

A Teoria de Aprendizagem Significativa no Ensino de Programação: um Mapeamento Sistemático da Literatura

Humberto A. P. Zanetti¹, Marcos A. F. Borges¹, Ivan L. M. Ricarte¹

¹Faculdade de Tecnologia – Universidade Estadual de Campinas (UNICAMP) – Limeira – SP – Brasil

h016304@dac.unicamp.br, {marcosborges, ricarte}@ft.unicamp.br

Abstract. *This paper aims to discuss the theory, applications and main didactic strategies for teaching programming based on the theory of Meaningful Learning, developed by David Ausubel. Research on this topic is recent and there are few reports of application of this theory in the context of teaching programming and even less research studying the convergence of the results of these reports. This lack of references and examples may generate insecurity or doubts in the teacher or researcher who is looking for practices based on Meaningful Learning in the context of teaching programming. In this scoping review, 26 articles published since 2010 were evaluated, with the aim of presenting a critical view of the most recent research. With this study it was possible to map the main strategies in which it was tested and the target audiences teaching premises of Meaningful Learning in programming learning.*

Resumo. *Esta pesquisa tem como objetivo discutir as aplicações e as estratégias didáticas para ensino de programação baseada na teoria de Aprendizagem Significativa, desenvolvida por David Ausubel. Pesquisas sobre esse tema são recentes e há poucos relatos de aplicação da teoria no contexto de ensino de programação, com ainda menos pesquisas estudando a convergência do resultado desses relatos. Essa falta de referências e exemplos pode gerar insegurança ou dúvidas no professor e pesquisador que busca práticas baseadas na Aprendizagem Significativa no contexto de ensino de Programação. Neste mapeamento sistemático da literatura, foram avaliados 26 artigos publicados desde 2010, com o objetivo de apresentar uma visão crítica sobre as pesquisas mais recentes. Com este estudo foi possível mapear as principais estratégias adotadas e os públicos-alvo em pesquisas que seguem premissas da Aprendizagem Significativa no ensino de programação.*

1. Introdução

As disciplinas de “Programação de Computadores” e “Algoritmos” estão presentes em todos os cursos de formação técnica e superior na área de Informática e Computação. Elas trabalham conhecimentos fundamentais e essenciais para o desenvolvimento acadêmico dos alunos durante o decorrer de formações na área. As disciplinas introdutórias de programação apresentam um primeiro obstáculo aos iniciantes, devido ao seu caráter técnico e exigência de um alto nível de abstração.

O desenvolvimento de algoritmos e a prática de programação são conhecimentos difíceis de ser ensinados, exigindo uma grande dedicação do aluno e abordagens

adequadas para sua aprendizagem. Muitas aplicações dessas disciplinas focam em ensinar questões técnicas da linguagem de programação sem focar em práticas para a resolução de problemas [Gomes e Mendes 2007]. Uma das grandes dificuldades dos iniciantes em programação é correlacionar conceitos anteriormente aprendidos e sedimentados com os novos apresentados, de modo a serem utilizados em conjunto nas soluções. Moström *et al.* (2008) destacam o problema em determinar quais recursos são necessários e qual a sequência de utilização deles para a composição de uma solução algorítmica. Gomes e Mendes (2007) enfatizam que os alunos têm dificuldades em estabelecer relações entre problemas anteriormente vivenciados e conhecimentos prévios com os novos conhecimentos e suas aplicações em novos problemas.

Trabalhos como o de Piteira e Costa (2013) discutem a dificuldade de alunos de cursos de computação em entender certas estruturas de dados e suas aplicações na resolução de problemas reais. A ausência de um elo que conecte o recurso apresentado ao contexto do problema a ser resolvido pode levar à desmotivação do aluno. Várias pesquisas, como de Möstrom *et al.* (2008), Gasaymeh *et al.* (2016) e Almeida *et al.* (2017), mostram que estratégias de ensino que promovam a aplicação de conceitos de forma mais concreta e significativa podem auxiliar no desenvolvimento da habilidade de abstração e na associação de conceitos para construção de uma solução.

Ausubel (2012) propôs a teoria da Aprendizagem Significativa (AS), na qual defende que a aprendizagem se torna mais significativa à medida que um novo conhecimento ou conteúdo é agregado às estruturas de conhecimento previamente existentes em um aprendiz, incorporando mais significado à sua estrutura cognitiva. Essa teoria questiona o aprendizado “mecânico” ou repetitivo, no qual um novo conhecimento é isolado ou tem associações arbitrárias com os anteriores.

Pelizzari *et al.* (2002) ressaltam o impacto positivo que a AS proporciona na construção e enriquecimento da estrutura cognitiva do aprendiz, do ponto de vista dos conceitos anteriormente apresentados e sua utilização na aquisição de novos conceitos e suas conexões. Esse processo de correlação e integração de conceitos, por meio da identificação de suas similaridades e diferenças, é fundamental para a ocorrência de uma aprendizagem mais significativa. Na AS, essas etapas são chamadas de *diferenciação progressiva*, apresentando inicialmente conceitos mais gerais e evoluindo, progressivamente, para mais específicos, e de *reconciliação integrativa*, que buscando correlacionar os conceitos apresentados [Moreira e Mansini 2002].

