

# Arquitetura Pedagógica Baseada no Pensamento Computacional para a Compreensão do Problema & Portfólio de Aprendizagem

## Pedagogical Architecture Based on Computational Thinking for Problem Understanding & Learning Portfolio

Francisco Xavier da Silva, IFMS - Campus Coxim - Coxim - MS – Brasil,  
[francisco.silva@ifms.edu.br](mailto: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](mailto: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](mailto:alberto@icomp.ufam.edu.br), <https://orcid.org/0000-0002-1752-8667>

***Abstract.** The article presents a Pedagogical Architecture Based on Computational Thinking for Understanding the Problem and the construction of a Learning Portfolio (API). A case study was carried out in which students were presented with scenarios and built a cooperative portfolio. The analysis of the scenarios revealed how the students developed new schemes in the elaboration and review of the problems. Therefore, API proves to be a valuable Pedagogical Architecture for teaching programming supported by computational thinking.*

***Keywords:** Computational Thinking; Pedagogical Architecture; teaching programming*

***Resumo.** O artigo apresenta uma Arquitetura Pedagógica Baseada no Pensamento Computacional para a Compreensão do Problema e a construção de um Portfólio de Aprendizagem (API). Foi realizado um estudo de caso em que os estudantes foram apresentados a cenários e construíram um portfólio cooperativo. A análise dos cenários revelou como os estudantes desenvolveram novos esquemas na elaboração e revisão dos problemas. Portanto, a API se mostra uma Arquitetura Pedagógica valiosa para o ensino de programação apoiada no pensamento computacional.*

***Palavras-chave:** Pensamento Computacional; Arquitetura Pedagógica; ensino de programação*

### 1. Introdução

A disciplina de Introdução à Programação está presente em vários cursos técnicos de computação, e, normalmente, é oferecida no primeiro ano, tendo como objetivo oportunizar o desenvolvimento do aluno à lógica de programação, tornando-o capaz de construir algoritmos corretos e otimizados. Porém, aprender a programação é muito

mais que escrever um conjunto de linhas de código numa dada linguagem e requer uma grande diversidade de pensamentos, compreensões e ritmos, ou seja, é uma arte, pois existem muitas maneiras de entender, combinar, reconhecer padrões e codificar a solução dos problemas, com alguma criatividade (KNUTH, 1978).

As disciplinas responsáveis por apresentar a programação para os alunos, na sua maioria, não consideram os conhecimentos prévios dos alunos, deixando-os frustrados. Dentre os motivos para essa frustração, destacam-se: dificuldade em compreender o problema; a inadequação dos métodos pedagógicos aos estilos de aprendizagem dos alunos; preocupação excessiva com detalhes de sintaxe da linguagem utilizada; punição ao erro, inviabilizando o pensamento crítico e criativo. Essas dificuldades culminam em baixa motivação, pouca autoestima e em um elevado índice de reprovação e desistência (PROULX, 2000).

Na busca por contribuir para resolução desses problemas, diferentes abordagens pedagógicas têm sido estudadas e propostas, na literatura, tais como; salas de fuga educacionais (LÓPEZ-PERNAS *et al.*, 2019); uso de linguagens especiais, como Scratch (MORALES-URRUTIA *et al.*, 2021); sala de aula invertida e programação no contexto de histórias digitais (ZHA *et al.*, 2020); aprendizagem cooperativa (GARCIA, 2021); aprendizagem baseadas em jogos (GIANNAKOULAS e XINOGALOS, 2018).

Dessa forma, nossa contribuição está estruturada em uma Arquitetura Pedagógica (AP) (CARVALHO, NEVADO E MENEZES, 2005) ancorada nos pressupostos teóricos da epistemologia genética (PIAGET, 2016) e baseada no pensamento computacional (WING, 2006), mediada pela revisão por pares (MAZUR, 1999). Com o objetivo primordial de ajudar os estudantes do ensino médio na construção de conhecimento sobre programação propomos uma Arquitetura Pedagógica Baseada no Pensamento Computacional para a Compreensão do Problema & Portfólio de Aprendizagem que faz parte da Arquitetura Pedagógica Baseada no Pensamento Computacional para Apoio à Aprendizagem Introdutória de Programação (APAP)

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 Compreensão do Problema & Portfólio de Aprendizagem e, finalmente, na Seção 4, são expostas às considerações finais.

