</li></ul>	<p><u>Professor</u></p> <p>-Analisar os objetivos elencados em cada tarefa a fim de elaborar as matrizes de análise que servirão de auxílio para o professor acompanhar o aprendizado dos alunos durante o ano letivo.</p> <p><u>Tarefa 1</u></p> <p>Esta tarefa será desenvolvida na linguagem de programação C, em que, os alunos devem identificar qual a estrutura ministrada até o momento (<i>if...else</i> ou <i>switch...case</i>) é a melhor opção para desenvolver o problema proposto, interpretar corretamente o que foi solicitado na tarefa e saber realizar os cálculos matemáticos necessários para o desenvolvimento da atividade.</p> <p><u>Objetivos a serem verificados pelo professor ao final da tarefa:</u></p> <ul style="list-style-type: none"><li>- Analisar o código para detectar a existência de erros lógicos e/ou sintáticos.</li><li>- Analisar a interpretação do aluno referente ao que foi exposto para ele (leitura correta do enunciado programa).</li><li>- Identificar se o discente consegue perceber os comandos necessários a serem utilizados no desenvolvimento do programa, utilizando, para isso, a linguagem de programação C.</li><li>- Analisar se o aluno consegue desenvolver os cálculos necessários para fazer com que o programa desempenhe sua função corretamente.</li><li>-Analisar a escolha feita pelo usuário da estrutura utilizada para desenvolver o programa.</li></ul>
--	--

Fonte: Autores.

### 2.3 Material do aluno

Após desenvolver o material do professor, elaborou-se a estrutura do material do aluno, que foram utilizadas no desenvolvimento de todas as ADs, alterando somente as tarefas propostas em cada atividade. Dessa forma, cada AD desenvolvida tem como objetivo investigar os conceitos básicos de cada conteúdo ministrado, analisando-o individualmente e de forma integrada aos conteúdos já trabalhados, fazendo com que os alunos consigam desenvolver a cada conteúdo ministrado algoritmos mais elaborados por meio de uma base sólida.

A fim de compor as ADs, foi considerado os objetivos a serem alcançados pelos alunos descritos no material do professor, bem como, as habilidades a serem avaliadas: interpretação de texto, abstração, raciocínio lógico e noções matemáticas. E, ao final de cada AD, foi inserido um questionário, que servirá de auxílio para que o professor construa a próxima atividade didática a ser trabalhada e verifique se a percepção dos alunos corrobora com as ADs desenvolvidas, levando em consideração o empenho, a aprendizagem, o foco nos estudos, e o trabalho despendido, conforme a Figura 1.

Após definir os aspectos que farão parte do material do aluno, sua construção começa a ser desenvolvida. O material é composto pelo cabeçalho, o qual é constituído de: nome da turma que participará da pesquisa, um espaço para ser inserido o nome do aluno, a data em que a atividade será realizada, um *feedback* qualificado a ser preenchido pelo professor, o nível de participação do aluno nas ADs (completa, parcial ou não realizada), as tarefas a serem desenvolvidas e para finalizar um questionário. Vale ressaltar que 7 ADs foram produzidas com base nos conteúdos trabalhados na disciplina de Programação I (AD1 - introdução aos conceitos básicos de programação, AD2 - comandos de seleção (*if...else*), AD3 - comandos de seleção (*switch...case*), AD4 - laços de repetição (*while*), AD5 - laços de repetição (*do...while*), AD6 - laços de repetição (*for*) e, por último, AD7 - estruturas homogêneas (vetor)). Como todas as ADs possuem a mesma estrutura, abaixo será apresentada a AD3 desenvolvida na forma resumida (Figura 3).

**Figura 3** – Modelo reduzido do material do aluno referente a AD3, comando *switch...case*, contendo a tarefa 1 e a letra (a) da tarefa 2.

**TURMA INFO XX**  
Nome: \_\_\_\_\_  
Data: \_\_\_\_/\_\_\_\_/\_\_\_\_  
Feedback qualificado: \_\_\_\_\_  
Atividade Desenvolvida:  completa  parcial  não realizada  
**Orientações:** Todas as tarefas devem ser enviadas pelo ambiente de aprendizagem SIGAA<sup>1</sup>.

**Tarefa 1:** Analisando o exercício abaixo, identifique a melhor estrutura estudada até o momento e construa o programa utilizando a linguagem de programação C. Justifique sua escolha.

a) Uma loja fornece 10% de desconto para funcionários e 5% de desconto para clientes vips. Faça um programa que calcule o valor total a ser pago por uma pessoa. O programa deverá ler o valor total da compra efetuada e um código que identifique se o comprador é um cliente comum, funcionário ou vip.

