

Utilização do Software VisuAlg no Ensino da Lógica de Programação

Using the VisuAlg Software in Teaching Programming Logic

Fabício Hartmann Borba^{*a}; Miriam Ines Marchi^a; Márcia Jussara Hepp Rehfeldt^a

^aUniversidade do Vale do Taquari, Programa de Pós-Graduação Stricto Sensu em Ensino. RS, Brasil.

*E-mail: fabriciohart@gmail.com

Resumo

A crescente demanda de profissionais de Tecnologia da Informação em empresas, universidades e outras instituições é notória e, para tal, é necessária a formação de programadores para darem suporte à criação e manutenção de sistemas computacionais. O algoritmo, base para a criação dos *softwares*, pode ser ensinado de forma a facilitar o aprendizado dos estudantes, utilizando-se o Português Estruturado. Nesse sentido, foi realizada uma prática pedagógica, na qual se utilizou o *Software VisuAlg* como ferramenta auxiliar no ensino de algoritmos. O objetivo foi analisar as potencialidades do *VisuAlg* como recurso didático nas aulas de Algoritmos. A pesquisa, que é de cunho qualitativo, tem aproximações com estudo de caso. Os dados coletados foram analisados tendo como base a Análise Textual Discursiva. Os resultados encontrados apontam que: a) a utilização de diferentes técnicas e ferramentas auxiliou no ensino da lógica de programação, pois instigou os estudantes a experimentar e testar os algoritmos; b) a criação de algoritmos na Língua Portuguesa ajudou os estudantes a pensarem em sua lógica, não sendo necessário compreender comandos em inglês; c) a motivação dos estudantes aumentou, pois pode-se comprovar que os códigos criados estavam funcionando; d) o *VisuAlg* permitiu encontrar os erros nos códigos facilmente. Portanto, o *Software VisuAlg* pode ser considerado uma ferramenta que auxiliou no ensino da lógica de programação e estimulou o interesse dos estudantes da área de programação.

Palavras-chave: Recurso Didático. Algoritmos. Português Estruturado.

Abstract

The growing demand for professionals of Information Technology (IT) in companies, universities and other institutions is notorious, and for that, it is necessary the programmers formation to creation support and computer systems maintenance. The algorithm, the basis for the software creation, can be taught in a way that facilitates the student's learning, using Structured Portuguese. In this sense, a pedagogical practice was performed, in which VisuAlg Software was used as an auxiliary tool in the algorithms teaching. The objective was to analyze the VisuAlg potential as a didactic resource in the Algorithms classes. The research, which is qualitative, has approximations with case study. The collected data were analyzed based on the Discursive Textual Analysis. The results show that: a) the use of different techniques and tools helped in the teaching of programming logic, as it instigates students to experiment and test the algorithms; b) the algorithms creation in Portuguese Language helped students to think in their logic, not being necessary to understand commands in English; c) the students' motivation increase, because it is possible to verify that the codes created are working, as well as the facility of being able to find the errors. Therefore, the use of tools that help teaching the programming logic, and stimulate the students' interest in this area, is increasingly used by different teachers.

Keywords: Didactic Resource. Algorithms. Structured Portuguese.

1 Introdução

Empresas, universidades e outras instituições de diversos setores da sociedade, cada vez mais, necessitam profissionais da área de informática e computação. Isso ocorre em função da crescente informatização e automatização de serviços que, antigamente, eram realizados de forma manual. Portanto, pessoas com habilidades e conhecimentos são indispensáveis para desenvolver e dar suporte para máquinas e sistemas que suprem o avanço tecnológico e a inevitável modernização das organizações.

Diante desse contexto, o presente trabalho discorre acerca da inserção e da utilização de ferramentas como recurso didático para auxiliar no desenvolvimento da lógica de programação. Tal lógica é necessária para a construção

da estrutura dos algoritmos, utilizados em disciplinas de Algoritmos e Programação de Computadores, que são a base para a criação de sistemas computadorizados.

Nesse sentido, foram analisadas as contribuições do uso do *Software VisuAlg*, que utiliza a Língua Portuguesa para a escrita dos algoritmos. Essa forma de escrita é conhecida como Português Estruturado, Portugol ou Pseudocódigo, pois todos os comandos e funções podem ser descritos em português e são adaptações genéricas da linguagem de programação. Dessa forma, a utilização da língua materna dos estudantes pode contribuir para a compreensão da lógica inicial necessária na resolução de problemas computacionais, conforme destaca Carvalho (2007).

A justificativa para realizar tal estudo está embasada

no fato de alguns estudantes precisarem de um tempo maior para resolverem os problemas computacionais e para compreenderem a lógica de programação, bem como o processo de escrita dos algoritmos apresentados em aula. Além disso, observou-se que os códigos eram desenvolvidos com “papel e caneta” e os estudantes apresentavam dificuldades em compreender o resultado do algoritmo construído. Supunha-se que isso pudesse estar relacionado ao fato de ser um método de construção abstrato, no qual o estudante precisa entender a lógica do que é escrito para “prever” o resultado final. Em outras palavras, por meio da utilização do “teste de mesa”, o estudante tentava comprovar se a construção se apresentava correta ou não.

Acredita-se que, além dos motivos citados, outro obstáculo da compreensão da lógica de programação é que as linguagens de programação são, geralmente, escritas na Língua Inglesa. Isso se torna um empecilho, visto que alguns estudantes podem não entender o significado de certos termos em tal idioma.

Essas percepções também são apresentadas por Souza (2009), professor da Universidade Severino Sombra - RJ e do Centro de Ensino Superior de Valença - RJ, ao expor os objetivos que o levaram a criação do *Software VisuAlg*. O autor descreve que, por meio de sua experiência, notou que escrever algoritmos no papel é abstrato demais para a compreensão de alguns estudantes. Além disso, o autor menciona que o contato inicial com a linguagem de programação, na Língua Inglesa, exige um rigor na criação do código, o que se apresenta como uma barreira no aprendizado dos conceitos. Souza (2009, p. 2) cita ainda que: “[...] o fato de que o aluno ainda enfrenta o obstáculo de entender o problema, criar por si uma solução ou compreender aquela apresentada por seus colegas ou pelo professor, isto é, a lógica de programação em si”.

