

## METODOLOGIAS DE ENSINO DE LÓGICA DE PROGRAMAÇÃO E ALGORITMOS EM CURSOS DE GRADUAÇÃO

Rodrigo Oliveira Zacarias<sup>1</sup>

Pós-graduando em Docência do Ensino Superior

Denise Ribeiro Barreto Mello<sup>2</sup>

Doutora em Saúde Mental pelo IPUB/UFRJ.

**Resumo:** Um número considerável de alunos apresenta dificuldades de aprendizagem de lógica de programação e algoritmos nas fases iniciais dos cursos de graduação, tendo em vista a alta capacidade de raciocínio e de abstração exigidos. Face isso, o objetivo deste artigo é realizar uma revisão bibliográfica sobre metodologias de ensino que mais contribuem para o processo de ensino-aprendizagem de tais disciplinas. Foi realizado um levantamento bibliométrico no Google Acadêmico® e selecionados 10 artigos científicos que abordassem propostas pedagógicas inovadoras e experiências bem-sucedidas na área. A partir disso, foi feita uma análise comparativa sobre trabalhos com base nos resultados averiguados. Por fim, constatou-se que as metodologias fornecem os suportes para a escolha da estratégia de ensino mais adequada, sendo que as mesmas podem ser aplicadas de forma isolada ou em conjunto, conforme a evolução do aluno.

**Palavras-chave:** Ensino-Aprendizagem; Lógica; Algoritmos; Ensino Superior.

**Abstract:** A considerable number of students present difficulties in learning programming logic and algorithms in the initial stages of undergraduate courses, due the high reasoning and abstraction skills required. In this context, the purpose of this article is to carry out a bibliographic review on teaching methodologies that contribute most to the teaching-learning process of such disciplines. A bibliometric survey was conducted in Google Academic® and selected 10 scientific articles that addressed innovative pedagogical proposals and successful experiences in the area. From this, a comparative analysis about the papers based on the results obtained was made. Finally, it was verified that the methodologies

---

<sup>1</sup>Atua na área de Gestão e de Análise e Desenvolvimento de Sistemas de Informação, [rodrigozacarias.ti@gmail.com](mailto:rodrigozacarias.ti@gmail.com)

<sup>2</sup>Coordenadora do curso de Psicologia do Centro Universitário Redentor, [coordpsicologia@redentor.edu.br](mailto:coordpsicologia@redentor.edu.br)

provide the support for the choice of the most appropriate teaching strategy, and they can be applied in isolation or together, according to the student's evolution.

**Keywords:** Teaching-Learning; Logic; Algorithms; Higher Education.

## 1. INTRODUÇÃO

O processo de ensino-aprendizagem de algoritmos e de lógica de programação tem sido uma temática de ampla discussão nas áreas de Educação e Informática, principalmente, com a criação de novos recursos para apoiar esse procedimento. No Brasil, a aprendizagem de princípios de programação acontece, em sua maioria, durante os cursos de graduação da área de informática ou afins. Em outros países, como Estados Unidos e Estônia, esses conceitos são praticados nas séries iniciais com o objetivo de preparar os alunos desde cedo para lidar com os avanços tecnológicos e desenvolver a capacidade de adaptação a mudanças (SILVA *et al.*, 2015).

Esse contato tardio com os princípios de lógica de programação e algoritmos acarreta em grande dificuldade enfrentada por parte dos estudantes nas disciplinas introdutórias de programação, o que ocasiona em desistência e, conseqüentemente, em um alto índice de evasão (BELCHIOR; MEIRELES; FERREIRA, 2016). O conceito de lógica, essencial para a programação de computadores, pode ser definido como a “arte de bem pensar”. Considerando que a forma mais complexa do pensamento é o raciocínio, a lógica estuda a “correção do raciocínio”, ordenando o pensamento e a razão (FORBELLONE; EBERSPÄCHER, 2005).

Segundo Manzano e Oliveira (2016), algoritmo pode ser definido como uma descrição sequencial de passos para a realização de uma ação dentro de um tempo pré-definido ou programado e com um resultado esperado. É considerado uma das bases da computação, sendo um dos principais mecanismos pelos quais os computadores realizam suas operações. Além disso, Jesus e Brito (2009) ressaltam que não existe uma única solução para um problema que envolve algoritmos. Caminhos diferentes podem levar ao mesmo resultado, sendo a solução por algoritmos uma questão bastante subjetiva.

As disciplinas sobre algoritmos e lógica de programação, muito comuns em cursos de graduação relacionados à Computação ou Engenharias, exigem que os estudantes absorvam uma grande quantidade de conteúdo, habilidades e competências nas fases

iniciais dos cursos, tais como raciocínio lógico, capacidade de abstração da solução e resolução de problemas, além de uma representação em linguagem formal ou computacional (JESUS; BRITO, 2009).

Durante as aulas dessas disciplinas, o aluno precisa apresentar uma capacidade alta de abstração para que possa obter uma compreensão completa do problema. De fato, apresentar essa habilidade aliadas às demais não é uma tarefa trivial, o que acaba criando uma série de obstáculos, como a dificuldade para entender lógica computacional, falta de motivação e desânimo gerados pelo despreparo para cursar as disciplinas. Com isso, essas disciplinas frequentemente apresentam altos índices de reprovação e, conseqüentemente, evasão de estudantes, sendo elas um dos fatores que mais dificultam a continuidade dos alunos no curso (BELCHIOR; MEIRELES; FERREIRA, 2016).

