



Arquitetura Pedagógica Baseada no Pensamento Computacional para a Resolução Cooperativa de Problemas

Francisco Xavier da Silva, IFMS - Campus Coxim - Coxim - MS – Brasil,

francisco.silva@ifms.edu.br, <https://orcid.org/0000-0001-7028-7623>

Crediné Silva de Menezes, PPGIE - UFRGS - Porto Alegre, RS – Brasil,

credine@gmail.com, <https://orcid.org/0000-0002-2709-7135>

Alberto Nogueira de Castro Junior, IComp - UFAM - Manaus, AM – Brasil,

alberto@icomp.ufam.edu.br, <https://orcid.org/0000-0002-1752-8667>

Resumo. Este artigo apresenta uma Arquitetura Pedagógica baseada no Pensamento Computacional para a resolução cooperativa de problemas. Foi realizado um experimento de ensino em que os estudantes, de posse de problemas, constroem soluções individuais, revisam as soluções dos seus pares e, em seguida, em grupo, constroem uma solução única. A análise das soluções apresentadas revelou como os estudantes desenvolveram novos esquemas na elaboração de suas soluções, na revisão das soluções dos problemas de seus pares e na reconstrução da solução do grupo. Portanto, a Arquitetura Pedagógica apoiada no Pensamento Computacional utilizada se mostrou valiosa para o ensino e a aprendizagem de programação.

Palavras-chave: Pensamento Computacional; Arquitetura Pedagógica; Aprendizagem de Programação

Pedagogical Architecture Based on Computational Thinking for Cooperative Problem Solving

Abstract. This article presents a Pedagogical Architecture based on Computational Thinking for cooperative problem solving. A teaching experiment was carried out in which students, in possession of problems, build individual solutions, review the solutions of their peers, and then, as a group, build a unique solution. The analysis of the solutions presented revealed how the students developed new schemes in the elaboration of their solutions, in the review of the solutions of their peers' problems and in the reconstruction of the group's solution. Therefore, the Pedagogical Architecture supported by Computational Thinking used proved to be valuable for the teaching and learning of programming.

Keywords: Computational Thinking; Pedagogical Architecture; Learning the Programming

1. Introdução

Oliari (2021), em sua pesquisa, mostrou que, para aprender programação, é importante não apenas entender a teoria, mas também interpretar e criar problemas com base em algum método de aprendizagem ativo. Portanto, a integração da teoria com a prática é eficaz se o estudante construir ativa e interativamente sua aprendizagem.

Com o objetivo de ajudar os estudantes do Ensino Médio Técnico em Informática na construção de conhecimento sobre programação, propomos uma Arquitetura Pedagógica baseada no Pensamento Computacional para a resolução cooperativa de



problemas, que faz parte da Arquitetura Pedagógica baseada no Pensamento Computacional para Apoio à Aprendizagem de Programação (APAP).

Este artigo, descreve as estratégias propostas para a aprendizagem de programação, incluindo o Pensamento Computacional como meio para que os estudantes reflitam sobre resolução de problemas e tenham um melhor entendimento das tecnologias que nos cercam (Wing, 2006). Além disso, nas Arquiteturas Pedagógicas segundo Carvalho, Nevado e Menezes (2007), essa concepção pedagógica se apoia em três pilares: i) o construtivismo proposto por Jean Piaget (2016), ii) o paradigma pedagógico concebido por Paulo Freire (2014) e sintetizado na Pedagogia da Pergunta, e iii) o uso de Tecnologias Digitais de Informação e Comunicação (TDICs).

Este artigo está organizado da seguinte forma: a Seção 2 contém a fundamentação teórica; a Seção 3 apresenta a Arquitetura Pedagógica baseada no Pensamento Computacional para a cooperação e resolução de problema e, finalmente, a Seção 4 expõe as considerações finais.

2. Fundamentação teórica

Nesta seção, serão apresentados os conceitos fundamentais que formam a base teórica da proposta, com ênfase nas interações na aprendizagem sob a perspectiva construtivista de Piaget, no Pensamento Computacional e na Arquitetura Pedagógica.