Em outro estudo, Junior, Vieira e Vieira (2015) apresentam uma pesquisa realizada no Centro Universitário de Volta Redonda - RJ, em que alguns dos motivos que podem levar os estudantes a não compreenderem a lógica de programação, ou até reprovarem em disciplinas de *Algoritmos e Programação*, envolve a dificuldade de raciocínio lógico necessário para a construção dos códigos, a capacidade de abstração e a leitura e interpretação dos textos dos enunciados. Em síntese, saber o que é preciso fazer e os conhecimentos matemáticos necessários para os testes e comparações que são realizados em algoritmos são alguns dos motivos citados pelos autores.

Dessa forma, conforme já referido, aprimorar as técnicas para o ensino da programação e criar *softwares* que auxiliem na construção da lógica dos estudantes podem ser fatores importantes, pois com a evolução e a modernização da sociedade são cada vez mais necessários programadores e desenvolvedores de sistemas em empresas e nas diferentes áreas acadêmicas e profissionais. Com isso, a utilização de

instrumentos e de ferramentas, que auxiliem no ensino e na aprendizagem, estimulando o interesse dos estudantes, nesta área, são bem-vindos aos professores.

É neste contexto que se vislumbra a possível contribuição deste estudo, que se propôs a analisar como a utilização do *Software VisuAlg* contribuiu para a aprendizagem da lógica de programação, no ensino de algoritmos e nas aulas iniciais de programação. Portanto, o objetivo deste artigo é apresentar os resultados obtidos durante a prática pedagógica descrita na dissertação do Mestrado em Ensino, após análise dos dados coletados, apresentando as percepções dos estudantes no uso e na utilização de *softwares* e ferramentas como auxiliares no ensino de algoritmos e aulas iniciais de programação.

A presente pesquisa foi desenvolvida a partir da prática pedagógica realizada com alunos do curso Técnico em Informática, na disciplina de Algoritmos, do Módulo III - Assistente em Programação, da Escola Estadual de Educação Profissional Estrela, no município de Estrela/RS.

2 Material e Métodos

O presente artigo resultou da dissertação apresentada ao Programa de Pós-Graduação Mestrado em Ensino, na qual foi descrita e analisada a prática pedagógica realizada com os sete estudantes da disciplina de Algoritmos. A pesquisa, que apresenta características de um estudo de caso, tem um cunho qualitativo e os resultados foram analisados com base na Análise Textual Discursiva, conforme propõem Moraes e Galiazzi (2013).

O estudo de caso é um método de pesquisa que pode ser utilizado, conforme mencionado por Yin (2015, p. 4): “[...] em muitas situações, para contribuir ao nosso conhecimento dos fenômenos individuais, grupais, organizacionais, sociais, políticos e relacionados”, e permite que o pesquisador possa observar características de “eventos da vida real”. Além disso, o estudo de caso ajuda a entender fenômenos sociais complexos e:

[...] permite que os investigadores foquem um “caso” e retenham uma perspectiva holística e do mundo real - como no estudo dos ciclos individuais da vida, o comportamento dos pequenos grupos, os processos organizacionais e administrativos, a mudança de vizinhança, o desempenho escolar, as relações internacionais e a maturação das indústrias (YIN, 2015, p. 4).

Richardson (1999) menciona que, com uma pesquisa qualitativa, pode-se interpretar os dados coletados de forma a compreender e vivenciar as situações analisadas com o grupo de estudantes foco da prática pedagógica, não se limitando às informações genéricas de algumas perguntas estruturadas. Moreira (2011, p.51) afirma que:

O pesquisador enriquece sua narrativa com trechos de entrevistas, excertos de suas anotações, vinhetas, exemplos de trabalhos de alunos, entremeados de comentários interpretativos procurando persuadir o leitor, buscando apresentar evidências que suportem sua interpretação e, ao mesmo tempo, permitem ao leitor fazer julgamentos de

modo a concordar ou não com as asserções interpretativas do pesquisador.

Durante a prática pedagógica, os dados foram coletados por meio de: a) questionário de sondagem, com o qual foram verificados os conhecimentos dos estudantes em relação à programação; b) anotações no diário de campo, no qual foram registrados o desenvolvimento das aulas, as atividades, os acontecimentos ocorridos durante a prática, bem como as falas de alguns estudantes, gravação de áudio e os códigos dos algoritmos criados para a resolução dos exercícios propostos; e c) questionário de avaliação da prática, para saber como as atividades desenvolvidas com o *Software VisuAlg*, contribuíram para o ensino de algoritmos.

Seguindo preceitos da Análise Textual Discursiva (MORAES; GALIAZZI, 2013), os dados coletados foram divididos em categorias. De acordo com os autores, os dados podem ser organizados da seguinte forma: fragmentação e análise do material coletado com detalhes; estabelecimento das relações ou categorização, agrupando os dados correlatos; captação do emergente, observação e análise do que surgiu de novo durante o processo; e auto-organização, deixando que novas ideias sejam inseridas na compreensão do todo.

Moraes e Galiazzi (2013) descrevem, ainda, que é necessário ter presente a relação entre leitura e interpretação ao iniciar a análise qualitativa. Os autores se referem à Análise Textual Discursiva como metodologia de análise de dados e informações de natureza qualitativa com a finalidade de produzir novas compreensões sobre fenômenos e discursos. Isso se deve ao fato de que todo texto possibilita multiplicidade de leituras, relacionadas às intenções e entendimentos do autor e do leitor, baseando-se em perspectiva teórica, consciente ou não.

Além disso, convém referir que os materiais coletados podem ter sido produzidos a partir de: entrevistas, registros de observação, depoimentos e diários diversos (MORAES; GALIAZZI, 2013). Os autores citados descrevem o processo da análise textual discursiva em três etapas (Quadro 1).

Quadro 1 - Etapas da Análise Textual Discursiva

Etapas	Característica
Unitarização	Nesta etapa ocorre a desmontagem dos textos, que consiste em explorar os dados em busca de proposições e conceitos relativos ao estudo em questão.
Categorização	Consiste em construir relações entre as unidades de base, combinando-as e classificando-as, reunindo esses elementos unitários na formação de conjuntos que congregam elementos próximos, resultando daí sistemas de categorias (MORAES; GALIAZZI, 2013).
Construção do metatexto	Consiste na elaboração de novos textos que representem o conjunto de informações coletadas, constituídos de descrição e interpretação, representando o conjunto um modo de teorização sobre os fenômenos investigados (MORAES; GALIAZZI, 2013).

Fonte: Moraes e Galiazzi (2013).