O bom desempenho dos alunos está atrelado a diversos fatores, com destaque ao fator motivacional. Muitos autores ressaltam a necessidade de aplicação de metodologias que auxiliem o desenvolvimento de todos os níveis de conhecimento e habilidades dos alunos, baseando-se em muitos casos na Taxonomia de Bloom. Criada pelo pedagogo e psicólogo Dr. Benjamin Samuel Bloom, essa sistemática de aprendizagem apresenta seis níveis interdependentes: conhecimento, compreensão, aplicação, análise, síntese e avaliação. Com isso, os alunos embasam a construção do conhecimento a partir da decomposição e análise dos problemas, propondo soluções como agentes ativos no processo de ensino-aprendizagem (JESUS; BRITO, 2009).

Nesse contexto, o uso de metodologias ou estratégias de ensino que contribuam para um melhor processo de ensino-aprendizagem dessas disciplinas iniciais da Computação tem atraído a atenção dos profissionais que trabalham nos cursos de graduação que contêm essas disciplinas, principalmente, quando tais métodos abrangem uma perspectiva pedagógica inovadora, com foco na motivação e no protagonismo do aluno. O desafio para os professores é traçar novas estratégias de ensino que tornem a aprendizagem mais dinâmica, ágil e prazerosa para os estudantes, solidificando os ensinamento da área (BELCHIOR; MEIRELES; FERREIRA, 2016).

A partir desse cenário, o objetivo deste trabalho é realizar uma revisão bibliográfica sobre as metodologias ou estratégias de ensino aplicadas em sala de aula que mais contribuem para uma melhor aprendizagem em disciplinas de lógica de programação e algoritmos em cursos de graduação. Com isso, apresentar um comparativo entre as

metodologias que apresentaram os melhores resultados no processo de ensino-aprendizagem.

## 2. METODOLOGIA DE PESQUISA

Conforme Gil (2017), esta pesquisa é classificada como aplicada em sua natureza, abordando aspectos exploratórios, pelo intuito da geração de resultados ainda desconhecidos, e bibliográfica quanto ao procedimento, tendo em vista a busca trabalhos já presentes na literatura. A estratégia de pesquisa foi organizada e estruturada em quatro etapas sequenciais, conforme apresentado na Figura 1.



**Figura 01 - Etapas da estratégia de pesquisa.**

Fonte: Elaboração própria

Na etapa inicial deste trabalho, foi definido o Google Acadêmico® (<https://scholar.google.com>) como a base de dados para a bibliometria e a seleção dos trabalhos científicos para a revisão bibliográfica. Essa base permite a busca dos mais variados trabalhos científicos como artigos, ensaios, monografias, dissertações e teses, tanto nacionais quanto internacionais, o que configura uma grande abrangência para o tema a ser pesquisado (GOOGLE, 2019).

Na segunda etapa, é realizada a pesquisa bibliométrica na base de dados. O objetivo é aferir o número de publicações científicas existentes sobre os termos relacionados à temática deste artigo, bem como obter o conjunto de trabalhos que serão analisados. Para isso, são utilizadas palavras-chave e tesouros referentes ao tema conforme a Figura 2.



**Figura 02 - Palavras-chave, tesouros e termos correspondentes utilizados na busca na base de dados.**

Fonte: Elaboração própria

Neste caso, os termos de pesquisa são Metodologias de Ensino, Lógica de Programação, Algoritmo e Ensino Superior. Os tesouros foram definidos para que a pesquisa abranja o máximo de trabalhos que retratem as principais temáticas de investigação em seus conteúdos. A utilização de termos na língua inglesa auxilia na identificação de trabalhos internacionais. Com base nos termos elencados, foi formulada a estratégia de consulta apresentada no Quadro 1.

```
(
TITLE-ABS-KEY (metodologia ensino OR teaching methodology OR learning technique) AND
TITLE-ABS-KEY (lógica programação OR programming logic OR computational logic) AND
TITLE-ABS-KEY (algoritmo OR algorithm) AND
TITLE-ABS-KEY (ensino superior OR graduação OR graduation course OR undergraduate)
)
```

