

SoPensa: um framework de pensamento computacional e soft skills para o ensino de programação na percepção docente

Antonio Rege^{1,2}, José Viterbo¹, Luciana Salgado¹

¹Instituto de Computação – Universidade Federal Fluminense (UFF)
Av. Gal. Milton Tavares de Souza, s/nº – Niterói – RJ – Brasil

²Grupo de Informática para Pesquisa em Computação – Instituto Federal do Acre (IFAC)
Rio Branco – AC – Brasil

{regejs}@gmail.com, {viterbo, luciana}@ic.uff.br

Abstract. *This paper presents SoPensa, a conceptual and operational framework for teaching programming in vocational technical education, grounded in computational thinking and soft skills, with theoretical foundations in meaningful learning and constructionism. Given the scarcity of proposals that integrate these axes into teacher mediation, SoPensa structures this practice into four phases. Its application in two first-year integrated technical education classes in Informatics was examined through reflexive thematic analysis of interviews with the two teachers who applied it, complemented by a questionnaire answered by the students. From the teachers' perspective, the framework organizes previously diffuse practices and favors decomposition and algorithms; abstraction persists as a difficulty and perceptions of soft skills diverge according to observation conditions.*

Resumo. *Este artigo apresenta o SoPensa, framework conceitual e operacional para o ensino de programação na educação profissional técnica, ancorado no pensamento computacional e nas soft skills, com fundamentação na aprendizagem significativa e no construcionismo. Diante da escassez de propostas que integrem esses eixos na mediação docente, o SoPensa estrutura essa prática em quatro fases. A aplicação em duas turmas do 1º ano do ensino técnico integrado em Informática foi examinada por análise temática reflexiva das entrevistas com os dois professores aplicadores, complementada por questionário respondido pelos estudantes. Na percepção docente, o framework organiza práticas antes difusas e favorece decomposição e algoritmos; a abstração persiste como dificuldade e as percepções sobre soft skills divergem conforme as condições de observação*

1. Introdução

No ensino de programação, dominar a sintaxe de uma linguagem é parte do aprendizado. O estudante precisa decompor problemas, abstrair informações e formular algoritmos logicamente. Nesse contexto, o Pensamento Computacional (PC) consolidou-se como habilidade fundamental para orientar processos de raciocínio na resolução de problemas [Wing 2006, Wing 2014]. Paralelamente, competências interpessoais, como criatividade, colaboração, comunicação e pensamento crítico, reunidas sob o termo Soft Skills (SS), têm recebido atenção crescente diante do avanço da inteligência artificial e da automação

de tarefas técnicas [Tomić et al. 2019, Wang and Herzog 2023]. A SBC destaca que o mercado demanda profissionais capazes de combinar competências técnicas e SS desde o início da formação [SBC 2025].

No Brasil, esse desafio assume contorno particular na Educação Profissional Técnica de nível médio (EPTNM), que reúne formação geral e técnica em um mesmo percurso [Ramos 2008]. Nas turmas iniciais de Lógica de Programação, as dificuldades dos estudantes [Araújo et al. 2025] e a resistência docente a novos métodos [SBC 2025] recaem sobre o professor, que precisa mediar esse processo sem apoio sistematizado. Essa condição situa a mediação docente como foco deste trabalho.

A literatura sobre o ensino de programação, PC e as SS cresceu nos últimos anos, mas ainda são escassas as propostas que integrem esses dois eixos de forma sistematizada. Parte dos estudos indicam o potencial dessa relação. [Wong and Cheung 2020] observaram ganhos em criatividade e colaboração com Scratch, enquanto [Jalinus et al. 2024] propõem a aprendizagem baseada em projetos, com rendimentos nas competências do 4C¹. Apesar desses resultados, [Maitz et al. 2022] apontam a raridade de intervenções que desenvolvam PC e SS de forma integrada, e [Ciancarini et al. 2019] mostram que o PC ainda é tratado como uma habilidade individual. Essa limitação também aparece em [Tsortanidou et al. 2023], que indicam a falta de estudos empíricos que relacionem PC, criatividade e colaboração. Quando o recorte se volta para o ensino técnico integrado ao ensino médio brasileiro, com foco na prática docente, o Mapeamento Sistemático da Literatura (MSL) realizado por [Rege et al. 2025] não identificou propostas voltadas ao professor que integrem PC e SS nessa modalidade.

Diante dessa lacuna, este artigo responde à seguinte questão de pesquisa: de que modo a aplicação do framework SoPensa reorganiza a mediação docente em turmas de Lógica de Programação do ensino técnico integrado ao ensino médio? O objetivo é apresentar o SoPensa, proposta integrada de PC e SS para a mediação docente na EPTNM, e examinar percepções sobre sua aplicação a partir de entrevistas com os dois professores aplicadores, complementadas por um questionário respondido pelos estudantes. A avaliação ocorreu em duas turmas do primeiro ano, ofertadas no Instituto Federal do Acre (IFAC). Como eram as únicas turmas da modalidade no período, a análise priorizou a profundidade das percepções docentes, e não a generalização estatística.

O estudo contribui com um framework conceitual e operacional, com fundamentação teórica explícita em cada fase e foco na prática docente da EPTNM. Metodologicamente, utiliza a Análise Temática Reflexiva (ATR) para avaliar um framework educacional. Empiricamente, caracteriza percepções e ajustes pedagógicos decorrentes da aplicação inicial em duas turmas do ensino técnico.

2. Fundamentação Teórica

Esta seção apresenta os fundamentos do SoPensa: dificuldades no ensino de programação, PC, SS e teorias de aprendizagem que orientam o framework.

2.1. Dificuldades no ensino e na aprendizagem de programação

Aprender a programar exige o uso combinado de capacidades como abstração, decomposição, reconhecimento de padrões e construção de algoritmos. Estudos mostram

¹Framework de Habilidades de Aprendizagem e Inovação, composto pelas competências de criatividade, pensamento crítico, comunicação e colaboração [Partnership for 21st Century Learning 2019].

que os estudantes muitas vezes dominam a sintaxe da linguagem, mas não conseguem aplicá-la na resolução de problemas [Moraes et al. 2020]. [Sim and Lau 2022] apontam a abstração como uma das partes mais desafiadoras desse percurso para o iniciante.