Diante disso, os registros no diário de campo, as gravações e os códigos salvos dos estudantes durante a prática pedagógica foram agrupados, conforme as suas características em comum. Para Moraes e Galiazzi (2013, p. 22), a categorização: “[...] é um processo de comparação constante entre as unidades definidas no momento inicial da análise, levando a agrupamentos de elementos semelhantes”. A relação entre os dados coletados foram se estabelecendo a partir da escuta e transcrição dos áudios gravados durante a prática, análise dos algoritmos desenvolvidos pelos estudantes e percepções anotadas no diário de campo.

A dissertação foi dividida em três categorias: conhecimentos sobre algoritmos, na qual foram descritos os conhecimentos sobre algoritmos dos estudantes; as contribuições do *Software VisuAlg* no ensino de algoritmos, em que se buscou compreender como os estudantes fazem a relação entre a lógica de programação apresentada e a construção dos algoritmos em português estruturado; e percepções dos estudantes sobre o *Software VisuAlg*, análise de como o *VisuAlg* auxiliou na construção da lógica necessária para o desenvolvimento de algoritmos.

Nesta publicação, é descrita a segunda categoria: contribuições do *Software VisuAlg* no ensino de algoritmos. Sendo assim, apresenta-se como o *software* em questão contribuiu para o ensino da programação, durante a prática pedagógica desenvolvida.

3 Resultados e Discussão

3.1 Português Estruturado no Aprendizado da Lógica de Programação

É importante saber que, para os *hardwares* dos computadores funcionarem, é preciso que sejam desenvolvidos os seus *softwares*, sendo necessárias pessoas com conhecimento técnico sobre a área de programação para criá-los. Celestino (2015) apresenta que os profissionais responsáveis por criarem os sistemas são chamados de programadores ou desenvolvedores, tornando os computadores acessíveis e funcionais aos diferentes tipos de usuários. O que diferencia os desenvolvedores dos usuários é, principalmente, as suas funções em relação ao uso computador. Os desenvolvedores criam os sistemas para suprir as necessidades dos usuários em relação às várias possibilidades do seu uso.

Ao partir da abordagem de que o estudante tem a possibilidade, segundo Valente (1993, p.3), de “ensinar” o computador a fazer ou resolver o que se propõe, pode, também, utilizar programas de linguagem computacional que “não prenda o usuário as suas funcionalidades, mas sim proporcione um ambiente para desenvolver as suas atividades com o auxílio do mesmo”. Com isso, o computador, por meio dos *softwares*, pode auxiliar no ensino e na aprendizagem e possibilitar o desenvolvimento de conhecimentos para a construção da lógica de programação nos estudantes.

Com base nessa abordagem, estudantes de cursos de Informática e Computação podem criar sistemas e programas para auxiliar outras pessoas e empresas no uso do computador. Utilizam-se os *softwares* e, conseqüentemente, o computador, na construção da lógica de programação, necessária para a criação de algoritmos capazes de solucionar problemas e, assim, posteriormente, o desenvolvimento de novos programas.

Os programas de computador são algoritmos e processos escritos, em uma linguagem de programação, sendo que após a sua compilação é gerada a linguagem de máquina, que executa determinadas tarefas, conforme especificações anteriormente definidas. Algumas das linguagens de programação mais conhecidas, citadas por Puga e Riseti (2016), são: Fortran, Cobol, Pascal, C, C++, Visual Basic, Delphi, Java, entre outras.

Para o desenvolvimento dos programas, é necessário o conhecimento tanto da lógica envolvida em sua criação como da linguagem de programação, que é a representação dos códigos responsáveis por executar os diferentes processos realizados pelo programa. Puga e Riseti (2016, p. 261) apresentam que: “a linguagem de programação é formada por palavras”, as quais são agrupadas em frases com um “determinando significado”, resolvendo algum problema proposto. Dessa forma, consideram-se as palavras utilizadas como “palavras-chave” e a frases como “estrutura de programação”.

Puga e Riseti (2016) descrevem que seguir tal lógica é importante para que qualquer pessoa que trabalhe com programação possa compreender e fazer alterações, quando necessário. Com isso, escrever o código com uma lógica clara para criar ou interpretar as estruturas dos programas é fundamental para o seu correto funcionamento.

A lógica é importante para quem deseja trabalhar com programação e desenvolvimento de sistemas, pois permite criar a percepção da construção da seqüência necessária para a estruturação do código e resolver os problemas apresentados. Moraes (2000, p. 4) define que seqüência lógica são os pensamentos para poder descrever: “[...] uma seqüência de instruções, que devem ser seguidas para se cumprir uma determinada tarefa”. O autor supracitado também descreve que a:

[...] instrução é a informação que indica a um computador uma ação elementar a executar. Convém ressaltar que uma ordem isolada não permite realizar o processo completo, para isso é necessário um conjunto de instruções colocadas em ordem seqüencial lógica (MORAES, 2000, p.4).

Para compreender sobre a utilização da lógica na construção de algoritmos, é necessário saber interpretar os problemas a serem resolvidos na programação e, com isso, identificar quais as possibilidades de soluções.

O aprendizado e o aprimoramento da lógica são essenciais para a formação de um bom programador, pois serve de base para o desenvolvimento de sistemas em qualquer linguagem

de programação. Forbellone e Eberspächer (2005, p.2-3) explicam que a lógica de programação “[...] pode ser concebida pela mente treinada e pode ser representada em qualquer uma das inúmeras linguagens de programação existentes”. Isso ocorre pelo fato de que as linguagens de programação são diferentes em alguns aspectos, pois foram desenvolvidas com objetivos diversos. No entanto, com a utilização de algoritmos se pode apresentar de maneira mais fácil a lógica que se deseja programar, sem que seja necessário especificar qual a linguagem pretendida. De acordo com Forbellone e Eberspächer (2005, p. 3): “o objetivo principal do estudo da Lógica de Programação é a construção de algoritmos coerentes e válidos”.

Para entender mais sobre os algoritmos é fundamental saber que a ideia de algoritmo é antiga. Puga e Riseti (2016) relatam que matemáticos e filósofos já discutiam sobre esses no século XVII, sendo Gottfried von Leibniz um estudioso proeminente no assunto. Puga e Riseti (2016, p. 8) descrevem que Gottfried já previa: “máquinas universais de calcular e estranhas linguagens simbólicas para representar ideias complexas por meio de sinais convencionais”. Nessa época, os algoritmos eram utilizados na matemática clássica para substituir “os números por letras” e definir “símbolos que são manipulados por meio de regras práticas”.