**Na linguagem C:**

**Justificativa:**

**Tarefa 2:** Analise os códigos abaixo e reescreva-os utilizando o comando Switch... Case.

a)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int x;
    printf("Escolha o código do produto\n");
    printf("1 - Vestuario\n");
    printf("2 - Higiene Pessoal\n");
    printf("3 - Produto perecivel\n");
    printf("4 - Produto nao perecivel\n");
    scanf("%d",&x);
    if (x==1) {
        printf("Voce quer comprar uma blusa?\n");
    }
    else
    if(x==2) {
        printf("Voce quer comprar um creme dental?\n");
    } else
    if(x==3) {
        printf("Voce quer comprar um kg de carne?\n");
    }else
    if(x==4){
        printf("Voce quer comprar uma lata de oleo ?\n");
    }
}
```

**Reescrevendo o código:**

**Questionário**

Classifique as atividades didáticas segundo os critérios abaixo. Em relação ao/a:

Empenho na disciplina a atividade didática desenvolvida:  
1 (Não contribuiu) 2 (Contribuiu pouco) 3 (Contribuiu) 4 (Contribuiu muito) 5 (Contribuiu por completo)

Aprendizagem:  
1 (Não contribuiu) 2 (Contribuiu pouco) 3 (Contribuiu) 4 (Contribuiu muito) 5 (Contribuiu por completo)

Foco nos estudos:  
1 (Não contribuiu) 2 (Contribuiu pouco) 3 (Contribuiu) 4 (Contribuiu muito) 5 (Contribuiu por completo)

Trabalho despendido para sua resolução:  
1 (Nada trabalhosa) 2 (Pouco trabalhosa) 3 (Trabalhosa) 4 (Muito trabalhosa) 5 (Excessivamente trabalhosa)

Fonte: Autores.

## 2.4 Implementação da metodologia proposta

As ADs foram realizadas no ano letivo de 2021, no curso técnico em informática integrado ao ensino médio de uma Instituição Federal do Estado do Rio Grande do Sul. Nesse ano, ingressaram no curso 56 alunos pertencentes ao primeiro ano do curso. Em 2021, o mundo se encontrava em um momento difícil da pandemia do covid-19 e, no Brasil, as instituições de ensino foram mantidas fechadas, de modo que as aulas eram ministradas de forma remota. Conforme decidido no ano de 2020, enquanto a instituição estivesse fechada para a realização das aulas presenciais, as disciplinas seriam organizadas em módulos, cuja carga horária foi calculada de acordo com o número de horas de cada matéria. Eram ofertados, no máximo, de forma simultânea para os alunos e, conforme os módulos foram acabando, outros se iniciavam. Essa estratégia foi organizada pela instituição para que não houvesse um acúmulo de trabalhos a serem desenvolvidos pelos alunos, pois a presença era computada pela entrega das atividades semanais. Na disciplina de Programação I, por ter um elevado número de horas-aula (120 h anual), ficou destinado um tempo correspondente a dois meses e meio de aula em cada semestre.

Dentro dos módulos que estariam sendo ministrados, a instituição organizou um horário e um tempo para que as aulas síncronas fossem ministradas semanalmente. Na disciplina de Programação I, as aulas eram ministradas de forma síncrona duas

vezes por semana, com um tempo de duração de 1 h e 15 min, sendo que, no restante do tempo, o aluno tinha acesso, por meio do Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), aos materiais necessários para compor o seu aprendizado — como gravação das aulas dadas, exercícios, livros, entre outros —, complementando o conhecimento de forma assíncrona. A disciplina foi organizada e o conteúdo dividido da seguinte forma: no primeiro semestre, trabalhou-se, inicialmente, com fluxogramas, comandos de decisão (*if...else e switch...case*) e a linguagem de programação C; no segundo semestre, foram abordados os conteúdos dos laços de repetição (*while, do...while e for*) e estrutura homogênea, *vetor*.

Como os momentos síncronos junto aos alunos ocorriam duas vezes por semana, decidiu-se que, no primeiro encontro, o professor ministraria o conteúdo referente ao tópico a ser abordado naquele momento e, no final da aula, disponibilizaria uma lista de exercícios para os alunos resolverem. No encontro seguinte, o professor esclareceria as dúvidas que os alunos trouxessem e analisaria a necessidade de dispor mais tarefas antes de aplicar a AD referente ao conteúdo abordado. As ADs eram aplicadas, no momento em que os alunos não relatavam mais dúvidas e conforme eles já compreendessem o conteúdo. Isso oscilou de uma a duas semanas, dependendo do conteúdo a ser abordado, bem como sua complexidade.