2.1 Epistemologia segundo Piaget

A teoria epistemológica de Piaget (2016) ajuda na solução de problemas, explicando de forma satisfatória a construção do conhecimento, partindo do conhecimento que as pessoas têm antes de qualquer formalização. Assim, o que a estrutura mental orgânica do sujeito epistêmico é, em certo momento, resulta da interação entre o sujeito e o meio em um momento anterior. Aplicando a teoria de Piaget à construção de soluções para problemas, definimos um referencial teórico a ser seguido.

Essa teoria oferece um modelo para explicar a construção do conhecimento que pode ser usado em todos os domínios e em todos os níveis de desenvolvimento. Os pontos centrais desta teoria envolvem a transição de um nível inferior de conhecimento para um nível considerado superior (Piaget, 2016).

Essa transição de um conhecimento inferior para um superior acontece, segundo Piaget (1973), quando o sujeito capta o ponto que se interessa aprender, tendo em conta a função e a base psicológica do mesmo, ou seja, ao agir, o sujeito epistêmico incorpora cada vez mais elementos às suas ações.

Para o autor, esses esquemas de ações são chamados de “assimilação”, pois quando os esquemas de ação de um organismo não conseguem assimilar uma situação ou quando o organismo (mente) desiste de se esforçar para assimilar uma situação ou se modifica, criando novas ações, ou seja, aperfeiçoa seus esquemas ou cria novos esquemas; Piaget chama de “acomodação” essa mudança na forma da ação. Assim, a acomodação nada tem de passiva, é um processo realizado ativamente pelo sujeito que modifica suas próprias formas de ação (Piaget, 1973).

Em linhas gerais, é possível afirmar que os esquemas de ação de um sujeito em um determinado instante determinam as ações disponíveis para ele naquela situação específica. À medida que as estruturas mentais se desenvolvem por meio das interações com o meio, o sujeito cognitivo passa a distinguir e integrar suas ações, o que acarreta diferenciação e coordenação de seus padrões de comportamento. Assim, seu sistema de



esquemas de ação vai se complexificando, desde o nascimento até a idade adulta (Piaget, 1973).

De forma geral e esquemática, os períodos de formação do sistema de esquema de ações e operações e algumas de suas características gerais, estudado em detalhes por Piaget (2016), são:

- No primeiro estágio, denominado sensório-motor, cada ação integra-se como um todo indivisível de percepção e movimento. Há a constituição do sistema de esquemas de ações sensório-motoras. É durante essa fase que a criança adquire conhecimento prático, caracterizado como um "saber fazer"

- O desenvolvimento do sujeito epistêmico segue uma progressão na qual, no estágio pré-operacional, demonstra a capacidade de representar situações e objetos. Nesse estágio, o comportamento do sujeito reflete essa capacidade, como evidenciado quando uma criança representa uma situação de almoço ao brincar de dar comida à sua boneca ou simula uma situação de guerra mediante uma batalha entre guerreiros. Além disso, o sujeito é capaz de narrar histórias sobre suas experiências diárias, demonstrando assim sua capacidade representativa.

- No estágio operatório concreto da Epistemologia Genética de Piaget, o sujeito epistêmico atinge a capacidade de realizar operações internas sobre suas representações mentais, permitindo uma representação mais precisa da realidade circundante. Isso abrange a compreensão aprimorada do espaço, da causalidade, da conservação (incluindo substância, peso, números, classes lógicas, relações de magnitude, entre outros). Piaget denomina essa capacidade de realizar ação interior de "operação".

- Por fim, no estágio formal da Epistemologia Genética de Piaget, o sujeito atinge a capacidade de representar mentalmente qualquer objeto por meio de signos, como palavras ou letras. Essa habilidade marca o desenvolvimento da capacidade de operar internamente sobre esses signos. Nesse estágio, ocorre uma transição crucial onde o pensamento se torna mais abstrato e simbólico, permitindo ao sujeito lidar com conceitos complexos e realizar operações mentais sobre eles. É nessa fase que se evidencia uma maior autonomia do pensamento e a capacidade de raciocínio lógico formal, possibilitando a manipulação de ideias de forma mais sofisticada e independente.

Para Pádua (2009), a principal diferença entre os estágios está na forma como o pensamento do sujeito se desenvolve e nas capacidades cognitivas que ele adquire em cada fase, passando do âmbito da motricidade do primeiro estágio às atividades representativas do segundo, até o terceiro e o quarto com o pensamento operatório, a diferença entre eles é constatada pelo fato de que no terceiro, o pensamento operatório ainda esteja ligado ao concreto, enquanto que no quarto, este mesmo pensamento tem ligação ao abstrato e formal.