**Quadro 01 - Pesquisa em base de conhecimento.**

Fonte: Elaboração própria

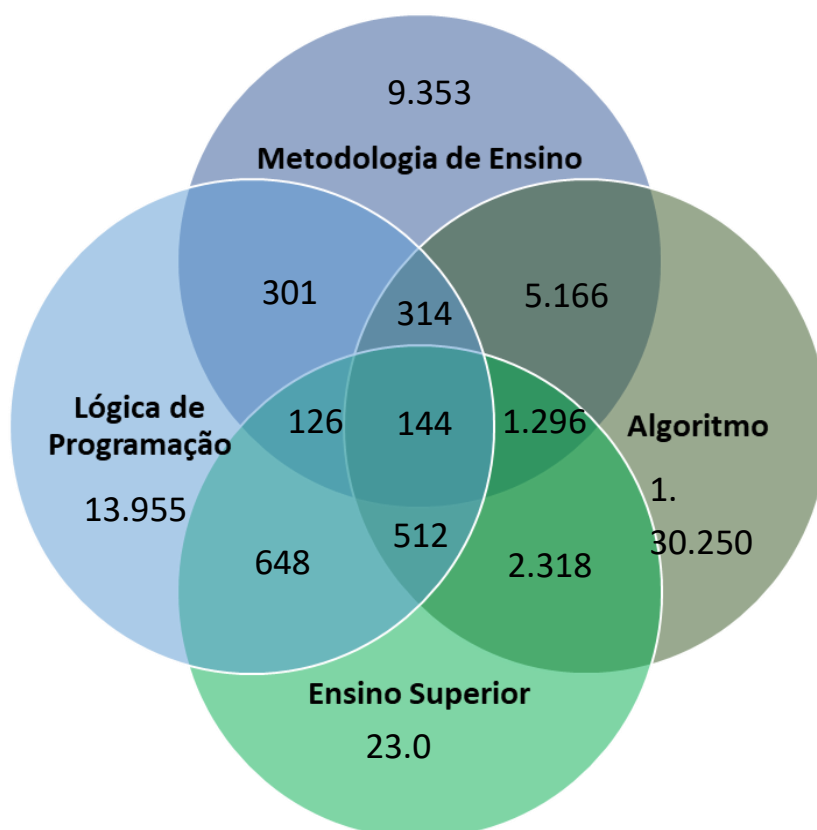
Na terceira etapa, é realizada a seleção dos trabalhos científicos. A escolha é feita por meio da leitura dos resumos e/ou *abstracts* para identificar as metodologias e os resultados obtidos com sua aplicação. O objetivo é obter uma variedade de metodologias aplicadas em cursos de graduação para o ensino de lógica de programação e algoritmos

que alcançaram bons resultados. O recorte temporal contempla os trabalhos publicados no período de 2013 a 2018.

Na última etapa, é feita a análise com a leitura completa dos trabalhos científicos selecionados. Em seguida é realizado um comparativo entre os métodos e os resultados apresentados por cada trabalho com a discussão de quais metodologias podem ser destacadas para o uso frequente no ensino superior.

### 3. BIBLIOMETRIA E SELEÇÃO DOS TRABALHOS CIENTÍFICOS

Conforme ilustrado na Figura 2 **Error! Reference source not found.**, foi realizada uma busca na base de dados Google Acadêmico® utilizando os tesouros dos termos de pesquisa. No primeiro momento, a busca foi realizada consultando-se um termo de cada vez e, posteriormente, relacionando-os entre si. Como resultado, se obteve o diagrama de *Venn* apresentado na Figura 3, ilustrando a quantidade trabalhos identificados em cada tema de pesquisa e suas interseções.



**Figura 03 - Diagrama de Venn com os resultados da pesquisa nas bases de dados do Google Acadêmico®.**

Fonte: Elaboração própria

Dos quatro conceitos, algoritmo é o tema mais explorado, com uma quantidade muito superior a metodologia de ensino, lógica de programação e ensino superior. Considerando os trabalhos que abordam os quatro temas pesquisados foram identificados 144 documentos no Google Acadêmico®, dentre livros, capítulos de livros, ensaios, artigos científicos, resenhas, etc.

É possível perceber que a quantidade de trabalhos relacionando os quatro conceitos teve um número menos expressivo, se comparado com as demais interseções de temas. Isso demonstra que metodologias de ensino de lógica de programação e algoritmos ainda são uma área a ser explorada.

Em seguida, foi efetuada a seleção de trabalhos científicos a serem analisados dentre os 144 documentos identificados, dando preferência aos artigos científicos. A escolha foi feita por meio da leitura dos resumos e/ou *abstracts*. Assim, foram elencados 10 (dez) artigos científicos que são apresentados no Quadro 2 de forma resumida, com destaque ao objetivo, metodologia e resultados alcançados.

**Avaliando Aspectos Motivacionais do Uso da Ferramenta Scratch para Ensino de Programação: Um Relato de Experiência (BELCHIOR; MEIRELES; FERREIRA, 2016)**

**Objetivo:** Verificar a viabilidade da utilização do Scratch como ferramenta auxiliar para o ensino de disciplinas iniciais de computação e, com isso, melhorar o processo de ensino-aprendizagem dos conceitos de lógica de programação.

**Metodologia:** No primeiro momento foi efetuado o planejamento da oficina, contendo tópicos de preparação, execução e avaliação. A oficina, com carga horária de 10h, foi realizada com 15 alunos do curso de graduação de Sistemas de Informação, utilizando a ferramenta Scratch para o ensino de lógica de programação. As aulas foram monitoradas pelo professor da disciplina. No fim, foi aplicado um questionário para que os alunos pudessem avaliar a ferramenta, considerando os aspectos de aprendizagem.

**Resultados:** Grande parte dos alunos correspondeu de forma favorável à experiência. Cerca de 80% dos alunos consideraram que o Scratch facilita a compreensão, reforça a teoria repassada em sala de aula e é apropriado para os estudos da disciplina. Eles relataram também que houve maior facilidade para compreender as situações que exigem o raciocínio lógico, além de elogiarem o visual da ferramenta.

