

---

# Proposta de Aprendizagem Integrada de Matemática e Programação com Abordagens do Pensamento Computacional no Jogo Robocode

## Integrated Learning Proposal for Mathematics and Programming with Approaches to Computational Thinking in the Robocode Game

---

**MATHEUS CARVALHO MEIRA**

Universidade Estadual de Campinas - UNICAMP

**KARLAN RICOMINI ALVES**

Instituto Federal São Paulo – IFSP

**LUIS ANTONIO TAVARES**

Universidade Estadual de Campinas - UNICAMP

**RAFAEL FERNANDO DIORIO**

Instituto Federal São Paulo – IFSP

**SERGIO FERREIRA DO AMARAL**

Universidade Estadual de Campinas - UNICAMP

**Resumo:** Estudantes das novas gerações, imersos num contexto digital, podem demonstrar desmotivações quando inseridos somente em ambientes considerados como tradicionais para ensino de disciplinas como matemática. Planejar ambientes com jogos digitais podem ser uma estratégia lúdica para ensino de matemática e programação de computadores. A presente proposta apresenta desafios no ambiente do jogo educacional com simulador de robôs, Robocode. Os desafios foram construídos a partir das abordagens do Pensamento Computacional e apresentam aportes entre conceitos matemáticos e de programação de computadores para determinar estratégias de comportamento e a programação dos robôs dentro do jogo. Os conteúdos pertinentes à matemática estão relacionados ao ensino e aprendizagem do plano cartesiano e são integrados à linguagem de programação com conceitos introdutórios na orientação à objetos. Os resultados apontam que a integração das disciplinas em um ambiente de jogo, devidamente planejado, pode apoiar o ensino e aprendizagem dos conteúdos matemáticos e computacionais de modo lúdico.

**Palavras-chave:** Pensamento Computacional. Ensino e Aprendizagem. Robocode. Matemática. Linguagem de Programação.

**Abstract:** Students of new generations, immersed in a digital context, can demonstrate demotivation when inserted only in environments considered as traditional for teaching subjects such as mathematics. Planning environments with digital games can be a playful strategy for teaching mathematics and computer programming. The present proposal presents challenges in the educational game environment with robot simulator, Robocode. The challenges were built

---

from the approaches of Computational Thinking and present contributions between mathematical concepts and computer programming to determine behavior strategies and the programming of robots within the game. The contents relevant to mathematics are related to the teaching and learning of the Cartesian plane and are integrated into the programming language with introductory concepts in object orientation. The results show that the integration of disciplines in a game environment, properly planned, can support the teaching and learning of mathematical and computational content in a playful way.

**Keywords:** Computational Thinking. Teaching and Learning. Robocode. Mathematics. Programming Language.

MEIRA, Matheus Carvalho; ALVES, Karlan Ricomini; TAVARES, Luís Antônio; DIORIO, Rafael Fernando; AMARAL, Sérgio Ferreira. Proposta de Aprendizagem Integrada de Matemática e Programação com Abordagens do Pensamento Computacional no Jogo Robocode. *Informática na Educação: teoria & prática*, Porto Alegre, v. 23, n. 3, p. 100-117, set./dez. 2020.

## 1 Introdução

Métodos de ensino e aprendizagem com envolvimento de jogos digitais educacionais fazem parte de muitas das ações desenvolvidas atualmente para incentivar o aprendizado de conteúdos presentes nas áreas da matemática e da computação. A robótica e o Pensamento Computacional são tidas como ótimas ferramentas para o desenvolvimento da pedagogia da ciência, tecnologia, engenharia e matemática (PLAZA et al., 2019). Questões matemáticas e computacionais, em especial as linguagens de programação, são assuntos de distintas disciplinas das quais muitos temas podem ser unificados e estudados de modo integrado. Os jogos educacionais, por sua vez, tendem a estimular a atração dos jovens, cercados em tecnologia a cada instante mais presente em seus cotidianos. A programação tem o potencial para despertar a criatividade daqueles que fazem seu uso (VALENTE, 1999). Estudantes inscritos nas disciplinas de linguagem de programação ou mesmo admiradores das tecnologias, em muitos casos, têm o desejo de compreender a lógica empregada nas aplicações contemporâneas.

A presente abordagem pretende exibir e discutir contextos significativos para incentivo na aprendizagem de conteúdos matemáticos com apoio da linguagem de programação entre estudantes de cursos técnicos integrados de informática. Dentre essas abordagens encontra-se o Pensamento Computacional, como uma forma de pensar de modo estratégico e organizado sobre os problemas, valorizando diferentes formas de investigação e expressão, em especial aqueles inerentes à computação.

Como é possível promover a aprendizagem matemática quando se aprende concepções na programação de computadores? As hipóteses salientam o desenvolvimento do Pensamento Computacional e na análise de ideias para determinações de resoluções de problemas a partir de questões abordadas dentro de jogos educacionais. Neste contexto, destacar situações que trabalhem estratégias de jogos para promover o ensino e aprendizagem de conteúdos matemáticos com base em linguagem de programação. Investigação da ocorrência do desenvolvimento do raciocínio condicional quando se aprende a programar e aplicar no desenvolvimento de conceitos matemáticos.

“A utilização das tecnologias de ensino e aprendizagem é um tema recorrente dentro da comunidade de educadores brasileiros, em particular dos educadores matemáticos.” (BORBA; LACERDA, 2015, p. 490). No âmbito do conhecimento matemático existem trabalhos correlatos que apontam para o desenvolvimento de jogos eletrônicos, a partir do *software* Scratch<sup>1</sup>, e suas relações na contribuição dos processos de construção do conhecimento matemático em alunos do curso fundamental (SÁPIRAS; VECCHIA; MALTEMPI, 2015).

O aprendizado de linguagem de programação, citando a área da Computação, pode ser um meio alternativo para desenvolvimento de disciplinas que envolvam a Educação Matemática. O estudo apontado por Moraes et al. (2017) constatou que aprender a programar fomenta a aprendizagem de matemática.

O trabalho consistiu na prática de dinâmicas de aprendizagem apoiado na associação do Pensamento Computacional, com jogos educacionais digitais, para propor práticas a partir de um ambiente estabelecido nos conceitos da aprendizagem de linguagem de programação com suporte da matemática. Os conceitos a serem desenvolvidos na área da matemática foram focados à temática do Sistema de Coordenadas Cartesianas, conhecido como Plano Cartesiano, para determinação de estratégias com base nos pontos nos eixos verticais e horizontais que compõem o cenário do jogo. O jogo educacional digital Robocode integra o ambiente das aplicações seguidas nas abordagens do Pensamento Computacional. O Robocode consiste em um jogo com envolvimento de linguagem de programação com objetivo programar comportamentos de um robô virtual, mesclado por um ambiente de desenvolvimento, que permite ensinar linguagem de programação (Java e C#) de modo lúdico (KUCCHAR et al., 2019).