Para simplificar, conforme Guedes (2014, p.3), um algoritmo é uma “seqüência de passos que devem ser realizados para alcançar determinado objetivo”, ou seja, um passo a passo dos procedimentos necessários para a resolução de uma tarefa. Nesse não é respondida a pergunta ‘o que fazer?’, mas sim ‘como fazer?’.

Além dos algoritmos que estão presentes no dia a dia de todos, esses também são utilizados na área de programação, quando é necessário descrever as etapas que serão efetuadas para a execução das tarefas de um programa. Isso facilita a compreensão de quem está no início do aprendizado da programação, pois são normalmente genéricos e se consegue adaptá-los a qualquer linguagem.

Forbellone e Eberspächer (2005, p. 3) destacam que um “algoritmo tem por objetivo representar mais fielmente o raciocínio envolvido na Lógica de Programação”. Dessa forma, pode-se abstrair a lógica e o processo envolvido na criação de detalhes computacionais que as linguagens de programação requerem, o que pode ocasionar uma maior compreensão da resolução dos problemas propostos.

A construção correta de um algoritmo é importante, pois uma vez proposta uma solução essa pode ser transcrita para qualquer linguagem de programação, conforme as especificações de cada uma. Na criação de um algoritmo, segundo Forbellone e Eberspächer (2005), é necessário descrever “ações claras e precisas”, e ao partir de um “estado inicial” chega-se a um “estado final e bem definido”. Isso significa que, após a criação de um algoritmo esse deve ser capaz de, ao inserir dados iniciais, gerar um resultado e sempre que forem inseridos os mesmos dados precisa

retornar resultados iguais.

Existem, basicamente, três tipos de representação de algoritmos: descrição narrativa, fluxograma e pseudocódigo (PUGA; RISSETTI, 2016; GUEDES, 2014). Guedes (2014) apresenta uma breve descrição de cada tipo: a) Descrição narrativa que consiste em analisar o problema e escrever, utilizando uma linguagem natural, os processos para a sua resolução; b) Fluxograma que consiste em analisar o problema e escrever, utilizando símbolos gráficos predefinidos, os passos para a sua resolução; c) Pseudocódigo que consiste em analisar o problema e escrever, por meio de regras, os passos para sua resolução.

Na pesquisa aqui apresentada são realizadas a análise e o estudo das contribuições do “pseudocódigo”, também chamado de “português estruturado”, que utiliza estruturas de código escritos em Língua Portuguesa para representar os algoritmos na aprendizagem dos estudantes. Tal tipo de estrutura é a base utilizada nas aulas das disciplinas de Algoritmos e Programação de Computadores, nas quais os estudantes têm o primeiro contato com a programação.

Berg (1998) e Carvalho (2007) expõem que a escrita dos algoritmos por meio do Português Estruturado é bastante simplificada, mas pode representar a maioria dos comandos, testes e laços de repetição que são utilizados nas linguagens de programação, de maneira bem semelhante. O Português Estruturado tem a característica de ser genérico e permite que seja adaptado a praticamente qualquer linguagem de programação, sendo essa estruturada ou não.

Isso não significa, conforme Carvalho (2007), que resolver os problemas algorítmicos por meio do Português Estruturado seja uma tarefa simples, mas que não é necessário tanto rigor nos detalhes como nas linguagens de programação. A autora descreve, ainda, que “o algoritmo não deixa de funcionar porque nos esquecemos de colocar um ‘;’ (ponto e vírgula) por exemplo, já um programa não funcionaria” (CARVALHO, 2007, p.7).

Por meio da experiência empírica do primeiro autor desta pesquisa se percebeu que alguns professores já utilizam a criação de algoritmos em português há vários anos, nos quais os comandos eram escritos com papel e caneta. No entanto, percebendo as dificuldades que alguns estudantes apresentavam, decidiu-se realizar uma busca para encontrar algum *software* que pudesse auxiliar esse processo, podendo o estudante visualizar o resultado do algoritmo após a sua construção.

Sendo assim, para o desenvolvimento da prática pedagógica aqui analisada, foi utilizado o *Software VisuAlg*. Os autores Almeida (2013), Costa (2010), Guedes (2014), Leite et al. (2013) e Souza (2009) descrevem experiências e vantagens do recurso, demonstrando que este é um *software* muito utilizado por diferentes professores das disciplinas de Algoritmos e Programação de Computadores em diversas instituições de ensino, principalmente, na Educação

Superior. Como citado por Guedes (2014), esse é um: “[...] bom recurso para quem está iniciando o aprendizado de algoritmos”. Além disso, o *VisuAlg* foi escolhido por estar disponível sob a licença de uso *Freeware* e por apresentar várias opções para controle e visualização dos dados trabalhados.

O *VisuAlg*, abreviação de “Visualizador de Algoritmos”, é um *software* simples de domínio público, com licença *Freeware*, que pode ser executado sem a sua instalação, ocupando aproximadamente 1,2 *megabytes* de espaço em disco. Isso possibilita a sua rápida utilização em qualquer computador, tanto com sistema operacional *Linux* ou *Windows*. Foi criado por Cláudio Morgado de Souza, professor da Universidade Severino Sombra - RJ, do Centro de Ensino Superior de Valença - RJ.

Além do exposto, a empresa Apoio Informática Ltda. (2018, texto digital), que ajudou a desenvolver o *VisuAlg*, menciona que “A linguagem que o *VisuAlg* interpreta é simples: é uma versão portuguesa dos pseudocódigos largamente utilizados nos livros de introdução à programação”. Como descrito no *site* da referida empresa, o *VisuAlg* surgiu da necessidade de “[...] uma ferramenta que permitisse aos alunos iniciantes em programação o exercício dos seus conhecimentos num ambiente próximo da realidade”. Em outras palavras, o recurso foi desenvolvido com o objetivo de exercitar a criação de algoritmos em um *software* que tivesse o processo parecido ao de uma linguagem de programação, além de codificar e executar algoritmos em uma linguagem simples, na Língua Portuguesa, e com facilidade para edição do que foi escrito.