**Um Curso de Programação a Distância com Metodologias Ativas e Análise de Aprendizagem por Métricas de Software (OLIVEIRA *et al.*, 2017)**

**Objetivo:** Favorecer a aprendizagem de programação por meio de um curso à distância com metodologia centrada no desenvolvimento da aprendizagem do aluno e com o uso de recursos tecnológicos para auxiliar a prática e a avaliação em uma perspectiva em

diversas dimensões.

**Metodologia:** Foi realizada a estruturação de atividades associadas a habilidades de programação, tais como: análise, compreensão, abstração, sequência lógica e construção. Atrelado a isso, houve a disponibilização de um fórum tira-dúvidas em que todos podiam assumir um papel de tutor para auxiliar os colegas na resolução de problemas, sendo bonificados pela qualidade da resposta. Por fim, foram utilizadas tecnologias para o mapeamento de perfis e realização de avaliações das dificuldades apresentadas pelos alunos.

**Resultados:** Os experimentos foram realizados com um grupo de 25 alunos, dentre estudantes de ensino médio até de pós-graduação. As atividades centradas nas habilidades e competências necessárias para a programação fizeram com que 78,3% dos alunos considerassem a estrutura do curso adequada para ambientes de educação à distância. 65,2% dos alunos consideraram a aplicação de teoria juntamente com exercícios práticos um elemento essencial para a aprendizagem. Os autores realizaram procedimentos de avaliação diagnóstica por meio de métricas de software para detectar problemas de aprendizagem entre alunos e possíveis plágios nas respostas encaminhadas, sendo uma técnica eficaz e um diferencial para ações pedagógicas personalizadas.

#### **LogicBlocks: Uma Ferramenta para o Ensino de Lógica de Programação (MARCON JÚNIOR; BONIATI, 2015)**

**Objetivo:** Desenvolver uma ferramenta web intuitiva, o LogicBlocks, com acesso facilitado para os interessados em aprender programação e que pudesse ser aplicada em sala de aula. Com isso, diminuir os fatores que desestimulam os alunos durante a fase de aprendizagem de programação, como erros de lógica ou de sintaxe.

**Metodologia:** Com base em outras metodologias alternativas, foi desenvolvida a ferramenta web para que os professores pudessem elaborar exercícios de programação e que os alunos pudessem praticar, por meio de um sistema de conexão de artefatos lógicos através de blocos visuais, e com isso construir o algoritmo, seja em computadores ou dispositivos móveis. Foi realizado um estudo de caso com alunos do ensino médio e do curso de graduação para avaliar o potencial da ferramenta. No fim, foi aplicado um questionário de avaliação da ferramenta.

**Resultados:** Durante o estudo de caso, os 35 alunos foram apresentados a exercícios já resolvidos e puderam resolver dois desafios propostos. 62% dos alunos consideraram a ferramenta intuitiva e 93% consideraram que elemento gráfico proporcionados pelos blocos potencializou o aprendizado. Por fim, 64% deles destacaram que a prática é um fator muito importante para o aprendizado de programação.

#### **Relato sobre o Uso de uma Ferramenta de Desenvolvimento de Jogos para o Ensino Introdutório de Lógica de Programação (COUTINHO; BONATES; MOREIRA, 2018)**

**Objetivo:** Relatar e discutir o uso de *Game Maker*, ferramenta de desenvolvimento de jogos digitais baseada na visão de blocos, para o ensino de conceitos de lógica de programação para alunos ingressantes em cursos de graduação. Assim, verificar o potencial motivador da ferramenta e a possibilidade de aplicar os conceitos aprendidos na prática.

**Metodologia:** Inicialmente, duas turmas de cursos superiores testaram a ferramenta



*Game Maker* durante as aulas para o desenvolvimento completo de jogos juntamente com os professores. Após a experiência, foi aplicado uma pesquisa de satisfação online não obrigatória com perguntas fechadas, avaliadas por meio da escala de Likert e também perguntas abertas, tanto para os alunos quanto para os professores.

Resultados: Ao final, somente 29 dos 56 alunos responderam à pesquisa. Os resultados apurados foram bem diversificados. Grande parte dos alunos achou que a ferramenta não é de fácil utilização e que não dá total suporte para a parte de codificação. Porém, a maioria concordou que a ferramenta traz motivação e auxilia bastante no entendimento dos conceitos de lógica de programação, porém consideraram que há necessidade de instrução durante seu uso. No ponto de vista dos professores, o uso da ferramenta trouxe alguns avanços significativos, porém seria necessário um tempo maior de experimentação para avaliar os reais impactos na aprendizagem dos alunos.

#### **Integração de uma Metodologia de Ensino Presencial de Programação com um Sistema Tutor Inteligente (GONZALEZ; TAMARIZ, 2014)**

Objetivo: O trabalho se propõe a integrar prática pedagógica de ensino de programação com recursos tecnológicos baseados em jogos computacionais para auxílio no ensino presencial. Desse modo, foi desenvolvido o HALYEN, um ambiente virtual que funciona com um tutor inteligente, que avalia e sugere uma prática pedagógica com base no perfil e na emoção de cada aluno.