Após essas duas etapas serem cumpridas, a AD foi aplicada, sendo disponibilizada no horário da aula síncrona, e o aluno teve até o final do dia para retornar sua avaliação pelo SIGAA. Essa alternativa foi implementada junto aos alunos porque muitos não puderam estar presentes no momento em que as aulas online estavam ocorrendo e, pelas normativas, os discentes não eram obrigados a comparecerem nos encontros previstos no horário das aulas síncronas. Sendo assim, o professor deveria proporcionar outros momentos para os alunos cumprirem com suas obrigações escolares.

## 2.5 Matriz de análise

Após a construção das ADs, do material do professor e do aluno serem estabelecidos, foi desenvolvido o material de análise, denominado matrizes de análise (Tabela 1). Esse material auxilia o docente a realizar o *feedback* e analisar o desempenho e o desenvolvimento dos seus alunos de acordo com os critérios instituídos. Com o propósito de construir a matriz de análise, a Tabela 1 foi desenvolvida para as turmas cujos dados coletados foram analisados a partir dos objetivos elencados dentro de cada AD no momento de sua elaboração, levando em consideração as 4 habilidades propostas neste artigo. Essas habilidades foram identificadas por escalas qualitativas e quantitativas, utilizando os conceitos “satisfatório” (maior que 70%), “parcialmente satisfatório” (50 – 69%) e “insatisfatório” (abaixo de 50%).

Ainda é possível perceber que, na Tabela 1, o professor avalia o nível de compreensão e entendimento dos alunos de acordo com cada habilidade: (I) interpretação de texto, (A) abstração, (RL) raciocínio lógico e (NM) noções matemáticas. Dessa maneira, é possível acompanhar cada aluno nos diferentes graus de percepção necessários à aprendizagem de programação.



**Tabela 1** – Matriz de análise das ADs.

Alunos	Interpretação de texto - I			Abstração - A			Raciocínio lógico - RL			Noções matemáticas - NM		
	S	PS	I	S	PS	I	S	PS	I	S	PS	I
Aluno 001												
Aluno 002												
Aluno 003												
Aluno 004												
Aluno 005												
Resultado por critério num total de XX alunos	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

S – Satisfatório; PS – Parcialmente Satisfatório; I – Insatisfatório. Fonte: Autores.

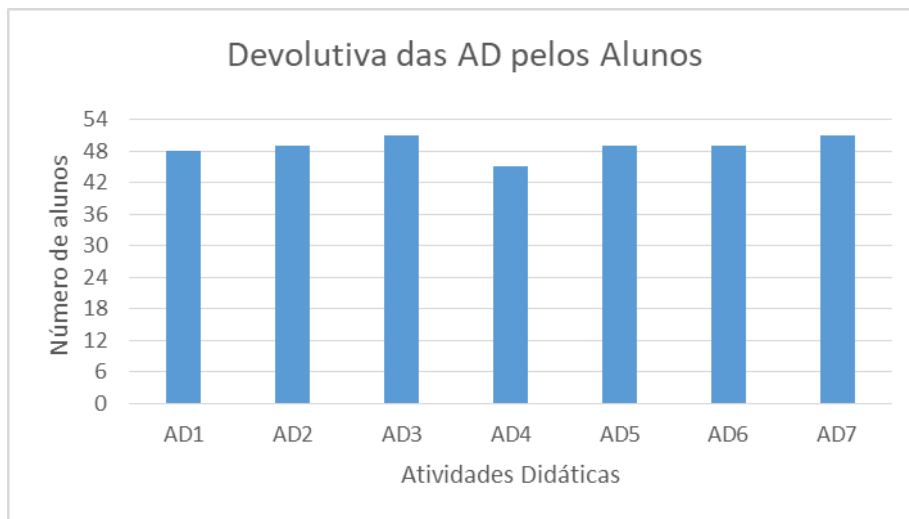
Dessa forma, ao longo do ano, alunos e professores puderam analisar os percentuais alcançados em cada pilar, ao desenvolverem as ADs. Por meio do *feedback*, o professor proporcionará, aos discentes, uma retomada do conteúdo nas lacunas observadas e o aluno pode focar seus estudos em pontos importantes do conteúdo a fim de que, na etapa seguinte, essa lacuna não seja um obstáculo que os impeça de prosseguir em seu aprendizado. A importância em se dar um *feedback* qualificado aos alunos é relatada por Shadiev *et al.* (2013) que citam em seu trabalho que nem sempre os estudantes conseguem obter um *feedback* instantâneo ou comentários do professor, fazendo com que muitas vezes o aluno iniciante em programação seja forçado a aprender uma linguagem de programação sozinho, raramente interagindo com os colegas.