Por apresentar facilidade para digitar, corrigir, executar e depurar os pseudocódigos, o *VisuAlg* pode ser utilizado por professores nas diversas etapas da construção do conhecimento sobre a lógica de programação, nos diferentes níveis de ensino. Para Guedes (2014), o *software* é considerado “um bom recurso” para ser utilizado no meio acadêmico, não somente para aprender algoritmos, mas também para “melhor entender sua execução”. O uso do *VisuAlg* também é defendido por Costa (2010, p.6), que cita a existência de “vários recursos didáticos que auxiliam na explicação do funcionamento dos programas”, auxilia na execução do passo a passo do código e na análise do conteúdo das variáveis.

Para Souza (2009), resolver um problema de lógica, por meio da escrita de algoritmos na Língua Portuguesa, com o auxílio do *Software VisuAlg*, facilita a compreensão e aprendizagem dos estudantes sobre o processo de criação da estrutura do código. De acordo com o autor, com o conhecimento dos principais comandos, que são em português, desenvolve-se um código de fácil entendimento. O estudante tem a possibilidade de realizar testes para saber se existem erros de codificação. Com isso, pode-se executar o “programa” criado e visualizar o seu comportamento passo

a passo e analisar os dados resultantes.

Para que se possa fazer uma melhor comparação entre a estrutura de criação de um algoritmo com a utilização do *VisuAlg* e o mesmo código escrito no *Dr. Java*, na Figura 1A e na Figura 1B são apresentadas as principais diferenças entre as duas abordagens.

Figura 1 - Comparativo entre Português Estruturado e a Linguagem Java a partir do *Software VisuAlg*. (B) a partir do *Software Dr. Java*)

(A)

```
ALGORITMO "exemploVisualg"
VAR
  idade : INTEIRO
INICIO
  ESCREVA ("Digite a sua idade: ")
  LEIA (idade)
  SE (idade >= 18) ENTAO
    ESCREVA ("Você é maior de idade.")
  SENAO
    ESCREVA ("Você é menor de idade.")
  FIMSE
FIMALGORITMO
```

(B)

```
class exemploJava
{
  public static void main (String arg[])
  {
    int idade;
    idade = Entrada.leiaInt ("Digite a sua idade: ");
    if (idade >= 18)
      System.out.println ("Você é maior de idade.");
    else
      System.out.println ("Você é menor de idade.");
  }
}
```

Fonte: os autores.

Observa-se que, tanto na Figura 1A como na Figura 1B, a estrutura do algoritmo é parecida, mas na segunda imagem é preciso compreender um pouco da Língua Inglesa para saber quais comandos utilizar. É necessário estar atento a alguns detalhes que são específicos da linguagem *Java*, como o ponto e vírgula (;) ao final de algumas linhas do código, abrir e fechar parênteses e chaves corretamente, usar letras maiúsculas e minúsculas nos comandos e palavras reservadas do sistema.

Pode-se perceber que a utilização de uma linguagem de programação, como o *Java*, não é uma função simples, pois é preciso ter atenção a detalhes que são importantes para o seu correto funcionamento. Já na utilização do Português Estruturado, o foco principal não é cuidar a escrita, mas sim a resolução do problema e o desenvolvimento da lógica necessária para resolvê-lo. Mesmo assim, é necessário seguir uma lógica para a criação dos algoritmos, de maneira que todos compreendam o que foi escrito. Por isso, é preciso utilizar uma estrutura mínima para o seu correto funcionamento.

Dessa forma, os estudos apresentados sobre a utilização de diferentes ferramentas para auxiliar no ensino de

programação vêm complementar e dar suporte teórico à prática pedagógica desenvolvida. Esses ilustram, também, que os professores já percebem essas dificuldades há algum tempo e sugerem utilizar diferentes ferramentas para dar suporte ao ensino de programação. Dentre essas ferramentas estão os *softwares* que auxiliam no ensino de algoritmos e que utilizam códigos em português. Sendo assim, os estudantes podem escrever os programas, testá-los e analisá-los passo a passo para encontrar onde estão os erros e inconsistências.

3.2 Relato da Prática Pedagógica

Ao iniciar a prática pedagógica, foram apresentadas algumas situações cuja resolução poderia ocorrer por meio de algoritmos. Com isso, buscou-se situar os estudantes nos assuntos que seriam trabalhados no decorrer das aulas, tais como: tipos de dados primitivos; conceito de variáveis e constantes; operadores relacionais, lógicos e de atribuição; comandos de entrada (LEIA) e saída de dados (ESCREVA); comando condicional SE (simples, composto e encadeado) e comandos de repetição PARA e ENQUANTO.

Com base na experiência do primeiro autor desta publicação, uma situação que costumeiramente acontece quando esses conteúdos são trabalhados é que alguns estudantes esquecem de fechar os comandos. Por exemplo, eles não lembram que o SE precisa do FIMSE ao final para funcionar corretamente. Sendo assim, um aspecto favorável da utilização do *VisuAlg* é que ele apresenta mensagens informando que esperava encontrar o comando de fechar. Por meio de mensagens como essas, os estudantes conseguem acompanhar se o código digitado está correto e, também, encontrar dicas de como resolver o erro. Sendo assim, as mensagens apresentadas pelo *Software VisuAlg* podem ser uma referência na hora de desenvolver os algoritmos propostos.

Dando prosseguimento à prática desenvolvida, evidenciou-se que, à medida que os estudantes terminavam de resolver os exercícios propostos, um auxiliava o outro na resolução dos problemas e nas dúvidas encontradas. Além disso, tentavam fazer alterações no código para ver qual resultado apareceria.

Uma situação interessante que aconteceu foi quando dois estudantes estavam conversando e rindo, o que chamou a atenção dos outros. Ao conferir o que estavam fazendo, percebeu-se que haviam criado um novo algoritmo. Neste, ao serem digitadas algumas informações, era mostrado um resultado. Conforme as respostas eram escritas, o algoritmo apresentava mensagens divertidas. O algoritmo criado foi nomeado pelos estudantes de "Testando", como pode ser visto na Figura 2.

Figura 2 - Algoritmo “Testando”, criado pelos estudantes

algoritmo "Testando"

```
var
nome : caracter
idade : inteiro
trabalho : caracter

inicio

  escreval ("BEM-VINDO AO TESTANDO...")

  escreva ("Qual o seu nome? ")
  leia (nome)

  escreval (nome, ", este nome parece meio estranho.")

  escreva ("Qual a sua idade? ")
  leia (idade)

  se (idade<18) entao
    escreval ("Idade de bebê?")
  senao
    escreval ("Idade OK!")
  fimse

  escreva ("Você trabalha (S/N)? ")
  leia (trabalho)

  se (trabalho="N") entao
    escreval ("Folgado!")
  senao
    escreval ("Tudo certo!")
  fimse
```

fimalgoritmo

Fonte: dados da pesquisa.