Metodologia: Por meio do HALYEN, sistema baseado em agentes inteligentes, é oferecido um questionário diagnóstico para averiguar as motivações de aprendizagem. Depois, alunos interagem com jogos em nível crescente de dificuldade, sempre havendo um momento de mediação do professor na sala de aula fora do ambiente virtual, entre um nível e outro. Na terceira parte, é realizada a construção do algoritmo de forma colaborativa e mediada pelo professor. Durante a execução das atividades, o sistema vai inferindo o perfil do aluno e sua motivação com base no seu desempenho e no seu questionário. O próprio sistema sugere práticas pedagógicas para o professor, que possui um perfil personalizado, para assistir àquele aluno e também exibe mensagens motivadoras. O professor também pode propor atividades por meio de sua interface.

Resultados: O HALYEN foi aplicado em uma oficina de participação voluntária com 33 alunos de curso de graduação em Computação. Em relação ao perfil inicial, 57,14% dos alunos foram classificados como orientado ao desempenho e 37,14% como orientado à aprendizagem. Foi observado que alunos ficaram muito motivados com a utilização dos jogos. Para a maioria dos alunos de ambos os perfis, as estratégias pedagógicas sugeridas pelo sistema foram úteis para o aprendizado e para a resolução dos desafios. Os aspectos emocionais negativos, como tristeza ou frustração, se mantiveram baixos durante toda oficina e não impactou no desempenho dos alunos. As hipóteses conseguiram ser verificadas, tendo em vista que a emoção esperada foi manifestada pelos alunos em cada atividade.

#### **Lógica de Programação: Iniciação Lúdica com App Inventor for Android (GOMES; MELO, 2013)**

Objetivo: Com base em práticas metodológicas tecnológicas, o trabalho tem a proposta de ensino de lógica de programação por meio da criação de jogos de baixa complexidade para dispositivos móveis Android.

Metodologia: A utilização do App Inventor é base da proposta metodológica de integrar os paradigmas da programação com a criação de jogos, tudo de uma forma divertida e

motivadora. A abordagem é dividida em quatro módulos que abrangem desde os estágios iniciais de desenvolvimento até a publicação dos aplicativos na web.

Resultados: Com o fim do curso, os alunos puderam criar quatro jogos que abrangiam os tópicos de programação. O fato de não haver a preocupação com uma sintaxe facilitou a absorção do conceito de lógica por parte dos alunos. Para os mesmos, a interface visual merece destaque para o processo de aprendizagem, pela fácil usabilidade e por ser atraente e motivadora, tendo em vista as possibilidades de personalização e inovação, o que contribuiu para maior engajamento por parte dos alunos.

**O uso da técnica de narrativa OCC-RDD como apoio para elaboração e ou roteirização de conteúdo das aulas ministradas nas disciplinas dos cursos de computação (COELHO; VEGA, 2018)**

Objetivo: Estudar e aplicar a técnica de ensino OCC-RDD (Objetivo, Contratempo, Catástrofe, Reação, Dilema e Decisão), que se baseia em histórias narrativas como forma de contribuir para a redução das dificuldades de aprendizagem em disciplinas de cursos de computação e para aumento da motivação por parte dos alunos.

Metodologia: Consiste na aplicação da técnica OCC-RDD para a elaboração dos planos de aula para o ensino de disciplinas de Computação. Foi realizado um experimento com alunos de curso técnico com a disciplina de banco de dados. A partir do OCC-RDD, são criados cenários ou histórias com personagens que precisam solucionar algum problema que envolve os tópicos abordados na aula. As principais situações de cada cenário foram objetivo, contratempo e catástrofe. Com isso, os alunos deviam se pôr no lugar dos personagens e interpretar possíveis soluções para os desafios a partir das informações narradas. Os alunos também preencheram um questionário sobre os conceitos de aula antes depois da aplicação da técnica para efeitos comparativos de aprendizagem.

Resultados: O questionário diagnóstico, contendo 10 questões, foi preenchido por 38 alunos antes da aplicação do OCC-RDD. O número de questões respondidas incorretamente pela maioria dos alunos foi discrepante em relação ao número de corretas. Na sequência, a técnica OCC-RDD foi aplicada durante duas semanas durante as aulas e o mesmo questionário foi reaplicado após o período. Com isso, houve um aumento considerável de respostas corretas, com cerca de 50% dos alunos respondendo corretamente. Isso contribuiu de forma positiva para o processo de aprendizagem dos conteúdos ministrados.

**Logirunner: um Jogo de Tabuleiro como Ferramenta para o Auxílio do Ensino e Aprendizagem de Algoritmos e Lógica de Programação (CASAROTTO *et al.*, 2018)**

Objetivo: O trabalho propõe a utilização de um jogo de tabuleiro, o Logirunner, com cartas e peças físicas, para apoiar o ensino de algoritmos e lógica de programação. Seu principal foco é o papel motivador para com os alunos no processo de aprendizagem desses conceitos, de forma descontraída e divertida.

Metodologia: Após um levantamento bibliográfico sobre os conceitos e jogos aplicados ao ensino de algoritmos e lógica de programação, o jogo de tabuleiro Logirunner começou a ser construído. O foco estava relacionado na abordagem dos conceitos atrelada à fácil usabilidade e portabilidade, sem perder as características de um jogo de tabuleiro. Com o protótipo pronto, foi realizado um experimento com vinte alunos dos cursos de Ciência da Computação e de Sistemas de Informação. Foi utilizada a estratégia

MEEGA+ para a avaliação das estratégias empregadas por meio do jogo e um questionário para a avaliação da receptividade por parte dos alunos.