### 3. Resultados e Discussão

No decorrer do ano letivo, foram aplicadas sete ADs, às quais foram atribuídas notas e *feedbacks* qualificados para o desenvolvimento dos alunos. A atribuição de notas às ADs ocorreu no momento de pandemia, quando a comunicação com os alunos estava prejudicada e muitos não se faziam presentes durante as aulas síncronas. Essa não era a única avaliação proposta aos alunos; pois, no semestre, é obrigatório oferecer aos alunos pelo menos três formas diferentes de avaliação. Vale salientar que as notas foram publicadas em um arquivo separado no intuito de desmistificar o ensino-aprendizado baseado exclusivamente em notas, em que o aluno se vê realizando a tarefa devido pela nota e não pela relação com o aprendizado.

Com o propósito de acompanhar o índice de participação dos alunos nas ADs, foi considerado o envio das atividades pelo ambiente SIGAA. A Figura 4 mostra o número de alunos das turmas que desenvolveram as ADs no decorrer da pesquisa.

**Figura 4** – Participação dos alunos em cada atividade didática no ambiente SIGAA.

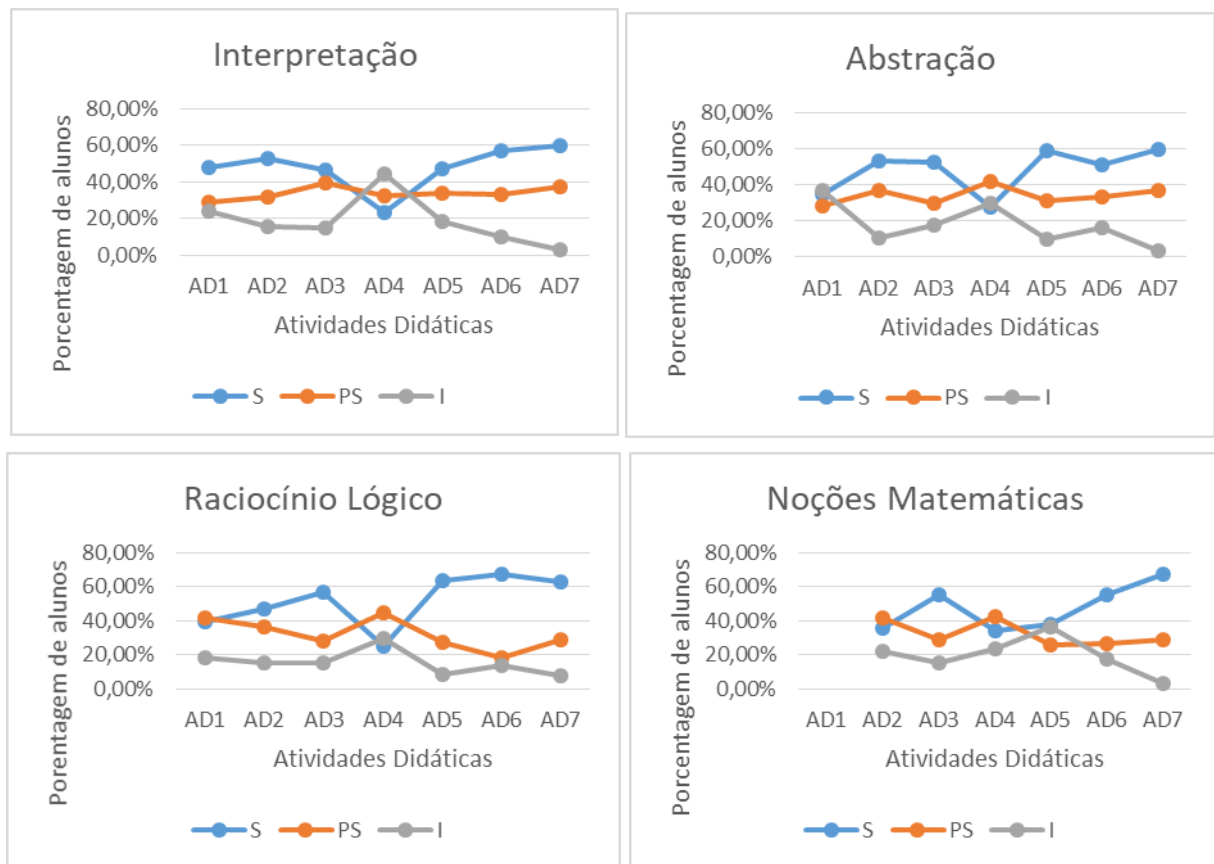


Fonte: Autores.