O objetivo principal deste trabalho é desenvolver práticas, com abordagens do Pensamento Computacional, que envolvam conceitos matemáticos para estratégias no desenvolvimento de jogos com linguagem de programação no Robocode. Os conceitos específicos à matemática estão relacionados na aprendizagem do plano cartesiano.

O aumento da retenção dos estudantes nos primeiros anos dos cursos técnicos integrados de informática retrata pensamentos de preocupações nas Instituições Ensino (IE). É possível apontar para o Instituto Federal de Educação Ciência e Tecnologia de São Paulo (IFSP), ações como Programa de Auxílio de Permanência (PAP) que busca combater situações de retenção ou evasão. Contudo, quando se tratam de ações pedagógicas, estudos desenvolvidos pela FEPESP (2011) indicam a existência poucas iniciativas, principalmente em disciplinas relacionadas à área de exatas, como a matemática e a linguagem de programação, das quais demonstram elevadas taxas de retenção. A retenção por conteúdo ocorre no momento em que estudantes apresentam dificuldades de assimilação e podem provocar a evasões escolares (IFSP, 2019).

---

<sup>1</sup> Ambiente desenvolvido no MIT (*Massachusetts Institute of Technology*) Media Lab com objetivo de apoiar o ensino de lógica de programação com bases construcionistas, de modo a estimular a criatividade, raciocínio sistêmico e o trabalho colaborativo. Possui interface com blocos de comandos que devem ser arrastados e encaixados para formar a sintaxe dos programas que podem ser representados em jogos, animações e histórias interativas compartilhadas em comunidades *online* (MALONEY; RESNICK; RUSK, 2010; SCRATCH, 2021).

Indicar alternativas em relação aos métodos tradicionais de ensino e mesclar variados métodos, ensino e aprendizagem, que possam estimular os estudantes a conquistarem os objetivos propostos nos objetivos do plano da disciplina são pontos que orientam o presente trabalho. As IEs podem conquistar prerrogativas com os avanços da tecnologia e dos *softwares* relacionados à educação para contribuir com métodos de ensino e aprendizagem. Neste sentido, podem desfrutar de *softwares* educacionais, como jogos digitais, associados com processos para criar universos de aprendizagem em disciplinas de linguagem de programação com aportes dos conceitos da matemática.

A Lei de Diretrizes e Bases da Educação Nacional (LDB, Lei nº 9.394/1996) dispõe de uma normativa atualizada em 2017 denominada BNCC (Base Nacional Comum Curricular), que “define o conjunto orgânico e progressivo de aprendizagens essenciais que todos os alunos devem desenvolver ao longo das etapas e modalidades da Educação Básica” (BNCC, 2017, p. 7). Aspecto intrínseco na presente pesquisa consiste em aproximar intervenções metodológicas às orientações fundamentadas na BNCC (Base Nacional Comum Curricular), em específico das trilhas de aprendizagens suportadas na matemática. A BNCC (2017, p. 9) dispõe de características no exercício da “curiosidade intelectual [...], incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas”. Essa disposição observada na orientação da BNCC privilegia as relações entre o pensamento computacional e as metodologias de aprendizagens matemáticas.

## 2 Pensamento Computacional e Aprendizagem em Matemática

O pensamento computacional começa a ser idealizado na obra de Papert (1996) e seu esforço para utilizar os computadores no processo de aprendizagem das pessoas. Para Papert, “os computadores deveriam ser utilizados para que as pessoas pudessem “pensar com” as máquinas e “pensar sobre” o próprio pensar”. Papert propôs utilizar os computadores na educação antes dos computadores serem populares e estarem presentes nos lares, isso reflete a visão ambiciosa e visionária deste pesquisador (VALENTE, 2016) .

Já para Wing (2006 apud VALENTE, 2016) o pensamento computacional é limitado ao poder computacional e de seus processos. O “pensamento computacional se baseia no poder e nos limites dos processos computacionais, quer eles sejam executados por um ser humano ou por uma máquina”. Isso limita o pensamento computacional, o que não se aplica se utilizamos um conceito mais amplo, onde a abstração e outras características não são limitadas apenas aos processos e ao poder computacional, desvinculando o pensamento computacional do próprio computador e o relacionando a competências e habilidades que são desenvolvidas no ser humano.

No trabalho de Mannila et al (2014) temos a apresentação de nove conceitos que dão base ao pensamento computacional. A autora dividiu em três grupos esses conceitos, primeiro grupo: coleção de dados, análise de dados e representação de dados, nesta parte através dos dados

---

compreendemos o cenário que está sendo trabalhado, seja para solucionar um problema ou para aprender novos conteúdos não aplicados na vida cotidiana. No segundo grupo: decomposição de problemas, algoritmo e abstração, se trabalha a decomposição de problemas, isso pode levar ao desenvolvimento de algoritmos que necessariamente leva o desenvolvimento da abstração. E, terceiro grupo: simulação, automação e paralelização, aqui trabalhamos com a visualização da informação, sendo que a paralelização pode ajudar a chegar num mesmo objetivo percorrendo caminhos diferentes.

De modo prático, o pensamento computacional destina-se ao desenvolvimento de habilidades e competências. O pensamento computacional consiste em uma habilidade para resolver problemas, pensamos que esse é um dos cenários que o pensamento computacional pode ajudar as pessoas. Mas, não podemos limitá-lo a ser somente um instrumento de solução de problemas. Pois, como ainda é um tema carente de uma definição precisa, e ainda está em desenvolvimento. Sendo que o pensamento computacional pode expandir e contribuir com diversas áreas do conhecimento além da possibilidade de aplicação em diversos contextos sociais. Contudo, podemos afirmar que o pensamento computacional pretende desenvolver as capacidades humanas, sendo uma possibilidade de aplicação à educação, que é em essência o grande transformador de vidas e realidades (BARR; STEPHENSON, 2011).

*International Society for Technology in Education (ISTE)* e a *American Computer Science Teachers Association (CSTA)* (2011) identificaram nove conceitos para o Pensamento Computacional, dentre eles: (1) coleta de dados; (2) análise de dados; (3) representação de dados; (4) decomposição de problema; (5) abstração; (6) algoritmos; (7) automação; (8) paralelização e simulação.

O pensamento computacional consiste no processo de resolução de problema, com as seguintes características: formulação de problemas de uma forma que permita usar um computador e outras ferramentas para ajudar a resolvê-los; organização lógica e análise de dados; representação de dados através de abstrações como modelos e simulações; automação de soluções através do pensamento algorítmico (a série de passos ordenados); identificação, análise e implementação de soluções possíveis com o objetivo de alcançar a mais eficiente e efetiva combinação de etapas e recursos; e generalização e transferência desse processo de resolução de problemas para uma ampla variedade de problemas (CSTA; ISTE, 2011).