[Cheah 2020] aponta que o estudante encontra obstáculos já no projeto da solução, por não conseguir dividir o problema em partes menores. Tal dificuldade é frequentemente associada à fragilidade dos conhecimentos prévios em matemática e lógica, cuja ausência intensifica os obstáculos iniciais [Medeiros et al. 2021, Pureza et al. 2025]. A esses fatores cognitivos somam-se aspectos comportamentais, como a desmotivação [Moraes et al. 2020] e a ansiedade na disciplina, associada à dificuldade de compreender o conteúdo e iniciar a resolução de problemas [Chang et al. 2024]. Sem mediação pedagógica intencional, essas dificuldades tendem a se retroalimentar, com reflexos nos índices de reprovação e evasão das disciplinas introdutórias [Moraes et al. 2020].

Na educação profissional, essas dificuldades convivem com condições estruturais próprias. [Moura 2008] descreve os desafios históricos da modalidade no Brasil, ligados à organização curricular, à formação docente e à infraestrutura, como herança da separação entre formação acadêmica e formação para o trabalho. No ensino de programação dentro dessa modalidade, os obstáculos cognitivos e comportamentais tornam esse cenário mais complexo, com impacto direto sobre o trabalho do professor.

2.2. Pensamento Computacional

O PC ganhou destaque com [Wing 2006], que o apresentou como forma de resolver problemas e projetar sistemas a partir de conceitos da Ciência da Computação. A autora refinou a definição ao caracterizá-lo como o processo mental de formular um problema de modo a admitir solução executável por humanos ou máquinas, e apontou a abstração como o processo mais importante e de mais alto nível do PC [Wing 2014]. [Montuori et al. 2024] reforçam essa leitura e relacionam o ensino de programação ao desenvolvimento da resolução de problemas. Neste estudo, o PC é operacionalizado a partir de quatro componentes recorrentes na literatura: decomposição, reconhecimento de padrões, abstração e elaboração de algoritmos [Grover and Pea 2013, Shute et al. 2017].

2.3. Soft Skills

As SS abrangem competências socioemocionais e interpessoais, como comunicação, colaboração, pensamento crítico e gestão das emoções, associadas à adaptação em ambientes em transformação [Kapustina and Matyushina 2023, Dadelo 2025]. [Wang and Herzog 2023] registram o crescente reconhecimento, no meio educacional, de habilidades pouco quantificáveis, como colaboração, liderança e criatividade. No ensino de programação, o programador precisa combinar competências técnicas e interpessoais, como colaboração, comunicação e pensamento crítico [Tomić et al. 2019, Figueiredo and García-Peñalvo 2022]. [Mohammed and Ozdamli 2024] observam uma lacuna entre o domínio técnico dos egressos e o desenvolvimento dessas habilidades, e a [SBC 2025] registra, no contexto brasileiro, a formação inicial insuficiente dos professores de Computação para trabalhar SS e empregar metodologias ativas.

2.4. Teorias de aprendizagem ancoradoras

O SoPensa fundamenta-se em duas teorias complementares. A Aprendizagem Significativa (AS), de Ausubel, destaca o papel do conhecimento prévio do aprendiz

na aquisição de novos conhecimentos [Ausubel 2003]. Para que ela ocorra, são necessários materiais potencialmente significativos e a predisposição do estudante para aprender [Ausubel et al. 1980]. A teoria descreve dois processos de organização cognitiva, a diferenciação progressiva, em que conceitos mais inclusivos se especializam, e a reconciliação integrativa, em que diferenças aparentes entre conceitos são superadas pela identificação de relações mais amplas. O Construcionismo, de Papert, entende a aprendizagem como construção ativa do conhecimento por meio da interação com o meio e da criação de artefatos significativos [Papert 1980, Resnick et al. 2009]. Nesse contexto, o erro tem papel formativo, e as tentativas e revisões fazem parte do processo.

3. O Framework SoPensa

Esta seção apresenta o SoPensa, com sua visão geral, seus princípios, as quatro fases, o fluxo de aplicação e as ações do professor.

3.1. Visão geral e princípios

O SoPensa é um framework conceitual e operacional para a mediação docente no ensino de programação na EPTNM. É conceitual porque organiza, em estrutura teoricamente fundamentada, o desenvolvimento conjunto de habilidades de PC e SS. É operacional porque associa a cada fase ações concretas do professor, instrumentos e competências. O framework não prescreve materiais didáticos nem linguagens específicas, mantendo-se compatível com os recursos que o docente já adota.

Quatro princípios estruturam o framework. PC e SS operam juntos em cada fase, sem tratamento separado. A ancoragem se dá em duas teorias complementares, a AS e o Construcionismo, com função distinta atribuída a cada uma. A mediação docente é central, com o professor como mediador do processo em todas as fases. O fluxo é iterativo, com retroalimentação entre etapas conforme o progresso da turma. A Figura 1 apresenta a estrutura geral do framework.