A fim de analisar os dados, calculou-se a média de participação dos alunos em cada AD, a qual ficou em 48,9 discentes. Também se computou a moda,  $M_0 = 49$  e verificou-se que o número de participantes em cada AD é o mesmo da média, sinalizando uma ótima homogeneidade entre os dados. Por último, analisou-se o desvio padrão, no intuito de verificar o grau de variação das devolutivas e evidenciou-se que houve uma pequena variação,  $DP = 1,88$ . Analisando a Figura 5 e calculando a média, moda e desvio padrão das devolutivas dos alunos, observa-se que somente na AD4 ocorre uma pequena variação, pois o número de alunos participantes dessa atividade está um pouco abaixo do desvio padrão calculado. Isto ocorreu quando os alunos se depararam com uma estrutura mais complexa — laço de repetição (*while*) — e junto a ela era necessário utilizar comandos condicionais, aumentando a complexidade dos programas a serem desenvolvidos. Contudo, no geral, ocorreu um alto índice de participação nas ADs (média ~ 87%), permitindo que o professor tivesse uma percepção mais eficiente quanto ao rendimento do aluno e identificasse as lacunas peculiares para retomar o(s) diferente(s) conteúdo(s) específicos, individualmente ou em grupo.

Para analisar a capacidade das ADs na aprendizagem dos alunos segundo as habilidades interpretação de texto, abstração, raciocínio lógico e noções matemáticas, foi realizada uma média geral das tarefas 1 e 2 em cada AD, onde esses critérios foram sendo trabalhados e desenvolvidos por meio dos *feedbacks* qualificados. No intuito de melhor analisar os dados, os resultados das avaliações das 3 turmas (1A, 1B e 1C) foram analisados de maneira unificada. Além disso, os dados serão discutidos de acordo com as habilidades, com base na faixa de desempenho dos alunos: satisfatório (S), parcialmente satisfatório (PS) e insatisfatório (I). Primeiramente, informa-se que na AD1, referente à competência denominada interpretação, a tarefa 1 foi desconsiderada, pois não havia texto para ser interpretado. Além disso, pelos mesmos motivos da tarefa 1 na AD1 da habilidade Interpretação, a AD1 (tarefa 1 e 2) referente a competência denominada Noções de Matemática, não foi avaliada. A Figura 5 mostra os resultados obtidos nos indicadores “satisfatório”, “parcialmente satisfatório” e “insatisfatório”, referente à média das tarefas em todas as ADs em relação às quatro habilidades analisadas.

**Figura 5** – Gráfico da média das tarefas 1 e 2 em todas as ADs levando em consideração as quatro habilidades analisadas.



Fonte: Autores.

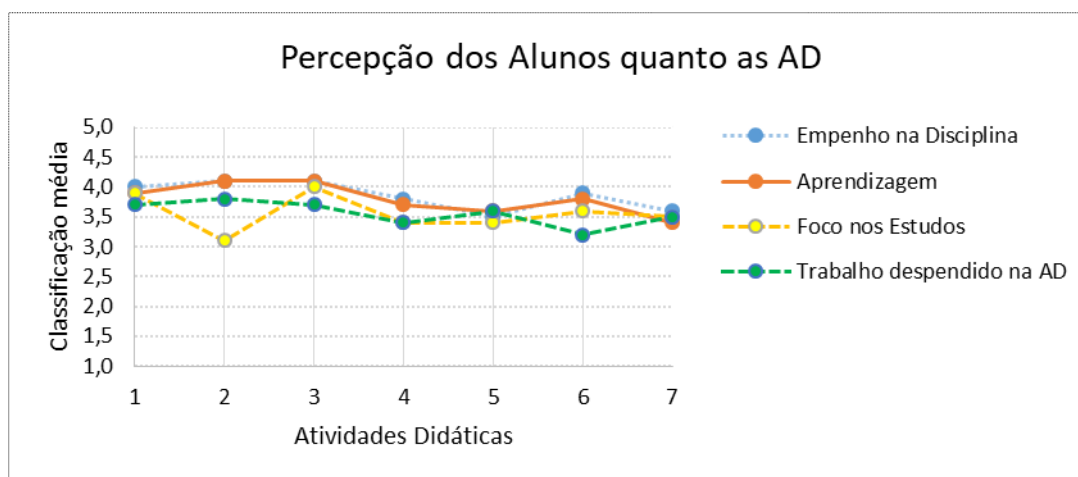
Em relação ao indicador satisfatório, as habilidades abstração e interpretação mostram tendências parecidas, pois após, a AD1, ocorre um aumento dessas habilidades e, em seguida, um pequeno decréscimo até a AD3. Logo em seguida, na AD4, observa-se um decréscimo dos alunos nessas habilidades, voltando a aumentar o índice nas próximas AD. Dessa maneira, consegue-se perceber que a variação da abstração está correlacionada com a interpretação. Outra competência que pode ser correlacionada com a abstração e a interpretação é o raciocínio lógico, o qual possui tendências parecidas, com exceção da AD2, a qual esperaríamos um valor acima de 51%, mas obteve-se 47,40%. Já a habilidade noções matemáticas possui um comportamento próprio, com aumento dessa habilidade da AD2 para a AD3, diminuindo na AD4 e voltando a crescer progressivamente até a AD final (AD7).