Ao tentar definir uma cultura de sala de aula que fosse mais propícia ao pensamento computacional, os participantes identificaram estratégias ou características que poderiam ser consideradas benéficas em toda a experiência de aprendizagem. Estes incluíram: (1) Maior uso por professores e alunos de vocabulário computacional, quando apropriado, para descrever problemas e soluções; (2) Aceitação por professores e alunos de tentativas de solução fracassadas, reconhecendo que a falha precoce muitas vezes pode colocá-lo no caminho para um resultado bem sucedido. Trabalho em equipe dos alunos, com uso explícito de: (1) Decomposição - quebrar problemas em partes mais pequenas que podem ser mais facilmente resolvidas, (2) Abstração - simplificando do concreto ao geral à medida que as soluções são desenvolvidas; (3) Negociação - grupos dentro da equipe trabalhando juntos para fundir partes da solução no todo,

---

e (4) Construção de consenso - trabalhando para construir a solidariedade coletiva por trás de uma ideia ou solução (BARR; STEPHENSON, 2011).

A aproximação da computação com a matemática é um caminho possível para a aplicação do pensamento computacional no ensino desta disciplina. Torna-se importante ressaltar que o pensamento computacional no seu âmago trabalha o desenvolvimento de capacidades, habilidades e competências para que as pessoas ampliem o "aprender a aprender". Sendo um fator decisivo para o processo de aprendizagem as perguntas que o indivíduo faz para determinar o caminho para a solução de problemas. Na simulação temos a possibilidade de permitir o erro durante a execução. Na coleção de dados através de recursos computacionais podemos filtrar e comparar diversos cenários. Através de algoritmo podemos propor soluções de diversas formas, utilizando da paralelização para compor soluções (NAKAMURA; KAWASAKI, 2019; PIRES et al., 2019).

No Brasil, no instante em que se destaca as trilhas matemáticas, torna-se pertinente analisar as orientações dispostas na BNCC (Base Nacional Comum Curricular), no aspecto para utilizar as tecnologias no apoio à formação dos estudantes. As orientações da BNCC encorajam trabalhos a partir dos conceitos presentes no pensamento computacional e, associações às competências, como as trilhas de aprendizagem matemáticas, para o desenvolvimento de projetos e atividades que proporcionem interações entre os estudantes com o uso de tecnologias digitais (LEAL; MALTEMPI, 2020). Pode ser notado, de modo claro, no texto normativo da BNCC, a inserção do pensamento computacional: "compreender, utilizar e criar Tecnologias Digitais de Informação e Comunicação (TDIC) de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva" (BNCC, 2017, p. 9).

As características intrínsecas do pensamento computacional, citando a formulação de problemas ou modelos de abstrações, podem privilegiar unidades, como os saberes, nos processos de aprendizagem matemática. O ambiente de formação em matemática pode ser compreendido nas experiências tecnológicas, científicas e também nos processos que envolvem a criatividade. Portanto, de acordo com as novas regulamentações da BNCC, essas experiências tecnológicas podem refletir nas atividades desenvolvidas na Educação Básica e possibilitar a identificação de habilidades relacionadas ao conceito de pensamento computacional, citando, por exemplo, as características do componente curricular da matemática. Estabelecido o componente da matemática como instrumento de formação, o pensamento computacional pode apoiar esse processo, no que tange características cognitivas associadas às situações cotidianas (AZEVEDO; MALTEMPI, 2020; LEAL; MALTEMPI, 2020).

Em estudos apontados por Silva (2018), foram observadas características dispostas na BNCC sustentados no arcabouço do pensamento computacional em um ambiente de ensino fundamental para desenvolver competências para soluções de tarefas matemáticas. Ainda de acordo com o autor, ao usar o pensamento computacional, os estudantes puderam incitar conhecimentos matemáticos e, também, interdisciplinares no empenho para investigar e

---

solucionar problemas utilizando as tecnologias digitais das linguagens de programação de computadores.

A concepção de Wing (2006), sobre o pensamento computacional e a aprendizagem matemática, não está apenas restrita às tecnologias digitais, como programações de *softwares*, vai muito além por estar associada às concepções e ideias utilizadas pelos sujeitos. Estas concepções permitem, aos sujeitos, a busca para determinar possíveis soluções às atividades propostas no currículo de matemática, de modo a agrupar aspectos que privilegiam a comunicação e interação entre estudantes, professores e pessoas envolvidas no processo de aprendizagem, que incluem as relações em um contexto escolar.

Neste aspecto, acredita-se que a presente pesquisa pode contribuir com a área de Educação Matemática no instante em que produzir pesquisas e reflexões em relação à concepção da abordagem pensamento computacional associada à aprendizagem matemática. O embasamento teórico relacionado às práticas pedagógicas poderá estabelecer condições para o ensino e aprendizagem de conteúdos matemáticos. Para compreender as contribuições do pensamento computacional do presente estudo nas práticas pedagógicas, junto aos currículos de matemática, foram adotados instrumentos de apoio pedagógicos providos das Tecnologias Digitais de Informação e Comunicação (TDIC), como o ambiente do Robocode.

### **3 Framework de Simulação “Robocode”**

O experimento do trabalho apresenta uma associação entre as abordagens do Pensamento Computacional e desafios em forma de problemas apoiados no software do tipo jogo educacional digital Robocode. Conforme a sugestão de Alaiba e Rotaru (2008), Robocode pode ser classificado com a visão de um *framework* de simulação acessível, descomplicado e de transparente manipulação, nativamente desenvolvido com objetivo no ensino das abordagens da Programação Orientada à Objetos.

Aspectos de conteúdos matemáticos podem ser observados no jogo Robocode, no instante em que o mesmo foi projetado como um plano cartesiano, representado pelo simulador das disputas entre os robôs. Destacam-se, para essa disposição no ambiente da disputa, um marco indicado no canto inferior esquerdo que forma a origem do plano cartesiano (0,0), e a altura e largura máxima do campo dependem das configurações previamente selecionadas. Portanto, a qualquer momento, um robô virtual pode recuperar as informações de localização no simulador pelos eixos “x” e “y”. Essa característica de jogo, permite a criação de desafios os quais permitem envolver diversificados conceitos matemáticos (ABDELLATIF; MCCOLLUM; MCMULLAN, 2018).

Fayek e Farag (2015) classificam a iniciativa do Robocode como um *software* livre e de código fonte aberto amparado nos termos EPL (*Eclipse Public License*). Essa licença de código aberto permite o desenvolvimento e as simulações das disputas virtuais envolvendo a programação de robôs nas linguagens Java e C#. O jogo tem como uma das suas características principais um ambiente de simulação, representado por uma arena, das quais os robôs virtuais desenvolvem as disputas, com o objetivo de fomentar as estratégias de programação. Para cada robô, torna-

---

se importante destacar a existência de algoritmos e estratégias próprias de comportamentos (LARSEN, 2013).