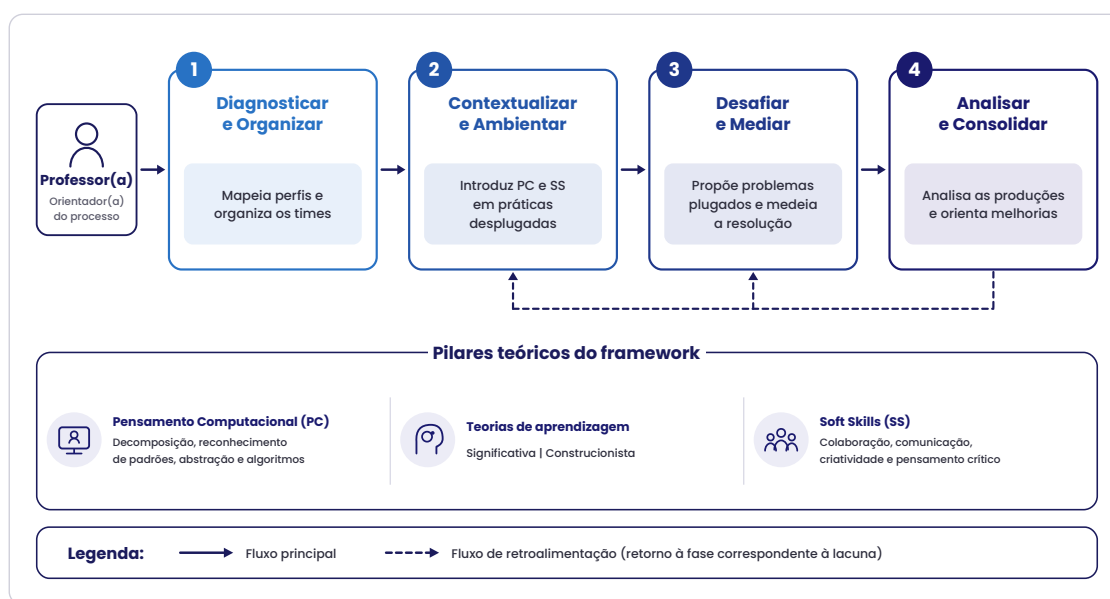


Figura 1. Estrutura do framework SoPensa

3.2. As quatro fases e sua fundamentação

As fases são sequenciais em sua lógica formativa, mas o fluxo não é estritamente linear. Cada uma reúne ações de PC e SS e apoia-se na teoria de aprendizagem que melhor descreve o movimento cognitivo predominante naquele momento.

A Fase 1, Diagnosticar e Organizar, mapeia os conhecimentos prévios dos alunos em PC e SS e organiza os times a partir da identificação de perfis cognitivos e interpessoais. Apoia-se na AS, que sustenta que novos conhecimentos se conectam quando há subsunções adequados nos quais ancorá-los [Ausubel 2003].

A Fase 2, Contextualizar e Ambientar, introduz os conceitos de PC e SS por meio de atividades práticas desplugadas, ambientando o aluno antes da transição para a codificação. Mobiliza as duas teorias, na conexão a conceitos prévios e na construção ativa do conhecimento [Ausubel et al. 1980, Papert 1980].

A Fase 3, Desafiar e Mediar, propõe desafios plugados que exigem a aplicação dos componentes do PC e o exercício das competências de SS, com o professor como mediador da exploração das soluções. Apoia-se no Construcionismo, pela experimentação ativa e pela criação de artefatos [Papert 1980, Resnick et al. 2009].

A Fase 4, Analisar e Consolidar, conduz a análise das produções dos times, a orientação de melhorias e a reflexão coletiva sobre o aprendizado. Mobiliza a diferenciação progressiva e a reconciliação integrativa da AS, em complemento ao aprendizado iterativo do Construcionismo, em que o aluno revisa, testa e aprimora suas soluções [Ausubel et al. 1980, Papert 1980].

3.3. Fluxo de aplicação

O fluxo operacional descreve, para cada fase, a entrada, a atividade, a saída, os artefatos e as competências, que não têm o mesmo papel em todas as etapas. Na Fase 1 elas são mapeadas, e não desenvolvidas, dado o caráter diagnóstico, apoiada no questionário diagnóstico e na formação dos times. Nas Fases 2 e 3 desenvolvem-se pelo ciclo de mediação Pensar–Agir–Refletir, ao qual se soma Comunicar na Fase 3, com apoio de atividades desplugadas na Fase 2 e de ambientes visuais e textuais, na Fase 3. Na Fase 4 o foco recai sobre a comunicação, no registro do feedback e da autoavaliação dos times, e sobre o pensamento crítico, exercido quando os estudantes avaliam os próprios acertos e erros, com apoio de rubricas qualitativas e fichas de observação. A retroalimentação entre fases ocorre quando a análise das produções na Fase 4 revela lacunas. Os times retornam às fases anteriores para retomar o que não foi consolidado e voltam à Fase 4 com as soluções revisadas. É por esse laço que componentes do PC ou competências de SS pouco mobilizados na primeira tentativa, voltam a ser trabalhados nas fases anteriores, e não dentro da Fase 4. Os exemplos das atividades desplugadas e plugadas conduzidas nas fases 2 e 3, bem como o desenho das práticas, são detalhados em trabalho correlato.

4. Metodologia da Avaliação

Esta seção descreve a avaliação do framework, organizada em quatro subseções: desenho da pesquisa, contexto e participantes, coleta de dados e ATR aplicada às entrevistas.

4.1. Desenho da pesquisa

A pesquisa adotou abordagem qualitativa predominante, com uso complementar de dados quantitativos descritivos para caracterizar os resultados [Mattar and Ramos 2021,

Creswell and Creswell 2017]. Quanto aos procedimentos técnicos, configurou-se como pesquisa-ação, caracterizada pela participação ativa dos envolvidos, pela flexibilidade das etapas e pelo caráter reflexivo [Thiollent 2022, Tripp 2005].

O desenho seguiu uma estrutura cíclica composta por quatro fases: Diagnóstico, Planejamento, Intervenção e Avaliação [Tripp 2005]. Na fase de Diagnóstico, um MSL [Petersen et al. 2008] e uma coleta empírica com 133 professores de informática dos Institutos Federais, seguida de entrevistas com 13 deles, permitiram identificar lacunas no desenvolvimento de competências de PC e SS, base empírica do framework. A fase de Planejamento definiu seus aspectos conceituais e operacionais; a Intervenção aplicou o framework em duas turmas de Lógica de Programação; e a Avaliação analisou as percepções dos participantes. Este artigo recorta a fase de Avaliação. Os demais objetivos são tratados em outro recorte da pesquisa.