É por meio de momentos como este que os estudantes criam significado para aquilo que estão aprendendo, pois se evidencia compreenderam o que foi explicado. Essa ideia é corroborada por Moran, Masetto e Behrens (2007, p.23):

Aprendemos quando descobrimos novas dimensões de significação que antes se nos escapavam, quando vamos ampliando o círculo de compreensão do que nos rodeia [...]. Aprendemos mais quando estabelecemos pontes entre a reflexão e a ação, entre a experiência e a conceituação, entre a teoria e a prática; quando ambas se alimentam mutuamente.

Outros dois comandos importantes para a criação de algoritmos mais complexos são os Laços de Repetição (PARA e ENQUANTO), que servem para criar *loopings* e repetir diferentes comandos ou mostrar mensagens. Para testar o correto funcionamento dos Laços de Repetição, o *VisuAlg* dispõe de uma ferramenta que executa o código passo a passo (linha por linha), continuando somente após ser pressionado o botão correspondente. Esta opção é muito útil para que os estudantes possam acompanhar a execução de seu código e identificar não só o local no qual pode estar acontecendo algum erro, como também os valores que as variáveis estão recebendo. Com isso, pode-se ter um maior controle do que é feito em cada linha do algoritmo.

Um dos algoritmos analisados pelos estudantes está representado na Figura 3, sendo que neste era preciso informar o que o código faz e descobrir o erro contido neste.

Figura 3 - Exercício de Laço de Repetição ENQUANTO com erro

ALGORITMO "Exercicio05"

```
VAR
numero : INTEIRO
contador : INTEIRO
soma : INTEIRO
resultado : INTEIRO

INICIO
contador <- 0
resultado <- 0

ENQUANTO (numero >= 0) FACA
  ESCREVA("Informe um número: ")
  LEIA(numero)
  SE (numero >= 0) ENTAO
    contador <- contador + 1
    soma <- soma + numero
  FIMSE
FIMENQUANTO

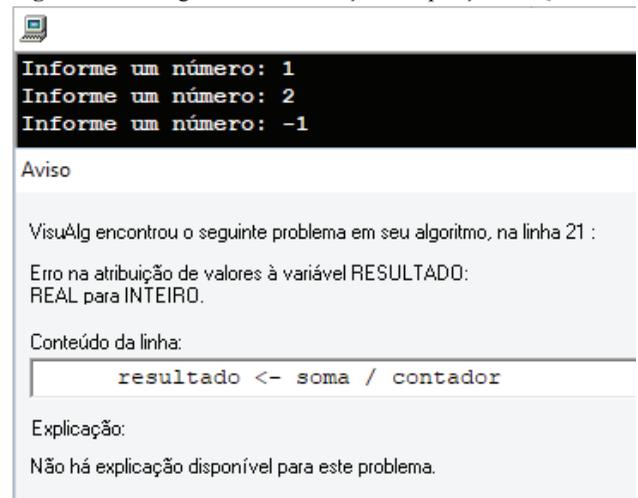
resultado <- soma / contador
ESCREVA("A média é: ", resultado)

FIMALGORITMO
```

Fonte: dados da pesquisa.

Ao testarem o código para ver o seu funcionamento, inicialmente, os estudantes não visualizaram nenhum erro. Contudo, perceberam que o algoritmo ficava rodando e solicitando números sem parar. Nesse momento, um estudante, ao analisar melhor o código, percebeu pelo teste “ENQUANTO (número = 0) FACA” que, para encerrar a execução, era necessário digitar um número negativo. Dando continuidade aos testes, constatou que, no momento do encerramento do laço de repetição, era apresentado erro no código, conforme apresentado na Figura 4.

Figura 4 - Mensagem de erro do Laço de Repetição ENQUANTO



Fonte: dados da pesquisa.

Ao analisar a mensagem de erro do *VisuAlg*, perceberam que são apresentadas dicas de como corrigi-lo. Isso facilitou aos estudantes encontrarem o erro e corrigir o problema. Hostins e Raabe (2007, p.98) defendem que as ferramentas utilizadas para o ensino de algoritmos precisam permitir:

maior grau de autonomia ao aluno fornecendo recursos de

interface que reduzem os enganos iniciais e que possibilitassem acompanhar detalhadamente a execução do programa, além de incluir um mecanismo simples de verificação da correção da solução.

Dessa forma, verificou-se que, a cada exercício desenvolvido pelos estudantes, menos auxílio eles precisavam. Essa autonomia se atribui ao fato de eles ajudarem uns aos outros a esclarecerem as suas dúvidas e, também, por estarem compreendendo os comandos que foram estudados para a criação dos códigos dos algoritmos. Leite et al. (2013, p.640) também comentam: “[...] pode-se observar que os estudantes de Programação [com o uso do *VisuAlg*] puderam aprender de maneira mais prática, rápida e agradável os conceitos de algoritmos”. Evidenciou-se esse mesmo aprendizado nos estudantes participantes da prática pedagógica desenvolvida.

Além disso, autores como Almeida (2013), Costa (2010), Guedes (2014), Leite et al. (2013), Puga e Rissetti (2016), Souza (2009), defendem o uso de alguma ferramenta para auxiliar no ensino de algoritmos, e citam o *VisuAlg* como um bom recurso para quem está iniciando na criação de algoritmos ou na programação. Costa (2010) ainda descreve que os recursos didáticos disponíveis auxiliam no ensino dos comandos em português, pois foram pensados justamente para auxiliar no processo de ensino da lógica de programação.

Conforme já referido, após a prática pedagógica, foi aplicado aos estudantes um questionário, que teve por objetivo analisar a impressão, os comentários e o aprendizado dos estudantes após construírem diferentes tipos de algoritmos com o auxílio do *Software VisuAlg*.