Resultados: O jogo foi utilizado durante dois dias, seguindo a estratégia MEEGA+ como forma de abordagem de aplicação. Por fim, os alunos preencheram um questionário sobre experiência do jogador, usabilidade, aspectos de confiança e satisfação. Mais de 70% das avaliações foram positivas em todos os quesitos. Interação, diversão e aprendizagem foram considerados os pontos fortes. Como sugestões estão a criação de novas cartas e desafios, além do acréscimo de tempo às jogadas. Por fim, destacou-se o engajamento dos alunos para o cumprimento das tarefas, o que permitiu considerar o experimento como satisfatório.

#### **Gamification technique for teaching programming (CARRENO-LEON *et al.*, 2018)**

Objetivo: Apresentar uma proposta metodológica baseada em *gamification*, elementos e técnicas de design de jogos, para o ensino de programação em disciplinas introdutórias em curso de graduação.

Metodologia: A técnica de gamificação foi utilizada, como suporte para aumentar a capacidade na resolução de problemas, em um curso básico de introdução à programação, por 14 alunos durante 4 semanas em 2 sessões de 2 horas cada, para resolver um total de 15 algoritmos. Como primeira atividade, a técnica de gamificação foi apresentada para os alunos. Três práticas foram inicialmente designadas, em nível crescente de dificuldade. No final da apresentação da técnica, os algoritmos que tinham que ser resolvidos em grupo usando os cartões que continham as estruturas de código. O progresso era quantificado pelo professor, como se os alunos estivessem em uma competição. No fim, os alunos responderam uma pesquisa sobre a experiência.

Resultados: Nos primeiros dois níveis, todos os grupos conseguiram resolver os algoritmos. No terceiro nível, o grupo 1 resolveu todos, o grupo 2 resolveu três, o grupo 3 não conseguiu nenhum e o grupo 4 resolveu dois. Somente o grupo 1 manteve o comportamento constante independente do nível de complexidade. Na pesquisa de satisfação, o grupo 1 aprovou a técnica com mais 90% de concordância, os grupos 2 e 4 com mais 80% e o grupo 3 por volta de 75%. Os autores consideraram que a técnica é válida para avaliar o comportamento humano durante o processo aprendizagem.

#### **Gradually Learning Programming Supported by a Growable Programming Language (CAZZOLA; OLIVARES, 2016)**

Objetivo: Face à dificuldade apresentada por muito estudantes para a aprendizagem de construtores e demais estruturas de programação, é proposta a aprendizagem gradual, onde um programa é escrito a partir de um modelo incremental onde aluno vai modificando a complexidade do código conforme aprende novas estruturas de linguagem de programação.

Metodologia: A partir de uma linguagem de programação escolhida, o professor modela um problema de forma que a solução possa ser incrementada gradualmente em nível de complexidade crescente. O modelo de ciclo de aprendizagem gradual é composto pelas fases de crescimento da linguagem, exposição de novos conceitos, prática dos alunos com incrementação do código. Um estudo de caso é realizado com alunos ingressantes na graduação para averiguar as potencialidades do método.

Resultados: Durante 14 semanas, os alunos, alguns já familiarizados com programação, foram expostos a poucos conceitos de programação de cada vez com o suporte de uma linguagem extensível cujo crescimento seguia a ritmo de ensino do professor. No fim, houve aplicação de um questionário anônimo. Os comentários foram positivos. Todos os alunos afirmaram que a abordagem aumentou o foco na resolução de problemas (50% dos alunos avaliaram como excelente, enquanto os outros 50% como muito bom, este último conjunto inclui todos os alunos já familiarizados com programação). Nesse sentido, uma estrutura de desenvolvimento que cresce com o conhecimento dos alunos ajuda a focar nos problemas e não na linguagem e os estudantes experientes são desafiados a resolver os problemas sem depender dos conhecimentos prévios e de uma perspectiva diferente.

**Quadro 02 – Resumos dos artigos científicos selecionados para análise e discussão.**

Fonte: Elaboração Própria

#### **4. ANÁLISE E DISCUSSÃO SOBRE OS TRABALHOS SELECIONADOS**

Os 10 (dez) artigos científicos selecionados abrangem uma diversidade de propostas pedagógicas para o ensino de lógica de programação ou algoritmos, seja com a aplicação de jogos, recursos tecnológicos ou estratégias de ensino. A análise e a discussão dos trabalhos são feitas de acordo com o recurso metodológico utilizado, a saber: software e aplicativos educacionais, desenvolvimento e aplicativo de jogos e métodos de ensino.

##### **➤ Software e Ferramentas Educacionais**

Os recursos tecnológicos baseados em software e aplicativos educacionais se mostraram ótimas opções pedagógicas para o ensino devido às facilidades proporcionadas para o desenvolvimento de atividades com avaliação em tempo real, o que permite um *feedback* imediato da aprendizagem do aluno. Além disso, interfaces amigáveis e intuitivas podem atuar como incentivo à execução das atividades propostas.

