

Mapeamento Sistemático em Metodologias de Ensino-aprendizagem de Programação

Cátia Mesquita Brasil Khouri
Dep. Ciências Exatas e Tecnológicas
Univ. Estadual do Sudoeste da Bahia
Vitória da Conquista-BA, Brasil
khouri@uesb.edu.br

Gidevaldo Novais dos Santos
Dep. Ciências Exatas e Tecnológicas
Univ. Estadual do Sudoeste da Bahia
Vitória da Conquista-BA, Brasil
gnovais@uesb.edu.br

Maria Silva Santos Barbosa
Dep. Ciências Exatas e Tecnológicas
Univ. Estadual do Sudoeste da Bahia
Vitória da Conquista-BA, Brasil
mssbarbosa@uesb.edu.br

Resumo—O processo de ensino-aprendizagem de programação vem atraindo mais e mais a atenção da comunidade acadêmica. Por um lado, devido aos altos índices de reprovação e repetência nas disciplinas de programação nos cursos de graduação da área tecnológica. Por outro lado, devido ao interesse crescente de inserir o pensamento computacional na educação básica, o que de fato vem acontecendo em vários países ao redor do mundo. No Brasil, até então, o ensino de programação só foi incluído nos cursos técnicos integrados ao ensino médio. Este mapeamento sistemático apresenta uma visão abrangente das estratégias metodológicas que têm sido propostas para o ensino de programação na educação básica – ensino médio. Foram coletados dados nas publicações em periódicos e anais de eventos de âmbito nacional que discutem questões relativas ao uso de tecnologias na educação, no período de 2014 a 2020. A identificação inicial de 983 artigos que trazem "ensino-aprendizagem de programação" em seu título ou resumo revelam a importância do tema no meio acadêmico. O estudo identifica uma série de ferramentas e estratégias experimentadas em cursos presenciais, a distância ou semipresenciais; na educação formal, em oficinas ou cursos livres. Observa-se uma preferência por atividades plugadas, ou seja, com utilização de algum dispositivo eletrônico, o que pode se justificar ao considerar a faixa etária do público-alvo e suas preferências na utilização destes, sendo possível considerar o acesso a tais recursos como algo relevante para a escolha. Justifica-se ainda o objetivo de utilizar uma metodologia que potencialize a ação do aprendiz, deslocando a centralidade das atividades de aprendizagem para o maior interessado no processo: o aluno - e considerando, por exemplo, os aspectos motivadores. Outro aspecto relevante identificado é que a pesquisa na área ainda se apresenta incipiente e carece tanto de maior aprofundamento teórico-conceitual quanto de experimentação metodologicamente controlada.

Index Terms—metodologias de ensino, programação, mapeamento sistemático

I. INTRODUÇÃO

Aprender a programar constitui-se numa tarefa complexa, independente do público interessado, pois envolve diversas singularidades do domínio da programação. É necessário o exercício constante do raciocínio lógico e da capacidade de abstração para resolver problemas – interpretar, analisar requisitos e apresentar soluções algorítmicas. Estas, por sua vez, exigem a compreensão de conceitos como estruturas de controle, recursão, ponteiros e outros. Enfrentadas essas dificuldades, o educando ainda precisa lidar com a sintaxe complexa das linguagens de programação e as ferramentas

normalmente utilizadas para edição, compilação/interpretação e depuração de códigos [12].

A complexidade inerente à programação resulta, muitas vezes, em fracasso no aprendizado, levando a altos índices de reprovação nas disciplinas de programação (introdutória ou avançada) em cursos de graduação superior e técnicos [4], [11]. As frequentes reprovações causam desânimo e desmotivação nos estudantes que acabam por evadir dos cursos [11].

Concorrentemente a essas dificuldades, vêm surgindo nos últimos anos diversas propostas de inserir-se o ensino de programação já na educação básica. Na realidade, o que tem sido proposto e, de fato, efetivado em diversos países é a alteração de currículos visando inserir o desenvolvimento do pensamento computacional na educação de crianças e adolescentes [1]. Entre outros aspectos, as alterações curriculares incluem o ensino de linguagens de programação para crianças. A visão por trás dessas iniciativas é a de que as habilidades envolvidas no pensamento computacional são fundamentais para indivíduos de qualquer área de atuação [25]. Além do que, a demanda por profissionais de computação tem aumentado significativamente em uma sociedade cada vez mais dependente das tecnologias da informação e comunicação.

No Brasil, ao contrário do movimento internacional, o ensino de programação ainda não faz parte do currículo da educação básica, exceto para os cursos técnicos integrados ao ensino médio. Assim, docentes que atuam nesse âmbito e pesquisadores com interesse no tema têm investido no desenvolvimento e aplicação de metodologias alternativas (distintas da abordagem tradicional) para vencer as dificuldades inerentes ao ensino e aprendizagem de programação. Apesar das iniciativas, o volume de evidências científicas sobre os reflexos da aplicação de tais metodologias ainda se apresenta incipiente e, portanto, inconclusivo.

É nesse contexto que se propõe a realização do presente Mapeamento Sistemático (MS) com o objetivo de identificar estudos primários desenvolvidos no Brasil dedicados às metodologias de ensino de programação na educação básica, em especial, aqueles voltados para o ensino médio. Pretende-se disponibilizar à comunidade acadêmica e científica um levantamento abrangente que apresente um quadro geral do cenário das pesquisas na área, apontando as lacunas existentes e a direção para novos esforços de pesquisa.

Considerando a estreita relação das habilidades concernentes ao pensamento computacional e a aprendizagem de programação, procuramos identificar nos trabalhos relatados o desenvolvimento de tais habilidades como um objetivo. No intuito de identificar as bases teóricas para as metodologias propostas, extraímos também dados relativos às teorias de aprendizagem empregadas em sua concepção. Por fim, fazemos uma breve discussão a respeito da validação dos resultados apresentados nos artigos selecionados. O escopo do mapeamento abrange os trabalhos publicados no período de 01 de janeiro de 2014 a 31 de março de 2020, nas revistas e anais de eventos nacionais conforme descritos na Seção III.

O presente artigo apresenta trabalhos relacionados na Seção II e a metodologia adotada na Seção III. Os resultados obtidos são apresentados e discutidos na Seção IV. Na Seção V são apresentadas as considerações finais e indicação de trabalhos futuros.

II. TRABALHOS RELACIONADOS

Estabelecer o estado da arte em torno de um tema de interesse é fundamental para o direcionamento de novas pesquisas que venham a agregar conhecimento à área. Assim é que, eventualmente, pesquisadores debruçam-se sobre estudos primários desenvolvidos em um determinado período para levantar produções cientificamente relevantes e questões em aberto. Sobre o tema abordado neste trabalho, encontramos alguns estudos de mapeamento sistemático, dos quais apresentamos a seguir os mais recentes.

O MS realizado por [8] levanta as dificuldades no ensino e na aprendizagem de programação. Eles trabalharam num escopo internacional buscando artigos publicados entre 2010 e 2014 numa única base de dados: SCOPUS¹. No nosso mapeamento, o interesse é o de levantar metodologias aplicadas no contexto brasileiro, que apresenta várias diferenças com relação ao internacional. Em especial, a não obrigatoriedade de oferta de disciplinas de ciência da Computação no ensino básico. Marcolino e Barbosa [7] conduziram um MS para identificar os softwares educacionais que apoiam o ensino de programação, no que tange o uso de novas TICs, modalidades de ensino e tecnologias de desenvolvimento. Enquanto o nosso trabalho busca identificar as abordagens metodológicas aplicadas ao ensino de programação, a motivação para o trabalho deles foi a criação de um catálogo de contribuições e a possível extração e reutilização de artefatos na concepção de novos produtos de software.

Caldeira e Vilela [6] apresentam um MS que se restringia a auxiliar na escolha de plataformas EAD para o ensino-aprendizagem de algoritmos e programação. Elas buscaram artigos publicados no período de 2011 a 2016 em algumas bases de dados nacionais e uma internacional. Após aplicação dos critérios de inclusão e exclusão foram eliminados ainda os estudos primários que não contemplavam a comparação entre plataformas ou não faziam uma análise do uso da plataforma como meio de auxiliar no processo de ensino-aprendizagem

¹<https://www.elsevier.com/pt-br/solutions/scopus>

em algoritmos e programação. Como resultado, foram identificados apenas dois trabalhos baseados, respectivamente, na plataforma MOODLE² e A.L.I.C.E.³. Entretanto, o MS realizado considerou estudos secundários em sua metodologia.

Já em trabalho mais recente [9], o MS faz uma avaliação histórica das publicações dos anos 2014 a 2017 sobre o ensino de programação no Brasil na educação superior e básica, sinalizando os tipos de metodologias e ferramentas utilizadas sem, contudo, citar seus nomes. Eles só consideraram o ensino formal, excluindo os estudos que relatam a experimentação de metodologias em cursos livres ou oficinas, e não apresentam uma discussão sobre os resultados alcançados com as metodologias.

A principal contribuição do MS aqui relatado é a apresentação de um panorama atualizado das metodologias de ensino de programação que têm sido propostas e relatadas nas publicações nacionais mais relevantes identificando diversos aspectos das abordagens utilizadas como gamificação, robótica educativa, desenvolvimento de jogos; ferramentas digitais e não digitais utilizadas; presencialidade; linguagens de programação empregadas, entre outros.

III. METODOLOGIA

Mapeamento Sistemático (MS), ou revisão de escopo, é um tipo de estudo secundário, que oferece uma visão ampla acerca de determinada área de estudo, permitindo verificar a existência de lacunas e evidências nas pesquisas da área. Em geral, um MS é empreendido quando se pretende detectar potenciais questões de pesquisa em aberto [13]. Para esse fim, o MS se encarrega de coletar e classificar estudos primários sobre um mesmo tema, mas que investigam questões de pesquisa distintas, identificando o local da publicação [21].