## **2. Fundamentação teórica**

Nesta seção, serão abordados conceitos essenciais que constituem a fundamentação teórica da proposta, destacando-se interações na aprendizagem, no ponto de vista Piagetiano, pensamento computacional e a arquitetura pedagógica.

### **2.1 Tríade da aprendizagem de Piaget**

Segundo (Piaget, 2016), o conhecimento prévio do aluno é fundamental, nas interações com os seus pares, nas trocas de informações e na forma de revisar as atividades; construindo, assim, situações de desequilíbrios momentâneos e reequilíbrios.

A epistemologia genética busca identificar leis gerais que permitam compreender o porquê dessa sucessão nas formas de conhecimento e dos mecanismos que as engendram. Piaget e Garcia (1982) identificam, na psicogênese e na história das ciências, mecanismos comuns na construção desses conhecimentos. Para esses autores,

o elemento que consideram de maior importância, nesse estudo comparativo, consiste na tríade dialética constituída pelas etapas INTRA, INTER e TRANS. Tais etapas ajudam no desenvolvimento de novos esquemas, dinâmicos e mutáveis, que crescem de acordo com certos mecanismos, desenvolvendo todos os domínios do pensamento humano.

Ainda, segundo Piaget e Garcia (1982), a generalidade que explica a tríade de aprendizagem, em todos os domínios do pensamento e em todos os níveis de formulação, são as condições que as leis gerais de assimilação e equilíbrio impõem a toda aquisição cognitiva. Todavia, a assimilação não pode ser pura porque ao incorporar novos elementos aos esquemas anteriores, a inteligência modifica sem cessar estes últimos para juntá-los aos novos dados. Contudo, inversamente, as coisas não são mais conhecidas em si mesmas, já que esse trabalho de acomodação não é possível a não ser em função do processo inverso de assimilação (PIAGET e GRECO, 1974).

Os processos de assimilação e equilíbrio descritos por Piaget e Garcia (1982), descrevem o desenvolvimento do pensamento humano. Com base nessa tríade, o pensamento computacional se destaca como uma abordagem que utiliza princípios lógicos e algorítmicos para promover a criação de novos esquemas cognitivos e a adaptação constante do pensamento humano às novas informações e desafios.

## **2.2 Pensamento Computacional**

O artigo seminal de Wing é "Computational Thinking," publicado em 2006 definiu o Pensamento Computacional (PC) como "um processo mental que generaliza um problema real a um problema computacional, o pensamento de solução do problema que envolve habilidades como abstração, decomposição, algoritmos, iteração e avaliação" que envolve a resolução de problemas, a concepção de sistemas, baseando-se nos conceitos fundamentais da computação. Como um processo de pensamento envolvido na formulação e reformulação de um problema e na expressão de sua(s) solução(s) de forma que um computador — humano ou máquina — possa efetivamente executar.

Ainda, segundo a autora, alguns problemas podem ser simples, enquanto outros são de natureza mais complexa, independentemente dos elementos envolvidos nos problemas podemos sempre questionar: "O quão difícil é de resolver?" e "Qual a melhor forma de resolvê-lo?", sendo assim podemos pensar no problema, supostamente difícil, de forma diferente, em um problema possível de resolver através da redução, incorporação, transformação ou simulação.

O PC tem se mostrado uma abordagem altamente eficaz tanto em contextos acadêmicos quanto em situações da vida real, conforme evidenciado por pesquisadores, entre eles Bati *et al.* (2018). É importante ressaltar que existem diversas definições do PC, provenientes de estudiosos como Curzon *et al.* (2009). No entanto, entre todas essas definições, o PC é comumente reconhecido como um componente fundamental para a comparação e análise de diferentes algoritmos, bem como para a resolução criativa de problemas por meio do pensamento crítico.