Nos estudos de Meira, Lima e Borges (2016), o objeto contendo as abordagens do jogo Robocode pode ser aplicado para apoiar o aprendizado de linguagens de programação, citando Java ou C#. Aplicações Robocode são suportadas nos mais variados ambientes, desde as acadêmicos representadas nas salas de aulas, laboratórios de informática até o planejamento de competições científicas online podem utilizadas como estratégias de ensino e aprendizagem de matemática ou programação de computadores (MEIRA; BORGES, 2017).

Na visão de Nidorf, Barone e French (2010), Robocode também pode ser enquadrado como *Serious Games*<sup>2</sup>, no instante em que se estabelece um ambiente próprio contendo o denominado agente de aprendizagem, com a possibilidade de interação e aprendizado. O agente de aprendizagem possibilita selecionar uma diversidade de comportamentos e mensurar a eficiência de cada ação durante os desafios estabelecidos ao longo do jogo.

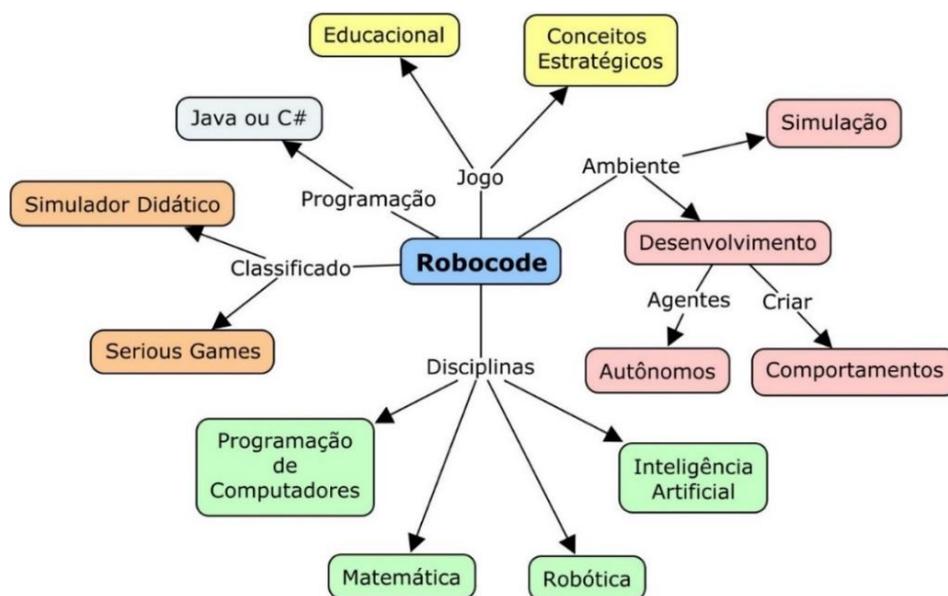
Para Recchia, Chung e Pochiraju (2014) o Robocode consiste num simulador de desafios sustentados na Inteligência Artificial (IA). Nas simulações destes desafios, caracterizadas como as disputas dentro do jogo, é possível constatar os comportamentos de cada robô virtual participante da partida. Como os desafios são realizados de modo *on-line*, no ambiente da plataforma, torna-se possível observar e comparar as programações idealizadas por todos desenvolvedores participantes.

No Jogo simulado, os robôs são destacados como entidades autônomas e seus programadores não possuem comando nas ações, contribuindo assim, para o desenvolvimento dos agentes inteligentes. Quando programado, um determinado robô virtual não têm a absoluta consciência do universo das disputas e seu conhecimento se restringem às informações de sensores das quais pode desenvolver a leitura (exemplo: posicionamento do robô virtual, ângulos estabelecidos para o radar e o canhão) (FAYEK; FARAG, 2015). A Figura 1 exibe um mapa conceitual para diferentes compreensões e classificações, desenvolvido a partir das referências citadas na presente seção, para variadas características do Robocode.

Figura 1 – Compreensões para Robocode

---

<sup>2</sup> O termo *Serious Games* passou a ser utilizado para identificar os jogos com um propósito específico, ou seja, jogos que transcendam a ideia de entretenimento e ofereçam outros tipos de experiências, como aquelas voltadas ao aprendizado e ao treinamento (BLACKMAN, 2005).



Fonte: Elaborada pelos autores

#### 4 Metodologia

Os métodos de pesquisa da presente proposta indicam abordagens qualitativas de natureza aplicada. As classificações da pesquisa quanto aos objetivos são exploratórias e descritivas. A pesquisa exploratória tem por objetivo o “estabelecimento de maior familiaridade com o problema a fim de torná-la mais explícito ou na construção de hipóteses.” (GIL, 2008). A pesquisa descritiva “exige do investigador uma série de informações sobre o que deseja pesquisar com a descrição aos fatos e fenômenos de uma determinada realidade” (TRIVIÑOS, 1987). O objeto de estudo consiste no processo de aprendizagem integrado de matemática e linguagem de programação Java em um ambiente estruturado nas abordagens do Pensamento Computacional.

#### 5 Ambiente dos Desafios com Robocode

Jogos de programação podem oferecer abordagens motivadoras aos alunos para introduzir conceitos matemáticos e de programação de computadores. Estimular alunos a aprender consiste em um importante tema de discussão entre professores. Neste contexto, os jogos educacionais, devidamente planejados em ambientes educacionais, podem se apresentar como abordagens para oferecerem novas possibilidades para contribuir e envolver estudantes no aprendizado de conceitos de programação. Podem ser considerados objetivos das práticas estruturadas no ambiente do Robocode: (1) aproximar disciplinas específicas do cotidiano dos alunos; (2) apresentar aplicações práticas na resolução de problemas reais ou simulados; (3) e envolver de modo natural o interesse dos alunos. Trabalhar com desenvolvimento e estratégias de jogos digitais, também, despertam novos vínculos dos alunos perante a escola e incentivam trabalhos em equipes com definições de estratégias cooperativas de aprendizagem. O desafio de um jogo trabalha valores afetivos como a autoconfiança e autoestima do aluno, na medida da evolução

dos desafios, associando-as a sua capacidade de interagir no grupo e resolver problemas (HAKULINEN, 2011).

O cenário de aplicação da presente proposta foi o Instituto Federal de Educação Ciência e Tecnologia do estado de São Paulo (IFSP), Campus IFSP Capivari, representado pelo curso Técnico Integrado em Informática. A turma selecionada foi o 2º (segundo) ano com aproximadamente 34 (trinta e quatro) alunos. A proposta foi aplicada durante o segundo semestre de 2018, entre os meses de outubro a novembro, em um total de 2 (dois) encontros em 8 aulas, totalizando cerca de 6 (seis) horas e 30 (trinta) minutos de dedicação integral. Os encontros foram mediados entre 2 (dois) professores (um representante da área de matemática e outro da computação, respectivamente) e toda turma do 2º ano do Técnico Integrado em Informática reunida em um laboratório de informática, devidamente equipados com os *softwares* determinados ao ambiente do Robocode. De modo a oportunizar os conceitos do Pensamento Computacional, a proposta focou a diversificação do ensino e aprendizagem. Para essa diversificação, abordagens do Pensamento Computacional foram utilizadas, como: (1) trabalhar a aprendizagem colaborativa com estudantes responsáveis pela aprendizagem; (2) com professores atuando como facilitadores e aconselhores. Para desenvolvimento das atividades, foram criados grupos de 2 (dois) a 4 (quatro) estudantes.