4.2. Contexto e participantes

A pesquisa foi realizada no IFAC, Campus Rio Branco, após aprovação pelo CEP, conforme o CAAE 85642524.5.0000.0233. O campo foi escolhido pela oferta de cursos técnicos em Informática e pela viabilidade de acesso ao contexto investigado. A aplicação ocorreu em duas turmas do primeiro ano do curso técnico integrado ao ensino médio, uma de Informática para Internet e outra de Informática em Redes, com 35 estudantes cada. As turmas foram escolhidas por constituírem um cenário introdutório, sem contato prévio com conteúdos de PC e SS no contexto da programação.

Além dos estudantes, participaram da aplicação dois professores, P1 e P2, selecionados por terem graduação em Computação, vínculo efetivo, experiência mínima de três anos em Lógica de Programação e assinatura do termo de consentimento. P1 leciona em Informática em Redes e P2 em Informática para Internet, com cinco e oito anos de docência, respectivamente. A aplicação ocorreu no primeiro semestre letivo de 2025. Antes da intervenção, o framework foi apreciado por dois docentes especialistas com experiência em Lógica de Programação, distintos dos professores responsáveis pela aplicação. A apreciação avaliou clareza conceitual, coerência pedagógica e viabilidade prática, resultando em ajustes incorporados à versão aplicada. Durante a intervenção, os professores conduziram as aulas e atividades, enquanto o pesquisador acompanhou as aulas como observador participante, prestando apoio técnico e metodológico aos professores e registrando as observações em diário reflexivo.

4.3. Coleta de dados

As fontes principais foram entrevistas semiestruturadas com P1 e P2, conduzidas ao término da aplicação. Orientadas pela pergunta de pesquisa, abordaram a percepção dos professores sobre a aplicação do framework, os efeitos percebidos no desenvolvimento de PC e SS, as condições de execução e os ajustes pedagógicos necessários. As entrevistas foram gravadas em áudio e transcritas integralmente.

Como fonte complementar, aplicou-se um questionário aos estudantes, em escala Likert de cinco pontos, organizado em torno da visão geral do framework, do desenvolvimento percebido em PC e SS e da recomendação de uso. Dos 70 alunos matriculados nas duas turmas, 64 responderam, correspondendo a 91,4% do universo. Os dados foram tratados por estatística descritiva. A combinação entre entrevistas e questionário configura

triangulação entre fontes complementares [Flick 2013], para ampliar a interpretação dos achados qualitativos.

4.4. Análise Temática Reflexiva

As entrevistas foram analisadas segundo a ATR [Braun and Clarke 2022], na versão reflexiva, e não nas variantes orientadas à confiabilidade da codificação. A escolha decorre do alinhamento entre os pressupostos da ATR e os objetivos da pesquisa, em que a subjetividade do pesquisador é recurso analítico, e não viés a eliminar. Não se buscou estabelecer relações causais entre o framework e os efeitos relatados, e as percepções foram compreendidas como situadas no contexto institucional e na trajetória dos professores.

A análise seguiu seis fases, com retornos iterativos entre etapas. A familiarização com o corpus envolveu três escutas das gravações, transcrição integral e revisão dos textos. Após a inserção das transcrições no Atlas.ti [Frieze 2019], a codificação combinou orientação dedutiva e indutiva nos níveis semântico e latente: parte dos códigos registrou conteúdos manifestos (Decomposição, Tempo, Engajamento) e parte, significados implícitos (Limitação avaliativa, Sistematização pedagógica, Defasagem formativa), derivados dos eixos PC e SS ou das próprias falas, como Pertencimento e Lacuna curricular. Quando um recorte expressava mais de um significado, utilizou-se codificação múltipla. A codificação percorreu quatro rodadas, reduzindo de 50 para 27 códigos até a estabilização. O corpus resultou em 72 trechos literais codificados (24 de P1 e 48 de P2), com 85 codificações em razão da codificação múltipla (Figura 2).

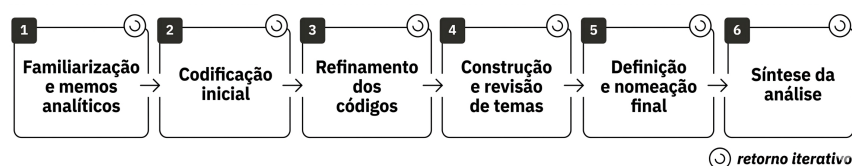


Figura 2. Percurso da Análise Temática Reflexiva

A análise foi conduzida pelo pesquisador-autor e discutida com outro pesquisador em quatro reuniões voltadas à problematização das decisões interpretativas, às fronteiras entre códigos e temas e ao refinamento da estrutura final. A posição do pesquisador como autor do framework foi tratada como parte da análise, com atenção ao risco de leituras favoráveis e preservação de trechos que expressavam limites e dificuldades relatadas pelos participantes.

A construção analítica resultou em cinco temas interpretativos e um eixo complementar. No Tema 4, a divergência entre os professores foi assumida como conceito organizador, e não como inconsistência. O eixo Potencial pedagógico registra a avaliação convergente quanto à recomendação do framework. As percepções foram postas em diálogo com os dados do questionário, para ampliar a interpretação, e não como validação cruzada [Mattar and Ramos 2021, Flick 2013].

5. Resultados e Discussão

A análise produziu cinco temas interpretativos e um eixo analítico complementar, sintetizados na Tabela 1. Cada tema é organizado por um conceito central, apresentado por trechos de P1 e P2 e posto em diálogo com a literatura sobre PC, SS e ensino de programação.