Com base num sistema de categorização do Concurso Internacional Bebras de Informática e Literacia Informática Dagiené e Futschek (2008) é possível identificar no modelo que a aprendizagem Bebras concentra-se em conceitos de informática, apoiando a compreensão e o desenvolvimento do pensamento computacional. O conceito central

introduzido nas tarefas Bebras e dividir amplamente o conteúdo da tarefa em uma destas cinco áreas (categorias):

- Algoritmos e programação
- Dados, estruturas de dados e representações (inclui gráficos, mineração de dados)
- Arquitetura e processos do computador (inclui tudo relacionado ao funcionamento do computador - agendamento, processamento paralelo)
- Comunicações e redes (inclui criptografia, computação em nuvem)
- Interação (Interação Humano-Computador, IHC), sistemas e sociedade

De forma semelhante, Korkmaz Çakir e Özden, (2017) identificou cinco habilidades distintas que pode ser compreendido como um guarda-chuva que abrange pensamento algorítmico, cooperatividade, criatividade, pensamento crítico e resolução de problemas. Essas perspectivas destacam a diversidade de habilidades envolvidas no PC e a sua capacidade de promover uma abordagem holística para lidar com desafios complexos do dia a dia.

### **2.3 Arquitetura Pedagógica**

Segundo Carvalho, Nevado e Menezes (2007), as APs são definidas como suportes estruturantes, maleáveis e adaptáveis a diferentes enfoques temáticos: pedagógicas relacionais, métodos ativos, abordagens cooperativas, tecnologias digitais, inteligência artificial e reconfigurações dos tempos e espaços de aprendizagem, entre outros.

Os elementos essenciais na descrição e concepção de uma AP são: O domínio de conhecimento; os objetivos educacionais; o conhecimento prévio; as dinâmicas interacionista-problematizadoras; suporte computacional (tecnológico); mediações pedagógicas distribuídas (aprendizagens cooperativas) e avaliação processual e cooperativa das aprendizagens (MENEZES, CASTRO-JUNIOR e ARAGON, 2020).

Neste ponto, analisando as afirmações Carvalho, Nevado e Menezes (2007), um dos mecanismos da AP pedagógicos que orienta a Arquitetura Pedagógica é a aprendizagem cooperativa das soluções e as possibilidades, segundo Piaget (1986) de explorar os possíveis não observáveis, informações cognitivas que não se dá através da percepção do objeto e o necessário que para ele é um produto das composições inferenciais do sujeito e também não é um observável. Com visita a diferentes soluções elaboradas por seus pares.

Para Pereira (2021), um dos mecanismos de mediação distribuída de uma AP é a revisão por pares que favorece os alunos na construção de novos conhecimentos. Apoiados, por uma participação ativa do processo, os discentes incubem-se de ler e revisar os argumentos de outros estudantes, analisando a partir dos seus conhecimentos e oferecendo provocações que lhes oportunizem caminhos para reflexão e reconstruções, tornando-os um mediador, papel, geralmente exercido pelo professor. Ainda, os alunos oferecem a seus pares um feedback em uma linguagem que, em geral, está mais próxima do que a dos professores.

### **3 Arquitetura Pedagógica Baseada no Pensamento Computacional para a Compreensão do Problema & Portfólio de Aprendizagem**

O presente estudo utiliza-se dos fundamentos científicos metodológicos do conhecimento aplicado na Design Science Research (DSR) concebidas no paradigma

epistemológico das Ciências do Artificial (PIMENTEL, FILIPPO e SANTOS, 2020), para a pesquisa no ensino aprendizagem de programação. Para avaliar o artefato proposto neste trabalho, foi acoplada ainda o método do Estudo de Caso (EC) descrito na perspectiva do Yin (2010), e a análise de conteúdo de (BARDIN, 2016) juntamente com Método de Exploração Crítica (INHELDER, BOVET E SINCLAIR, 1977), para sistematizar a produção de dados que gerariam os aportes para AP.

A Arquitetura Pedagógica Baseada no Pensamento Computacional para a Compreensão do Problema & Portfólio de Aprendizagem faz parte de uma Arquitetura Pedagógica chamada Arquitetura Pedagógica Baseada no Pensamento Computacional para Apoio à Aprendizagem Introdutória de Programação (APAP) que busca ajudar os alunos a assimilar, criar, construir habilidades para resolver problemas, pois suas assimilações, suas criações intelectuais, suas construções cognitivas têm um extraordinário potencial pedagógico na compreensão do problema.

Ao programar, o sujeito constrói o conhecimento quando compreende o problema, constrói modelo de solução mental e por fim implementa e o trata do erro (parte executável). A APAP se constitui em uma AP complexa e é composta de três AP simples que fluem umas para as outras na operação real do desenvolvimento da aprendizagem de programação.