A partir dos grupos devidamente formados, os professores mediadores se encarregaram em apresentar as atividades de modo a desenvolver o Pensamento Computacional. A atividade teve seu foco fundamentado na criação de situações problemas para abordagens no jogo educacional Robocode. Essas situações podem ser visualizadas na Figura 2, com a apresentação de um problema incompleto com características marcantes em apresentar conceitos necessários para sua resolução. O problema incompleto, denominado desafio, tem como principal finalidade evitar situações em que o robô bata na parede. O problema foi estruturado para envolver conceitos de plano cartesiano, da matemática e, também, conceitos introdutórios da linguagem de programação Java. No problema várias abordagens do Pensamento Computacional podem ser adotadas para sua resolução, como: coleta, análise e representação de dados; decomposição de problema.

Figura 2 – Estrutura do Problema para Envolver Coordenadas e Programação

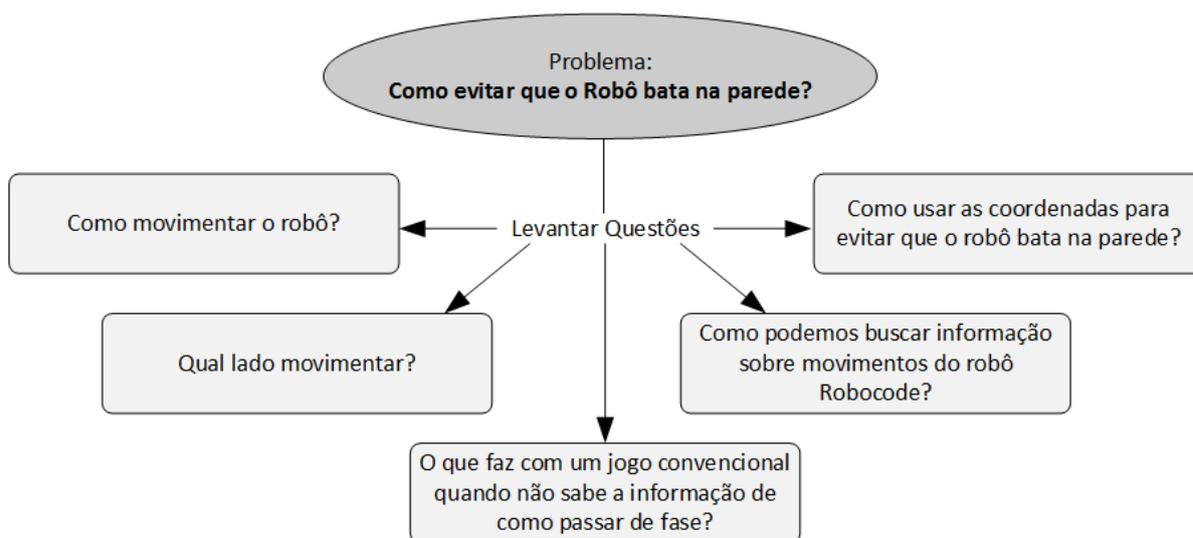


Fonte: Elaborada pelos autores

O desafio foi apresentado aos estudantes, das quais puderam desenvolver várias abordagens de conceitos presentes no Pensamento Computacional: (1) casos relacionados, pesquisar o comportamento de outros robôs, quando batem na parede; (2) raciocínio lógico, ações a serem tomadas na ocorrência dos eventos; (3) conhecimento prévio, com base em análise de disputas anteriores. O desafio exibido na Figura 3, exigiu dos estudantes uma pesquisa para aprofundar os conhecimentos das regras e características do Robocode. Perguntas relacionadas ao conhecimento prévio do aluno puderam ser instigadas: "O que você faz em um jogo convencional quando não sabe a informação de como passar de fase?" ou "Como usar as coordenadas do robô para evitar a colisão". Durante a aplicação do desafio, professores puderam compartilhar fontes de recursos para as pesquisas de acordo com o planejamento pedagógico do plano cartesiano e introdução a linguagem de programação Java.

Os conceitos tratados no jogo Robocode estão relacionados com os eventos de movimentação e orientação do robô. Para responder as perguntas instigadas na Figura 3, foram necessárias compreender quais são as informações necessárias para iniciar as estratégias de resolução ao desafio. Os grupos de estudantes puderam levantar informações necessárias para resposta ao desafio com base na Figura 4: (1) a arena da disputa, Figura 4 (a), é compreendida entre os eixos X (representa a largura, em pixels) e eixo Y (representa a altura, em pixels); (2) ainda na arena, Figura 4 (a), é possível observar a orientação da arena, que pode ser medida em graus ou radianos; (3) já o robô, Figura 4 (b), se orienta a partir das informações obtidas com comandos programáveis dos eixos X e Y em relação a arena da disputa, das quais podem originar diversas estratégias como respostas com base em cálculos a partir do sistema de coordenadas cartesianas.

Figura 3 – Características Codificadas da Estrutura do Problema

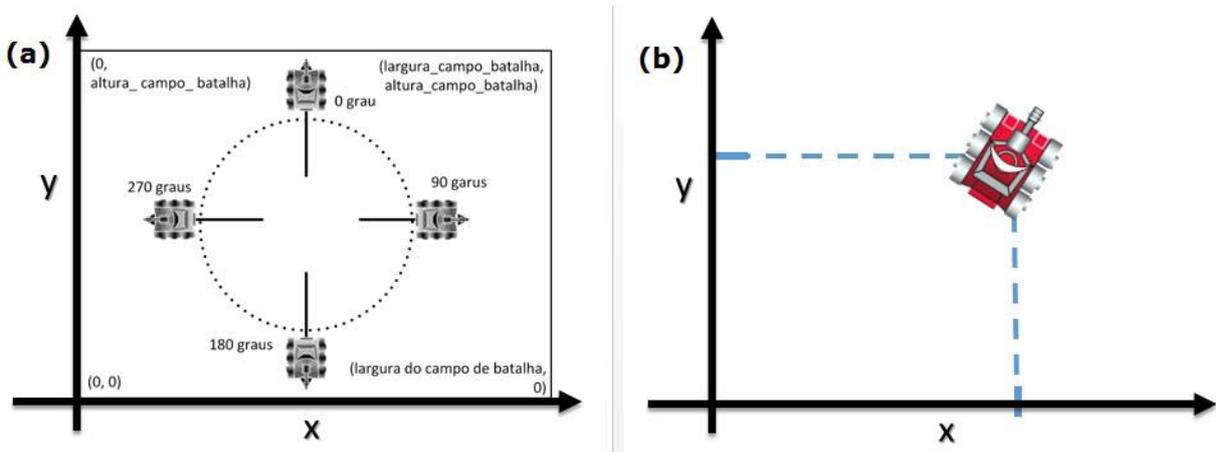


Fonte: Elaborada pelos autores

De posse das informações necessárias para orientação na arena de acordo com o plano cartesiano, os estudantes puderam novamente usar abordagens presentes no Pensamento

Computacional, como: (1) decomposição do desafio principal para posterior ordenação das partes de forma a conduzir os jogadores a formularem hipóteses que funcionem na resolução; (2) estabelecimento de uma sequência de lógica de resolução.