Segundo a Epistemologia Genética Piagetiana, o conhecimento, inclusive o mais abstrato, está intrinsecamente ligado a ações que podem ser realizadas. Para Piaget, qualquer aprendizado que ocorra sem compreensão do seu significado, ou seja, sem uma conexão com ações possíveis, não pode ser considerado como conhecimento.

2.2 Pensamento Computacional

Papert (1980) destacou a importância do uso dos objetos-de-pensar-como suporte à reflexão, permitindo que os sujeitos pensem sobre seu próprio pensamento e estendam essa capacidade para outras situações. Ao utilizar essa abordagem, as habilidades cognitivas dos estudantes são aprimoradas, pois estes são incentivados a refletir sobre o processo de pensamento, ampliando suas estruturas mentais.



Essa ênfase na reflexão sobre o pensamento de Papert (1980) se constitui no que Wing (2006) chamou de Pensamento Computacional (PC). Ela destaca o PC como uma habilidade essencial para o futuro, que vai além da programação e inclui diferentes níveis de abstração, resolução de problemas e pensamento lógico e sistêmico (Wing, 2017). A autora também enfatiza a relevância do PC em diversas áreas do conhecimento e sua aplicação em todos os níveis de ensino.

Assim, a abordagem de Papert (1980) e a visão de Wing (2017) convergem na importância da reflexão metacognitiva e do desenvolvimento de novas habilidades cognitivas. Papert (1980) afirma que um bom programa computacional eficaz é aquele estruturado em etapas que facilitem a compreensão e a correção de erros, inclusive utilizando a programação paralela para tornar o processo mais claro e compreensível.

Por outro lado, Wing (2017) ressalta que o PC não se restringe apenas a uma linguagem de programação, mas abrange processos mentais durante a solução e formulação de problemas, podendo ser expresso em diversas formas de comunicação linguística (linguagem visual, natural, de programação, ...) que seja compreendida e esteja em um formato acessível ao público que se destina.

Neste contexto, o PC pode contribuir para a formação de sujeitos preparados para as demandas do século XXI ao promover uma postura ativa, crítica e inventiva nos estudantes. Através do desenvolvimento do PC, os estudantes são incentivados a analisar, fazer escolhas, testar hipóteses e agir sobre o objeto do conhecimento, o que os prepara para lidar com os desafios e exigências da atualidade. Além disso, o PC estimula a criticidade, a habilidade de trabalhar em equipe e a criatividade, competências essenciais para os cidadãos do século XXI (Lago e Aragón, 2022).

Os autores Selby e Woollard (2013), Barr e Stephenson (2011), Dagiené, Futschek e Stupuriené (2019), forneceram contribuições essenciais que ajudaram na construção do agrupamento das habilidades de PC que integram a APAP, a saber: decomposição, reconhecimento de padrões, abstração, criatividade, modelagem e simulação, algoritmos, implementação, especificação, interpretação de dados, representação de dados e avaliação.

Dessa forma, ao integrar as ideias de Papert, Piaget e Wing, é possível criar um ambiente educacional dinâmico, que estimule a criatividade, a resolução de problemas e o desenvolvimento cognitivo dos estudantes, preparando-os para enfrentar os desafios do mundo atual de forma autônoma e crítica.

2.3 Arquitetura Pedagógica

Para Carvalho, Nevado e Menezes (2007), as Arquiteturas Pedagógicas (APs) são estruturas de aprendizagem que envolvem a confluência de diferentes componentes, como abordagens pedagógicas, *software* educacional, *Internet*, inteligência artificial, Educação a Distância, concepções de tempo e espaço.

A teoria de Paulo Freire (2014), Pedagogia da Incerteza, baseia-se na ideia de que o conhecimento surge da movimentação, da dúvida e da necessidade de buscar novas alternativas, sendo essencial traduzir ideias em práticas pedagógicas concretas, contribui para a concepção e implementação de APs que valorizam a participação ativa dos estudantes, a reflexão crítica, a construção cooperativa do conhecimento e a busca por soluções para problemas reais, promovendo uma educação mais dinâmica, participativa e significativa para os estudantes.