Neste trabalho, a metodologia foi definida com base nas orientações de [13] e [16]. Tendo em vista o objeto de pesquisa, adotamos o protocolo que passamos a descrever. Para classificar as metodologias que vêm sendo empregadas no ensino introdutório de programação no Brasil para estudantes do ensino médio, foram propostas as seguintes questões de pesquisa (QP):

- QP1: *Quais metodologias têm sido propostas para o ensino introdutório de programação no Ensino Médio?*
- QP2: *Em sua concepção, as metodologias buscam o desenvolvimento do pensamento computacional?*
- QP3: *As metodologias propostas são baseadas em alguma Teoria de Aprendizagem? Quais?*
- QP4: *Os estudos revelam melhoria na aprendizagem com a aplicação das metodologias propostas?*

O escopo definido para a pesquisa foi o corpo de publicações em anais de eventos e periódicos que discutem questões relativas ao uso de tecnologias na educação. No intuito de restringir as buscas às publicações mais relevantes, foram selecionados eventos de âmbito nacional ou internacional

²<https://moodle.com/>

³RAABE, André Luís A.; SILVA, Júlia M. C. da. Um Ambiente para Atendimento às Dificuldades de Aprendizagem de Algoritmos. XIII Workshop sobre Educação em Computação. São Leopoldo-RS, 2005.

realizados no Brasil, cancelados pela Sociedade Brasileira de Computação (SBC) que estivesse em sua 25ª edição ou posterior; e revistas nacionais avaliadas pelo sistema Qualis CAPES com conceito B1 ou superior. Para manter o estudo tão abrangente quanto possível e obter dados mais recentes, as buscas foram realizadas sobre fontes publicadas nos anos 2014 a 2020. Foram estas as bases de dados consideradas:

- Revista Brasileira de Informática na Educação (RBIE);
- Revista Novas Tecnologias na Educação (RENOTE);
- Anais do Simpósio Brasileiro de Informática na Educação (SBIE);
- Anais do Workshop de Informática na Escola (WIE);
- Anais do Workshop de Educação em Computação (WEI);

A. Processo de busca

A primeira etapa para a identificação dos estudos primários a serem examinados foi o processo de busca automática nas bases de publicações. As bases de dados foram distribuídas entre os pesquisadores para esse fim. No caso das quatro primeiras fontes citadas acima, os repositórios oferecem opções de filtros que possibilitaram realizar buscas pela chave:

(ensin* programa*) OR (lógica programa*) OR (aprend*programa*) OR (computa* ("ensino básico" OR "ensino médio")) OR (computa* "educação básica")

nos campos “Título” e “Resumo”, restringindo o escopo temporal de 01 de janeiro de 2014 a 31 de março de 2020. No caso do WEI, a busca automática foi processada repositório SOL (SBC Open Lib)⁴ apenas para os anos 2015 a 2020. Uma vez que as publicações do ano 2014 não estão disponíveis nesta base de dados, a busca foi feita manualmente para este período. Como resultado da fase busca foram obtidos 983 artigos. A Tabela I apresenta o quantitativo das publicações retornadas por fonte:

Tabela I
QUANTITATIVO DE ARTIGOS RETORNADOS NA BUSCA AUTOMÁTICA

Fonte	RBEI	RENOTE	SBIE	WEI	WIE	Total
#Artigos	149	154	293	159	228	983

Fonte: Os autores

B. Critérios de Inclusão e Exclusão

A partir dos 983 trabalhos selecionados, aplicamos os critérios de inclusão e exclusão que possibilitaria restringir o corpo de estudos primários àqueles que, potencialmente, contribuiriam para responder às questões de pesquisa postas. Assim, para que o artigo fosse incluído na pesquisa deveria ser artigo completo e que abordasse o ensino de programação no ensino médio (CI1)⁵. No entanto, deveriam ser excluídos:

⁴<https://sol.sbc.org.br/index.php/wei/issue/archive>

⁵Foram incluídos alguns artigos que propunham metodologias para o ensino de programação aplicadas no ensino médio e no ensino fundamental.

artigos que não apresentassem uma proposta de metodologia para o ensino de programação (CE1); (2) artigos que reportassem revisões ou mapeamentos sistemáticos – a fim de somente analisarmos estudos primários (CE2); e artigos que relatassem a continuidade de um estudo relatado em outro artigo também selecionado. Nesse caso, foi mantida a publicação mais recente sendo excluída(s) a(s) anterior(es) (CE3). Foram estes os critérios de inclusão e exclusão estabelecidos:

CI1: Artigos originais completos que abordam o ensino de programação na educação básica.

CE1: Artigos que não propõem, pelo menos, uma metodologia aplicada ao ensino de programação no ensino médio.

CE2: Artigos de revisão ou mapeamento sistemático de literatura.

CE3: Artigos referentes à mesma pesquisa (para evitar duplicações, foram excluídas as publicações menos recentes)

A aplicação dos critérios de inclusão e exclusão a cada artigo foi feita, inicialmente, com a leitura dos resumos, seguindo com a leitura da seção de metodologia e por fim, do restante artigo até que fossem identificados elementos suficientes para a correta aplicação. Cada pesquisador realizou a seleção dos artigos de um ou dois repositórios. Ao todo, foram selecionados 44 artigos para serem classificados. A Tabela II apresenta o número de artigos resultantes por fonte.

Tabela II
QUANTITATIVO DE ARTIGOS SELECIONADOS CONFORME CI1 E CE3

Fonte	RBEI	RENOTE	SBIE	WEI	WIE	Total
#Artigos	1	7	4	10	19	41

Fonte: Os autores

C. Classificação dos Estudos Primários

Nesta fase, os artigos selecionados foram distribuídos entre os pesquisadores de modo que cada um ficou responsável por classificar os estudos de uma fonte distinta daquela que lhe tinha sido atribuída na fase de seleção. Cada artigo passou então por uma leitura criteriosa acompanhada da anotação do título, autores, local e ano de publicação. Visando responder à primeira questão de pesquisa (QP1), destacamos aspectos relevantes para a classificação das metodologias de ensino propostas. Para Manfredi [14], não existe uma conceitualização única para metodologia de ensino, uma vez que ela está vinculada “explícita ou implicitamente a uma concepção epistemológica e a uma visão de mundo, pois as práticas científicas e pedagógicas são aspectos de uma totalidade maior: a prática social (praxis social).” [14]. No entanto, não é uma pretensão deste trabalho definir as metodologias utilizadas do ponto de vista epistemológico ou histórico-social. Ao invés disso, a classificação que fazemos dos estudos analisados diz respeito tão somente às técnicas/ferramentas/caminhos para aprendizagem mencionadas por eles. Para esse fim, atribuímos “etiquetas” aos artigos analisados, relativas aos métodos e

técnicas empregados, incluindo artefatos utilizados e a forma de aplicação. Como resultado, os artigos foram classificados com as seguintes etiquetas:

- 1) PLUGADA/DESPLUGADA/PLUGDES - Se propõe atividades plugadas, desplugadas ou de ambos os tipos, respectivamente;
- 2) RB - se baseada em robótica pedagógica;
- 3) COMPETIÇÃO - se envolve uma competição;
- 4) JOGO - se propõe o desenvolvimento de um jogo;
- 5) GAMIFICAÇÃO - se utiliza técnicas de gamificação;
- 6) PRESENCIAL/DISTÂNCIA/SEMI - se a metodologia é aplicada, respectivamente, presencialmente, a distância ou num estilo semipresencial (sem perda de generalidade, classificamos na modalidade semipresencial as metodologias que preveem atividades presenciais e atividades executadas pelos educandos sem a supervisão dos ministrantes, independente de carga horária).
- 7) Se a metodologia considera abordar conteúdos de programação em paralelo aos conteúdos de outras disciplinas, o estudo recebeu uma etiqueta com o nome da disciplina. Por exemplo, MATEMÁTICA e FÍSICA.
- 8) Se a abordagem proposta busca desenvolver determinados comportamentos sociais nos estudantes baseada no fato que tais comportamentos propiciam a aprendizagem, atribuímos a etiqueta correspondente. Por exemplo, MOTIVAÇÃO, COLABORAÇÃO, INTERATIVIDADE, etc.

Ainda com relação à (QP1), rotulamos os artigos também conforme as ferramentas utilizadas, os conteúdos de programação constantes no currículo abordado e as linguagens de programação utilizadas. Devido à quantidade desses rótulos, faremos sua apresentação junto com a apresentação dos dados coletados na Seção IV.

Um aspecto importante observado nesta fase da coleta é o fato de que alguns estudos analisados propõem utilizar-se do ensino de programação visando a desenvolver o pensamento computacional [25] nos estudantes. Por outro lado, outros estudos focam no pensamento computacional com o objetivo de potencializar a aprendizagem de programação. Assim, lançam mão de atividades (outras que não dizem respeito, diretamente, à programação formal de computadores) que desenvolvam o pensamento computacional no intuito de favorecer a aprendizagem da programação formal. Uma preocupação neste estudo é observar se as metodologias propostas levam em conta a influência das habilidades relativas ao PC na capacidade do aprendiz absorver os conteúdos diretamente relacionados à resolução de problemas computacionais através da elaboração de programas. Por isso propusemos a segunda questão de pesquisa (QP2). Estudos que levaram em conta essa influência do pensamento computacional receberam a etiqueta PC.