Figura 4 – (a) Coordenadas do Campo de Jogo (b) Coordenadas do Robô



Fonte: Elaborada pelos autores

## 6 Análise e Discussão dos Dados

As resoluções dos problemas com base nas abordagens do pensamento computacional permitiram às equipes desenvolverem estratégias de movimento no instante em que o robô bater na parede. Para promover discussões de resoluções, os estudantes sentiram a necessidade de estudar as regras do jogo para posteriormente desenvolver a programação do código, de acordo com as condições apresentadas durante as disputas.

O problema de movimentação do robô pôde ser decomposto em fases de resolução, da qual conceitos do pensamento computacional segundo Mannilla (2014) foram utilizados, como: coleção, análise e representação de dados. Neste ponto, os estudantes formularam os objetivos de aprendizagem a partir da discussão acerca dos dados apresentados no problema. Foram repetidos ciclos de rediscussões do problema a cada a cada nova partida disputada junto ao jogo Robocode para identificar os novos conhecimentos adquiridos.

A partir das abordagens do pensamento computacional, iniciou-se a fase de estudo dos conceitos matemáticos da orientação do robô com o sistema de coordenadas cartesianas (compreensão dos conceitos envolvidos) para que o resultado forneça o aporte necessário para criar o programa de movimentação com base nos conceitos linguagem de programação Java. Para linguagem de programação os alunos trabalharam a programação de métodos<sup>3</sup> (métodos *ahead*, *back*, *turnRight*, *turnLeft*, ex. *getBearing()* - ângulo em graus da parede batida em relação

<sup>3</sup> Os métodos são procedimentos ou funções, estabelecidas nas linguagens de programações de computadores, que realizam as ações de um determinado objeto especificado. No caso do Robocode, são representadas pelas as ações determinadas ao robô como: *ahead()* ou *back()* que permite movimentar o robô para frente ou para trás, a uma distância x indicado por parâmetro.

---

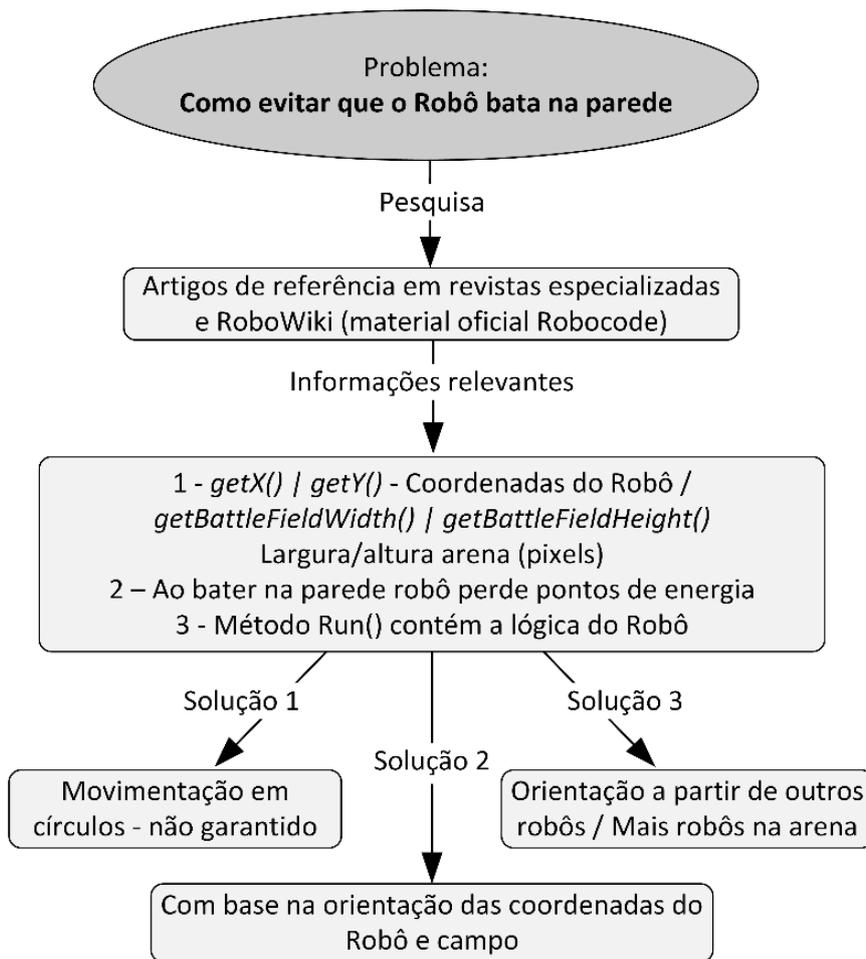
ao robô.), indiretamente tiveram acesso as noções de programação de eventos<sup>4</sup> (evento *onHitWall*) que dispara o método “quando robô bater na parede”. A Figura 5 apresenta algumas respostas de equipes para o problema de movimentação do robô.

Cada grupo de estudantes estruturou a resolução do problema de movimentação em 5 (cinco) passos de resolução: (1) pesquisa; (2) classificação; (3) comentários; (4) reunião e (5) sequência. Na pesquisa, foram discutidos aspectos como: o que se faria em um jogo convencional? Para essa resposta surgiram apontamentos indicando a leitura e estudo de manuais e artigos do jogo. Em seguida ocorreu a triagem das informações no sentido de classificar quais são relevantes para definir a movimentação do robô. A Figura 5 indica nas informações relevantes os conceitos dos sistemas de coordenadas cartesianas e, também, os métodos e eventos da programação que podem ser associados aos matemáticos e gerar respostas de movimentação. O problema apresentado aos estudantes, admite várias repostas igualmente válidas, com diferentes graus de abordagens para as soluções. Coube a cada equipe, desenvolver reuniões com objetivo de discussão das soluções apontadas como viáveis, se considerando, também a estratégia de jogo adotadas durante as disputas. A final de cada reunião, as informações foram transformadas em possíveis soluções válidas com o sequenciamento e justificativa dos passos apresentados em resposta ao desafio de movimentação.