Assim, segundo Souza e Menezes (2022) essas APs buscam educar para a autonomia e cooperação, fomentando a discussão de pontos de vista divergentes,



respeitando o pensamento próprio de cada indivíduo, mas também incentivando a interlocução e a cooperação entre os estudantes.

Para Silva *et al.* (2023), as APs buscam promover a aprendizagem como um processo ativo, cooperativo e reflexivo, onde os estudantes são incentivados a participar da construção do conhecimento, a pensar criticamente sobre os problemas apresentados e a interagir de forma significativa com os conteúdos estudados.

Nas APs, a cooperação funciona através da promoção de um ambiente de aprendizagem cooperativo, onde os estudantes são incentivados a trabalhar em conjunto para resolver problemas, criar soluções e compartilhar conhecimento (Souza e Menezes, 2022).

Deste modo, as APs incorporam os princípios do construtivismo ao proporcionarem um ambiente de aprendizagem que estimula a autonomia, a reflexão, a investigação e a cooperação dos estudantes, permitindo que construam ativamente seus conhecimentos e desenvolvam habilidades cognitivas e metacognitivas essenciais para sua formação.

3. Arquitetura Pedagógica Baseada no Pensamento Computacional para a Resolução Cooperativa de Problemas

A microarquitetura pedagógica baseada no Pensamento Computacional para a resolução cooperativa de problemas (AP2) faz parte da Arquitetura Pedagógica baseada no Pensamento Computacional para Apoio à Aprendizagem de Programação (APAP), uma AP complexa, que busca ajudar os estudantes a assimilar, criar, construir habilidades para resolver problemas, pois suas assimilações, criações intelectuais e construções cognitivas têm um extraordinário potencial pedagógico na compreensão do problema.

Este trabalho abordará as microarquiteturas que compõem a resolução do problema, especificamente a AP2, que integra a AP complexa denominada APAP. Além da AP2, a APAP inclui outras microarquiteturas pedagógicas como a compreensão do problema e construção do repositório de aprendizagem (AP1) e a depuração do algoritmo (AP3).

A APAP é descrita como uma estrutura complexa integrada por diferentes componentes que apoiam a aprendizagem construtivista da programação. Ela envolve a seleção consciente de problemas a serem utilizados no repositório de aprendizagem, bem como aspectos que auxiliam no planejamento e resolução desses problemas e na interação cooperativa entre os estudantes por meio de *Screencast* e da implementação em uma linguagem de máquina. Cada aspecto da APAP foi cuidadosamente projetado para desenvolver habilidades de resolução de problemas nos estudantes com base nos benefícios do PC, em um contexto de aprendizagem interativa e construtivista.

3.1. Estrutura da AP2

Na AP2, os estudantes construíram uma solução individual para o problema escolhido na AP1 e realizaram revisões em pares das soluções individuais e/ou em grupos, construindo, na sequência, uma solução do grupo, solução esta que será implementada na AP3 e/ou revisitada propiciando novas reflexões (Figura 1).

A atividade de aprendizagem cooperativa da AP2 se desenvolve em, pelo menos, três etapas: na construção da solução individual, na primeira revisão por pares e na segunda revisão por pares, conforme segue:

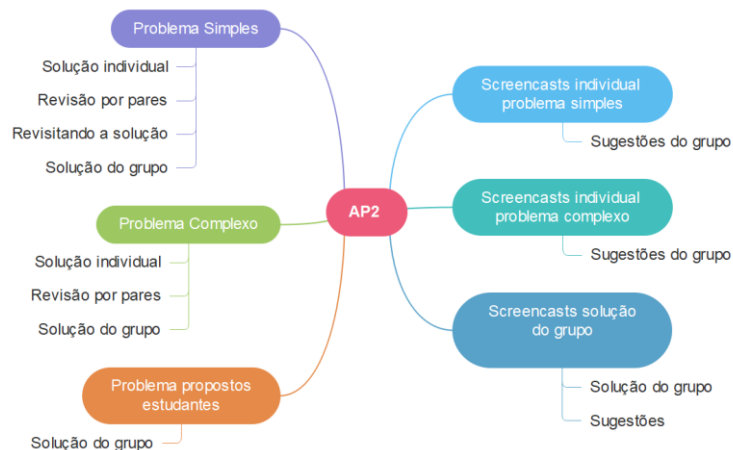


Figura 01: Estrutura da AP2
Fonte: Elaborado pelos autores (2024).