Para responder à terceira questão (QP3), buscamos identificar nos artigos uma fundamentação teórica baseada em teorias da aprendizagem. Em caso positivo, reportamos a respectiva teoria atribuindo uma etiqueta com o nome de seu autor. Por exemplo, estudos baseados na Teoria do Construcionismo,

receberam a etiqueta PAPERT. Para não deixar margem de dúvida com relação ao procedimento de classificação, estudos que não se basearam em uma teoria de aprendizagem receberam uma etiqueta NAOTEO. Por fim, a quarta questão (QP4) é direcionada a averiguar os efeitos das metodologias propostas sobre a aprendizagem do público-alvo no que diz respeito à programação formal de computadores e à possibilidade de generalização dos resultados obtidos. Nesse caso, a metodologia utilizada no estudo primário foi analisada considerando o tamanho e a composição da amostra — público-alvo — observada, aplicação do método, resultados obtidos em termos do aproveitamento dos estudantes na elaboração de programas. Em termos quantitativos, foi levantada a carga horária das atividades realizadas e o número de replicações do experimento.

Após a primeira rodada de leitura dos artigos, uma nova rodada foi executada. Para essa etapa, os estudos foram redistribuídos entre os pares que fizeram uma nova leitura a fim de detectar possíveis divergências entre os mesmos. Nos casos em que houve divergências foram discutidas entre ambos os leitores que decidiram sobre a questão divergente.

Ainda nesta etapa foi excluído um artigo cujas metodologia não havia sido aplicada experimentalmente e duas duplicações de estudos reportados em duas fontes distintas. Conforme definido o CE3, foram mantidas as versões mais recentes. Todos os artigos excluídos foram publicados nos anais do SBIE. Restaram, então, 41 estudos na etapa final do mapeamento. A Tabela III mostra a configuração final da distribuição dos artigos analisados por fonte.

Tabela III
CONFIGURAÇÃO FINAL DOS ARTIGOS ANALISADOS

Fonte	RBEI	RENOTE	SBIE	WEI	WIE	Total
#Artigos	1	7	4	10	19	41

Fonte: Os autores

IV. RESULTADOS E DISCUSSÕES

Apresentamos aqui os dados coletados com o fim da caracterizar as metodologias propostas. Em cada subseção será abordada uma questão de pesquisa.

A. Metodologias de ensino propostas

Conforme explicitado na seção 3.3, registramos as ocorrências das técnicas/ferramentas/ caminhos para aprendizagem em cada estudo primário. Assim, para responder à QP1: *Quais metodologias têm sido propostas para o ensino introdutório de programação no Ensino Médio?* Contabilizamos as diversas etiquetas atribuídas aos estudos e apresentamos a seguir uma síntese dos resultados.

1) *Computação plugada/desplugada:* A computação desplugada é uma técnica utilizada para ensinar conceitos de computação sem uso do computador. Brackmann [1] defende a utilização desta técnica baseado na teoria construcionista [19], que estimula a movimentação física do educando na

manipulação de materiais concretos. Entre os artigos examinados, em apenas 6 (15%) são empregadas atividades desplugadas, enquanto a maioria (85%) só propõe atividades com uso do computador. O Gráfico 1 apresenta a distribuição das metodologias de acordo com esse aspecto.

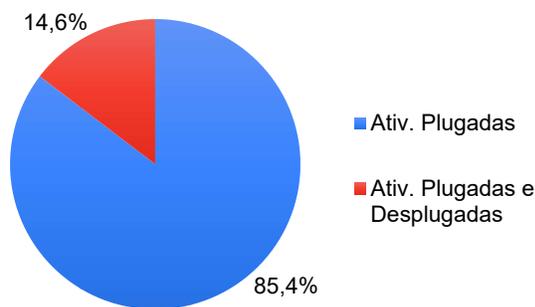


Figura 1. Metodologias quanto ao tipo de atividades propostas: Plugadas/Desplugadas.

As ferramentas digitais citadas nos artigos foram categorizadas de acordo com suas funcionalidades. As Tabelas IV e V mostram todas as ferramentas e o número de estudos em que elas são aplicadas e o Gráfico 2 mostra as ferramentas organizadas por categoria.

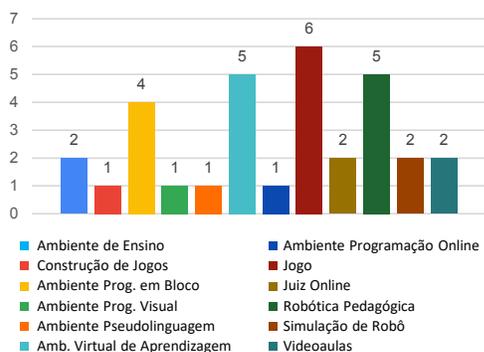


Figura 2. Ferramentas digitais utilizadas por categoria e o número de estudos em que foram aplicadas.

Tabela IV
FERRAMENTAS DIGITAIS

Ferramenta	#Estudos	Descrição
Hour of Code	1	Ambiente de ensino de programação – Lúdico, contém vídeos e tutoriais com personagens familiares aos estudantes.
Tri-Logic	1	Ambiente de ensino e aprendizagem de lógica de programação gamificado.
Construct2	3	Ambiente de programação visual para construção de jogos.
MIT App Inventor	8	Plataforma de programação visual (em blocos) para criação de aplicativos para Android. O site oferece muitos tutoriais, inclusive para iniciantes.
iVProg	1	Ambiente de programação em blocos. Pode ser integrado ao Moodle.
Blockly	1	Ambiente de programação em blocos online desenvolvido para ensinar programação a crianças. Enquanto o programa em blocos é construído vai aparecendo ao lado o código em uma das cinco linguagens: Python, Lua, JavaScript, PHP ou Dart.
SCRATCH	12	Ambiente de programação em blocos. É possível criar jogos, histórias, animações interativas.
Visual Programmer (ViP)	1	Ambiente de programação visual através de símbolos para ensino de programação a alunos surdos.
VisualG	3	Ambiente para edição e execução de algoritmos em pseudolinguaagem (palavras-chave em português) que mostra as mudanças de estado das variáveis enquanto o algoritmo é executado.
AGILE	1	Ambiente Virtual de Aprendizagem gamificado e interativo.
Edmodo	1	Ambiente Virtual de Aprendizagem.
AVA não identificado	1	Ambiente Virtual de Aprendizagem.
Canvas	1	Ambiente Virtual de Aprendizagem. Oferece uma opção de utilizar o ambiente gratuitamente com as funcionalidades essenciais de gerenciamento e aprendizagem.
MOODLE	1	Ambiente Virtual de Aprendizagem de código aberto, permite a instalação de pluggins.
VPL	1	Ambiente de programação textual online (C, C++, PHP, Java e Fortran). Permite edição e execução de código e pode ser integrado ao AVA Moodle.
Code.org	1	Jogo - Plataforma que disponibiliza jogos para aprendizagem de programação.
Problemas de Lógica	1	Ambiente online que disponibiliza vários jogos tipo desafio para desenvolvimento do raciocínio lógico para vários níveis/idades - Teste de Einstein, Quizzes de várias disciplinas, passatempos em geral.
Cod Combat	1	Jogo que aborda conceitos de programação através de uma aventura medieval. Oferece ambiente de programação próprio – Python, Javascript e CoffeScript (experimental).
CodeMonkey	1	Jogo para ensinar programação na linguagem CoffeeScript. A versão gratuita contempla condicionais, repetição, funções e variáveis.
Light-Bot	3	Jogo online que permite ao usuário controlar um robô através de um conjunto de passos.

Tabela V
FERRAMENTAS DIGITAIS - CONTINUAÇÃO

Ferramenta	#Estudos	Descrição
Monster Coding	1	Jogo em que o usuário cria um personagem e o comanda através de blocos - booleanos, matrizes, condicionais e repetição.
URI Online Judge	1	Juiz - Plataforma que disponibiliza diversos problemas para serem resolvidos através de programas e um juiz online que corrige os programas com resposta imediata.
MOSS	1	Juiz - Serviço de detecção de plágio. Pode ser integrado ao ambiente URI.
ARDUINO	3	Robótica. Plataforma aberta de prototipagem que pode ser programado no ambiente Processing em uma linguagem de programação própria.
Kit de Robótica LEGO	4	Robótica. O kit contém componentes para montar robôs programáveis pelos alunos. Disponibiliza um caderno de atividades.
Kit de Robótica Fischertechnik - ROBO TX Explorer	1	Robótica. kit com componentes para montagem de robôs. Possui um ambiente integrado para programação em forma de fluxograma que depois é transferido para o robô via USB.
Mblock	1	Robótica. Ambiente de programação gráfica baseado no editor Scratch 2.0. Traduz o programa para a linguagem Arduino e o transmite para a placa.
SCRATCH para Arduino-S4A	1	Robótica. Uma adaptação do Scratch para programação (em blocos) do Arduino.
Robocode	1	Ambiente de simulação de Robôs. Simula uma batalha entre robôs em que se joga programando em Java. Os alunos adquirem conhecimentos de POO.
RoboMind	2	Ambiente de simulação de Robôs. O usuário programa o comportamento de um robô para resolver desafios.
Videoaulas	3	Videoaulas com conteúdos sobre (1) programação de jogos sem qualquer intervenção/orientação externa; (2) Idem; (3)

2) *Robótica*: O uso da robótica na educação é um ponto central da teoria do Construcionismo [18]. Silva [2] define robótica pedagógica como “o conjunto de processos e procedimentos envolvidos em propostas de ensino-aprendizagem que tomam os dispositivos robóticos como tecnologia de mediação para a construção do conhecimento.” Dentre os artigos analisados, encontramos 7 (17%) que utilizam de dispositivos robóticos para a mediação do ensino-aprendizagem (Gráfico 3).