Figura 5 – Respostas ao Problema

---

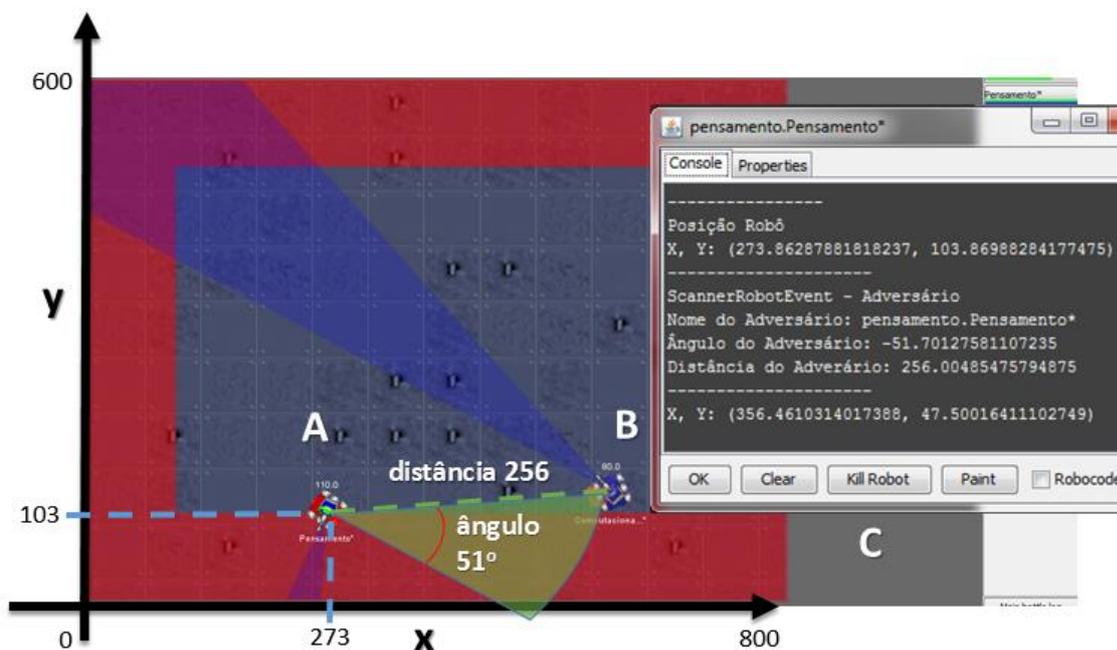
<sup>4</sup> Os eventos são chamados quando acontece algo específico no decorrer do jogo como o robô bater na parede denominado de evento *onHitWall*.



Fonte: Elaborada pelos autores

A Figura 6 apresenta informações das quais as equipes puderam analisar ao longo das disputas realizadas no Robocode. Os conceitos do sistema de coordenadas cartesianas puderam ser analisados nos eixos "x" e "y" do campo de disputa. Assim como o campo, os estudantes analisaram a orientação dos robôs "A" e "B". Os alunos levantaram as informações necessárias dos robôs como apresentado na Figura 6 na janela "C": (1) posição dos robôs a partir dos eixos (x e y); (2) ângulo do adversário (3) distância do adversário. De posse dessas informações o exemplo da Figura 6 indica (cor vermelha) o perímetro desejável de movimentação do robô a evitar a colisão com a parede. Ao final das análises foi possível que cada equipe sugerisse diversas estratégias de movimentação: com base na orientação do campo ou mesmo com base no posicionamento do adversário. Esse conjunto de informações pôde ser trabalhado no aspecto matemático para cálculos de coordenadas. Posteriormente aos cálculos, houve a integração com a linguagem de programação, para converter as estratégias adotados em códigos de instruções para os robôs.

Figura 6 – Coordenadas Cartesianas no Robocode



Fonte: Elaborada pelos autores

## 7 Conclusão

Com o Pensamento Computacional aplicado em jogos educacionais, espera-se incentivar novos currículos com integrações de matemática, com os conceitos de planos cartesianos, com programação de computadores. Alunos versus a complexidade de linguagens de programação (Java), da qual muitos especialistas consideram desafiante. Além do incentivo, pretende-se alinhar as práticas pedagógicas com a normativa da BNCC, em quesitos como despertar a prática intelectual e investigação associadas as Tecnologias Digitais de Informação e Comunicação. Espera-se, também, organização de currículos com cursos de programação coerentes e embutidos em práticas de computação mais ampla, que pudessem envolver outros conceitos matemáticos.

No processo de aplicação da proposta, um fator considerado decisivo no processo de aprendizagem, foram as eliciações desenvolvidas pelos grupos, em um ambiente colaborativo, para determinar possíveis caminhos para solução do desafio proposto.

Nos conceitos matemáticos, os estudantes trabalharam a posição e orientação do robô com base em um planejamento a partir do plano de coordenadas. Nos desafios propostos, grupos definiram a orientação, em graus, da parede da qual o evento ocorreu em relação ao robô. De posse dos cálculos matemáticos retornados da aplicação, da análise e resposta do jogo em relação ao plano cartesiano, foi possível que os estudantes pudessem desenvolver as programações do robô. Esses conceitos de programação envolveram os estudantes a pesquisar os eventos e programar os métodos do robô, de modo a integrar os resultados matemáticos com o desenvolvimento da lógica e da programação na linguagem de programação Java. Os

estudantes também puderam perceber que as referências matemáticas fornecem resultados como subsídios para desenvolvimento de estratégias do jogo e conseqüentemente responder aos desafios propostos.

## Referências

ABDELLATIF, A. J.; MCCOLLUM, B.; MCMULLAN, P. Serious games quality characteristics evaluation: The case study of optimizing robocode. *SIIE 2018 - 2018 International Symposium on Computers in Education, Proceedings*, p. 1-4, 2018.

ALAIBA, V.; ROTARU, A. Agent architecture for building robocode players with SWI-Prolog. *Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2008*, v. 3, n. 2, p. 3-7, 2008.

AZEVEDO, G. T. DE; MALTEMPI, M. V. Processo de Aprendizagem de Matemática à luz das Metodologias Ativas e do Pensamento Computacional. *Ciência & Educação (Bauru)*, v. 26, p. 1-18, 2020.

BARR, B. V.; STEPHENSON, C. Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, v. 2, n. 1, p. 48-54, 2011.

BLACKMAN, S. Serious Games ... and Less! *Computer Graphics*, v. 39, n. February, p. 12-16, 2005.

BNCC. *Base Nacional Comum Curricular*. Disponível em: <<http://basenacionalcomum.mec.gov.br>>. Acesso em: 6 out. 2020.

BORBA, M. DE C.; LACERDA, H. D. G. L. Políticas Públicas e Tecnologias Digitais: Um Celular por Aluno. *Educação Matemática Pesquisa*, v. 17, n. 3, p. 490-507, 2015.