Etapa 1 - Construção da solução: O estudante é convidado a elaborar uma solução para o problema escolhido em um documento *online* e a produzir um *Screencast* explicando sua solução.

Etapa 2 - Primeira revisão por pares: O professor organiza os estudantes em duplas, buscando nessa fase evitar a formação de ilhas de revisões. Em seguida, distribui as soluções individuais aos grupos para que os estudantes analisem as respostas dos colegas, sem a identificação dos autores, propondo que busquem verificar se não existem inconsistências e ofereçam provocações e *feedback*.

O professor, então, devolve as soluções ao estudante autor, que, com base nas revisões recebidas e em seu próprio julgamento, pode fazer as alterações pertinentes na sua solução, se assim desejar.

Ao finalizar a revisão individual, os estudantes leem suas novas soluções individuais. Em seguida, o professor propõe que os estudantes formem pares e construam uma solução única do grupo baseando-se nas soluções individuais e suas revisões. Cada grupo de autores submete ao professor uma versão final da solução do problema, recebendo confirmação do professor.

Etapa 3 - Segunda revisão por pares: Após a entrega da solução do grupo, o professor inicia uma nova distribuição para revisão, novamente evitando a formação de ilhas de revisão. Ao revisar, o grupo deve oferecer provocações, considerando os pontos fracos e fortes da solução revisada, possibilitando ao grupo revisado a reflexão de modo que os ajude a construir uma solução melhor, submetendo a solução do grupo ao professor, recebendo sua confirmação.

Conforme a percepção do professor sobre a qualidade das soluções produzidas até este ponto, essa etapa pode se repetir com novas configurações para propiciar novos refinamentos. Por fim, os pares devem elaborar uma nova solução e apresentá-la aos participantes para que, em conjunto, possam construir uma ou mais soluções.

3.2 Organização do Experimento

Foi realizado um experimento de ensino para validação APAP. Inicialmente, foi feito um convite verbal a algumas turmas do 2.º semestre do curso Técnico Integrado em Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul, *campus* Coxim, para que os estudantes participassem como



voluntários do curso *online* “Programação: Aprendendo a Resolver Problemas”. Nove estudantes se inscreveram, sendo oito homens e uma mulher.

Este experimento foi mediado integralmente pelas tecnologias digitais. Para iniciarmos, foi disponibilizado um *link* do *Google Meet* por e-mail aos estudantes para o primeiro encontro síncrono. Todos os estudantes inscritos no curso compareceram, demonstrando grande interesse. Durante esse encontro, apresentamos a estrutura e o funcionamento do curso. Estabelecemos que teríamos dois encontros síncronos por semana, cada um com duração máxima de duas horas e teríamos atividades assíncronas. Também apresentamos as principais ferramentas que utilizaríamos ao longo do curso, como o *Moodle* para a plataforma de aprendizado, o *Google Meet* para videoconferências, o *Google Docs* para compartilhamento de documentos, o *WhatsApp* para comunicação e o *Loom* para recursos visuais.

3.3 O Experimento de Ensino

Iniciamos na AP1 com os vídeos de apoio sobre “*O que é Pensamento Computacional?*” e “*O que é Pensamento Crítico?*”, seguido de um fórum sobre o que os estudantes entenderam de “Como resolver problemas usando o Pensamento Computacional?”. Em seguida, apresentamos problemas escolhidos pelo mediador do teste *Bebras Australia Computational Thinking Challenge*, traduzidos para a Língua Portuguesa, dos anos de 2018, 2020 e 2022.

A escolha dos problemas do Desafio Bebras se constitui em uma importante ferramenta, tendo em vista que os problemas foram selecionados de um evento internacional, utilizando habilidades do PC.

Imediatamente, solicitamos aos estudantes que, individualmente, escolhessem por ordem de prioridade um dos seis problemas do teste *Bebras Australia Computational Thinking Challenge* propostos pelo mediador.

Durante sete encontros síncronos e atividades assíncronas, os estudantes participaram da microarquitetura pedagógica AP2. Essa abordagem incluiu a utilização de fóruns de discussão, repositórios cooperativos, vídeos de apoio que exploraram os conceitos de “Pensamento Computacional” e “Pensamento Crítico”, além da apresentação das soluções individuais, revisão por pares e solução do grupo dos problemas propostos.