Considerando o indicador parcialmente satisfatório, novamente as habilidades abstração e interpretação mostraram uma tendência de uma faixa de variação (abstração – 28,25 – 37,00% e interpretação – 28,60 – 39,25%; com exceção da AD4 na competência abstração, que possui um valor de 42,10%). Já o raciocínio lógico e as noções matemáticas possuem uma variação no comportamento, exatamente inversa à variação visualizada no indicador satisfatório, com exceção da variação encontrada em noções matemáticas, a qual mostra um comportamento semelhante ao indicador satisfatório a partir da AD5.

Completando, o indicador insatisfatório apresenta tendências semelhantes nas variações das ADs para todas as habilidades. De uma maneira geral, em todas as ADs, o indicador “satisfatório” foi o mais observado entre os alunos e, conseqüentemente, esses alunos atingiram maiores níveis de aprendizado, com exceção da AD4, cujo indicador PS foi o mais observado. Foi apresentado aos alunos, pela primeira vez, a estrutura de laço de repetição (*while*), e eles tiveram dificuldade em compreender o modo de funcionamento desse comando.

A fim de analisar a percepção dos alunos quanto às ADs desenvolvidas (Figura 6), foi elaborado e aplicado um questionário (Figura 1) que considerou os itens empenho na disciplina, aprendizagem, foco nos estudos e trabalho despendido na AD. Esses itens permitem analisar, respectivamente: o interesse do aluno em relação ao desenvolvimento das ADs; o entendimento dos alunos frente aos assuntos abordados; a evolução no ritmo de estudo; e o tempo despendido para realizá-las. Além de realizar as respectivas análises, os itens considerados permitem auxiliar o professor a construir a próxima atividade e verificar se a percepção dos alunos corrobora com as ADs desenvolvidas.

**Figura 6** – Gráfico que demonstra a variação de Empenho, Aprendizagem, Foco nos Estudos e Trabalho Despendido nas ADs.



Fonte: Autores.

Observando as respostas dos alunos, distribuídos em 342 questionários, verifica-se que o item trabalho despendido nas ADs possui uma correlação com as outras percepções dos alunos da AD1 até a AD4; mas a partir da AD5, mostra um comportamento inversamente proporcional. Isto corrobora que duas visões podem ser observadas pelo professor: o tempo despendido para realizar a tarefa e o nível de dificuldade do aluno perante cada AD. Com isso, da AD1 – AD4, sugere-se que a variação do trabalho despendido foi influenciada pelo tempo despendido, pois, quanto maior o tempo despendido, maior foi o empenho e a aprendizagem. No entanto, a variação entre AD4 – AD7 foi influenciada pelo nível de dificuldade (ADs com maior grau de dificuldade), pois, quanto maior o empenho e a aprendizagem, menor o trabalho despendido e vice e versa. Quanto ao foco, sua percepção é muito parecida com as percepções de empenho e aprendizagem, mostrando que o aluno mais focado tende a ter uma melhor aprendizagem e um melhor empenho nas ADs. A exceção é a atividade AD2, pois nela os comandos referentes à linguagem de programação C foram apresentados ao aluno pela primeira vez. Ao sentir uma maior dificuldade no conteúdo trabalhado, o aluno teve um menor foco em realizar a AD. Dado que o empenho e a aprendizagem possuem a mesma tendência de variação, pode-se concluir que a diminuição dos índices nestes itens, possivelmente, está correlacionada com a dificuldade dos discentes quanto a esses tópicos trabalhados e assim verificando que, nas próximas vezes que esses assuntos forem discutidos, devem-se usar metodologias complementares de aprendizagem.