A AS pode ser base para métodos e estratégias no contexto do ensino de programação, com um objetivo claro de promover meios de aprendizagem contextualizados, progressivos e com associações dos conceitos que são aplicados em qualquer linguagem de programação. A preocupação por buscar meios de aprendizagem de programação que possam ser mais significativos, compreensíveis e contextualizados não é recente. Shneiderman (1977) explorou a temática, associando a AS para discutir abordagens didáticas que pudessem auxiliar na compreensão da sintaxe e semântica das linguagens de programação.

Este artigo tem o objetivo de realizar um mapeamento sistemático da literatura e analisar trabalhos que exploram estratégias de ensino que se baseiem nas premissas da AS. Essas pesquisas são direcionadas para o ensino-aprendizagem de programação de computadores, lógica de programação e algoritmos, discutindo os objetivos desses trabalhos e potenciais contribuições que eles podem trazer para a área. O mapeamento

tem o objetivo de identificar trabalhos publicados que possam atestar a eficácia e resultados da aplicação dessas práticas baseadas em AS, e proporcionar meios definir quais estratégias são aplicadas. Por meio desta análise, objetiva-se também encontrar lacunas de pesquisas e demandas para criação de um método que siga diretrizes da AS.

O texto está organizado da seguinte forma: a Seção 2 apresenta o contexto e objetivos que conduziram esse mapeamento sistemático; a Seção 3 mostra como foi feito o processo de busca e seleção dos trabalhos analisados, assim a categorização dos mesmos; a Seção 4 apresenta resultados e discussões sobre os trabalhos e questões de pesquisa, assim como apontamentos de lacunas para novas pesquisas; e, por fim, na Seção 5, são feitas as considerações finais.

2. Contexto e objetivo do mapeamento

Atualmente, muitos pesquisadores e educadores procuram novas maneiras de tornar o ensino de programação mais dinâmico, motivador e significativo ao aluno [Raposo e Dantas 2016]. Transmitir um novo conhecimento requer planejamento, usando como base o que já foi apresentado (e aprendido), buscando consolidar o conhecimento, tornando-o mais atraente e significativo ao aluno. Segundo Moreira (2012), o papel do educador é proporcionar, de maneira gradativa e progressiva, recursos para que o aluno possa construir sua estrutura cognitiva, organizando e relacionando o conhecimento previamente adquirido com os novos, que estão em processo de aquisição.

O mapeamento sistemático, segundo Dermeval, Coelho & Bittencourt (2019), tem como objetivo responder questões de pesquisa sem grande aprofundamento, apenas para que se tenha uma visão geral sobre o tema. A escolha por esta forma de revisão, por meio de mapeamento, se deu pela necessidade de categorizar trabalhos de interesse para trabalhos futuros que possam colaborar com a temática envolvida. Este mapeamento tem como objetivo principal analisar trabalhos que possam apontar quais são as práticas encontradas na literatura que seguem como premissa diretrizes da AS para o ensino de programação, assim como quais são os públicos-alvo atingidos. O resultado dessa análise visa contribuir com futuros trabalhos que possam desenvolver métodos de ensino em programação que possam se apoiar nos pilares da AS, promovendo meios de auxiliar a aquisição de conhecimento por parte do aluno.

2.1. As questões de pesquisa

A definição das questões de pesquisa (QP), que delimitam essa pesquisa, foi baseada em dois aspectos que são importantes em práticas de ensino de programação: o público-alvo a ser atendido e quais estratégias são adotadas. Esses aspectos foram utilizados para categorizar os trabalhos e com isso, conduzir as análises e discussões. Para este mapeamento, duas questões foram formuladas:

- **QP1:** *Qual é o público-alvo presente em pesquisas que utilizam práticas de ensino de programação baseado em Aprendizagem Significativa?*
- **QP2:** *Quais estratégias são utilizadas em práticas de ensino de programação que promovam a Aprendizagem Significativa?*

3. Métodos aplicados na Revisão Sistemática da Literatura

Para esta RSL, foi utilizado o protocolo PRISMA (*Preferred Reporting Items for Systematic reviews and Meta-Analyses*), uma metodologia que estabelece procedimentos, como a lista de itens que devem ser considerados para a revisão (PRISMA *checklist*) e o fluxo de exclusão e inclusão de artigos (PRISMA *flow diagram*) [Page et al. 2021].

Para a construção da *string* de busca, foram escolhidos três conceitos principais para definição dos termos, que são: a própria AS e a inclusão do termo “teoria”, para tirar a desambiguação com o termo geral “aprendizagem significativa”; a inclusão do nome do criador da teoria, também por critérios de desambiguação e, por fim, o conceito de ensino de programação. A *string* de busca padrão foi inicialmente definida a partir dos principais termos em inglês e seus sinônimos, sendo definida na seguinte forma: (*"meaningful learning" OR "meaningful learning theory") AND ("david ausubel" OR "ausubel") AND ("introductory programming education" OR "introductory programming" OR "teach* programming" OR "learn* programming" OR "computer programming" OR "novice programming" OR "coding education" OR "coding" OR "software engineering education" OR "cs1" OR "computer science education" OR "introductory computer science"*). Essa *string* também foi adaptada para as línguas portuguesa e espanhola.