Na “compreensão e resolução do problema (AP2)” os estudantes construíram uma solução individual para o problema escolhido (Diário secreto – disponível no *link*: <https://shorter.me/BLZGI>) e realizaram revisões em pares das soluções individuais e/ou em grupos, construindo na sequência uma solução do grupo, solução esta que será implementada na AP3 e/ou revisitada propiciando novas reflexões.

Nessa AP os participantes receberam informações que os subsidiaram na construção da solução individual do problema. Sobre isso, P3 e P9 registraram as suas soluções em um documento *online* e na produção dos *Screencasts* individuais do problema escolhido:

P3: “Primeiro temos que ver como elas acharam o padrão para decifrar o nome do irmão dela, segundo pegar esse padrão e usar para descobrir o outro nome, terceiro após descobrir o nome fazer um novo alfabeto com os códigos para entender o tipo de linguagem.”

P9: “Primeiro eu tentei entender o Problema, observei as linhas, imaginei que não era o formato do código, pois tinha a tabela, então comecei a ligar as linhas, contanto na vertical e na horizontal, logo após isso, descobri a resolução do Problema, o nome nesse código criptografado era Josef.”



P2 e P4 registraram também:

P2: “Primeiro eu tentei entender o Problema, observei as linhas, imaginei que não era o formato do código, pois tinha a tabela, então comecei a ligar as linhas, contando na vertical e na horizontal, logo após isso, descobri a resolução do Problema, o nome nesse código criptografado era Josef.”

P4: “Resolvi o problema contando as linhas horizontais e verticais e deu no nome Josef”

É interessante destacar que ambos, P2 e P4, buscaram relacionar conhecimentos prévios na busca de construir a solução para o problema ao evidenciar a decomposição quando *“dividi o problema para entender o problema criptografado em etapas menores, quando observa as linhas e conta as colunas para descobrir a resolução do problema”* (P2), o reconhecimento de padrões e a especificação ao *“Ao usar padrões nas informações do problema criptografado para entender que não se tratava do formato do código, mas sim de ligar as linhas e contar na vertical e horizontal e ao definir o problema com precisão, identificar os requisitos e decompor o processo em etapas gerenciáveis”* (P4), e por fim quando busca representar e interpretar os dados ao lidar com *“a representação simbólica dos dados no problema criptografado e no código com o nome “Josef” e extrair significado dos dados criptografados para descobrir o nome “Josef” como a solução para o problema”* (P4).

De modo geral, os participantes demonstraram uma transformação cognitiva ao passar da interpretação das pistas para a elaboração de um método ou algoritmo para decifrar o código e encontrar a solução, quando buscaram resolver o problema envolvendo uma análise individual das pistas e a construção de uma estratégia sequencial de resolução, utilizando seus conhecimentos prévios para entender o problema.

3.3.1 Revisão por Pares

A distribuição de revisões, levou em conta as diferentes fases do desenvolvimento do pensamento formal (Piaget, 1973) dos participantes, distribuindo-os, de modo a favorecer a interação entre revisores e revisados, com o intuito de incentivar o desenvolvimento cognitivo.

Após as revisões o mediador solicitou aos participantes que revisassem suas soluções e as contribuições dos seus revisores e participassem de um fórum após assistir os *Screencasts* dos seus pares, que explicavam como haviam pensado na solução do problema e quais dificuldades encontraram para construir a solução.

Em seguida, de posse das sugestões dos seus pares, reescreveram sua nova solução, com base na(s): a) própria solução; b) na solução dos seus revisores; c) solução de seus revisados; d) revisões que fizeram sobre cada um dos revisados, tendo a oportunidade de refletir sobre esse material e construir o seu posicionamento final.

Os indícios observados em P1 mostram que as revisões resultaram na reconstrução das soluções iniciais. A solução original foi enriquecida com reescritas após as revisões.