As ferramentas Scratch, apresentado por Belchior *et al.* (2016), e LogicBlocks, apresentado por Marcon Junior e Boniati (2015), auxiliam na aprendizagem com o uso de blocos de programação visuais. Essa característica permite que os alunos foquem na lógica do algoritmo, sem se preocupar com a sintaxe de linguagem, um fator essencial para a construção de um programa. Por promover exercícios práticos, as duas ferramentas foram bem recebidas pelos alunos, obtendo resultados satisfatórios em seus experimentos.

Considerando o suporte para avaliações, o sistema HALYEN de Gonzalez e Tamariz (2014) pode ser destacado por permitir uma análise completa do perfil e do sentimento do aluno de forma automatizada e em tempo real. A iniciativa proposta de HALYEN atuar como suporte à avaliação diagnóstica concede ao professor a autonomia para traçar uma

estratégia em resposta ao desempenho do aluno durante o desenvolvimento da atividade proposta de forma personalizada, trazendo mais dinamismo para a sala de aula.

Assim, software e ferramentas educacionais são recursos tecnológicos cada mais indispensáveis ao professor devido à flexibilidade, dinamismo e atratividade proporcionados para a prática na resolução dos problemas de lógica e programação.

### ➤ **Desenvolvimento e Aplicativos de Jogos**

Os jogos, principalmente os digitais, são elementos de alta atratividade para os jovens. A sua aplicação em aulas já capta bastante a atenção dos alunos, pois permite aliar a diversão aos desafios, impulsionando a superação por parte dos estudantes. Além disso, os jogos exigem bastante atenção, percepção e alto processamento cognitivo o que auxilia no pensamento lógico mais ágil.

Esses aspectos podem ser verificados nas experiências que apostam na Gamificação, como artigos de Coutinho *et al.* (2018), Gomes e Melo (2013), Casarotto *et al.* (2018) e Carreno-Leon *et al.* (2018). As propostas de construção de jogos, sejam digitais ou físicos no caso de Casarotto *et al.* (2018), exigem que o aluno trabalhe seu raciocínio com base na sequência lógica necessária para que o jogo funcione ao final de seu desenvolvimento. Além disso, as estratégias também são um elemento-chave para o desenvolvimento cognitivo no campo de lógica e de tomada de decisão, situação semelhante à codificação de programação.

Dessa forma, os alunos trabalham as competências e habilidades necessárias, de forma direta ou indireta, para a construção de algoritmos para resolução de problemas nas aulas num contexto lúdico, divertido e atrativo. Isso faz com que os jogos sejam uma forte opção para motivar os estudantes durante às disciplinas.

### ➤ **Métodos de Ensino**

As diferentes formas de construção de conhecimento exigem que professor seja flexível, dinâmico e que aposte em métodos de ensino que fujam dos padrões convencionais e mais tradicionais. Oliveira *et al.* (2017) defende que uma aprendizagem colaborativa por meio de fóruns de discussão incentivados podem ser uma boa estratégia para o ensino à distância. Isso estimula a pesquisa e a participação dos alunos na resolução dos problemas.

Coelho e Veja (2018) apresentam a técnica de narrativa de histórias chamada OCC-RDD para auxiliar na resolução de problemas de lógica. Essa abordagem fora dos métodos

tradicionais permite expor os conceitos em uma outra forma de comunicação. Com isso, os alunos podem ter contato com termos técnicos através de uma visão diferenciada e mais contextualizada, além de poderem propor as soluções ativamente se pondo no lugar das personagens.

A aprendizagem gradual de Cazzola e Olivares (2016) se destaca por utilizar um princípio intrínseco da área de desenvolvimento de sistemas para o ensino dos próprios algoritmos. Ao implementar modificações evolutivas no código a medida que novos conceitos são aprendidos pelos alunos, instaura uma perspectiva real do que é feito na prática. Além de facilitar a aprendizagem em estágios, já familiariza os alunos indiretamente com o que acontece no cotidiano dos desenvolvedores.

O fator em comum presente nos trabalhos é a preocupação no sentido motivacional para com o aluno, estimulando-o a sempre continuar superando os obstáculos no processo de aprendizagem, a partir do incentivo às atividades práticas durante todo o percurso, e tornando-o o protagonista na construção do conhecimento. Os estágios de aprendizagem explicitados pela Taxonomia de Bloom podem ser verificados em diversos aspectos nos métodos elencados.

Os métodos de ensino e aplicativos educacionais oferecem amplo suporte para os estágios de conhecimento e compreensão, pois tem o enfoque na percepção e entendimento do conteúdo. Nesse mesmo sentido, os jogos facilitam a aplicação, análise e síntese do que é ensinado por meio da efetiva práxis que, por fim, complementa a etapa de avaliação, que é contínua nas experiências relatadas. Isso pode configurar o conjunto de fatores que contribuíram para as experiências positivas e satisfatórias, tanto para os pesquisadores quanto para os alunos, nos estudos de casos efetuados.

## **5. CONSIDERAÇÕES FINAIS**

A presença das disciplinas de lógica de programação e de algoritmos é bem comum em cursos de graduação da área de Ciências Exatas, principalmente nas áreas de Computação e Engenharia. Muitos alunos costumam apresentar dificuldades na compreensão e na aplicação dos conceitos relativos devido à exigência de habilidades e competências prévias bem desenvolvidas. Isso pode acarretar em problemas de motivação, reprovações e até em evasão, dificultando a continuidade dos alunos nos referidos cursos.