Tendo em vista que cada turma e cada aluno teve características peculiares, todos esses quesitos nos possibilitaram fazer ajustes na forma do ensino-aprendizagem das temáticas abordadas. Com este trabalho, foi possível verificar onde estão as possíveis fragilidades na compreensão dos alunos durante o ano letivo, bem como orientar o professor a utilizar metodologias complementares de ensino-aprendizagem. Tudo isso, nunca esquecendo que, por meio do *feedback* do professor, é possível aumentar e disseminar um melhor ritmo de estudo dos alunos. Com essas ações desenvolvidas, ao final do ano letivo, percebe-se que as mesmas foram bem distribuídas e uma uniformidade em seu desenvolvimento foi estabelecida e isso certamente contribuiu para amenizar o índice de evasão e reprovação que ficaram em torno de 1% e 15%, respectivamente.

#### 4. Conclusão

Nesta pesquisa, foi apresentado um conjunto de sete ADs que inclui todo o conteúdo do ensino da disciplina introdutória de Programação I do curso técnico em informática integrado ao ensino médio, no intuito de avaliar os alunos com perspectivas baseadas nas habilidades de interpretação, abstração, raciocínio lógico e noções matemáticas. As ADs tiveram como propósito desenvolver um *feedback* qualificado e dar ritmo de estudo ao aluno. Ao analisar as habilidades investigadas nesse trabalho, foi verificado que a interpretação, abstração e raciocínio lógico possuem tendências parecidas sendo importante realizar um *feedback* para elas a fim de analisar a aprendizagem dos alunos no ensino inicial de programação. A habilidade noções matemáticas, mesmo tendo um comportamento diferente das demais, também é um importante critério a ser investigado, pois a disciplina em análise aborda conceitos básicos essenciais ao ensino introdutório de programação.

Considerando a participação dos alunos no desenvolvimento das ADs e levando em consideração o cenário de pandemia, obteve-se uma alta adesão, onde a maior parte dos alunos envolvidos com a disciplina desenvolveram as atividades propostas, possibilitando que o professor realizasse uma retomada dos conteúdos focada nas lacunas da maioria dos discentes que estavam empenhados em melhorar seu aprendizado. Isso só foi possível com o *feedback* do professor em determinar um maior ritmo de estudos para diversos alunos que necessitavam de auxílio extra para avançar na compreensão da disciplina.

A percepção dos alunos das ADs, foram observados em todos os questionários distribuídos em sete ADs, onde foi observada a variação do empenho na disciplina, a aprendizagem, o foco nos estudos e o trabalho despendido na AD. Em relação ao empenho e a aprendizagem, os resultados encontrados se mantiveram muito próximos, tendo uma leve queda no final do ano, momento em que sempre ocorre um acúmulo maior de atividades a serem realizadas envolvendo todas as disciplinas do curso, mostrando que quanto maior o empenho, maior é a aprendizagem. O foco nos estudos se manteve constante durante o ano, tendo uma leve queda na AD2, indicando que, no geral, os alunos mantiveram um ritmo de estudo. O item referente ao trabalho despendido se manteve variando entre 3 e 3,7; em alguns momentos, os alunos entenderam que as ADs poderiam ser menos trabalhosas. Entende-se que, no contexto de aulas remotas, os alunos precisam dedicar mais tempo em atividades extraclasse e que somente os encontros síncronos realizados duas vezes por semana com duração de 2 horas e 30 minutos não são suficientes para que o aluno consiga ter um bom aprendizado. Dessa maneira, dentro do contexto analisado as ADs contribuíram para que os professores pudessem acompanhar frequentemente o desenvolvimento e desempenho dos alunos de acordo com as habilidades exigidas na área de programação.

Em síntese, conclui-se que as ADs apresentadas neste trabalho se mostraram uma alternativa viável para o ensino introdutório de programação. Isso porque o acompanhamento contínuo por meio das sete ADs e do *feedback* qualificado fez com que os alunos adquirissem um melhor ritmo de estudo e, conseqüentemente, um bom nível de entendimento dos conteúdos trabalhados. Também é importante que ressaltar as ADs foram importantes durante a pandemia de covid-19, pois a baixa interação e comunicação com o aluno foram importantes fatores a serem superados. Futuramente essa metodologia poderá ser aplicada em outras instituições de ensino, a fim de analisar sua viabilidade em contextos diferentes.

#### Referências

- Ambrósio, A. P. L., Almeida, L. S., Macedo, J., Santos, A., & Franco, A. H. (2011). Programação de Computadores: compreender as dificuldades de aprendizagem dos alunos. *Revista Galego-Portuguesa de Psicoloxía e Educación*, 19 (1), 185–197.
- Aparício, J., & Costa, C. (2018). A virtual robot solution to support programming learning an open source approach. *Iberian Conference on Information Systems and Technologies, CISTI. 2018-June*, 1-6. IEEE Computer Society.