Tabela 1. Síntese dos temas e códigos da análise temática

Tema	Códigos
T1. O SoPensa reorganiza a prática docente	Facilitação (P1, P2); Diagnóstico de perfil (P1, P2); Aplicabilidade flexível (P1); Integração teoria-prática (P1); Sistematização pedagógica (P2); Debate reflexivo (P2); Progressão didática (P2).
T2. O SoPensa aproxima o aluno da disciplina	Engajamento (P1, P2); Melhoria no desempenho (P1, P2); Iniciativa (P1); Pertencimento (P1); Preparação profissional (P2).
T3. O SoPensa e a estruturação do raciocínio antes da codificação	Decomposição (P1, P2); Abstração (P1, P2); Algoritmos (P1, P2); Resolução de problemas (P1, P2); Reconhecimento de padrões (P2).
T4. O SoPensa e a percepção divergente sobre as soft skills	Colaboração (P1, P2); Pensamento crítico (P2); Limitação avaliativa (P1).
T5. O SoPensa requer condições e ajustes pedagógicos	<i>Condições modificáveis:</i> Tempo (P1, P2); Nivelamento inicial (P2); Clareza metodológica (P2); Recursos didáticos (P2). <i>Limites estruturais:</i> Defasagem formativa (P2); Lacuna curricular (P2).
Eixo analítico complementar	Potencial pedagógico (P1, P2).

Nota: a indicação entre parênteses identifica o professor que sustenta cada código; (P1, P2) indica código mencionado pelos dois.
Fonte: Autoria própria.

5.1. T1. O SoPensa reorganiza a prática docente

P1 resume o efeito do framework em uma frase, “o framework facilitou, de fato, o processo de ensino-aprendizagem” (P1). P2 é mais específico sobre o que mudou, e a palavra que escolhe carrega o tema, “SoPensa organiza de maneira pedagógica, institucional, essa construção, esse incentivo de você trabalhar soft skills e pensamento computacional” (P2). Institucional, não pessoal. O que P2 descreve é a passagem de um esforço que dependia da iniciativa de cada um para algo com forma transmissível. A [SBC 2025] ajuda a situar esse relato, ao registrar que muitos professores da área não receberam, na formação inicial, subsídios para trabalhar SS ou metodologias ativas. Se a formação não os ofereceu, cada docente mobilizava repertório próprio, e não método partilhado, diferença que P2 nomeia ao falar em construção institucional.

A dificuldade não está só na origem da carreira. [Medeiros et al. 2018] registram que a falta de métodos e ferramentas apropriados é a questão mais citada pelos professores no ensino introdutório, e [Hudin 2023] acrescenta que o professor nem sempre sabe quais recursos e métodos didáticos usar nesse contexto, situando o relato no cotidiano da prática. A sistematização também trouxe uma exigência inicial, e P2 relata pausas para explicar o método aos alunos, dificuldade que cedeu ao longo do semestre. O que os dois descrevem, então, não é a chegada de conteúdo novo, mas a organização de práticas que cada um já ensaiava.

5.2. T2. O SoPensa aproxima o aluno da disciplina

P1 identifica um sinal de engajamento que emerge fora da atividade formal, “alguns jargões da lógica eles utilizam como piada, como trocadilhos na sala de aula, e eu acredito que isso representa um interesse acima da média” (P1). O interesse aparece onde não é avaliado: na brincadeira com os termos da disciplina fora da tarefa. P2 observa movimento semelhante e o estende ao professor: “foi um elemento motivador tanto para o aluno quanto para o professor” (P2). O relato ganha força quando se considera o ponto de partida da turma, pois P2 registra dificuldades prévias de interação, atribuídas em parte ao efeito da pandemia. Nesse contraste, o interesse descrito pelos dois professores representa um deslocamento, não uma disposição já existente.

[Chang et al. 2024] reforçam essa leitura ao registrarem que a proporção de estudantes que rejeitam o aprendizado de programação supera a dos que apreciam. O componente lúdico relatado por P1 se aproxima de [Ribeiro et al. 2020], para quem o sentido de evolução, sustentado por feedback e pela dimensão lúdica, opera como recurso de engajamento. Como o pesquisador é autor do framework e conduziu a formação inicial dos professores, parte do vocabulário com que P1 e P2 descrevem o engajamento pode ter origem nessa formação, condição tratada na Seção 6. Assim, os relatos são lidos como percepção situada, e não como medida de efeito. Essa leitura se completa com os dados do questionário, em que a contribuição geral do framework foi avaliada favoravelmente por 67,2% dos estudantes. O entusiasmo docente convive, portanto, com uma adesão estudantil mais contida, diferença que o próprio caráter informal do indicador de P1 já antecipava.

5.3. T3. O SoPensa e a estruturação do raciocínio antes da codificação

Para os dois professores, decomposição, abstração e algoritmos passaram a anteceder a escrita do código, e não a decorrer dela. P1 descreve a decomposição já incorporada à rotina dos alunos, “o aluno precisava decompor o problema em trechos e a partir desses trechos resolver determinadas situações” (P1). O relato é consistente com [Cheah 2020], que situa a dificuldade inicial antes da codificação, quando o estudante ainda não consegue dividir o problema em funcionalidades menores. A abstração concentra a tensão do tema. P1 a identifica como o principal obstáculo, “a maior dificuldade deles hoje é a abstração” (P1). P2 relata uma melhoria, mas a associa menos ao domínio da abstração e mais ao reconhecimento de sua necessidade, “houve essa melhoria, sim, porque primeiro despertou a necessidade de se pensar de maneira abstrata” (P2). A leitura da fala de P2 sugere que, em sua percepção, o ganho esteve em os alunos reconhecerem que precisavam abstrair, o que antecede o domínio da habilidade.

Essa percepção é coerente com [Sim and Lau 2022], que aponta a abstração do problema como a parte mais desafiadora para o iniciante. Se a abstração é o passo mais difícil, o reconhecimento de sua necessidade, relatado por P2, pode ser entendido como uma etapa anterior, e não como sinal de que a dificuldade foi superada. A leitura é coerente com a divergência interna do tema, em que P1 segue vendo a abstração como o maior obstáculo enquanto P2 percebe um movimento inicial em sua direção. As duas falas descrevem menos um avanço consolidado e mais uma mudança inicial, os alunos passaram a reconhecer a necessidade de abstrair. No contexto investigado, o T3 mostra que o framework favorece o uso do PC antes da codificação, com avanço percebido em decomposição e algoritmos, enquanto a abstração permanece como ponto crítico.