Trataremos, neste trabalho da AP simples, que compõe a escolha dos problemas a serem utilizados no portfólio de aprendizagem construtivista (AP1), que compõe a APAP juntamente com os aspectos que apoiam a planejamento e resolução do problema (AP2), e a interação cooperativa entre os estudantes em formato de video (screencast) (AP3). Conforme (Figura 01).



**Figura 01: Estrutura da APAP**

Cada uma das AP simples que compõem a APAP foi cuidadosamente projetada para criar uma escolha consciente para construção de soluções de problemas baseados nos conceitos e benefícios de cada uma das habilidades do PC nos alunos por meio de diferentes atividades realizadas em ciclos interativos. A APAP foi desenvolvida para responder a perguntas-chave que ajudam o aluno a entender como aprender programação.

A construção do conhecimento se constitui com os agrupamentos possíveis da APAP, não se restringindo em modelos fixos, mas possibilita muitas generalizações para auxiliar no desenvolvimento e na aprendizagem dos estudantes. Neste contexto, destacamos AP que compõe a escolha dos problemas a serem utilizados e o portfólio de aprendizagem construtivista.

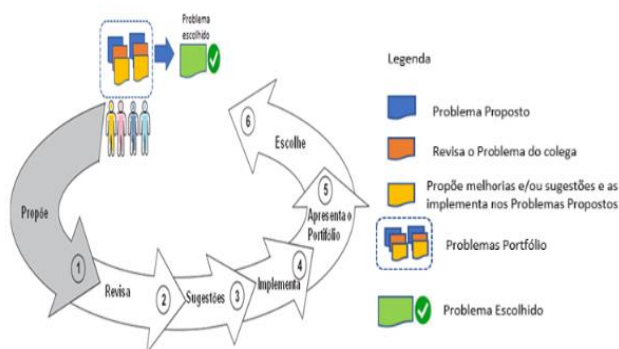
Essa AP se constitui das competências do pensamento computacional, o professor e/ou estudantes podem formular problemas, ou buscar identificar problemas, que serão escolhidos conforme a temática de interesse do tópico a ser estudado, as escolhas farão parte de um portfólio de problemas. Após realizar essa formulação ou identificação dos problemas, os alunos cooperativamente escolherão, dentre o portfólio, os problemas que serão resolvidos.

### 3.1 Estrutura da AP1.

A AP1 é composta por apresentação de cenários, Compreensão do Problema e construção do Portfólio de Aprendizagem Construtivista. Através da apresentação de cenários, os estudantes são incentivados a pensar sobre valores e críticas sociais, para abordar a realidade do dia a dia dos jovens e evidenciar que o PC pode ser uma estratégia interdisciplinar de formalização do pensamento.

Em seguida, os estudantes registram suas aspirações, propõe cenários e elaboraram um "Portfólio cooperativo", que é um documento compartilhado online onde todos os participantes escrevem suas propostas de problemas, e definem quais iram ser resolvidos, para construir o conhecimento sobre a temática. Esse portfólio é examinado pelos pares e pelo professor mediador, que levantam questionamentos que contribuem para a qualificação dos problemas propostos por cada estudante.

Nessa perspectiva, a (Figura 02) apresenta os ciclos de relevância portfólio cooperativo, que se inicia com as propostas individuais dos problemas a serem resolvidos, passando por revisões e sugestões dessas propostas individuais por seus pares de forma cooperativa até a escolha do grupo por um problema a ser resolvido por todos, desta forma, o ciclo da relevância portfólio construtivista e a representação da conexão do conhecimento individual com as trocas de experiências de seus colegas, quando o estudante é incentivado a propor um problema, revisar e sugerir melhorias aos problemas dos colegas, oferecendo considerações, perguntas, características e exemplos para melhorar a proposta a ser implementada na AP subsequente, ou se escolhe descartar o problema e iniciar outro ciclo de relevância.



**Figura 02: Ciclo da relevância portfólio cooperativo**

A atividade é realizada de forma síncrona e assíncrona para favorecer a construção cooperativa do portfólio. Os estudantes são incentivados a pensar além de seus livros didáticos e gerar problemas que ajudem seus colegas a pensarem e aplicarem os conceitos básicos de programação.