3) *Uso de Jogos*: Os jogos são explorados sob duas perspectivas em alguns dos estudos analisados. Uma delas é levar os estudantes a aprenderem programação enquanto desenvolvem jogos. Ferramentas como Scratch e Construct 2 viabilizaram algumas dessas propostas. A outra é conduzir a aprendizagem através de jogos, isto é, os educandos aprendem enquanto jogam. Esta alternativa é propiciada por ferramentas digitais como Cod Combat, CodeMonkey, LightBot e Monster Coding. O Gráfico 4 apresenta os quantitativos de estudos que utilizaram estas abordagens, respectivamente – 14 estudos propõem o desenvolvimento de jogos, sendo que 2 desses propõem também o uso de jogos; 3 estudos contam apenas com o uso de jogos e 24 não adotam qualquer uma dessas

abordagem.

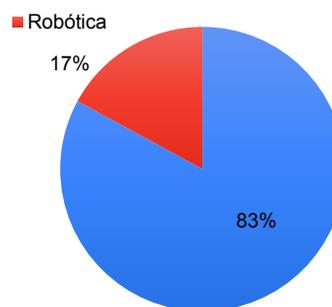


Figura 3. Percentual de estudos que propuseram robótica pedagógica.

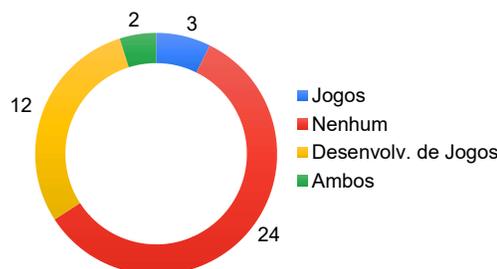


Figura 4. Número de estudos quanto ao uso de jogos e desenvolvimento de jogos.

4) *Gamificação*: O termo “gamificação” refere-se à ideia de usar elementos de design de jogos em contextos não relacionados a jogos para motivar e aumentar a atividade e a retenção do usuário [5]. Nas últimas décadas, o uso de gamificação na educação vem se intensificando como uma alternativa no ensino. Conforme mostra o Gráfico 5, entre os artigos analisados, 8 utilizam técnicas de gamificação, seja em ambientes virtuais de aprendizagem, seja nos procedimentos adotados.

5) *Modalidade de Ensino - Presencial, A Distância ou Semipresencial*: A maioria dos estudos é voltado para o ensino presencial – 38 (93%); e apenas 1 é destinado a cursos a distância. Neste, um curso é proposto para ensino de programação na linguagem C, baseado em metodologias ativas e que inclui uma ferramenta para gerar representações de perfis dos estudantes com base nos códigos produzidos por eles. Foram encontrados também 2 estudos cujas metodologias compreendem algumas atividades presenciais e outras a distância, que objetivam complementar o aprendizado do estudante no tempo em que lhe for mais apropriado. O Gráfico 6 mostra os percentuais de cada modalidade.

6) *Envolvimento de outras disciplinas*: Conforme mostra o Gráfico 7, a maioria dos estudos aborda o ensino de progra-



Figura 5. Percentual de metodologias que utilizam atributos de gamificação.

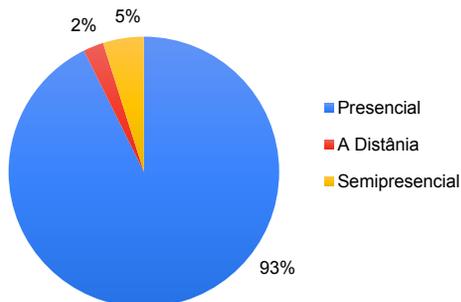


Figura 6. Percentual de metodologias aplicadas ao ensino presencial, a distância e semipresencial.

mação sem fazer relação com outras disciplinas curriculares. Entre aqueles que combinam o ensino de programação com outras disciplinas, a quase totalidade o faz com disciplinas das ciências exatas – matemática e física – 8 e 4 estudos, respectivamente. Um estudo aborda a biologia e 3 abordam várias disciplinas. Entretanto, ressaltamos que estes últimos casos não implicam em metodologias desenvolvidas em caráter interdisciplinar. Apenas uma ou outra atividade envolve conteúdos de outras disciplinas.

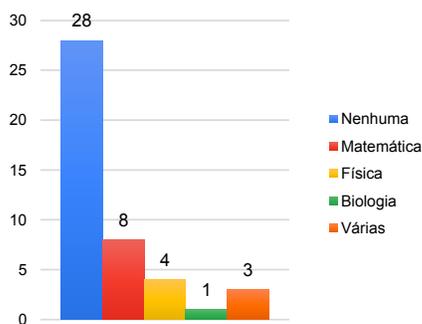


Figura 7. Número de estudos que envolveu outras disciplinas na metodologia proposta.

7) *Aspectos comportamentais*: Em alguns estudos analisados, aspectos comportamentais são levados em conta no planejamento das atividades. Por exemplo, metodologias que envolvem o desenvolvimento de jogos visam motivar os alunos. Atividades desenvolvidas em grupo valorizam a colaboração e a interatividade. A interatividade é o aspecto mais valorizado, sendo considerado em 14 artigos. O Gráfico 8 apresenta o número de ocorrências dos aspectos comportamentais citados. 14 estudos não se preocuparam em abordar aspectos comportamentais.

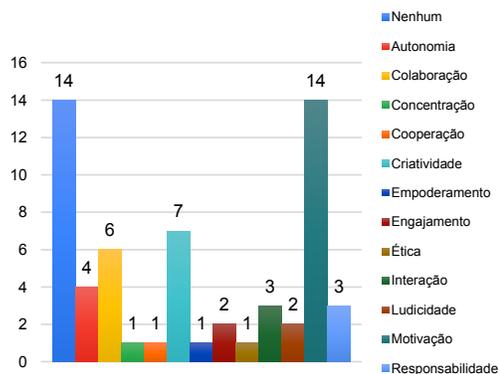


Figura 8. Número de estudos que focaram em aspectos comportamentais.

8) *Tecnologias não digitais*: Nesta seção apresentamos uma breve descrição das tecnologias não digitais mencionadas nos estudos. Algumas referem-se aos recursos utilizados na aplicação das metodologias. Outras descrevem a própria metodologia aplicada.

- *Metodologias centradas na aprendizagem do aluno*. Referem-se a metodologias de aprendizagem ativas orientadas a problemas experimentadas num curso a distância. A aprendizagem é direcionada nos seguintes eixos: (1) Conteúdos para ler, assistir e ouvir; (2) Atividades interativas para discutir, perguntar e ensinar; (3) Atividades para compreender problemas; (4) Atividades para criar programas; e (5) Atividades para revisar e depurar programas. A análise da produção dos estudantes é realizada a partir do mapeamento de seus perfis compostos de 348 métricas.
- *BOOTSTRAP: ÁLGEBRA*⁶. O estudo desenvolve uma metodologia baseada no currículo Bootstrap – um currículo de ciência da computação proposto por pesquisadores da Brown University (EUA). Consiste em 4 módulos, entre eles o Bootstrap: Álgebra, o qual se propõe a ensinar conceitos algébricos a estudantes de 12 a 16 anos através da programação de jogos. No estudo primário analisado, o autor utiliza programação funcional ao invés de imperativa.
- *Adaptação da metodologia LEGO*⁷. Segundo os autores, a metodologia sugerida no guia de utilização do kit

⁶<https://www.bootstrapworld.org/>

⁷<https://www.lego.com/pt-br/themes/mindstorms>

LEGO não leva em conta o desenvolvimento do pensamento computacional. Assim, eles propõem uma adaptação dessa metodologia. Infelizmente, não descrevem a metodologia no artigo.

- *Simulação Corporal*. Uma simulação corporal objeti-va representar conceitos com o próprio corpo. No estudo que propôs a atividade foram simulados jogos eletrônicos para representar e abstrair conceitos de funções: modularização de problemas, divisão de encargos para tarefas específicas e interação entre módulos do sistema.
- *Light-Bot Logicamente – Adaptação desplugada do LightBot*⁸. Jogo de tabuleiro inspirado no Lightbot - jogo eletrônico que se propõe a ensinar comandos básicos de programação com blocos, exercitando o raciocínio lógico nos aprendizes.
- *Peer Instruction*. Baseada na aprendizagem por pares, é uma metodologia que visa a interação entre os educandos que colaboram ajudando-se. No experimento, foram colocados na mesma turma estudantes que já sabiam programar e outros que não.
- *Jogos Boole*⁹. Jogos em forma de enigmas ou problemas que são postos através de histórias e resolvidos com cartas. São disponibilizados em revistas impressas.
- *Lobogames*¹⁰. Projeto da UFRGS que disponibiliza diversos materiais para impressão e uso livre em atividades de ensino – tabuleiros de jogos.
- *Metodologia Didático Simbólica*. Integra material didático adaptado para surdos e um ambiente visual de programação simbólica ViP.

9) *Conteúdos de programação abordados*: A maioria das propostas é voltada para o paradigma imperativo de programação. As exceções são a metodologia baseada no currículo Bootstrap (artigo 30), que utiliza programação funcional, e a abordagem baseada no ambiente de simulação de batalha de robôs Robocode (artigo 1) que, pela própria natureza da ferramenta, é voltada para programação orientada a objetos.