CSTA, C. S. T. A.; ISTE, I. S. FOR T. IN E. *Computational Thinkink. Teacher Resources*. 2. ed. [s.l.] National Science Foundation, 2011. v. 2

FAYEK, M. B.; FARAG, O. S. HICMA: A human imitating cognitive modeling agent using statistical methods and evolutionary computation. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIHLI 2014: 2014 IEEE Symposium on Computational Intelligence for Human-Like Intelligence, Proceedings*, 2015.

FEPESP. Evasão na Educação Profissional. *Fórum da Educação Profissional do Estado de São Paulo - FEPESP, 31 de maio de 2011, São Paulo. Anais eletrônicos.*, p. 1-1, 2011.

GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. 6. ed. São Paulo: Atlas, 2008.

HAKULINEN, L. Survey on Informatics Competitions: Developing Tasks. *Olympiads in Informatics*, v. 5, p. 12-25, 2011.

IFSP. Programa de Auxílio de Permanência. *Edital N. SPO.006 de 15 de fevereiro de 2019*, 2019.

KUCHAR, K. et al. Supervised Learning in Multi-Agent Environments Using Inverse Point of View. *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, p. 625-628, 2019.

LARSEN, F. N. *ReadMe for Robocode*. Disponível em: <<https://robocode.sourceforge.io/docs/ReadMe.html>>. Acesso em: 15 jul. 2020.

LEAL, L.; MALTEMPI, M. V. Matemática, Pensamento Computacional e BNCC: desafios e potencialidades dos projetos de ensino e das tecnologias na formação inicial de professores. *RBECM - Revista Brasileira de Ensino de Ciências e Matemática*, v. 3, n. 3, p. 748-776, 2020.

MALONEY, J.; RESNICK, M.; RUSK, N. The Scratch programming language and environment. *ACM Transactions on Computing Education*, v. 10, n. 4, p. 1-15, 2010.

MANNILA, L. et al. Computational Thinking in K-9 Education. *Proceedings of the Working Group Reports of the 2014 on Innovation 38; Technology in Computer Science Education Conference*, p. 1-29, 2014.

MEIRA, M.; BORGES, M. Aprendizagem de Linguagem de Programação com Metodologia PBL em Competições

---

Científicas com Robocode. *VI Congresso Brasileiro de Informática na Educação (CBIE 2017) Anais do XXIII Workshop de Informática na Escola (WIE 2017)*, p. 195–204, 2017.

MEIRA, M. C.; LIMA, M. S. S.; BORGES, M. A. F. Torneios Baseados em Robocode para Incentivar Jovens a Aprender Programação. *XXXVI Congresso da Sociedade Brasileira de Computação (CSBC 2016) WEI - 24o Workshop sobre Educação em Computação*, p. 2403–2412, 2016.

MORAIS, A. D. DE; VINICIUS, M.; BASSO, D. A. Educação Matemática & Ciência da Computação na escola : aprender a programar fomenta a aprendizagem de matemática? \* Mathematics Education and Computer Science in school: does learning to code foster the learning of Mathematics? *Ciênc. Educ.*, v. 23, n. 2, p. 455–473, 2017.

NAKAMURA, T.; KAWASAKI, T. Computer science unplugged for developing computational thinking and mathematical thinking. *Proceedings - International Joint Conference on Information, Media, and Engineering, IJCIME 2019*, p. 305–308, 2019.

NIDORF, D. G.; BARONE, L.; FRENCH, T. A comparative study of NEAT and XCS in robocode. *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, p. 1–8, 2010.

PAPERT, S. *The Connected Family: Bridging the Digital Generation Gap*. Atlanta: Taylor Trade Publishing, 1996.

PIRES, F. et al. Gamification and engagement: Development of computational thinking and the implications in mathematical learning. *Proceedings - IEEE 19th International Conference on Advanced Learning Technologies, ICALT 2019*, p. 362–366, 2019.

PLAZA, P. et al. Multiplatform educational robotics course to introduce children in robotics. *Proceedings - Frontiers in Education Conference, FIE*, v. 2018- Octob, 2019.

RECCHIA, T.; CHUNG, J.; POCHIRAJU, K. Performance of heterogeneous robot teams with personality adjusted learning. *Biologically Inspired Cognitive Architectures*, v. 7, p. 87–97, jan. 2014.

SÁPIRAS, F. S.; VECCHIA, R. D.; MALTEMPI, M. V. Utilização do Scratch em sala de aula. *Educ. Matem. Pesq.*, São Paulo, v. 17, n. 5, p. 973–988, 2015.

SCRATCH. *Create Stories, Games, and Animations Share With Others Around the World*. Disponível em: <<http://scratch.mit.edu>>. Acesso em: 5 jan. 2021.

SILVA, E. C. DA. *Pensamento Computacional e a formação de conceitos matemáticos nos Anos Finais do Ensino Fundamental: uma possibilidade com kits de robótica (Dissertação de Mestrado)*. [s.l.] Programa de Pós-Graduação em Educação Matemática - Universidade Estadual Paulista (UNESP), 2018.

TRIVIÑOS, A. *Introdução à Pesquisa em Ciências Sociais: a Pesquisa Qualitativa em Educação*. São Paulo: Atlas, 1987.

VALENTE, J. A. *O Computador na Sociedade do Conhecimento*. Campinas, São Paulo: OEA\_NIED/UNICAMP, 1999.

VALENTE, J. A. Integração do Pensamento Computacional no Currículo da Educação Básica: Diferentes Estratégias Usadas e Questões de Formação de Professores e Avaliação do Aluno Training and Student's Assessment. *Revista e-Curriculum*, v. 14, n. 3, p. 864–897, 2016.

WING, J. Pensamento Computacional – Um conjunto de atitudes e habilidades que todos , não só cientistas da computação , ficaram ansiosos para aprender e usar . *Revista Brasileira de Ensino de Ciência e Tecnologia.*, v. 9, n. 2, p. 1–10, 2006.

Recebido em julho de 2020.

Aprovado para publicação em dezembro de 2020.

**Matheus Carvalho Meira**

Discente de doutorado da Faculdade de Educação da UNICAMP.  
Professor do Instituto Federal de São Paulo  
m108479@dac.unicamp.br

**Karlan Ricomini Alves**

Mestrado em Educação UNISAL  
Professor do Instituto Federal de São Paulo  
karlan.ricomini@ifsp.edu.br

**Luis Antonio Tavares**

Discente de doutorado da Faculdade de Educação da UNICAMP.  
Professor do Instituto Federal do Sul de Minas Gerais  
luis.tavares@ifsuldeminas.edu.br

**Rafael Fernando Diório**

Doutor em Tecnologia e Inovação pela Faculdade de Tecnologia da UNICAMP  
Professor do Instituto Federal de São Paulo  
rafael.diorio@ifsp.edu.br

**Sergio Ferreira do Amaral**

Coordenador do Laboratório de Inovação Tecnológica Aplicada na Educação - LANTEC  
Professor Titular Colaborador da UNICAMP  
amaral@unicamp.br