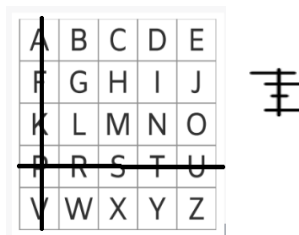
Solução P1 antes das revisões: “Neste problema, encontramos um “Exemplo” de solução onde não nos explicam a forma de resolvê-lo e logo em seguida nos dão um objetivo. Onde, ele deve explicar a forma de solucionar o problema. Em minha percepção, as linhas na horizontal e vertical dadas pelo problema, podem trazer a solução, já que a decodificação está presente nelas. A forma de descobrir como solucionar a decodificação está no cruzamento entre as linhas horizontais e verticais de acordo com a tabela em ordem alfabética.”



Revisor R1: “Bom para ser mais prático acho que primeiro teríamos que observar o já resolvido e buscar uma semelhança, por exemplo observando o já resolvido percebemos que as linhas verticais é qual coluna está a letra e as horizontais em qual quadradinho daquela coluna está a letra assim descobriremos qual o nome, no caso ali o nome é Josef.”

Revisor R2: “A resolução na minha opinião está quase boa, para mim precisava só de mais uma explicação como você usou as linhas para achar o nome.”

Revisor R3: “você explicar melhor como resolver, você não está falando todo o processo por exemplo tem 4 linhas verticais e 1 horizontal então faremos assim que daria na letra .”



As percepções de P1 após as revisões sinalizam acréscimos e reescritas em suas soluções:

P1 após as revisões: “Para elas resolverem isso elas precisam pensar nas pistas que elas tem até agora. A primeira pista que elas têm é que Luciana usou linhas horizontais e verticais usando a tabela de letras, pensando nessa tabela como uma matriz. Já que elas conseguiram descobrir o nome do irmão (pavel), então pensando nisso pode pensar que as linhas horizontais e verticais são usadas para selecionar as letras na tabela. Podemos pensar - Analisar a tabela - Podemos entender na tabela como as letras estão organizadas. A tabela está em ordem alfabética seguindo esse padrão. Observar - observar as linhas horizontais e verticais que usaram para criptografar a palavra . Elas devem contar quantas linhas horizontais e verticais foram traçadas. Mapear as letras selecionadas - Pensando nas linhas horizontais e verticais traçadas, elas devem começar a mapear as letras selecionadas. Assim formando a palavra. Seleção - Depois de selecionar todas as letras, elas devem juntar as letras na ordem correta para formar uma palavra, assim formando o nome do cachorro. O nome do cachorro é: Josef”

As habilidades do PC identificadas em P1 após as revisões foram: decomposição, reconhecimento de padrões, abstração, criatividade, modelagem e simulação, algoritmos, especificação e avaliação. Quando dividiu o problema em problemas menores, buscou o reconhecimento de padrões, interpretação da representação dos dados e extração de significado da informação contida na representação da palavra "JOSEF". Esses indícios são percebidos quando o estudante constrói sua solução, revisa as dos seus pares e também é revisado.

A solução do problema traz algumas informações percebidas por P1, dentre elas: A primeira pista é que Luciana usou linhas horizontais e verticais utilizando a tabela de letras, pensando nessa tabela como uma matriz. Quando se concentrou em relacionar informações e identificar padrões a partir das pistas, observou as linhas horizontais e verticais que usou para criptografar a palavra e mapear as letras selecionadas. Assim, P1 construiu estruturas que o levou à abstração, modelagem e aplicação de padrões identificados para a resolução final do problema.



Buscando consolidar uma única solução para o grupo, solicitamos que, entre todas as propostas apresentadas, os estudantes desenvolvessem uma solução única que seria adotada por todos. A solução escolhida está descrita a seguir:

"Inicialmente, vamos identificar o padrão usado para decifrar o nome do irmão, observando as linhas e colunas utilizadas na tabela. Criando assim um novo alfabeto com base nesse padrão e aplicá-lo para desvendar o outro nome. Pois, ao analisar as linhas e colunas, é possível determinar a posição das letras. Portanto, vamos seguir esses passos: entender o padrão inicial, aplicá-lo para encontrar o segundo nome e, por meio disso, decifrar a linguagem utilizada. A figura abaixo nos mostra como foi criado esse padrão"

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	R	S	T	U
V	W	X	Y	Z

A solução do grupo demonstra a transição para a modelagem e abstração ao considerar a organização alfabética da tabela e criar um novo padrão para decifrar o nome.