Apresentamos no Gráfico 9 os conteúdos conforme mencionados nos artigos e o número de artigos em que eles são citados. Sob o rótulo “Conceitos introdutórios” estão incluídos conceitos de: algoritmo, fluxograma, tipo de dados, variável, operadores aritméticos e lógicos. Os conteúdos referentes ao paradigma imperativo aparecem no gráfico na ordem em que normalmente são abordados em cursos de programação, por ordem de complexidade: Conceitos introdutórios, Estruturas sequencial, de seleção e de repetição, e assim sucessivamente.

A partir do Gráfico pode-se observar que 15 estudos não mencionaram os conteúdos abordados nas intervenções, o que nos impossibilita avaliar com precisão o escopo alcançado pelas metodologias nesse aspecto. Percebe-se também que 18 artigos relatam abordar estruturas de repetição enquanto apenas 10 relatam abordar estrutura sequencial e 15, estrutura de seleção. Isso pode significar uma falha no relato dos artigos

⁸<https://lightbot.com/>

⁹<https://jogosboole.com.br/>

¹⁰<http://www.inf.ufrgs.br/lobogames/>

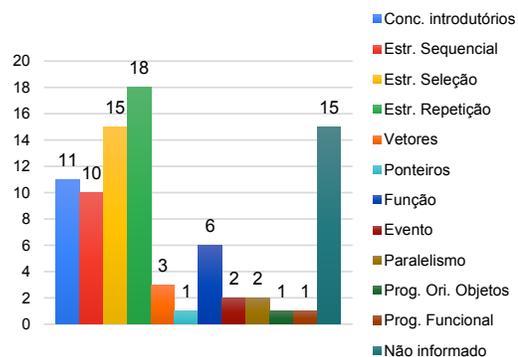


Figura 9. Número de estudos e os conteúdos de programação abordados

envolvidos. O fato de que muitas intervenções consistem em cursos livres ou oficinas de pequena carga horária, explica as baixas ocorrências de conteúdos mais complexos.

10) *Linguagens ou Ambientes de Programação Utilizados*: O Gráfico 10 apresenta as linguagens de programação utilizadas nas intervenções relatadas. Entre as abordagens não tradicionais, a abordagem mais utilizada é a programação em blocos que foi a alternativa adotada em 19 trabalhos. Assim como estas, as metodologias baseadas em ferramentas como Arduino, Robomind e LOGO, obviamente, utilizaram as linguagens específicas desses recursos. Entre as linguagens textuais tradicionais a utilização variou de 1 a 4 estudos. Destacamos aqui o ambiente Visual Programmer, integrante da proposta de metodologia de ensino de programação para surdos que dispõe de uma linguagem baseada em símbolos.

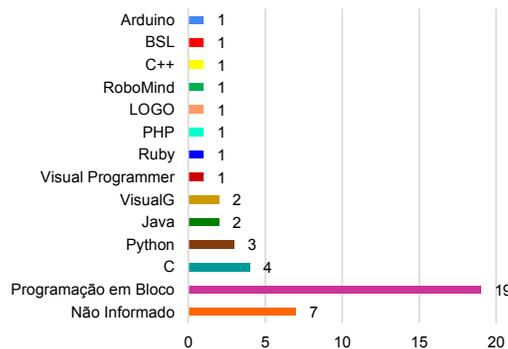


Figura 10. Linguagens/Ambientes de programação e o número de estudos que as utilizaram

B. Pensamento Computacional

Uma década e meia atrás, Jeanette Wing popularizou o conceito de “pensamento computacional” [17], [25], o qual reflete as ideias que Papert e Solomon lançaram em seu trabalho muitos anos antes [20]. Desde então, inúmeras definições têm sido propostas para o conceito de pensamento computacional e, de fato, não parece existir um consenso a respeito [1].

Neste trabalho, entendemos pensamento computacional como um dos eixos da área de computação conforme definido nas Diretrizes para ensino de Computação na Educação Básica elaboradas pela SBC [3], que se refere à “capacidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática, através da construção de algoritmos.” Nesse sentido, a elaboração de programas apresenta-se como uma competência intrinsecamente relacionada ao pensamento computacional.

Para elaborar a solução de um problema através de um algoritmo, é necessário lançar mão de técnicas e habilidades como decomposição (quebrar um problema complexo em partes menores – menos complexas), reconhecimento de padrões (identificar soluções conhecidas para problemas parecidos), generalização (tomar uma solução específica e estendê-la para casos mais gerais), análise (identificar se um problema é passível de solução computacional e, ainda, determinar o custo de uma solução e verificar se ela pode ser melhorada), abstração (desviar-se de distrações e focar apenas nas informações relevantes para a solução do problema).

Na elaboração de um programa com objetivo de resolver um problema, é necessário construir um algoritmo (um objeto abstrato) e então materializá-lo em forma de programa, considerando para isso as possibilidades e limitações dos sistemas computacionais. De fato, o pensamento computacional e a elaboração de programas estão imbricados de tal forma que não há como envolver-se nesta atividade sem empregar pensamento computacional.

Essa relação tem levado pesquisadores a propor metodologias de ensino de programação que direcionem esforços para, além de abordar os conteúdos específicos de programação diretamente, incluir a realização de tarefas que contribuam para o desenvolvimento do pensamento computacional dos estudantes. Para responder a QP2: *Em sua concepção, as metodologias buscam o desenvolvimento do pensamento computacional?* verificamos nos estudos analisados a existência (ou não) da intenção explícita de trabalhar o pensamento computacional dos educandos para além da construção de programas exclusivamente.

Conforme apresentado no Gráfico 11, 68% dos trabalhos visam o desenvolvimento do pensamento computacional, uma quantidade significativa de trabalhos.

Apesar de optarem por esta abordagem, não está muito claro na maioria dos estudos como se dá a transição do pensamento computacional para a habilidade de construir programas. Ou em que medida as atividades direcionadas ao desenvolvimento do pensamento computacional influenciam positivamente a aprendizagem de programação.

C. Teorias de Aprendizagem

Para responder à QP3: *As metodologias propostas são baseadas em alguma Teoria de Aprendizagem? Quais?* buscamos levantar as teorias que embasaram as metodologias propostas nos estudos primários.

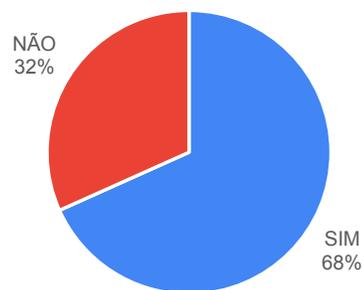


Figura 11. Percentual de estudos que focaram no desenvolvimento do pensamento computacional.

Surpreendentemente, dos 41 casos estudados, 29 (70,7%) não fazem referência a qualquer teoria de aprendizagem. Por outro lado, 12 trabalhos analisados (29,3%) apresentaram em suas propostas uma teoria de aprendizagem subjacente (Gráfico 12). Estes estudos foram fundamentados nas teorias de sete estudiosos que se encaixam, predominantemente, numa filosofia cognitivista. Um dos estudos apresenta como base uma teoria de aprendizagem delineada na perspectiva da filosofia humanista. A seguir são apresentados, de forma breve, os sete estudiosos e suas teorias correspondentes.

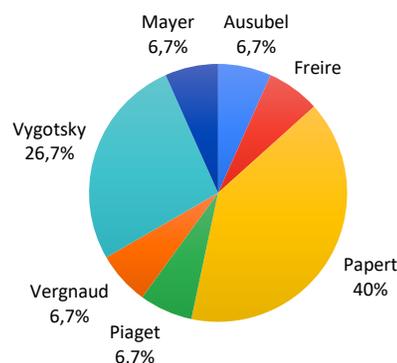


Figura 12. Teorias de aprendizagem por número de estudos fundamentados sobre elas.

- **David Ausubel e a teoria da aprendizagem significativa.** Embora reconheça a importância da experiência afetiva, Ausubel "propõe uma explicação teórica do processo de aprendizagem, segundo o ponto de vista cognitivista" [15]. Para Ausubel (1982), a aprendizagem significativa acontece quando uma informação se apoia em conceitos já existentes na estrutura cognitiva do aprendiz. Sua principal preocupação é como essa aprendizagem se dá no dia a dia, em sala de aula, e o que o aluno já sabe – chamado de subsunção – é fundamental para proporcionar novo aprendizado e deve ser identificado pelos docentes [15]. A teoria da aprendizagem significativa, tem uma grande contribuição do pesquisador Joseph. D. Novak,

responsável por elaborações, refinamentos e divulgações, ao ponto de figurar ao lado de Ausubel como proponente desta teoria de aprendizagem. Neste estudo as abordagens relativas à Ausubel-Novak representam 7% do total de trabalhos que apresentaram teorias de aprendizagem.