- Baeza, Y. R. A. (1995). Teaching Algorithms. *SIGACT News*, 26(4), 51-59.
- Berssanette, J. H. (2016). Ensino de Programação de Computadores: Uma Proposta de Abordagem Prática Baseada em Ausubel. Dissertação (Mestrado Profissional em Ensino de Ciência e Tecnologia) – UTFP/Ponta Grossa, Paraná.

- BRASIL. (2018). Ministério da Educação. Base Nacional Comum Curricular. Brasília. [http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518-versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518-versaofinal_site.pdf)
- Falkembach, G. A. M., Amoretti, M. S. M., Tarouco, L. R., & Vieiro, F. (2003). Aprendizagem de Algoritmos: Uso da Estratégia Ascendente de Resolução de Problemas. *Anais: 8º Taller Internacional de Software Educativo*. Santiago, Chile. [http://tise.cl/2010/archivos/tise2003/papers/aprendizagem\\_de\\_algoritmos.pdf](http://tise.cl/2010/archivos/tise2003/papers/aprendizagem_de_algoritmos.pdf)
- Gomes, A., & Mendes, A. (2015). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. *Proceedings - Frontiers in Education Conference, FIE. 2015-February*. Institute of Electrical and Electronics Engineers Inc.
- Gossmann, T. A. (2017). Utilização de algoritmos nos processos de ensino e de aprendizagem de programação de computadores em cursos técnicos. Monografia (Especialização em Docência na Educação Profissional) – Univates/Lajeado, Rio Grande do Sul.
- Halley, W. A. S., & Gondim, A. P. A. (2008). Esboço de Fluxogramas no Ensino de Algoritmos. *Anais do XXVIII Congresso da SBC*. Belém/PA. <http://www2.sbc.org.br/csbc2008/pdf/arg0121.pdf>.
- Koliver, C., Dorneles, R. V., & Casa, M. E. (2004). Das (muitas) dúvidas e (poucas) certezas do ensino de algoritmos. *Anais do XII Workshop de Educação em Computação – WEI*. Salvador, BA. <https://docplayer.com.br/82889605-Das-muitas-duvidas-e-poucas-certezas-do-ensino-de-algoritmos.html>
- Menzies, T. (1996). Applications of Abduction: Knowledge-Level Modeling. *International Journal of Human Computer Studies*. <http://menzies.us/pdf/96abkl.pdf>.
- Miranda, H. J. (2016). Estratégias de Leitura como Instrumento na Formação do Leitor Competente. Dissertação (Mestrado Profissional em Letras em Rede Nacional) – UFP/ Belém, PA.
- Moura, M. O. de. (2007). Matemática na infância. In *Educação matemática na infância: abordagens e desafios*. Vila Nova de Gaia: Gailivro.
- Morais, C. G. B., Mendes Neto, FM, & Osório, A. J. M. (2020). Dificuldades e desafios no processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática da literatura. *Pesquisa, Sociedade e Desenvolvimento*, 9 (10), e9429109287. <https://doi.org/10.33448/rsd-v9i10.9287>
- Nobre, I. A. M. N., & Menezes, C. S. (202). Suporte à Cooperação em um Ambiente de Aprendizagem para Programação (Samba). XIII Simpósio Brasileiro de Informática na Educação – SBIE. São Leopoldo, RS. <http://jacarepagua.dcc.ufrj.br/~ladybug/artigos/PereiraJr.pdf>.
- Rodrigues, M. C. J. (2002). Como Ensinar Programação? *Jornal Computação Brasil da Sociedade Brasileira de Computação*. <http://www.unit.br/methanias/artigos.htm>.
- Shadiev, R., Hwang, W., Yeh, S., Yang, S., Wang, J., Han, L., & Liu, C. (2013). Applying unidirectional versus reciprocal teaching strategies in web-based environment and their effects on computer programming learning. *Proceedings - 2013 IEEE 13th International Conference on Advanced Learning Technologies, ICALT 2013*, 5-9.
- Souza, E. R. C. (2015). Schopenhauer e os conhecimentos intuitivo e abstrato: uma teoria sobre as representações empíricas e abstratas. *Cultura Acadêmica*, 97-146. <https://static.scielo.org/scielobooks/g3p7n/pdf/souza-9788579836879.pdf>
- Schultz, M. R. (2003). Metodologias para Ensino de Lógica de Programação de Computadores. Monografia (Especialização em Ciência da Computação) – UFSC/Florianópolis, SC.
- Zacarias, R. O., & Mello, D. R. B. (2019). Metodologias de Ensino de Lógica de Programação e Algoritmos em Cursos de Graduação. *Revista Interdisciplinar do Pensamento Científico*, 5(2), 29-44.