Foram escolhidas: a base de dados *Scopus*, a base de pesquisa Periódicos CAPES e o motor de busca Google Acadêmico, para os mecanismos de consulta e busca por trabalhos. A busca foi feita por publicações a partir de 2010, buscando uma análise mais contemporânea do cenário envolvendo a pesquisa. A Tabela 1 apresenta a quantidade de trabalhos por etapa e o total selecionado para a análise.

Tabela 1. Resumo do resultado da busca pelos trabalhos

Bases de Dados	Qtd. de trabalhos	Após 1ª triagem	Após 2ª triagem
Periódicos CAPES	136	26	12
Google Acadêmico	114	48	21
Scopus	204	29	10
Total c/ trabalhos duplicados			43
Total s/ trabalhos duplicados			26

Os critérios de inclusão e exclusão estão apresentados resumidamente na Tabela 2. A prioridade foi a análise de trabalhos que abordassem práticas didáticas que envolvam programação como objetivo principal ou nas quais a programação fosse de suma importância para a execução da atividade. Toda essa fase de inclusão e exclusão ocorreu em duas etapas: 1) com a primeira triagem, aplicando o filtro por data (aplicando o critério I1) e a seleção de artigos completos (aplicando o critério I2); 2) com a segunda triagem, com a leitura primeiramente do título, palavras-chave e resumo (aplicando os critérios E1, E2 para exclusão) e, posteriormente, o texto do artigo (aplicando os critérios I3 e I4 de inclusão e critérios E3, E4 e E5 de exclusão). Também foi realizada uma filtragem por trabalhos em que o termo “aprendizagem significativa” estivesse relacionado à teoria desenvolvida por David Ausubel, como um critério de desambiguação, sendo que os alguns trabalhos em que o termo não fazia referência à teoria da AS, foram descartados (que foi a base para definir o critério de exclusão E3).

Tabela 2. Critérios de inclusão e exclusão

Critérios de Inclusão	Critérios de Exclusão
I1. Publicados a partir do ano de 2010.	E1. Apresentem apenas revisão da literatura.
I2. Trabalhos completos	E2. Sem fonte de publicação
I3. Abordem práticas pedagógicas e/ou proponham alguma metodologia e/ou ferramenta didática e/ou avaliem instrumentos didáticos.	E3. Usem o termo “aprendizagem significativa”, mas sem relação com a teoria de Ausubel.
I4. Abordem os temas ensino de programação e AS.	E4. Não contemplem a questão de pesquisa
	E5. Abordem práticas de Computação sem programação

Após a seleção dos trabalhos e leitura dos textos, foi realizado o processo de classificação, extração e mapeamento dos dados. Em uma planilha, foram organizados os trabalhos primeiramente por público-alvo. Após isso, foram anotados e relacionados quais eram as práticas e ferramentas descritas em cada um dos trabalhos. Com essas anotações, foi possível a incidência comum entre os artigos e definidos os agrupamentos finais para a análise.

3.1. Análise temática dos trabalhos

Com o objetivo de investigar tendências e lacunas dentro da temática de aplicação de AS no ensino de programação, foi realizada uma análise temática com o objetivo de identificar temas e abordagens na área. Segundo Souza (2019), a análise temática utiliza uma estratégia indutiva e baseada nas informações presentes nos textos, levando o pesquisador a descobrir padrões semelhantes e tendências entre os trabalhos analisados, podendo criar classificações. Nesta análise, foram utilizadas duas temáticas, que são convergentes nos trabalhos: a do público-alvo e a de estratégias de ensino.

Em uma classificação por seu público-alvo, foi possível notar que há trabalhos para *Ensino Superior* (17 trabalhos, 65,4%), *Ensino Médio* (4 trabalhos – 15,4%), *Ensino Fundamental* (3 trabalhos – 11,5%) e exclusivamente para o *Ensino Técnico* (2 trabalhos – 7,7%). A identificação de um público-alvo ajuda a entender qual é o objetivo da prática no trabalho analisado. Trabalhos que focalizam o ensino superior e técnico possuem um perfil de formação profissional, enquanto nos trabalhos dos demais grupos a aprendizagem de programação tem outros objetivos.

Para esse mapeamento, é considerada estratégia de ensino qualquer aplicação de algum método ou prática de ensino, ou adição de alguma ferramenta ou recurso de apoio para o educador. Dentre os estudos selecionados, foram identificadas algumas estratégias de ensino, sendo que em alguns trabalhos é possível notar mais do que uma estratégia utilizada. Pode-se observar que a maioria dos trabalhos utilizaram *Mapas Conceituais* (7 trabalhos - 26,9%), um recurso utilizado para representação de conhecimento e associação de conceitos. Outra estratégia que é associada ao relacionamento de conceitos foi a *Apresentação gradativa de conceitos* (4 trabalhos – 15,4%), sem adotar uma ferramenta ou outra estratégia de ensino específica. A apresentação gradativa do conhecimento é uma das premissas da AS, mas, para análise feita neste artigo, foi considerado como uma estratégia em particular, pois alguns trabalhos deixavam clara sua adoção. Na AS, a apresentação gradativa de conceitos está baseada principalmente nos princípios de diferenciação progressiva e a reconciliação integradora [Moreira, 2012]. *Recursos Multimídias* e *Novas Linguagens de Programação* são utilizados em 3 trabalhos cada (11,5%). Recursos como *Jogos Digitais*, *Robótica Pedagógica*, *Aprendizado Baseado em Problemas (PBL - Problem-Based Learning)* e *Scratch*, comuns em práticas didáticas, aparecem em apenas 2 trabalhos cada uma (7,7%). Apenas um trabalho (3,8%)

aplica *Metodologias Ativas*. A Tabela 3 organiza os trabalhos separados por estratégias de ensino aplicadas na pesquisa e os respectivo público-alvo.