- **Paulo Freire** e suas **pedagogias** retratam uma proposição de ensino-aprendizagem baseado na problematização e no diálogo. A educação dialógica mostra-se como essência da educação como prática da liberdade [10]. Esse diálogo é iniciado na busca pelo conteúdo a ser trabalhado em sala de aula. O diálogo não é a discussão polêmica entre sujeitos com a tentativa de impor suas verdades – na percepção de Freire, "não há diálogo se não há um profundo amor ao mundo e aos homens" [15] – ressaltando, nesta sentença, a filosofia humanista das suas pedagogias. Ao destacar as questões de ensino e de aprendizagem em Freire, suas pedagogias refletem pensamentos de outros tantos teóricos da área, como Carl Rogers, Lev Vygotsky, David Ausubel e Gérard Vergnaud. As pedagogias freirianas também aparecem com um percentual de 7% do total de teorias estudadas neste documento.
- **Seymour Papert** - O construcionismo proposto por Seymour Papert em 1980 tem como meta encontrar formas de aprendizagem na construção mental do sujeito, ancorada em sua própria construção de mundo. Para Papert, o uso do computador leva o aluno à construção do seu próprio conhecimento. Papert foi o responsável pelo desenvolvimento da linguagem Logo que foi escrita no livro *Mindstorms - Children, Computers, And Powerful Ideas* [17] e tratada como uma filosofia, pois por trás daquela linguagem, estava representada uma concepção de escola, de educação, de aprendizagem e de ensino. O livro foi publicado nos EUA em 1980, e a versão em português saiu em 1985. Naquela época representou um salto para o uso da Informática na escola. Até hoje ainda se tem Papert como líder dos trabalhos desenvolvidos, fato que se comprova com este estudo onde 40% das produções foram desenvolvidas baseadas na teoria do construcionismo de Papert. A Linguagem Logo apresenta uma metáfora que bem representa a proposta de Papert: ao utilizar a linguagem Logo, o aprendiz ensina uma tartaruga a fazer os movimentos na tela do monitor, desenhando as formas que desejar. Quando o comando não é executado de maneira correta, há um *feedback* e o aprendiz é orientado a tentar novamente, refazendo o comando, de acordo com as orientações, até que a tartaruga faça o que ele deseja. Neste processo de ensinar a tartaruga e depurar os erros, quando ocorrem, refazendo o comando, o estudante elabora/constrói o conhecimento daquilo que lhe foi proposto a aprender.
- **Jean Piaget** é frequentemente lembrado por uma posição filosófica de que o conhecimento humano é uma construção do próprio homem. Embora esta visão seja muito antiga - a do construtivismo - a obra de Piaget é confundida com a própria teoria construtivista. Embora existam outras visões construtivistas, o enfoque dado por Piaget é, sem dúvida, o mais lembrado [15]. Na **Teoria do Desenvolvimento Cognitivo**, de Piaget, destacam-se os períodos de desenvolvimento mental (sensório-motor, pré-operacional, operacional-concreto e operacional-formal) e alguns conceitos-chaves (assimilação, acomodação e equilíbrio), sendo que estes últimos formam o chamado "núcleo-duro" da teoria de Piaget. Vale salientar que a teoria mencionada não representa uma teoria de aprendizagem, mas de desenvolvimento, uma vez que Piaget não concordava com a definição usada para aprendizagem, por não acreditar ser coerente com a sua teoria [15].
- **Gérard Vergnaud** - A **teoria dos campos conceituais** de Vergnaud é uma teoria psicológica cognitivista que mostra o estudo do desenvolvimento e da aprendizagem das competências complexas, Vergnaud [24]. Esta teoria supõe que o núcleo do desenvolvimento cognitivo é a conceitualização [15]. Moreira [15] resume campo conceitual como "um conjunto informal e heterogêneo de problemas, situações, conceitos, relações, estruturas, conteúdos e operações de pensamento conectados uns aos outros", mais adiante o autor refere-se ao fato de que, "como são as situações que dão sentido aos conceitos", pode-se definir campo conceitual naturalmente como sendo "um conjunto de situações". Ou seja, na perspectiva de Vergnaud, o estudante aprende a partir do contato com situações-problemas, dominando o campo conceitual progressiva e lentamente. A teoria dos campos conceituais aparece em 7% dos estudos analisados;
- **Lev Vygotsky** Os resultados das pesquisas do psicólogo russo indicavam que o desenvolvimento cognitivo do estudante, e sua aprendizagem, acontecem por meio da interação social, portanto o papel do professor nesse contexto é o de provocar essa interação – utilizando-se da linguagem como mediação dos processos de ensino-aprendizagem – onde ele aprende também com a produção do outro ou do seu grupo. Para os autores das publicações utilizadas neste trabalho, Vygotsky tem a preferência de 27% do total das produções que utilizaram alguma teoria de aprendizagem. O **sociointeracionismo** de Vygotsky é uma referência para estudos que buscam atividades que visam a colaboração nos processos de ensino-aprendizagem e buscam, nestas condições, utilizar os conceitos de zona de desenvolvimento potencial e proximal. Para atuar no processo de mediação, o professor deve identificar nos estudantes o estágio em que se encontram ao tratar de algum conteúdo específico - os alunos conseguem ou não resolver sozinhos os problemas propostos? Precisam de ajuda, indicações de caminhos a seguir? Ao responder estas questões, identifica-se as zonas de desenvolvimento real e potencial, trabalhando em seguida, planejando-se atividades, para utilizar a **Zona de Desenvolvimento Proximal**, que é definida como a diferença entre o nível de desenvolvimento real do indivíduo e o nível de desenvolvimento potencial [15].
- **Richard Mayer** A **Teoria Cognitiva da Aprendizagem**

Multimídia (TCAM), criada por Richard Mayer em 2001, justifica o aprendizado através da palavra - escrita ou em discurso e imagens — estáticas ou dinâmicas. As imagens estáticas podem estar representadas através de ilustrações e fotos e as imagens dinâmicas através de animações e vídeos. A multimídia compreende a combinação de vários recursos tecnológicos com o objetivo de fornecer a informação por meio de diversos formatos, atingindo várias áreas sensoriais [23]. A TCAM está baseada em três pressupostos: Capacidade limitada, Canal duplo e processamento ativo. A capacidade limitada é a limitação do ser humano em processar informações simultaneamente através dos canais (duplo) auditivo e visual. O processamento ativo é o recebimento das informações recebida organizando-as e convertendo em conhecimento. Para Santos e Tarouco [22], a utilização dos recursos multimídia interativos são vistos pelos docentes como um meio de estimular os estudantes para uma melhor aprendizagem. No entanto, para Schnotz e Lowe [23], o uso da multimídia precisa estar de acordo com a habilidade cognitiva dos estudantes e também do conteúdo que se deseja que ele aprenda.

D. Reflexos das metodologias no aprendizado de programação

As atividades de pesquisa que objetivam a criação de propostas para a melhoria dos processos de ensino-aprendizagem têm sido uma realidade em diversas áreas. Em Computação observa-se um crescente interesse em elaborar propostas que venham a contribuir para minimizar alguns aspectos que tornam o aprendizado de conteúdos considerados difíceis por suas características de abstração e complexidade. Programação entra no rol de conteúdos difíceis, responsáveis por retenção de estudantes dos conteúdos introdutórios e avançados.

Para responder à Q4: *Os estudos revelam melhoria na aprendizagem com a aplicação das metodologias propostas?* verificamos os dados disponíveis a respeito dos resultados obtidos com as intervenções.

Uma vez criada uma nova estratégia metodológica, cujo objetivo seja a melhoria dos processos de ensino-aprendizagem, espera-se que os resultados obtidos com a aplicação da metodologia proposta, deem bons resultados naquele contexto para o qual foi criada.

Assim, os resultados obtidos na avaliação dos estudos analisados surpreendem (ver Gráfico 13) por ainda apresentar o número de 46% de estudos que não avaliaram os resultados da aprendizagem com o uso da metodologia proposta. Embora o número de estudos que tenham avaliado os resultados da aprendizagem seja maior, 54%, esperava-se que houvesse uma preocupação com resultados concretos da proposta, com publicação de resultados também completos.

Um indicativo destes resultados pode ser a grande quantidade de trabalhos publicados em eventos (8 de 41, conforme Tabela III), o que pode expressar uma tendência de publicação de trabalhos com resultados parciais de pesquisas em

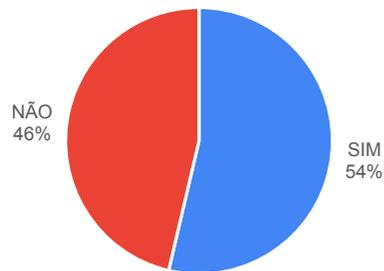


Figura 13. Percentual de estudos que relatam avaliação da aprendizagem de programação.

andamento, enquanto os trabalhos publicados em periódicos são trabalhos de pesquisa finalizados.

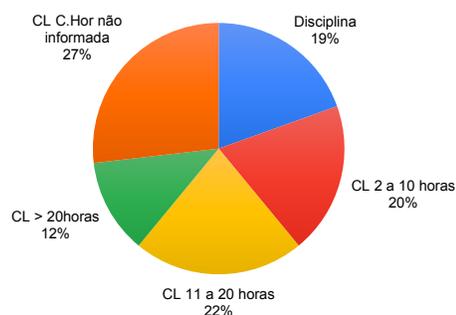


Figura 14. Ambiente de aplicação das metodologias – disciplinas formais ou cursos livres com cargas horárias.

Outro fator que pode dificultar a avaliação da aprendizagem é o fato de que 81% das intervenções foi realizada através de cursos livres ou oficinas enquanto que em apenas 19% a metodologia foi aplicada em disciplinas formais de programação. Adicionalmente, entre os cursos livres, apenas 12% teve carga horária superior a 20 horas (ver Gráfico 14).

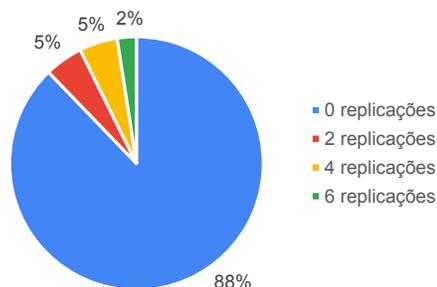


Figura 15. Número de replicações executadas das metodologias relatadas.

Os estudos analisados foram ainda avaliados quanto ao número de intervenções realizadas ou aplicações em situações

reais para avaliação da proposta metodológica. Neste sentido, os resultados apontaram para uma única intervenção, já considerando a aplicação única como válida para o relato da experiência (88%), enquanto que um total de 12% fizeram entre 2 e 6 intervenções (Gráfico 15).