Na Questão 1 - “Qual forma de construção dos algoritmos você preferiu, digitando no editor de texto ou utilizando o Software VisuAlg? Quais as diferenças que você percebeu ao escrever os algoritmos das duas maneiras?”, todos os estudantes escreveram que preferem utilizar o VisuAlg. Percebeu-se que os estudantes tiveram uma boa impressão com a utilização deste software, pois descreveram a facilidade de criar os algoritmos e a possibilidade poder testá-los, identificando os possíveis erros. Tal aspecto ficou evidente na resposta de um estudante: “A forma que eu preferi foi a utilização do Software VisuAlg, pois com esse caso ocorra algum erro o software irá informar e já no editor de texto não se tem certeza absoluta se o algoritmo está totalmente correto”. O mesmo é percebido na resposta de outro estudante, que descreveu que achou mais fácil aprender ao utilizar o VisuAlg: “Preferi pelo Software VisuAlg, pois foi uma maneira mais fácil de aprender”.

Souza (2009, p.8) corrobora com as impressões apresentadas pelos estudantes:

A experiência mostra que ele facilita o entendimento de como um programa de computador funciona, possibilita um feedback imediato sobre a correção e exatidão do pseudocódigo digitado, e o que se considera mais importante, instiga o aluno a experimentar e ver o resultado de suas alterações imediatamente.

Com a Questão 2 - “Aprender os algoritmos na Língua

Portuguesa auxiliou ou não na compreensão dos códigos e comandos? Explique, por quê? ”, pretendeu-se saber qual a percepção dos estudantes em desenvolver os algoritmos em língua portuguesa, se isso auxiliou ou não na compreensão da lógica de programação. Pôde-se perceber que ajuda sim a compreender a construção dos códigos, conforme descrito pelo estudante em sua resposta: “Auxiliou sim, pois o português é o nosso dia a dia e a compreensão ao se ler um código que esteja em português fica mais viável, ainda mais para nós que somos iniciantes”. Outro estudante também escreveu que facilitou o entendimento dos códigos, “[...] pois pode ser aprendido a lógica de forma mais fácil”.

Com a Questão 3 - “O uso do *Software VisuAlg* auxiliou ou não na compreensão e criação dos algoritmos? Explique, por quê?”, pretendeu-se que os estudantes escrevessem as suas percepções sobre o uso do *VisuAlg*. Analisando as respostas, percebeu-se que todos julgaram o aprendizado dos algoritmos satisfatório com este *software*. Eles destacaram as seguintes vantagens: observar o funcionamento do algoritmo desenvolvido; encontrar os erros com facilidade por meio das mensagens que indicam o problema e algumas sugestões para correção; “rodar” passo a passo o algoritmo; acompanhar o seu funcionamento e os valores salvos nas variáveis.

Valente (1993) defende que existem basicamente dois tipos de utilização dos *softwares*: de um lado está o estudante que aprende por meio do computador, e do outro o estudante que “ensina” o computador a como fazer determinada ação. Com isso, baseando-se no que foi exposto por Valente (1993, p.3), de que: “[...] o computador pode ser visto como uma ferramenta que permite ao aprendiz resolver problemas ou realizar tarefas”, pode-se classificar o *VisuAlg* como um *software* em que o estudante cria o seu algoritmo instruindo o computador sobre o que precisa ser feito. Além disso, pode-se considerar tal *software* como uma “ferramenta educacional”.

Ao continuar com a análise das respostas dos estudantes ao questionário, a próxima questão tratou sobre como os estudantes avaliam as aulas ministradas e, principalmente, a criação dos algoritmos em português com o uso do VisuAlg. Para isso, com a Questão 4 - “Você acharia interessante que as aulas de algoritmos sempre utilizassem algum software em português para criar os códigos? Justifique sua resposta.”, procurou-se conhecer a opinião e percepção dos estudantes sobre a metodologia de ensino proposta e utilizada nas aulas.

Observou-se que os estudantes consideram que o uso de softwares para aprender a criação de algoritmos pode auxiliar no ensino da lógica de programação, utilizando códigos em português. Isso fica comprovado na resposta de um estudante, que escreveu: “Sim, pois a maioria dos alunos aprenderia melhor assim”. Outro estudante também destacou as potencialidades do uso de softwares para encontrar os erros em seus algoritmos: “Acho essencial, pois com um software podemos aprender com nossos próprios erros”.

No geral, analisou-se que praticamente todos os estudantes avaliaram, positivamente, a utilização do Software VisuAlg

como ferramenta para o ensino da lógica de programação, por meio da criação de algoritmos escritos em Língua Portuguesa. Almeida et al. (2002), sobre o uso de softwares, ponderam que:

A motivação dos alunos aumentou sensivelmente, uma vez que as aulas em laboratório, acrescida da possibilidade de escrever os programas/algoritmos diretamente no computador e da facilidade de testar o seu funcionamento, são bem mais estimulantes e ilustrativas do que na forma tradicional, onde os algoritmos são escritos no papel (ALMEIDA et al., 2002, p.9).

Também se teve esta percepção, pois os estudantes se mostraram interessados e motivados na resolução dos exercícios propostos, ao implementar códigos mais elaborados e criando funcionalidades extras para melhorarem os seus algoritmos. Percebeu-se, ainda, que todos julgaram válida a utilização do *VisuAlg* para a criação dos algoritmos em português, principalmente, pela possibilidade de poderem testar e visualizar o resultado ou os erros do código, por meio das dicas e sugestões apresentadas. Isso facilitou a resolução dos problemas computacionais apresentados nos exercícios propostos durante a prática pedagógica.

3 Conclusão

Após a prática se constatou que os estudantes compreenderam os comandos e algoritmos trabalhados, pois conseguiram analisar os problemas apresentados nos exercícios e criaram a codificação necessária para resolvê-los. Em certas atividades, alguns estudantes criaram formas diferentes de resolução e propuseram algoritmos extras, demonstrando que estavam compreendendo a lógica necessária para o início da programação.

Além disso, podem ser destacados os seguintes aspectos: os estudantes desenvolveram novos algoritmos para simular situações e brincadeiras, demonstrando que estavam compreendendo a utilização do *VisuAlg* para a criação dos códigos; a utilização do *VisuAlg* se mostrou muito útil na criação dos algoritmos, tendo em vista que os estudantes puderam testar o código digitado encontrando facilmente os erros apresentados.