Tabela 3. Distribuição dos trabalhos por estratégia de ensino e público-alvo

	Fundamental	Médio	Técnico	Superior
<i>Apresentação gradativa</i>	-	-	-	[Berssanette e Francisco 2018], [Buriticá 2011], [Buriticá 2013], [Buriticá 2014]
<i>Jogos Digitais</i>	-	-	-	[Barbosa, Fernandes e Campos 2011], [Hernandez <i>et al.</i> 2010]
<i>Novas Linguagens</i>	[Wang, Mendori e Xiong 2014]	[Alexandron <i>et al.</i> 2017], [Silva <i>et al.</i> 2020]	-	-
<i>Mapas Conceituais</i>	-	-	-	[Bishop <i>et al.</i> 2017], [Brito, Rodriguez e Pellicer 2016], [Menon e Kovalchick 2020], [Minakshi e Sona 2018], [Mühling 2016], [Sharma e Chawla 2019], [Soto <i>et al.</i> 2013]
<i>Metodologias Ativas</i>	-	-	-	[Shitsuka <i>et al.</i> 2018]
<i>PBL</i>	-	-	-	[Aires <i>et al.</i> 2021], [Lima, Diniz e Eliasquevic 2014]
<i>Recursos Multimídia</i>	-	-	[Astolfi e Lopes Junior 2015], [Astolfi e Lopes Junior 2016]	[Matthews, Hin e Choo 2014]
<i>Robótica Pedagógica</i>	[Kaloti-Hallak, Armoni e Ben-Ari 2019]	[Yepes, Barone e Porciuncula 2021]	-	-
<i>Scratch</i>	[Suárez 2015]	[Gómez-Zermeño e Mejia 2020]	-	-

A presença majoritária de trabalhos em nível de escolaridade profissionalizante (superior e técnico) aponta uma preocupação de pesquisadores em apresentar conceitos de programação nessa etapa de formação. Em nível superior, há poucos trabalhos que abordam conceitos de Programação Orientada a Objetos (POO) e/ou conceitos mais avançados, como Estrutura de Dados e Programação Funcional, sendo apenas 6 estudos (Bishop *et al.* (2017); Brito, Rodriguez e Pellicer (2016); Buriticá (2013); Buriticá, (2014); Hernandez *et al.* (2010); e Müling (2016)). Os demais trabalhos destinados ao ensino superior abordavam conceitos fundamentais de programação. Os 2 trabalhos para o público de formação técnica envolviam POO, e todos os trabalhos de nível médio e fundamental abordavam os conceitos básicos de Algoritmos e Programação Estruturada.

4. Resultados obtidos e discussões

Nesta seção são discutidos os resultados obtidos na análise dos trabalhos que compõem essa RSL. A Seção 4.1 discute sobre o público-alvo; na Seção 4.2 são apresentadas discussões sobre as estratégias adotadas; e, por fim, na Seção 4.3 há uma discussão mais

abrangente em torno do tema e o apontamento de lacunas que podem ser base para novas pesquisas.

4.1. Os públicos-alvo presentes nos trabalhos (QP1)

A partir das análises apresentadas na seção 3.1 observa-se a maior presença de trabalhos no Ensino Superior. Nesse nível de formação, a apresentação de conceitos de programação é mais aprofundada e a necessidade do aluno em ter base de conhecimentos prévios para ir adiante nos estudos é mandatório. O paradigma de POO e as aplicações de Estruturas de Dados, por exemplo, exigem um domínio nas noções básicas de programação. Mas a identificação de poucos trabalhos que abordem conceitos avançados de programação pode indicar que ainda há demanda de pesquisa de aplicação da AS no ensino formal de programação.

A maior presença de trabalhos dedicados ao ensino nos níveis de formação profissional (Superior e Técnico) apontam uma preocupação por novas estratégias de ensino nesses níveis de formação. A falta de engajamento, dificuldades em abstrair e aprender conceitos alunos em cursos de formação em Computação é presente, como apontam Souza, Batista e Barbosa (2016). Com essa análise é possível identificar que há uma demanda de pesquisa em que o objetivo seja atingir a formação profissional. Em cursos formais de programação, é comum a alta taxa de evasão, principalmente em disciplinas iniciais. Também são notáveis fatores como desmotivação e falta de engajamento em alunos iniciantes, sendo uma questão relevante a ser resolvida pela academia [Hoed 2016].

A formação profissional em Computação em nível superior, presente na maior parte dos trabalhos analisados, possui o desafio de motivar e engajar um público adulto. Várias pesquisas apresentam soluções dinâmicas e lúdicas para o público mais jovens (crianças e adolescentes), que não se enquadram em um público mais maduro [Souza, Moretti & Podestá 2008; Medeiros, da Silva & Aranha 2013]. A base fundamental da AS, que é a correlação de conceitos aplicados por meio de um instrumento que auxilie sua visualização, como Mapas Conceituais (presente nos trabalhos), pode trazer um recurso que ampare essa percepção do aluno. Mas falta ainda a associação de um processo instrucional que seja mais efetivo na associação das diretrizes da AS, como mostram alguns trabalhos analisados, que definem uma apresentação gradativa dos conceitos, mas não se alinham a nenhum método conhecido na academia. Poucos trabalhos trazem esse alinhamento, como aqueles que usam PBL e Metodologias Ativas.