### 3.2 Organização do Experimento

Foi realizado um experimento piloto, para validação AP1, inicialmente, foi feito um convite verbal a algumas turmas do 2.º semestre do curso técnico integrado em desenvolvimento de sistemas do Instituto Federal do Mato Grosso do Sul campus Coxim, para que os estudantes participassem como voluntários do curso online, “programação: Aprendendo a Resolver Problemas”, escolha desses semestres se deu por apresentar na matriz curricular do curso as disciplinas de introdução de Lógica de Programação de Computadores 1 e Linguagem de Apresentação e Estruturação de Conteúdos 1.

Este experimento foi mediado integralmente pelas tecnologias e para iniciarmos foi disponibilizado um link do Meet por e-mail aos estudantes, para o primeiro encontro síncrono, e 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 expomos as principais ferramentas que utilizaremos ao longo do curso, como o Moodle para a plataforma de aprendizado, o Meet para videoconferências, o Google Docs para compartilhamento de documentos, o WhatsApp para comunicação e o Loom para recursos visuais.

### 3.3 O Experimento

Durante seis encontros síncronos e atividades assíncronas, os estudantes participaram da micro arquitetura pedagógica estruturada denominada “Compreensão do Problema & Portfólio de Aprendizagem (AP1)”, essa abordagem incluiu a utilização de fóruns de discussão, portfólios cooperativos, vídeos de apoio que exploraram os conceitos de "Pensamento Computacional" e "Pensamento Crítico", além da apresentação e criação de cenários tanto simples quanto complexos. Iniciamos a 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 “Como resolver problemas usando o pensamento Computacional?”

Como podemos observar, **Participante2 (P2)** e **Participante6 (P6)**, apresentaram seus entendimentos de conceitos, tais como: decomposição, generalização, abstração, reconhecimento de padrão e algoritmo, ou seja, criou novos esquemas mentais apoiados em esquemas adjacentes.

Vale ressaltar que serão conservadas nas transcrições dos protocolos as falas tais quais apresentadas pelos sujeitos.

**P2:** “Eu acredito que sim, pois é uma forma de organizar as atividades de sua rotina, e com isso uma melhor organização dos problemas do dia a dia. Um exemplo disso é uma receita, você tem que ter os ingredientes, ter as medidas e o modo de preparo, e com isso você conclui a receita.”

**P6:** “Sim, já que conseguimos focar no que é importante no problema e pensar como o computador entende o que pedimos.”

Com base nos textos, podemos inferir que o pensamento computacional foi percebido pelos estudantes como uma abordagem que vai além da computação, podendo

ser aplicado em vários contextos do cotidiano, quando expressam seu entendimento e a importância de decompor um problema em partes menores para facilitar a resolução (decomposição) e a capacidade de focar no essencial, ignorando o irrelevante (abstração), identificar padrões nos problemas e aplicar passo a passo, ou seja, sequências de etapas bem definidas para resolver cada parte do problema. Essa percepção evidenciou o interesse dos estudantes em construir novos caminhos para suas aprendizagens.

Em seguida, é apresentado cenários escolhidos por parte do mediador, do teste Bebras Austrália Computational Thinking Challenge, traduzidos para língua Brasileira, dos anos de 2018, 2020 e 2022. A escolha dos cenários do Desafio Bebras se constitui de uma importante ferramenta, tendo em vista que é um evento internacional que envolve os estudantes de maneira prática do nível fácil ao difícil a construir soluções de problemas utilizando habilidades do pensamento computacional, disponibilizados na aba Guias de soluções.

Na sequência, os estudantes responderam a um fórum sobre o que um problema precisa ter para ser considerado um desafio, ao ser questionados sobre suas respostas, P2, P6:

*P2: “Eu acredito que um problema que utiliza uma coisa mais cotidiana, fica mais interessante para a resolução, e também mais desafiador, a pessoa tenta utilizar todo seu conhecimento de sua vida, e com a resolução ele tem uma visão diferente daquele problema, com isso uma melhor lógica e ganho de tempo em sua vida cotidiana.”*

*P6: “Deve ser desafiador, ao mesmo tempo te fazer desenvolver em algum ponto seu raciocínio, ser impossível também é interessante, já que é interessante explicar porque é impossível.”*

Para os estudantes um problema/cenário precisa ser relevante, desafiador e capaz de estimular o pensamento crítico e criativo. De modo a Integrar diferentes esquemas mentais foram compartilhadas as escolhas individuais dos problemas/cenários apresentados pelo professor, foi sugerido que a atividade assíncrona seja a reflexão sobre as escolhas e a construção dos cenários propostos pelos estudantes.