5.4. T4. O SoPensa e a percepção divergente sobre as soft skills

P1 não percebeu evolução nas SS, e diz por quê, “com a Soft Skills eu não percebi nenhuma evolução ou diferença, até porque eu vejo eles com pouca frequência durante a semana” (P1). A segunda metade da frase é decisiva: P1 não afirma que o desenvolvimento não ocorreu, afirma que não teve como vê-lo. P2, que acompanhou a turma com frequência maior e em atividades em grupo, relata o contrário, “como houve trabalho conjunto, principalmente motivado pelo SoPensa, houve o melhor desenvolvimento dessa construção” (P2). As duas falas não se contradizem, marcam posições de observação distintas, possibilidade compatível com [Braun and Clarke 2022] quando o tema se organiza

em torno de uma divergência. [Wang and Herzog 2023] caracterizam as SS como habilidades pouco quantificáveis, e [Yurchenko et al. 2024] acrescentam que seus métodos de medição não são padronizados, o que torna previsível que P1 e P2, acompanhando turmas distintas e em condições diferentes, cheguem a leituras distintas. A fala de P2 sobre o trabalho conjunto contrasta com [Ciancarini et al. 2019], que registram pouca atenção ao trabalho em equipe em cursos introdutórios, dimensão diretamente ligada às SS. A divergência entre P1 e P2, longe de enfraquecer o tema, constitui o próprio achado interpretativo, a percepção sobre SS depende de onde se olha.

5.5. T5. O SoPensa requer condições e ajustes pedagógicos

Sobre o tempo, P1 e P2 convergem na avaliação, embora com ressalvas. P1 sintetiza essa posição ao afirmar que “o tempo no qual foi aplicado é suficiente, foi o mínimo” (P1). Suficiente, mas no limite. É P2 quem desloca a análise para uma condição curricular que o framework não criou nem resolve, “a construção dos nossos planos de ensino, nossas ementas, elas não estão focadas no desenvolvimento do pensamento computacional e de soft skills” (P2). O problema deixa de ser o tempo da aula e passa a ser o currículo. Isoladamente, essa queixa poderia soar como uma particularidade do campus. Contudo, [Medeiros et al. 2021] apontam a deficiência dos conhecimentos prévios, sobretudo em matemática e língua materna, como traço que separa o cenário nacional do internacional, aproximando o relato de P2 de uma dificuldade mais ampla, não apenas local. Quanto à lacuna curricular, [Ramos 2008] descreve o ensino médio integrado como projeto de articulação entre a formação geral e a técnica, relação que, na Lógica de Programação, ainda não alcançou o PC e as SS. Parte do que limita o framework, portanto, é anterior a ele, e o SoPensa funciona menos como correção dessas condições e mais como instrumento que as torna visíveis.

5.6. Eixo analítico complementar. Potencial pedagógico

O eixo complementar não constitui tema autônomo. Registra a avaliação convergente dos professores quanto à recomendação do framework, em conformidade com a orientação de [Braun and Clarke 2022] de não construir temas adicionais sem necessidade analítica. P1 amplia a recomendação para além da disciplina, “é um aspecto muito positivo e que pode e precisa ser replicado em outras disciplinas também” (P1). P2 mobiliza explicitamente o termo, “ele tem muito potencial para colaborar dentro do grupo acadêmico para a aula de programação, para as aulas de lógica” (P2). Os dois recomendam o framework, ainda que com condições distintas: P1 enfatiza a análise individual do perfil dos alunos, e P2, a aplicação prévia à ementa. Mesmo diante da divergência sobre SS (T4) e dos condicionantes estruturais (T5), a avaliação dos dois é positiva. A convergência se aproxima de [Maitz et al. 2022], que registram a raridade de intervenções que combinam programação, PC e habilidades socioemocionais, e de [Rege et al. 2025], que apontam a ausência de orientações para aplicar essas competências de forma sistemática.

5.7. Síntese e triangulação

Os dados do questionário aplicado aos 64 estudantes, detalhados em trabalho correlato, funcionam como camada complementar de triangulação com os achados qualitativos. A contribuição geral do framework foi avaliada favoravelmente por 67,2% dos estudantes, e a recomendação de uso atingiu 70,3%. Entre os componentes do PC, o reconhecimento de padrões teve a maior concordância (79,7%), seguido por decomposição (68,7%) e

algoritmos (60,9%), enquanto a abstração registrou o menor índice (48,4%), perfil que reforça o T3. Como já situado na Seção 2.2, a abstração ocupa o nível mais alto do PC [Montuori et al. 2024]; sua baixa concordância, no questionário e no relato docente, é compatível com uma habilidade mais complexa, que demanda mais tempo e mais base prévia, e não sinal de falha do framework.

Entre as SS, os menores índices de concordância couberam a colaboração (42,2%) e ao pensamento crítico (26,6%), que também registraram a maior neutralidade (46,9% e 48,4%, respectivamente), padrão que se aproxima da observabilidade desigual identificada no T4. A alta neutralidade é coerente com a leitura de que essas competências são difíceis de perceber e medir, e não necessariamente de que não tenham sido desenvolvidas. Vale notar que os números do questionário não acompanham o entusiasmo docente em todos os pontos: o engajamento descrito por P1 e P2 no T2 convive com índices estudantis mais moderados, o que reforça a leitura das percepções dos professores como situadas, e não como medida direta de efeito.

O conjunto dos achados pode ser lido à luz das duas teorias que fundamentam o framework. A AS sustenta que novos conhecimentos se conectam quando encontram subsunçores adequados, pressuposto que o T1 operacionaliza ao registrar que o framework permite partir do aluno real, e que o T3 tensiona ao mostrar a abstração comprometida onde há defasagem formativa (T5). O Construcionismo sustenta a aprendizagem como construção ativa por meio de artefatos, perspectiva compatível com o engajamento descrito no T2 e com a colaboração potencializada pelo trabalho conjunto relatada no T4.