4.2. As estratégias adotadas (QP2)

Sobre as estratégias adotadas nos trabalhos, a utilização de Mapas Conceituais teve a maior incidência. Esse recurso é comumente utilizado em trabalhos que abordam AS, tanto como ferramenta de ensino (apresentação dos conceitos e suas relações), como também artifício de discussão sobre o resultado da aprendizagem. Menon e Kovlachick (2020), Mühling (2016), Sharma e Chawla (2019) e Soto *et al.* (2013) apresentam explicitamente Mapas Conceituais que representam e correlacionam conceitos de programação de forma hierárquica para os alunos, criando um modelo do conhecimento que se espera ser adquirido.

Os trabalhos enquadrados na *Apresentação gradativa de conceitos* são pesquisas que mostram a intenção do pesquisador-educador em apresentar conceitos de

programação de forma gradativa, para auxiliar a construção de um modelo mental em que os conceitos iniciais se conectem e façam sentido para o conceito seguinte. Os trabalhos desse grupo, mesmo deixando clara a intenção de apresentação gradativa, não apresentam de maneira explícita a utilização de uma ferramenta ou estratégia encontrada na literatura para o auxílio à tarefa. Em alguns casos, como em Buritica (2013), os artigos descrevem sua prática didática como uma metodologia nova.

Entre as pesquisas que apresentam Novas Linguagens de Programação, destaca-se o de Silva *et al.* (2020), que apresenta a linguagem e ambiente de programação Calango, baseada em comandos em português, e o trabalho de Wang, Mendori e Xiong (2014), que desenvolvem uma linguagem de programação visual baseada em Mapas Conceituais. Em todos os trabalhos classificados desta forma, é possível identificar a demanda por criar ferramentas que possam ser mais lúdicas e que necessitam de menor grau de abstração na utilização de conceitos de programação.

As pesquisas que envolvem Jogos Digitais e Robótica Pedagógica também apresentam propostas de práticas com o objetivo de engajar os alunos de ensino superior e mostrar conceitos de programação em meios mais concretos. Os trabalhos envolvendo robótica tinham como público alunos de cursos de engenharia. A pesquisa de Barbosa, Fernandes e Campos (2011) propõe uma metodologia que mostra conceitos de programação através de um jogo no formato de desafios (*puzzles*). Já o trabalho de Hernandez *et al.* (2010) utiliza uma ferramenta de desenvolvimento de jogos digitais (*game engine*) para ensinar conceitos de POO.

Apenas três trabalhos alinhavam metodologias de ensino com AS. Em Aires *et al.* (2021) são discutidas práticas de desenvolvimento de aplicativos móveis baseadas em PBL. Em Lima, Diniz e Eliasquevici (2019) é proposta uma metodologia voltada ao ensino de Algoritmos usando resolução de problemas. Shitsuka *et al.* (2018) utilizam uma metodologia ativa em uma turma de Engenharia, devido à demanda dos alunos por mais dinamismo nas aulas de programação.

Em todos os trabalhos que utilizavam *Recursos Multimídias*, o principal objetivo era a associação de conceitos de maneira menos abstrata, usando recursos visuais, como vídeos e imagens. Os trabalhos com *Scratch* envolviam públicos de formação não profissional, tendo o ensino de programação como um recurso para criação de outras atividades didáticas.

Observa-se que há poucos trabalhos que utilizam estratégias que possam trazer a programação para um ambiente que ampare o exercício de abstração, como é necessário em POO, por exemplo. A utilização de Robótica Pedagógica, presente em apenas dois trabalhos, pode ser um recurso lúdico, interativa e multidisciplinar [Santos *et al.* 2019]. Os poucos trabalhos que abordam o recurso são voltados apenas para o ensino fundamental e médio, não abordando como esse ferramental poderia ser utilizado também para a formação profissional.

Poucos trabalhos propõem novas metodologias ou ferramentas para o ensino de programação, mostrando uma oportunidade para investigações na área com esse objetivo. A maior parte dos trabalhos aplicaram estratégias de apresentar conceitos de programação por meios que não fossem diretamente através de código-fonte. Essa preocupação aponta que usar as vantagens da AS pode favorecer na apresentação de conceitos que tenham maior significado, principalmente através de conceitos básicos

que formam ou complementam novos conceitos, de forma bem aparente (por exemplo, usando mapas conceituais) ou de forma gradativa.

4.3. Discussões gerais e lacunas para pesquisa

Este mapeamento sistemático mostra que ainda não se esgotam as possibilidades de novas pesquisas em torno no tema. Os resultados oriundos no mapeamento por público-alvo apontam uma maioria de trabalhos voltados à formação profissional. A correlação de conceitos de forma hierárquica e correlacionada, como defendida na AS, se mostra fortemente relacionada à estrutura didática encontrada no ensino de programação e áreas correlacionadas. Por exemplo, o encadeamento de conceitos fundamentais em POO ou Estrutura de Dados podem ser apresentados de forma ordenada e gradativa, com um reforço nas correlações de cada conceito e seu subsequente item: quando se é apresentada uma *classe* e sua relação direta com *objeto*, ou até mesmo as diferentes estruturas de dados, como *árvores*.