Imediatamente, foi solicitado no primeiro momento que individualmente eles escolhessem por ordem de prioridade um dos cenários simples e um complexo, do teste Bebras Austrália Computational Thinking Challenge, depois discutissem suas escolhas em grupo para consolidação de um cenário único, simples e um complexo, para todos. A escolha dos cenários do Desafio Bebras se constituiu de forma cooperativa em grupo, após uma seleção individual. Dos 6 cenários propostos pelo professor, os que foram escolhidos são **mesa de jantar** e **João e Maria**, conforme o link: <https://shorter.me/BLZGI>, e os que foram propostos pelos estudantes os escolhidos foram cenários, **sala de aula** e **falta de água**.

*Cenário Sala de aula: “Uma sala de aula com 20 alunos está sendo muito criticada por ser complicada de dar aula por questões de conversa. Os professores entraram em consenso e decidiram criar um mapa de sala, o mapa de sala foi criado para os alunos sentarem nas carteiras em ordem alfabética, porém existiam alguns alunos que não poderiam ficar perto um do outro, exatamente pelo fato de conversarem muito. (obs.: são 5 colunas e dentro de cada coluna existem 4 carteiras). Existiam 5 alunos que mais dificultavam o aprendizado em aula, são eles: Carlos, Fernando, João, Marcos e Roberto. Esses alunos não podiam ficar juntos na fileira nem na vertical, na horizontal e nem na diagonal. Para resolver esse cenário temos que A,B,C,D,E são para coluna, e 1,2,3,4 são para a linha. Exemplo: A3 = A primeira coluna da linha 3.”*



Ao analisar o cenário sala de aula escolhido, na perspectiva do pensamento computacional e da construção de conhecimento vivenciadas na interação dos estudantes com API, percebemos que o problema apresentado busca organizar os estudantes para evitar que os problemáticos ficassem próximos, seguindo um padrão, identificando as restrições dadas, a capacidade de dividir a sala em partes menores e mais gerenciáveis para encontrar melhores locais para organizar os estudantes nas carteiras conforme as restrições dadas, focando apenas nos aspectos relevantes e ignorando detalhes não essenciais de modo a fazer ajustes e melhorias para garantir que atenda a todas as restrições. Ou seja, (estágio intra) o problema está relacionado à organização espacial das carteiras, incorporando princípios de organização espacial e lógica para atender às restrições apresentadas.

*Cenário Falta de água: “A falta de acesso à água: Em uma comunidade rural eles sofrem com falta de água potável, e o único jeito deles conseguirem água é indo em um lago próximo ao vilarejo, sendo que no vilarejo existem 21 casais. Sendo assim, como faria para que todos os moradores do vilarejo possam ir buscar a água durante a semana (7 Dias), sem repetir nem um dia da semana, revezando entre eles.”*

O cenário Falta de água apresenta habilidades de desenvolvimento cognitivo e do pensamento computacional, são observáveis quando o problema pedi para dividir as tarefas em partes menores, ao planejar a distribuição da ida ao lago ao longo dos sete dias da semana. Também representa o problema de forma simplificada e desenvolve um plano ou algoritmo para distribuir os moradores de forma equitativa e eficiente ao longo dos dias, assegurando que todos tenham acesso à água potável (estágio inter). Levando em consideração a interação entre os moradores e o recurso limitado (água potável), bem como as decisões afetarão o sistema na totalidade.

### **3.3.1 revisão por pares dos cenários propostos pelos estudantes**

Partindo-se da premissa que para aprender é necessário fazer para compreender, apontada por Piaget (1973), os cenários **sala de aula** e **falta de água** propostos pelos estudantes, foram submetidos a uma revisão do grupo, visando favorecer as interações com as situações-problema e/ou com outros estudantes e melhorias no texto. Temos que, ao revisar um problema, os estudantes constroem autonomia e responsabilidade sobre seu processo de aprendizagem.