6. Limitações e Ameaças à Validade

Este artigo recorta a fase de Avaliação de uma pesquisa-ação organizada em quatro fases: Diagnóstico, Planejamento, Intervenção e Avaliação. As fases anteriores, em particular o MSL e a coleta empírica com 133 professores de informática vinculados aos Institutos Federais, sustentaram a construção do framework e são tratadas em outros recortes da pesquisa. As fragilidades discutidas a seguir dizem respeito ao artigo, e não à estrutura completa da investigação.

O número de participantes desta fase é reduzido. A análise mobilizou as percepções de dois professores que aplicaram o framework em duas turmas do primeiro ano do ensino técnico integrado de um único campus. Essa delimitação decorre do desenho do estudo, as turmas eram as únicas da modalidade oferecidas no campus no período, ministradas em paralelo, e a avaliação exigiu acompanhamento presencial ao longo de um semestre, com entrevistas, condição incompatível com desenho multi-institucional dentro do escopo do estudo. A profundidade interpretativa do material qualitativo, e não a representatividade estatística, é o objetivo da abordagem reflexiva da ATR [Braun and Clarke 2022]. A generalização é circunscrita ao contexto investigado, e a transferibilidade para outras instituições da Rede Federal depende de estudos adicionais, sobretudo porque os condicionantes estruturais identificados no T5, a defasagem formativa e a lacuna curricular, podem operar de modo distinto em contextos diversos.

A ameaça mais relevante diz respeito ao envolvimento do pesquisador. Em coerência com os pressupostos da pesquisa-ação, ele atuou como participante do processo investigativo, com inserção colaborativa no ambiente para acompanhamento da intervenção [Thiollent 2022, Tripp 2005]. O acompanhamento ocorreu por observação

participante, com apoio técnico e metodológico aos professores e registro contínuo em diário reflexivo, sem que o pesquisador assumisse a condução das aulas. Ainda assim, configura-se uma cadeia de envolvimento que precisa ser reconhecida. O pesquisador é autor do framework, foi responsável pela formação inicial dos professores aplicadores, executou a tabulação dos diagnósticos e a formação dos times, acompanhou as aulas, conduziu as entrevistas e a análise. A formação inicial envolveu duas reuniões com os professores, de aproximadamente 40 minutos cada, em que foram apresentados componentes, competências, fases e fluxos, e pode ter introduzido vocabulário e categorias que reaparecem nas falas, condição que precisa ser considerada na leitura de termos como decomposição, abstração, SS e PC.

Quatro mecanismos trataram essa cadeia: a apreciação prévia por especialistas (Seção 4.2), com validação independente antes da intervenção; a preservação, na análise, de trechos que expressam limites e dificuldades dos participantes, como a recusa de P1 quanto à evolução em SS (T4), os condicionantes estruturais (T5) e as pausas para explicar o framework (T1); as quatro reuniões de problematização interpretativa com outro pesquisador; a codificação combinada dedutiva e indutiva, com códigos extraídos das próprias falas. Esses mecanismos não eliminam a cadeia descrita, mas a tornam visível e mitigada.

Por fim, conforme a Seção 4.4, o estudo não busca relações causais: quando P1 e P2 atribuem ganhos à aplicação, registra-se a percepção dos participantes, e não uma relação causal, que exigiria controle experimental incompatível com a abordagem interpretativa [Mattar and Ramos 2021].

7. Considerações Finais

Na percepção dos dois professores, o valor do SoPensa está menos em garantir ganhos de aprendizagem e mais em reorganizar a prática docente e tornar visíveis condições antes difusas, da estruturação do raciocínio que antecede a codificação, com a abstração como ponto que persiste, até a defasagem formativa que ultrapassa a disciplina. Diante dos condicionantes estruturais, o framework atua mais como diagnóstico do que como solução. A divergência sobre as SS decorre das condições de observação de cada professor, e não de inconsistência. Ainda assim, ambos recomendam a proposta no contexto investigado.

Como trabalhos futuros, destacam-se a replicação em outras instituições da Rede Federal, para investigar a transferibilidade, e a aplicação do SoPensa antes da ementa curricular, como nivelamento prévio sobre as defasagens do T5, conforme sugerido por P2. Estudos com mais professores e de caráter longitudinal podem examinar a recorrência ou variação dos temas ao longo do tempo.

Declaração de Uso de Inteligência Artificial

Durante a elaboração deste trabalho, os autores utilizaram o modelo de linguagem Claude, da Anthropic, como ferramenta de apoio à revisão gramatical, ortográfica e textual. A ferramenta não foi utilizada para produzir autonomamente seções do artigo. Todo o conteúdo, incluindo a concepção do estudo, o delineamento metodológico, a coleta e análise dos dados e a interpretação dos resultados, é de responsabilidade dos autores.

Referências

Araújo, A. L. d. S., Oliveira, J. G. d. A., Souza, I. A. F. d., Azevedo, G. S. d., Batista, C. F. S., and Pereira, R. P. d. M. (2025). Aprendizagem e retenção em programação: um