4. Considerações finais

Neste artigo, apresentamos uma contribuição para responder à pergunta: “Como melhorar a aprendizagem de programação dos estudantes do Ensino Médio Técnico?” Para isso, concebemos a AP2 como parte integrante da composição da APAP, ancorada conceitualmente no *Framework* Arquiteturas Pedagógicas, Pensamento Computacional e Epistemologia Genética.

Após a aplicação do experimento de ensino delimitado nas AP1 e AP2, podemos concluir que foi possível verificar indícios de que a APAP, proposta como suporte ao ensino e aprendizagem de programação, auxilia aos estudantes a superar desafios, especialmente no entendimento e na proposição de problemas.

Entre as contribuições destacamos:

a) Ao integrar a APAP com o PC e a aprendizagem cooperativa em uma abordagem pedagógica, os estudantes vivenciaram uma experiência de aprendizagem mais abrangente, desenvolvendo habilidades tanto em resolução de problemas quanto em atividades cooperativas;

b) O desenvolvimento de habilidades específicas do PC, como decomposição, reconhecimento de padrões, abstração, criatividade, modelagem e simulação, algoritmos, especificação e avaliação. Quando os estudantes utilizaram essas habilidades em suas soluções para os problemas, estavam, de forma implícita, demonstrando a compreensão e aplicação dos princípios do PC.

Com esta constatação, podemos inferir que a AP2 pode auxiliar aos estudantes no desenvolvimento de habilidades de resolução de problemas, particularmente na programação de computadores, permitindo que construam suas soluções dos problemas, ao interagir, revisar e serem revisados por seus pares.



Referências

- BARR, Valerie; STEPHENSON, Chris. Bringing 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.
- CARVALHO, Marie Jane Soares; NEVADO, RA de; MENEZES, Crediné Silva de. Arquiteturas pedagógicas para educação a distância. **Aprendizagem em rede na educação a distância: estudos e recursos para formação de professores. Porto Alegre: Ricardo Lenz**, v. 1, p. 36-52, 2007.
- DAGIENÈ, Valentina; FUTSCHEK, Gerald; STUPURIENÈ, Gabrielè. Creativity in solving short tasks for learning computational thinking. **Constructivist Foundations**, v. 14, n. 3, p. 382-396, 2019.
- FREIRE, Paulo. **Pedagogia da autonomia: saberes necessários à prática educativa**. Editora Paz e terra, 2014.
- FREIRE, Paulo. **Por uma pedagogia da pergunta**. Editora Paz e Terra, 2014.
- LAGO, Muriel; ARAGÓN, Rosane. O pensamento computacional sob o olhar da microgênese cognitiva. **Revista Novas Tecnologias na Educação**, v. 20, n. 2, p. 217-227, 2022.
- OLIARI, Marco Antônio Milaneze et al. Introcomp: Reflexões de uma Década de Desafios e Conquistas no Ensino de Programação para a Rede Pública de Ensino. In: **Anais do Simpósio Brasileiro de Educação em Computação**. SBC, 2021. p. 173-182.
- PAPERT, Seymour; VALENTE, Jose Armando; BITELMAN, Beatriz. **Logo: computadores e educação**. Brasiliense, 1980.
- PIAGET, Jean. **L'epistemologia genetica**. Edizioni Studium Srl, 2016.
- PIAGET, Jean. **Biologia e conhecimento: ensaio sobre as relações entre as regulações orgânicas e os processos cognoscitivos**. 1973.
- SELBY, Cynthia; WOOLLARD, John. Computational thinking: the developing definition. 2013.
- SILVA, Francisco Xavier da; MENEZES, Crediné Silva de; CASTRO JUNIOR, Alberto Nogueira de. Arquitetura Pedagógica Baseada no Pensamento Computacional para a Compreensão do Problema & Portfólio de Aprendizagem. **Revista Novas Tecnologias na Educação**, v. 21, n. 2, p. 1-12, 2023.
- SOUZA, Patrícia Maurer; MENEZES, Crediné Silva de. Uma Arquitetura Pedagógica para o Desenvolvimento do Pensamento Computacional em Contexto Interdisciplinar. **Revista Novas Tecnologias na Educação**, v. 20, n. 2, p. 290-300, 2022.
- WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.
- WING, Jeannette. Computational thinking's influence on research and education for all. **Italian Journal of Educational Technology**, v. 25, n. 2, p. 7-14, 2017.