Após as revisões do grupo, o novo cenário **sala de aula**, foi modificado para considerar a distância física entre os alunos (estágio inter). Agora, o desafio é organizar os estudantes de forma que haja uma cadeira de distância entre os alunos problemáticos, e essa distância se aplica tanto nas posições horizontais quanto nas verticais e diagonais, a complexidade aumentou, já que além de resolver o problema espacial, os estudantes precisam levar em consideração a distância entre os problemáticos, envolvendo maior abstração e coordenação de condições adicionais (estágio trans).

E nas revisões do cenário falta de água, após as revisões, foi introduzido um elemento de organização rotativa e de estratégia entre as casas para buscar água, começando com uma casa por dia e seguindo um padrão e sequências ao problema original de falta de água específico de revezamento, desafiando os leitores do problema a pensar de forma mais abstrata e a aplicar conceitos de organização rotativa. Isso trouxe uma nova camada de complexidade ao problema original.

O problema foi simplificado para uma representação visual e uma fórmula que permite resolver a questão em um cenário mais genérico com um número variável de

casas. A problemática foi transformada em um problema mais abstrato, envolvendo uma organização rotativa das casas para buscar água, desafiando os leitores a pensar de forma mais abstrata e aplicar conceitos de organização rotativa (estágio trans)

No processo, é utilizado conhecimento instrumental (estágio intra), sem que as ações realizadas ou as alterações que elas impõem aos objetos sejam conscientes. Esse conhecimento instrumental foi previamente construído, a partir da interação do sujeito com o ambiente, após realizar tarefas de escolher um problema e construir outros. Os estudantes passaram por uma transformação, indo para um conhecimento conceitual (estágio inter) ao interagir com os elementos do PC presente nos problemas que foram apresentados e configurar um funcionamento e uma estrutura de pensamento corresponde à conceituação das mudanças que eles impõem ao construir novos problemas. O conhecimento é construído quando o estudante toma consciência das mudanças que suas ações provocam e refletem (estágio trans) sobre suas construções e escolhas, o que fica evidente quando ele explica aos seus pares suas escolhas.

#### **4 Considerações**

Neste artigo, apresentamos uma contribuição para dar resposta à nossa pergunta: como melhorar a aprendizagem de programação dos estudantes do ensino médio? Para tanto concebemos a AP1 que se constitui parte essencial da composição da APAP, ancorada conceitualmente no Framework Arquiteturas Pedagógicas, no Pensamento Computacional e na Teoria Piagetiana sobre a cognição.

Com base nas reflexões e análises sobre os fóruns e os cenários apresentados, é possível inferir que a AP1 contribui para que os estudantes aprendam a propor e construir problemas de forma mais estruturada, criativa e alinhada aos princípios do Pensamento Computacional e das teorias de desenvolvimento cognitivo na visão piagetiana de que o desenvolvimento cognitivo não é um acúmulo passivo de informações, mas um processo ativo de reestruturação conceitual.

Desta forma, quando Piaget (1976), enfatiza a importância do conhecimento prévio dos estudantes, mencionam a assimilação de novos elementos aos esquemas anteriores e a modificação contínua dos esquemas anteriores à medida que novos elementos são incorporados, das interações com os pares, refletindo a interação e a coordenação entre esquemas mentais que ocorre no processo de aprendizagem e da forma de revisar as atividades, levando a reorganizações e reconstruções mentais.

Isso implica que as atividades estão projetadas para promover o desenvolvimento cognitivo e a construção de novos esquemas. Refletem a progressão e a evolução do entendimento e da complexidade dos problemas apresentados, mostrando como os estudantes avançaram de níveis mais básicos (estágio intra) para níveis mais complexos e refinados (estágios inter e trans) na elaboração e revisão dos cenários, incorporando princípios do pensamento computacional e demonstrando maior abstração e organização em suas propostas de problemas.

Com esta análise, podemos afirmar que AP1 é capaz de auxiliar os estudantes no desenvolvimento de habilidades de programação e de resolução de problemas. Possibilitando, ainda, a utilização de um portfólio de cenários construídos pelos estudantes, que ajudam no planejamento de atividades que envolvam o reaproveitamento de cenários para construções de novos.