A apresentação dos conceitos de forma gradativa é uma diretriz da AS, mas como se trata de uma teoria educacional, não define um sistema metodológico de aplicação. Na AS, o processo de apresentação do conteúdo se inicia com um conceito inicial, que seja de domínio do aluno ou que seja inclusivo para ele, de fácil compreensão (chamado de ideia-âncora ou *subsunçor*) [Moreira e Mansini 2002]. Poucos trabalhos exploram artifícios que possam não somente ajudar na compreensão inicial, mas também em toda a cadeia de conhecimento. Ferramentas oriundas da Robótica Pedagógica e Computação Física, que apresentam artefatos tangíveis, interativos e programáveis, poderiam construir todo um cenário didático e aplicar conceitos de programação, tanto básicos quanto avançados (como em POO).

Uma demanda aparente na análise, e vinculada com o fato de ausência de um processo metodológico, é a de aplicação e desenvolvimento de métodos que conduzam a AS. Houve poucos trabalhos com a iniciativa de criar um meio sistemático de condução de práticas. É comum na comunidade acadêmica, vinculado ao ensino de programação, a busca por métodos que sejam replicáveis em sala de aula. Ainda existe demanda por trabalhos que proponham novas soluções nesse sentido.

Por meio do levantamento realizado nesta pesquisa, é notável a baixa incidência de trabalhos que alinhem métodos que amparem e estimulem o exercício de abstração com a AS. Futuras pesquisas podem propor métodos e estratégias que possam utilizar os conceitos de ideia-âncora e a progressão da apresentação de conceitos de maneira efetiva em atividades e práticas de programação. O uso de recursos tangíveis, como apresentados na Robótica Pedagógica, ou os recursos interativos oriundos de Jogos Digitais, podem promover ambientes que apoiem as principais premissas da AS, conduzindo práticas motivadoras junto aos alunos.

A Figura 1 apresenta um mapa mental das possíveis demandas sugeridas por este mapeamento, podendo ser explorados em novas pesquisas. Vale ressaltar que essas demandas não limitam a ocorrência de outras, mas baseiam-se nos resultados obtidos neste mapeamento, podendo conduzir pesquisadores a novos questionamentos e direcionando futuras pesquisas em torno da temática.

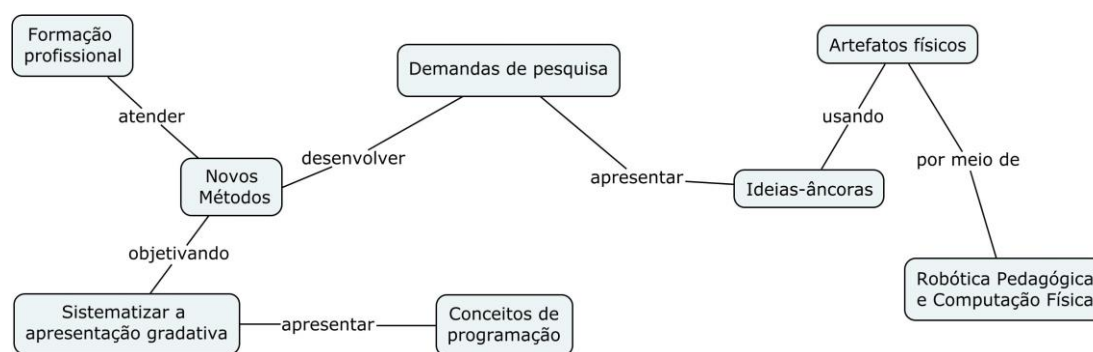


Figura 1. Mapa mental das demandas de pesquisa

5. Considerações finais

Estudos sobre a teoria da AS juntamente com o ensino de programação ainda são escassos, mesmo sendo uma teoria presente em diversas pesquisas de ensino em outras áreas, como matemática e ciência. A motivação em adotar premissas da AS vem principalmente das dificuldades que os educadores ainda encontram na apresentação de conceitos de programação para alunos iniciantes. Como contribuição, este artigo apresentou uma análise temática com o objetivo de orientar pesquisadores e novas investigações.

As estratégias de ensino de programação abordadas nessa análise trazem um cenário geral da pesquisa na área e algumas demandas por novas pesquisas. Por exemplo, a escassez de trabalhos que utilizem Robótica Pedagógica, Computação Física, Interfaces Tangíveis, entre outras ferramentas que possam ajudar na demonstração de conceitos com um menor nível de abstração e maior interatividade. Entre os trabalhos analisados, também se nota uma baixa incidência de pesquisas que abordem o ensino de programação posterior a Algoritmos e Lógica de Programação, como em POO ou Estrutura de Dados, que são comuns em formação profissional. Novas investigações podem sugerir aplicações, estratégias e ferramentas que possam ser aplicadas em cursos formais de programação, que sofrem um impacto direto quando há problemas na aprendizagem, como evasão e baixo aproveitamento acadêmico.