As respostas dos estudantes ao questionário expressam que: 1) os estudantes preferiram desenvolver os algoritmos utilizando o *Software VisuAlg* ao editor de textos, devido a facilidade de poder testar e encontrar os erros; 2) a aprendizagem da programação em português permitiu que os estudantes focassem na resolução dos problemas computacionais e na lógica necessária para resolvê-los, não em compreender o significados de comandos em inglês das linguagens de programação; 3) foram destacadas as funcionalidades do *VisuAlg* como ponto forte em sua utilização, pois se pode visualizar, por meio da indentação, qual comando está dentro de outro, identificando os erros apresentados nas mensagens, bem como dicas de como resolvê-los; 4) todos os estudantes que responderam ao questionário demonstraram saber qual

o resultado do algoritmo de exemplo apresentado e qual a função dos comandos deste; 5) os estudantes demonstraram que a prática pedagógica foi satisfatória e que eles puderam compreender e criar os algoritmos propostos sem muita dificuldade, recomendando o seu uso em outras turmas.

Com a experiência adquirida e os relatos apresentados pelos estudantes, na prática pedagógica, pôde-se perceber que a utilização de *softwares* e ferramentas auxilia no ensino de algoritmos nas aulas iniciais de programação. O uso de diferentes recursos no ensino da lógica de programação facilita o entendimento de como um programa de computador funciona. Além disso, instiga o estudante a testar o algoritmo que ele está desenvolvendo, para ver o seu resultado e acompanhar as alterações no conteúdo das variáveis no decorrer da depuração do código.

Ao concluir a pesquisa foi possível perceber que, conforme citado por diferentes autores, o uso de ferramentas que possam auxiliar o ensino e aprendizagem da lógica necessária para a criação de algoritmos é bem-vinda em sala de aula. Tais recursos proporcionam aos estudantes diferentes maneiras de aprender, estimulando-os a conhecerem cada vez mais sobre o desenvolvimento do código-fonte de programas. Além disso, essa abordagem possibilita aos estudantes compreenderem a lógica envolvida em sua criação e promove o desenvolvimento de habilidades imprescindíveis para atuar na área de Informática e Computação.

Referências

- ALMEIDA, E.S. et al. *AMBAP*: um ambiente de apoio ao aprendizado de programação. 2002. Disponível em: <https://www.researchgate.net/publication/266037757>. Acesso em: 19 maio 2021.
- ALMEIDA, R.S. *Aprendendo Algoritmo com VisuAlg*. Rio de Janeiro: Ciência Moderna, 2013.
- APOIO, Informática Ltda. *VisuAlg*. Disponível em: <http://www.apoioinformatica.inf.br/produtos/visualg>. Acesso em: 19 maio 2021.
- BERG, A.C.; FIGUEIRÓ, J.P. *Lógica de Programação*. Canoas: Ulbra, 1998.
- CARVALHO, F.P. *Apostila de Lógica de Programação - ALGORITMOS*. 2007. Disponível em: https://fit.faccat.br/~fpereira/apostilas/apostila_algoritmos_mar2007.pdf. Acesso em: 19 maio 2021.
- CELESTINO, A.L. *O limite entre o desenvolvedor e o usuário*. 2015. Disponível em: <https://www.professionaisti.com.br/2015/03/o-limite-entre-o-desenvolvedor-e-o-usuario/>. Acesso em: 19 maio 2021.
- COSTA, R.H. P.; FARIA, E.S.J.; YAMANAKA, K. Programação em duplas no aprendizado de programação de computadores em um curso de Engenharia de Produção: um estudo empírico. *REIC*, v.10, p. 4, 2010.
- FORBELLONE, A.L.V.; EBERSPÄCHER, H.F. *Lógica de programação: a construção de algoritmos e estruturas de dados*. São Paulo: Prentice Hall, 2005.
- GUEDES, S. *Lógica de programação algorítmica*. São Paulo: Pearson Education, 2014.

- HOSTINS, H.; RAABE, A. Auxiliando a aprendizagem de algoritmos com a ferramenta Webportugol. in: CONGRESSO DA SBC. 2007. Disponível em: <https://www.researchgate.net/publication/268344218>. Acesso em: 19 maio 2021.
- JUNIOR, J.A.T.L.; VIEIRA, C.E.C.; VIEIRA, P.P. *Dificuldades no processo de aprendizagem de Algoritmos: uma análise dos resultados na disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ - Campus Paracambi*. Cad. UniFOA 2015. doi: <https://doi.org/10.47385/cadunifoa.v10i27.293>.
- LEITE, V.M. et al. VisuAlg: estudo de caso e análise de compilador destinado ao ensino de Programação. *Rev. Nuevas Ideas Inform. Educ. TISE 2013*. Disponível em: <http://www.tise.cl/volumen9/TISE2013/637-640.pdf>. Acesso em: 19 maio 2021.
- MORAES, P.S. *Lógica de Programação*. 2000. Disponível em: <http://ftp.unicamp.br/pub/apoio/treinamentos/logica/logica.pdf>. Acesso em: 19 maio 2021.
- MORAES, R.; GALIAZZI, M.C. *Análise textual discursiva*. Ijuí: Unijuí, 2013.
- MORAN, J.M.; MASETTO, M.T.; BEHRENS, M.A. *Novas tecnologias e mediação pedagógica*. Campinas: Papirus, 2007.
- MOREIRA, J.V.; YNOGUTTI, C.A. *Piratas do Futuro - Ferramenta para suporte ao ensino de algoritmos e lógica de programação*. In: CONGRESSO DE INICIAÇÃO CIENTÍFICA DO INATEL - INCITEL 2012. Disponível em: <https://inatel.br/incitel/anais-incitel/incitel-2012-1/23-anais-2012-1/file>. Acesso em: 19 maio 2021.
- PUGA, S.; RISSETI, G. *Lógica de programação e estruturas de dados, como aplicações em Java*. São Paulo: Pearson Education, 2016.
- RICHARDSON, R.J. *Pesquisa Social: métodos e técnicas*. São Paulo: Atlas, 1999.
- SOUZA, C.M. VisuAlg - Ferramenta de Apoio ao Ensino de Programação. Universidade Severino Sombra, CECETEN. *Rev. Eletr. TECCEN*, v.2, n.2, 2009. doi: <https://doi.org/10.21727/teccen.v2i2.234>.
- VALENTE, J.A. *Diferentes usos do Computador na Educação*. Núcleo de Informática Aplicada à Educação - NIED/UNICAMP, 1993. Disponível em: http://www.nce.ufjf.br/ginape/iga502/Material_aulas/Diferentes%20usos%20do%20computador%20na%20educa%C3%A7%C3%A3o.pdf. Acesso em: 19 maio 2021.
- YIN, R.K. *Estudo de caso: planejamento e métodos*. Porto Alegre: Bookman, 2015.