## Referencias

- Bardin, L. (2016). Análise de conteúdo (Tradução Luís Antero Reto). *São Paulo, Brasil: Edições, 70*.
- Bati, K., Yetişir, M. I., Çalışkan, I., Güneş, G., & Gül Saçan, E. (2018). Teaching the concept of time: A steam-based program on computational thinking in science education. *Cogent Education, 5*(1), 1507306.
- Carvalho, M. J. S., de Nevado, R. A., & de Menezes, C. S. (2005, November). Arquiteturas pedagógicas para educação à distância: concepções e suporte telemático. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 1, No. 1, pp. 351-360).
- \_\_\_\_\_. (2007). Arquiteturas pedagógicas para educação a distância. *Aprendizagem em rede na educação a distância: estudos e recursos para formação de professores. Porto Alegre: Ricardo Lenz, 1*, 36-52.
- Curzon, P., Black, J., Meagher, L. R., & McOwan, P. W. (2009). cs4fn.org: Enthusiating students about Computer Science. *Proceedings of Informatics Education Europe IV*, 73-80.
- Dagienė, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *Informatics Education-Supporting Computational Thinking: Third International Conference on Informatics in Secondary Schools-Evolution and Perspectives, ISSEP 2008 Torun Poland, July 1-4, 2008 Proceedings 3* (pp. 19-30). Springer Berlin Heidelberg.
- Garcia, M. B. (2021). Cooperative learning in computer programming: A quasi-experimental evaluation of Jigsaw teaching strategy with novice programmers. *Education and Information Technologies, 26*(4), 4839-4856.
- Giannakoulas, A., & Xinogalos, S. (2018). A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Education and Information Technologies, 23*, 2029-2052.
- Inhelder, B., Bovet, M., & Sinclair, H. (1977). Aprendizagem e estruturas do conhecimento. *São Paulo: Saraiva*, 282.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior, 72*, 558-569.
- López-Pernas, S., Gordillo, A., Barra, E., & Quemada, J. (2019). Examining the use of an educational escape room for teaching programming in a higher education setting. *IEEE Access, 7*, 31723-31737.
- Mazur, E. (1999). Peer instruction: A user's manual.

- Morales-Urrutia, E. K., Ocaña, J. M., Pérez-Marín, D., & Pizarro, C. (2021). Can mindfulness help Primary Education students to learn how to program with an emotional learning companion?. *IEEE Access*, 9, 6642-6660.
- Pereira, A., da Silva, F. X., Müller, M. G., Lima, R. D. A. S., Jacaúna, R. D. P., & de Menezes, C. S. (2021). Arquitetura pedagógica debate de teses: critérios para seleção de teses. *RENOTE*, 19(2), 516-525.
- Piaget, J. (1973). Estudos sociológicos (R. Di Piero, Trans.). *Rio de Janeiro: Forense*. (Original published in 1965).
- \_\_\_\_\_. (1976). A Equilibração das Estruturas Cognitivas; o problema central do conhecimento. *Rio de Janeiro: Kahar Editores*.
- \_\_\_\_\_. (1986). O possível e o necessário (Vol. 2): Evolução dos necessários na criança (B. M. de Albuquerque, Trad.). *Porto Alegre*.
- \_\_\_\_\_. (2016). *L'epistemologia genetica*. Edizioni Studium Srl.
- Piaget, J., & García, R. (1982). *Psicogénesis e história de la ciencia*. Siglo xxi.
- Piaget, J., & Gréco, P. (1974). *Aprendizagem e conhecimento*. Freitas Bastos.
- Pimentel, M., Filippo, D., & dos Santos, T. M. (2020). Design Science Research: pesquisa científica atrelada ao design de artefatos. *RE@ D-Revista de Educação a Distância e eLearning*, 3(1), 37-61.
- Proulx, V. K. (2000). Programming patterns and design patterns in the introductory computer science course. *ACM Sigcse Bulletin*, 32(1), 80-84.
- Knuth, Donald E. The art of computer programming. Vol. 1: Fundamental algorithms. Reading, 1978.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Yin, R. K. (2010). Estudo de caso: planejamento e métodos. 4a Edição. *Sao Paulo: Artmed Editora S/A*.
- Zha, S., Jin, Y., Moore, P., & Gaston, J. (2020). Hopscotch into coding: introducing pre-service teachers computational thinking. *TechTrends*, 64, 17-28.