Pela análise realizada neste trabalho, é possível perceber que a utilização de conceitos da teoria da AS contribuem em práticas didáticas no ensino de programação, pois agregam recursos que podem auxiliar em alguns problemas comumente encontrados pelos alunos, como no exercício de abstração e na associação de conceitos. As estratégias apresentadas nos trabalhos utilizam-se de prerrogativas fundamentais da AS para levar novos recursos que possam auxiliar no processo de apresentação e construção do conhecimento por parte dos alunos. Há também trabalhos que conseguem alinhar a AS com alguns recursos como Jogos Digitais, Robóticas, PBL, entre outros, para criar estratégias e métodos de ensino.

Por fim, esta pesquisa não esgota todas as discussões acerca do uso da AS em ensino de programação, mas apresenta um cenário atual na área. Como trabalho futuro, a partir das análises realizadas, sugere-se o aprofundamento da pesquisa, com o intuito de criar um embasamento teórico para o desenvolvimento de metodologias de ensino que alinhem AS e com outras estratégias de ensino, promovendo novas contribuições na área de pesquisa.

Referências

- Aires, J. P., Aires, S., Pereira, M. J., & Alves, L. M. (2021). Active methodologies in incoming programming classes. In *Second International Computer Programming Education Conference (ICPEC 2021)* (Vol. 91). Schloss Dagstuhl--Leibniz-Zentrum für Informatik.
- Alexandron, G., Armoni, M., Gordon, M., & Harel, D. (2017). Teaching scenario-based programming: An additional paradigm for the high school computer science curriculum, Part 1. *Computing in Science & Engineering*, 19(5), 58-67.
- Astolfi, G., & Junior, D. L. (2015). Investigação sobre conhecimentos prévios de alunos do curso Técnico em Informática a partir da aplicação de organizadores prévios. Aprendizagem Significativa. *Revista/Meaningful Learning Review*, 15-28.
- Astolfi, G., & Junior, D. L. (2016,). Ensino de linguagem de programação com ênfase na aprendizagem significativa. In *Anais do XXIV Workshop sobre Educação em Computação* (pp. 2106-2115). SBC.
- Almeida, T., Netto, J. F., da Silva, R., & Custódio, T. (2017). Laboratório remoto de robótica como elemento motivador para a aprendizagem de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 28, No. 1, p. 665).
- Ausubel, D. P. (2012). *The acquisition and retention of knowledge: A cognitive view*. Springer Science & Business Media.
- Barbosa, L. S., Fernandes, T. C., & Campos, A. M. (2011). Takkou: Uma ferramenta proposta ao ensino de algoritmos. In *XVIII Workshop sobre Educação em Computação (WEI 2011)*.
- Berssanette, J. H., & Francisco, A. C. (2018). Proposta de abordagem prática para o ensino de Programação baseada em Ausubel. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 29, No. 1, p. 398).
- Bishop, M., Dai, J., Dark, M., Ngambeki, I., Nico, P., & Zhu, M. (2017). Evaluating secure programming knowledge. In *IFIP World Conference on Information Security Education* (pp. 51-62). Springer, Cham.
- Bosse, Y. (2020). Padrões de dificuldades relacionadas com o aprendizado de programação. Tese de Doutorado. Universidade de São Paulo. São Paulo, SP.
- Brito, M. G. L., Rodriguez, L. R. R. & Pellicer, Y. S. (2016). Concept Maps as a means university teaching assistant. In *International Conference on Concept Mapping*. Springer, Cham.
- Buriticá, O. I. T. (2011). Planeación de la asignatura programación i en un programa de ingeniería de sistemas a partir de la teoría del aprendizaje significativo. *Revista Educación en Ingeniería*, 6(12), 102-114.
- Buriticá, O. I. T. (2013). Estrategia metodológica para aproximar los paradigmas funcional, estructurado y orientado a objetos en ingeniería de sistemas a partir de aprendizaje significativo. *Avances Investigación en Ingeniería*, 10(2), 49-63.

- Buriticá, O. I. T. (2014). Relaciones de aprendizaje significativo entre dos paradigmas de programación a partir de dos lenguajes de programación. *Tecnura*, 18(41), 91-102.
- Derneval, D., Coelho, J. A. D. M., & Bittencourt, I. I. (2019). Mapeamento sistemático e revisão sistemática da literatura em informática na educação. *JAQUES, Patrícia Augustin; PIMENTEL, Mariano; SIQUEIRA, Sean; BITTENCOURT, Ig.(Org.) Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa. Porto Alegre: SBC.*
- Gasaymeh, A., AlJa'afreh, I. A., Al-Dmour, A., & Abu-Alrub, M. (2016). Higher education students' preferences for applying the principles of constructivism in learning programming languages with the use of ICTs. *Journal of Studies in Education*, 6(3), 168-187.
- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education-ICEE* (Vol. 7).
- Gomes, A., & Mendes, A. (2014). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1-8). IEEE.
- Gómez-Zermeño, M. G. & Mejía, D. B. (2020). Development of Significant Learning through Scratch Programming Logic of Secondary School Students. *The International Journal of Technologies in Learning*, 27(2), 21-36.
- Hernandez, C. C., Silva, L., Segura, R. A., Schimiguel, J., Ledón, M. F. P., Bezerra, L. N. M., & Silveira, I. F. (2010). Teaching programming principles through a game engine. *CLEI electronic journal*, 13(2), 1-8.
- Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação.
- Kaloti-Hallak, F., Armoni, M., & Ben-Ari, M. (2019). The Effect of Robotics Activities on Learning the Engineering Design Process. *Informatics in Education*, 18(1), 105-129.
- Lima, Á., Diniz, M., & Eliasquevici, M. (2019). Metodologia 7Cs: Uma Nova Proposta de Aprendizagem para a Disciplina Algoritmos. In *Anais do XXVII Workshop sobre Educação em Computação* (pp. 429-443). SBC.
- Matthews, R., Hin, H. S., & Choo, K. A. (2015). Practical use of review question and content object as advanced organizer for computer programming lessons. *Procedia-Social and Behavioral Sciences*, 172, 215-222.
- Medeiros, R. P. (2019). *Hello, world: uma análise sobre dificuldades no ensino e na aprendizagem de introdução à programação nas universidades*. Tese de doutorado. Universidade Federal de Pernambuco. Recife, PE.
- Medeiros, T. J., da Silva, T. R., & da Silva Aranha, E. H. (2013). Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. *Renote*, 11(3).
- Menon, P., & Kovalchick, L. (2020). Using a Concept Map to Represent the Composition of Knowledge in an Introductory Programming Course. *Information Systems Education Journal*, 18(3), 4-17.