- estudo no ensino médio técnico do instituto federal de educação, ciência e tecnologia do rio grande do norte, campus parelhas, brasil. *Revista Principia*, 62:e8545.
- Ausubel, D. P. (2003). *Aquisição e retenção de conhecimentos: uma perspectiva cognitiva*. Plátano, Lisboa.
- Ausubel, D. P., Novak, J. D., and Hanesian, H. (1980). *Psicologia educacional*. Intera-mericana.
- Braun, V. and Clarke, V. (2022). *Thematic Analysis: A Practical Guide*. SAGE Publications, London.
- Chang, L.-C., Lin, H.-R., and Lin, J.-W. (2024). Learning motivation, outcomes, and anxiety in programming courses—a computational thinking-centered method. *Education and Information Technologies*, 29(1):545–569.
- Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2):ep272.
- Ciancarini, P., Missiroli, M., and Russo, D. (2019). Exploiting agile practices to teach computational thinking. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pages 63–83. Springer.
- Creswell, J. W. and Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- Dadelo, S. (2025). Possibilities for evaluation to foster the soft skills of critical thinking, creativity, and communication in higher education. *Creativity Studies*, 18(1):13–29.
- Figueiredo, J. and García-Peñalvo, F. J. (2022). Strategies to increase success in learning programming. In *2022 International Symposium on Computers in Education (SIIE)*.
- Flick, U. (2013). *The SAGE handbook of qualitative data analysis*. sage.
- Friese, S. (2019). Qualitative data analysis with atlas. ti.
- Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational Researcher*, 42(1):38–43.
- Hudin, S. S. (2023). A systematic review of the challenges in teaching programming for primary schools’ students. *Online Journal for TVET Practitioners*, 8(1):75–88.
- Jalinus, N., Putra, R. R., et al. (2024). Implementation of project-based learning computational thinking models in mobile programming courses. *International Journal of Interactive Mobile Technologies*, 18(11).
- Kapustina, V. A. and Matyushina, M. A. (2023). Theoretical review of digital tools aimed to development soft skills during a learning process. In *2023 IEEE 24th International Conference of Young Professionals in Electron Devices and Materials (EDM)*, pages 1920–1923. IEEE.
- Maitz, K., Paleczek, L., and Danielowitz, C. (2022). Simultaneously fostering computational thinking and social-emotional competencies in 4th graders using scratch: A feasibility study. In *Proceedings of Mensch und Computer 2022*, pages 399–403.

- Mattar, J. and Ramos, D. K. (2021). *Metodologia da pesquisa em educação: abordagens qualitativas, quantitativas e mistas*. Edições 70.
- Medeiros, R., ao, T. F., and Ramalho, G. (2021). Comparação entre o panorama internacional e nacional sobre o ensino e a aprendizagem de introdução a programação no ensino superior. In *Anais do XXIX Workshop sobre Educação em Computação*, pages 478–487, Porto Alegre, RS, Brasil. SBC.
- Medeiros, R. P., Ramalho, G. L., and Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2):77–90.
- Mohammed, F. and Ozdamli, F. (2024). A systematic literature review of soft skills in information technology education. *behavioral sciences*, 14 (10), 894.
- Montuori, C., Gambarota, F., Altoé, G., and Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, 210:104961.
- Morais, C. G. B., Mendes Neto, F. M., and Osório, A. J. M. (2020). Difficulties and challenges in the learning process of algorithms and programming in higher education: a systematic literature review. *Research, Society and Development*, 9(10):e9429109287.
- Moura, D. H. (2008). A formação de docentes para a educação profissional e tecnológica. *Revista Brasileira da educação profissional e tecnológica*, 1(1):23–38.
- Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic books.
- Partnership for 21st Century Learning (2019). Framework for 21st century learning definitions. Acesso em: 8 jun. 2025.
- Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *12th international conference on evaluation and assessment in software engineering (EASE)*. BCS Learning & Development.
- Pureza, E. M., da Paz, D. P. A., Cardoso, A. V., do Nascimento Rodrigues, C., and do Nascimento, C. J. (2025). Dificuldades de aprendizagem em lógica de programação em cursos de geoprocessamento na educação a distância: Uma revisão bibliográfica (2006–2025). *IOSR Journal of Business Management (IOSR-JBM)*, 27(12):19–29.
- Ramos, M. (2008). Concepção do ensino médio integrado. *Texto apresentado em seminário promovido pela Secretaria de Educação do Estado do Pará nos dias*, 8:1–26.
- Rege, A., Salgado, L., and Viterbo, J. (2025). Pensamento computacional e soft skills no contexto do ensino de programação: Um mapeamento sistemático. In *Anais do XXXVI Simpósio Brasileiro de Informática na Educação*, pages 1389–1400, Porto Alegre, RS, Brasil. SBC.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.
- Ribeiro, R. B. S., Carvalho, L. S. G., Oliveira, D. B. F., de Oliveira, E. H. T., and Pessoa, M. (2020). Investigação empírica sobre os efeitos da gamificação de um juiz online em uma disciplina de introdução à programação. *Revista Brasileira de Informática na Educação*, 28:461–490.

- SBC (2025). *Grandes Desafios da Educação em Computação 2025-2035: Resumo Executivo*. Sociedade Brasileira de Computação (SBC), Porto Alegre.
- Shute, V. J., Sun, C., and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22:142–158.
- Sim, T. Y. and Lau, S. L. (2022). Review on challenges and solutions in novice programming education. In *2022 IEEE International Conference on Computing (ICOCO)*, pages 55–61. IEEE.
- Thiollent, M. (2022). *Metodologia da pesquisa-ação*. Cortez editora.
- Tomić, B., Jovanović, J., Milikić, N., Devedžić, V., Dimitrijević, S., Đurić, D., and Šeparac, Z. (2019). Grading students’ programming and soft skills with open badges: A case study. *British Journal of Educational Technology*, 50(2):518–530.
- Tripp, D. (2005). Pesquisa-ação: uma introdução metodológica. *Educação e pesquisa*, 31:443–466.
- Tsortanidou, X., Daradoumis, T., and Barberá, E. (2023). A k-6 computational thinking curricular framework: pedagogical implications for teaching practice. *Interactive Learning Environments*, 31(8):4903–4923.
- Wang, X. and Herzog, S. (2023). A conceptual design for a learning platform enhancing children and youth’s soft skills education. In *2023 3rd International Conference on Educational Technology (ICET)*, pages 149–153. IEEE.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Wing, J. M. (2014). Computational thinking benefits society. *40th anniversary blog of social issues in computing*, 2014:26.
- Wong, G. K.-W. and Cheung, H.-Y. (2020). Exploring children’s perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 28(4):438–450.
- Yurchenko, A., Drushlyak, M., Khvorostina, Y., Ostroha, M., Ponomarenko, V., and Semenikhina, O. (2024). The impact of team competitions on the development of soft skills in youth. 2024 47th mipro ict and electronics convention (mipro), 323–328.