Este trabalho se propôs a investigar metodologias ou estratégias de ensino para tais disciplinas com objetivo de facilitar o processo de ensino-aprendizagem e de dirimir os impasses mencionados. Foram selecionados 10 (dez) artigos científicos que trouxeram abordagens pedagógica inovadoras baseadas em software, aplicativos educacionais, jogos, entre outras estratégias, com aplicações bem-sucedidas.

As metodologias elencadas permitem aos professores traçarem a estratégia mais adequada para o ensino de lógica de programação e algoritmos, sendo que um método ou ferramenta pode ser aplicada de forma isolada ou complementar, conforme o aspecto motivacional e o processo de aprendizagem de cada aluno. Com isso, espera-se contribuir para o incentivo aos professores, para que possam difundir e aplicar cada vez mais diferentes práticas pedagógicas nessa área no Ensino Superior, seja com inspiração nos casos explicitados neste trabalho ou em abordagens inovadoras, face aos desafios relativos ao ensino e a ampla discussão no contexto educacional dessas disciplinas.

## REFERÊNCIAS

BELCHIOR, J. H. F.; MEIRELES, M. A. C.; FERREIRA, R. DOS S. Avaliando Aspectos Motivacionais do Uso da Ferramenta Scratch para Ensino de Programação: Um Relato de Experiência. **Nuevas Ideas en Informática Educativa**, v. 12, p. 618–623, 2016.

CARRENO-LEON, M. *et al.* **Gamification technique for teaching programming**. 2018 IEEE Global Engineering Education Conference (EDUCON). **Anais...** In: 2018 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON). Tenerife: IEEE, abr. 2018. Disponível em: <<https://ieeexplore.ieee.org/document/8363482/>>. Acesso em: 16 jan. 2019

CASAROTTO, R. I. *et al.* Logirunner: um Jogo de Tabuleiro como Ferramenta para o Auxílio do Ensino e Aprendizagem de Algoritmos e Lógica de Programação. **RENOTE**, v. 16, n. 1, 21 ago. 2018.

CAZZOLA, W.; OLIVARES, D. M. Gradually Learning Programming Supported by a Growable Programming Language. **IEEE Transactions on Emerging Topics in Computing**, v. 4, n. 3, p. 404–415, jul. 2016.

COELHO, U. M.; VEGA, Í. S. **O USO DA TÉCNICA DE NARRATIVA OCC-RDD COMO APOIO PARA ELABORAÇÃO E OU ROTEIRIZAÇÃO DE CONTEÚDO DAS AULAS MINISTRADAS NAS DISCIPLINAS DOS CURSOS DE COMPUTAÇÃO**. CONGRESSO INTERNACIONAL DE EDUCAÇÃO E TECNOLOGIAS. São Carlos: 2018

COUTINHO, E.; BONATES, M.; MOREIRA, L. O. **Relato sobre o Uso de uma Ferramenta de Desenvolvimento de Jogos para o Ensino Introdutório de Lógica de Programação**. VII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. Fortaleza, Ceará,

Brasil: 28 out. 2018Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/8291>>. Acesso em: 16 jan. 2019

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de Programação: a construção de algoritmos e estruturas de dados**. 3ª ed. São Paulo: Prentice Hall, 2005.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. Edição: 6ª ed. São Paulo: Atlas, 2017.

GOMES, T. C. S.; MELO, J. C. B. DE. **Lógica de Programação: Iniciação Lúdica com App Inventor for Android**. p. 7, 2013.

GONZALEZ, S. M.; TAMARIZ, A. D. R. INTEGRAÇÃO DE UMA METODOLOGIA DE ENSINO PRESENCIAL DE PROGRAMAÇÃO COM UM SISTEMA TUTOR INTELIGENTE. **Revista Brasileira de Informática na Educação**, v. 22, n. 02, p. 16, 23 nov. 2014.

GOOGLE. **About Google Scholar**. Disponível em: <<https://scholar.google.com.br/intl/pt-BR/scholar/about.html>>. Acesso em: 8 jan. 2019.

JESUS, A. DE; BRITO, G. S. CONCEPÇÃO DE ENSINO-APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES: A PRÁTICA DOCENTE. **Revista Varia Scientia**, v. 09, n. 16, p. 149–158, 2009.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. DE. **Algoritmos. Lógica Para Desenvolvimento de Programação de Computadores**. 28ª ed. São Paulo: Érica, 2016.

MARCON JÚNIOR, R. P.; BONIATI, B. B. **LogicBlocks: Uma Ferramenta para o Ensino de Lógica de Programação**. Anais do EATI - ENCONTRO ANUAL DE TECNOLOGIA DA INFORMAÇÃO E SEMANA ACADÊMICA DE TECNOLOGIA DA INFORMAÇÃO. Frederico Westphalen/RS: nov. 2015

OLIVEIRA, M. G. DE *et al.* Um Curso de Programação a Distância com Metodologias Ativas e Análise de Aprendizagem por Métricas de Software. **RENOTE**, v. 15, n. 1, 28 jul. 2017.

SILVA, T. R. DA *et al.* Ensino-aprendizagem de programação: uma revisão sistemática da literatura. **Revista Brasileira de Informática na Educação**, v. 23, n. 01, p. 182, 30 abr. 2015.