- Minakshi, S., & Sonal, C. (2018). Using Concept Map Network Based CLE for Teaching Learning and Evaluating the Knowledge Acquired by Learners. In *International Conference on Advances in Computing and Data Sciences* (pp. 148-157). Springer, Singapore.
- Moreira, M.A. (2012). O que é afinal, aprendizagem significativa? *Qurrriculum. Laguna*, v. 2, n. 3, p. 1-27.
- Moreira, M. A., & Masini, E. F. S. (2002). *Aprendizagem significativa: a teoria de David Ausubel*. Centauro.
- Moström, J. E., Boustedt, J., Eckerdal, A., McCartney, R., Sanders, K., Thomas, L., & Zander, C. (2008). Concrete examples of abstraction as manifested in students' transformative experiences. In *Proceedings of the Fourth international Workshop on Computing Education Research* (pp. 125-136).
- Mühling, A. (2016). Aggregating concept map data to investigate the knowledge of beginning CS students. *Computer Science Education*, 26(2-3), 176-191.
- Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., & McKenzie, J. E. (2021). PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. *Bmj*, 372.
- Pelizzari, A., Kriegel, M. D. L., Baron, M. P., Finck, N. T. L., & Dorocinski, S. I. (2002). Teoria da aprendizagem significativa segundo Ausubel. *revista PEC*, 2(1), 37-42.
- Piteira, M., & Costa, C. (2013). Learning computer programming: study of difficulties in learning programming. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 75-80).
- Raposo, E. H. S., & Dantas, V. (2016). O Desafio da Serpente-Usando gamification para motivar alunos em uma disciplina introdutória de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 27, No. 1, p. 577).
- Santos, R., Sousa, B., Raiol, A., Cerqueira, P., & Bezerra, F. (2019). Uma Proposta de Método de Ensino e Relatos de Experiências com a Robótica Educacional. In *Anais do XXVII Workshop sobre Educação em Computação* (pp. 111-120). SBC.
- Sharma, M. & Chawla S. (2018). Using Concept Map Network Based CLE for Teaching Learning and Evaluating the Knowledge Acquired by Learners. In *International Conference on Advances in Computing and Data Sciences* (pp. 148-157). Springer, Singapore.
- Shitsuka, D. M., Pereira, A. S., Shitsuka, R., & Boghi, C. (2019). Aprendizagem ativa de programação em turmas de engenharia: uma pesquisa-ação. *Research, Society and Development*, 8(3), 01-19.
- Shneiderman, B. (1977). Teaching programming: A spiral approach to syntax and semantics. *Computers & Education*, 1(4), 193-197.
- Silva, G., Santos, G., Canedo, E. D., Rissoli, V., Praciano, B., & Andrade, G. (2020). Impact of Calango language in an Introductory Computer Programming Course. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

- Soto, B. D. G., Ortiz, J. F. V., Moreno, I. R., & Ortiz, L. E. V. (2013). Experiencias de éxito en la aplicación de mapas conceptuales en la carrera de Ingeniería en Computación, México. *JETT*, 4(1), 73-84
- Souza, D. M., da Silva Batista, M. H., & Barbosa, E. F. (2016). Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, 24(1), 39.
- Souza, L. K. D. (2019). Pesquisa com análise qualitativa de dados: conhecendo a Análise Temática. *Arquivos Brasileiros de Psicologia*, 71(2), 51-67.
- Sousa, V. N. D., Moretti, A. C., & Podestá, V. A. D. (2008). Programação da grade de horário em escolas de ensino fundamental e médio. *Pesquisa Operacional*, 28(3), 399-421.
- Suárez, M. G. (2015). Efectos del proceso de aprender a programar con “Scratch” en el aprendizaje significativo de las matemáticas en los estudiantes de educación básica primaria. *Escenarios*, 13(2), 87-102.
- Wang, J., Mendori, T., & Xiong, J. (2014). A language learning support system using course-centered ontology and its evaluation. *Computers & Education*, 78, 278-293.
- Yepes, I., Barone, D. A. C., & Porciuncula C. M. D. (2021). Use of Drones as Pedagogical Technology in STEM Disciplines. *Informatics in Education*.