V. CONSIDERAÇÕES FINAIS

O ensino de computação tem sido considerado um importante objeto de pesquisa para desenvolvimento da área como um todo. Tratar das questões de ensino-aprendizagem tem levado os pesquisadores da área de Ciência da Computação, com interesse no tema, a trilhar caminhos multidisciplinares para sua formação e interdisciplinares para melhor compreender a ação do professor de forma mais consciente em sala de aula - compreende dessa forma, a responsabilidade do seu ato educativo, e também formativo, para os estudantes que chegam aos espaços formais da educação.

Ao identificarmos a grande quantidade de artigos publicados que trazem "ensino-aprendizagem de programação" em seu título ou resumo (983 artigos), há uma indicação da preocupação que existe em tratar do tema, tornando uma discussão relevante e atual nos meios acadêmicos da área. Entretanto, ainda é incipiente uma preocupação com os princípios que fundamentam o objeto de pesquisa, colocando os estudos em patamares de menor profundidade teórica. Para ilustrar esta ideia, retomamos os números reportados na QP3 – dos 41 estudos analisados, 29 destes não fazem referência a uma teoria de aprendizagem que fundamenta seus trabalhos de pesquisa, carecendo de uma afiliação teórica para balizar as suas escolhas, o que resulta ainda em uma baixa compreensão da importância do trabalho proposto – sendo necessário buscar mais informações em trabalhos externos para compreender precisamente as escolhas feitas.

Os resultados encontrados neste mapeamento indicaram uma preferência por atividades plugadas, ou seja, com utilização de algum dispositivo eletroeletrônico, o que pode se justificar ao considerar a faixa etária do público-alvo e suas preferências na utilização destes, sendo possível considerar o acesso a tais recursos como algo relevante para a escolha. Justifica-se ainda o objetivo de utilizar uma metodologia que potencialize a ação do aprendiz, deslocando a centralidade das atividades de aprendizagem para o maior interessado no processo: o aluno – e considerando, por exemplo, os aspectos motivadores.

AGRADECIMENTOS

Agradecemos a Daniel Brasil Khouri, graduando de Matemática da UFBA, por sua valiosa colaboração na coleta de dados.

REFERÊNCIAS

- [1] Christian Puhlmann Brackmann. *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. PhD thesis, Universidade Federal do Rio Grande do Sul, 2017.
- [2] Alzira Ferreira da Silva. *RoboEduc: uma metodologia de aprendizado com robótica educacional*. PhD thesis, Universidade Federal do Rio Grande do Norte, 2009.

- [3] SBC Sociedade Brasileira de Computação. Diretrizes para o ensino de computação na educação básica. 2018.
- [4] Carina Machado de Farias, Felli Pereira Azevedo, and José Elias de Jesus Dias. Uma abordagem gamificada para o ensino de lógica de programação: relato de experiência. In *Anais do XXVI Workshop sobre Educação em Computação*, Porto Alegre, RS, Brasil, 2018. SBC.
- [5] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining “gamification”. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek '11, page 9–15, New York, NY, USA, 2011. Association for Computing Machinery.
- [6] Jefta Caldeira e Ana Vilela. Um mapeamento sistemático para auxiliar na escolha de plataformas ead para o ensino-aprendizagem de algoritmos e programação de computadores. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 27(1):52, 2016.
- [7] Anderson Marcolino e Ellen Barbosa. Softwares educacionais para o ensino de programação: Um mapeamento sistemático. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 26(1):190, 2015.
- [8] Draylson Souza e Marisa Batista e Ellen Barbosa. Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, 24(1):39, 2016.
- [9] Lucas Blatt e Valdecir Becker e Alexandre Ferreira. Mapeamento sistemático sobre metodologias e ferramentas de apoio para o ensino de programação. *Anais do WIE 2017*, 23(1):815, 2017.
- [10] Paulo Freire. *Pedagogia do oprimido*. São Paulo: Paz e Terra, 2018.
- [11] Raphael Magalhães Hoed. Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação. Master's thesis, Universidade de Brasília, 2015.
- [12] Caitlin Kelleher and Randy Pausch. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.*, 37(2):83–137, June 2005.
- [13] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University and Durham University Joint Report, 2007.
- [14] Sílvia Maria Manfredi. Metodologia do ensino - diferentes concepções. e-Disciplinas USP. Versão preliminar.
- [15] Marco Antonio Moreira. *Teorias da Aprendizagem*. EPU, 1999.
- [16] Chitu Okoli, Traduzido por: David Wesley Amado Duarte, and Revisão técnica e introdução: João Mattar. Guia para realizar uma revisão sistemática de literatura. *EaD em Foco*, 9(1), abr. 2019.
- [17] Seymour Papert. *Mindstorms - Children, Computers, And Powerful Ideas*. BasicBooks, 1980.
- [18] Seymour Papert. *Constructionism: A new opportunity for elementary science education*. Massachusetts Institute of Technology, Media Laboratory, Epistemology and . . . , 1986.
- [19] Seymour Papert and Idit Harel. *Constructionism: research reports and essays, 1985-1990*. Ablex Pub. Corp, 1991.
- [20] Seymour Papert and Cynthia Solomon. Twenty things to do with a computer. Technical report, Massachusetts Institute of Technology, 1971.
- [21] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, page 68–77, Swindon, GBR, 2008. BCS Learning Development Ltd.
- [22] Leila Maria Araújo Santos and Liane Margarida Rockenbach Tarouco. A importância do estudo da teoria da carga cognitiva em uma educação tecnológica. In *Revista Novas Tecnologias*, volume 5, 2007.
- [23] Wolfgang Schnotz and Richard Lowe. External and internal representations in multimedia learning. *Learning and Instruction*, 13(2):117 – 123, 2003. External and Internal Representations in Multimedia Learning.
- [24] Gérard Vergnaud. La théorie des champs conceptuels. In *Recherches em Didactique des Mathématiques*, volume 10, pages 133–170, 1990.
- [25] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

VI. APÊNDICE

ESTUDOS PRIMÁRIOS SELECIONADOS PARA ANÁLISE

- 1) ANDRADE, Gil; HOSS, Diego ; BARBOSA, Ana ; GOMES, Lana . Metodologia Didático Simbólica como Alternativa para o Ensino de Programação de Computadores a Alunos Surdos. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27. , 2019, Belém. Anais do XXVII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2019 . p. 473-482. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2019.6652>.
- 2) BATISTA, Esteic Janaina Santos et al. Uso do Scratch no ensino de programação em Ponta Porã: das séries iniciais ao ensino superior. Anais do Workshop de Informática na Escola, [S.l.], p. 565, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/6863>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.565>.
- 3) BATISTA, Washington; CHALEGRE, Henderson; SENA, João Paulo; SANTOS, Allen; DOS SANTOS, David; RODRIGUES, Carlos; BITTENCOURT, Roberto. Oficinas de Aprendizagem de Programação em uma Escola Pública através do Ambiente Scratch. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 23. , 2015, Recife. Anais do XXIII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2015 . p. 306-315. ISSN 2595-6175.
- 4) BORDIN, Andrea; QUEPFERT, Wolgan. Projeto de ensino de programação para alunos de ensino médio: Uma análise do cenário e das percepções das oportunidades. Anais do Workshop de Informática na Escola, [S.l.], p. 205, out. 2018. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/7889>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2018.205>.
- 5) CAMBRUZZI, Eduardo; SOUZA, Rosemberg Mendes de. Robótica Educativa na aprendizagem de Lógica de Programação: Aplicação e análise.. Anais do Workshop de Informática na Escola, [S.l.], p. 21, out. 2015. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/4981>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2015.21>.
- 6) CÂNDIDO, Daniel et al. Estudo Comparativo de Abordagens Referentes ao Desenvolvimento do Pensamento Computacional. Anais do Workshop de Informática na Escola, [S.l.], p. 382, out. 2017. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/7256>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2017.382>.
- 7) Cavadas, Ana Fernandes, Raquel Teixeira, Leniah Motta, Claudia Marques, Carla de Oliveira, Carlo Emmanoel. (2019). O desenvolvimento de games como metodologia de ensino-aprendizagem e estratégia para promover o protagonismo juvenil. 853. 10.5753/cbie.sbie.2019.853.
- 8) CHAVES, Letícia Saraiva; SILVA, Abilio e; BEZERRA, Carla Ilane Moreira. Ensino de Programação em Escolas Públicas: Relato de uma Ação do PET-TI. Anais do Workshop de Informática na Escola, [S.l.], p. 667, nov. 2019. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/8562>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2019.667>.
- 9) DE FARIAS, Carina Machado; AZEVEDO, Fellipe Pereira; DIAS, José Elias de Jesus. Uma Abordagem Gamificada para o Ensino de Lógica de Programação: relato de experiência. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 26. , 2018, Natal. Anais do XXVI Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2018 . ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2018.3493>.
- 10) DE FARIAS, Carina Machado; DE OLIVEIRA, Anderson S.; SILVA, Everton Dias de A.. Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 26. , 2018, Natal. Anais do XXVI Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2018 . ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2018.3491>.
- 11) DE OLIVEIRA FERNANDEZ, Cassia et al. Uma proposta baseada em projetos para oficinas de Internet das Coisas com Arduino voltadas a estudantes do Ensino Médio. RENOTE-Revista Novas Tecnologias na Educação, v. 13, n. 2, 2015.
- 12) DE OLIVEIRA, Márcia Gonçalves et al. Um curso de programação a distância com metodologias ativas e análise de aprendizagem por métricas de software. RENOTE-Revista Novas Tecnologias Na Educação, v. 15, n. 1, 2017.
- 13) FERREIRA, Máverick; DE LIMA, Jakeline; SANTOS, Hígor; OLIVEIRA, Edjano; DO NASCIMENTO, Josefa. Computação para Ensino Médio na Modalidade Semipresencial: Uma Experiência da Disciplina de Estágio Supervisionado. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 23. , 2015, Recife. Anais do XXIII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2015 . p. 406-415. ISSN 2595-6175.
- 14) GARCIA, Victor Ronchi et al. Despertando Jovens Talentos com o Conhecimento da Computação. Anais do Workshop de Informática na Escola, [S.l.], p. 583, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/6865>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.583>.
- 15) GODINHO, Julia; TORRES, Kathleen; BATISTA, Gabriel; ANDRADE, Eduardo; GOMIDE, Janáina. Projeto Aprenda a Programar Jogando: Divulgando a Programação de Computadores para Crianças e Jovens. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 25. , 2017, São Paulo. Anais do XXV Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2017 . ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2017.3553>.
- 16) KARLING, Daniel Antonio; MONTE-ALTO, Helio. Investigação do Currículo Bootstrap no Ensino de Matemática por meio da Programação de Computadores. Anais do Workshop de Informática na Escola, [S.l.], p. 392, out. 2017. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/7257>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2017.392>.
- 17) LIMA, Priscila; VIEIRA, Paulo; BRANDÃO, Leônidas. Ensino de algoritmos, programação e matemática: panorama e estudo de caso para estudantes de escolas públicas brasileiras. Anais do Workshop de Informática na Escola, [S.l.], p. 697, nov. 2019. ISSN 2316-6541. Disponível em: <<https://www.brie.org/pub/index.php/wie/article/view/8565>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2019.697>.
- 18) LOPES, Alexandre; SANTANA, Thalia ; BRAGA, Adriano . Um relato de experiência sobre o ensino de programação de computadores no Ensino Básico por meio da Olimpíada Brasileira de Informática. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27. , 2019, Belém. Anais do XXVII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2019 . p. 151-160. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2019.6626>.
- 19) MARQUES, Wagner dos Santos; DE SOUZA, Paulo Silas Severo; MOMBACH, Jaline Gonçalves. Pensar para Programar: Projeto de Ensino no Curso Técnico em Informática. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO

- (WEI), 25. , 2017, São Paulo. Anais do XXV Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2017 . ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2017.3550>.
- 20) MATOS, Geisiane Souza et al. Desenvolvimento de Jogos para Incentivar Meninas na Área da Tecnologia de Informação: um estudo de caso no ensino médio de escola pública. Anais do Workshop de Informática na Escola, [S.l.], p. 715, nov. 2019. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/8567>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2019.715>.
- 21) MATTOS, Giorgia; DA SILVA, Danielle; MOREIRA, Josilene. A Utilização de Kits de Robótica como Ferramenta para o Ensino de Programação à Meninas do Ensino Médio. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 23. , 2015, Recife. Anais do XXIII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2015 . p. 376-385. ISSN 2595-6175.
- 22) MONTEIRO, David et al. Uma Experiência do Uso Do Hardware Livre Arduino no Ensino De Programação De Computadores. Anais do Workshop de Informática na Escola, [S.l.], p. 51, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/6602>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.51>.
- 23) NATAL, Maria Eduarda Contri et al. Tri-Logic: Um Ambiente Gamificado como Ferramenta de Auxílio ao ensino de aprendizagem de Lógica de Programação. RENOTE-Revista Novas Tecnologias na Educação, v. 16, n. 2, p. 41-50, 2018.
- 24) OLIVEIRA, Manassés Vitorino; RODRIGUES, Luciene Cavalcanti; QUEIROGA, Ana. Material didático lúdico: uso da ferramenta Scratch para auxílio no aprendizado de lógica da programação. Anais do Workshop de Informática na Escola, [S.l.], p. 359, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/6842>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.359>.
- 25) OLIVEIRA, Maria Angélica Figueiredo et al. Aplicação do método Peer Instruction no ensino de Algoritmos e programação de computadores. RENOTE-Revista Novas Tecnologias na Educação, v. 15, n. 1, 2017.
- 26) PANTALEÃO, Eliana; AMARAL, Laurence Rodrigues; BRAGA E SILVA, Gláucia. Uma abordagem baseada no ambiente Robocode para ensino de programação no Ensino Médio. Revista Brasileira de Informática na Educação, [S.l.], v. 25, n. 03, p. 95, out. 2017. ISSN 2317-6121. Disponível em: <<https://www.br-ie.org/pub/index.php/rbie/article/view/6452>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/rbie.2017.25.03.95>.
- 27) POLONI, Leonardo; DO SACRAMENTO SOARES, Eliana Maria; WEBBER, Carine G. PENSAMENTO COMPUTACIONAL NO ENSINO MÉDIO: PRÁTICAS MEDIADORAS UTILIZANDO A LINGUAGEM SCRATCH. RENOTE-Revista Novas Tecnologias na Educação, v. 17, n. 3, 2019.
- 28) RAMOS, Felipe; TEIXEIRA, Lilian da Silva. Significação da Aprendizagem Através do Pensamento Computacional no Ensino Médio: uma Experiência com Scratch. Anais do Workshop de Informática na Escola, [S.l.], p. 217, out. 2015. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/5024>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2015.217>.
- 29) RAMOS, Nadja; FREITAS, Carmen; AVILA, Sandra; COSTA, Paula; TESTONI, Vanessa; BORIN, Juliana. Ensino de Programação para Alunas de Ensino Médio: Relato de uma Experiência. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 23. , 2015, Recife. Anais do XXIII Workshop sobre Educação em Computação. Porto Alegre: Sociedade Brasileira de Computação, July 2015 . p. 386-395. ISSN 2595-6175.
- 30) RIBEIRO, Fabrício; MERLIN, Bruno; FÜLBER, Heleno. Avaliação do impacto de ambientes gamificados no processo de ensino-aprendizagem de programação de computadores: uma comparação entre elementos monousuário e multiusuário. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 803, nov. 2019. ISSN 2316-6533. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/8808>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.sbie.2019.803>.
- 31) RIBEIRO, Juliana Perez; MANSO, Marina Andrade; BORGES, Marcos. Dinâmicas com App Inventor no Apoio ao Aprendizado e no Ensino de Programação. Anais do Workshop de Informática na Escola, [S.l.], p. 271, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/6645>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.271>.
- 32) SANTANA, Beatriz Silva de; PEREIRA, Claudia Pinto. Aproximação de alunas do ensino básico do pensamento computacional: relato de experiência de uma oficina de eletrônica. Anais do Workshop de Informática na Escola, [S.l.], p. 69, nov. 2019. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/8493>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2019.69>.
- 33) SANTOS, Jarles Gomes; SANTOS, Jaian. Primeiro contato com a programação através do Software Scratch: experiência no ensino técnico. Anais do Workshop de Informática na Escola, [S.l.], p. 362, out. 2017. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/7254>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2017.362>.
- 34) SILVA, Laíza; CARVALHO, Darlinton. Saberes Davó: Uma Abordagem para o Ensino de Programação no Ensino Médio. Anais do Workshop de Informática na Escola, [S.l.], p. 1, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/6597>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.1>.
- 35) SILVA, Leonardo Soares e; CAVALCANTI, Elmano Raimundo. Avaliação Experimental do Robomind no Ensino de Programação com Estudantes do Curso Técnico em Informática Integrado ao Ensino Médio. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 288, out. 2018. ISSN 2316-6533. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/7983>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.sbie.2018.288>.
- 36) SILVA, Thiago Reis da et al. Investigando dois formatos de videoaulas de programação de jogos digitais para alunos do ensino médio. Anais do Workshop de Informática na Escola, [S.l.], p. 187, out. 2015. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/5019>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2015.187>.
- 37) SILVA, Thiago Reis et al. Um relato de experiência da aplicação de videoaulas de programação de jogos digitais para alunos da educação básica. Anais do Workshop de Informática na Escola, [S.l.], p. 141, nov. 2016. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/6611>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2016.141>.
- 38) SOUZA, Daniella et al. Lightbot Logicamente: um game lúdico amparado pelo Pensamento Computacional e a Matemática. Anais do Workshop de Informática na Escola, [S.l.], p. 61, out. 2018. ISSN 2316-6541. Disponível em: <<https://www.br-ie.org/pub/index.php/wie/article/view/7874>>. Acesso em: 09

-
- maio 2020. doi:<http://dx.doi.org/10.5753/cbie.wie.2018.61>.
- 39) SOUZA, Isabelle Maria Lima de; RODRIGUES, Rivanilson da Silva; ANDRADE, Wilkerson. Explorando Robótica com Pensamento Computacional no Ensino Médio: Um estudo sobre seus efeitos na educação. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 490, nov. 2016. ISSN 2316-6533. Disponível em: <<https://www.brie.org/pub/index.php/sbie/article/view/6730>>. Acesso em: 09 maio 2020. doi:<http://dx.doi.org/10.5753/cbie.sbie.2016.490>.
- 40) VASCONCELLOS, Isadora Lopes Barbosa; TAMARIZ, Anabell Del Real; BATISTA, Sílvia Cristina Freitas. Planejamento, desenvolvimento e avaliação de um Ambiente Virtual de Aprendizagem gamificado. RÊNTE-Revista Novas Tecnologias na Educação, v. 17, n. 1, p. 21-30, 2019.
- 41) VENTORINI, André Eduardo; FIOREZE, Leandra Anversa. Funções e programação no Scratch. RÊNTE-Revista Novas Tecnologias na Educação, v. 16, n. 2, p. 576-585, 2018.