

Framework SoPensa para o ensino de lógica de programação: um relato de experiência no ensino técnico integrado

Antonio Rege^{1,2}, Cássio Noronha³, 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 – Brazil

³Instituto Federal da Bahia – (IFBA) – Campus Eunápolis
Eunápolis – Bahia – BA – Brasil

{regejs, cassioac12}@gmail.com, {viterbo, luciana}@ic.uff.br

Abstract. *This paper reports the application of the SoPensa framework, aimed at integrating computational thinking and soft skills into introductory programming teaching. The experience was conducted in two first-year classes in an integrated technical course in Computing, with 70 students, during the 2025 academic semester. The intervention followed four phases, combining unplugged and plugged activities, supported by a formative cycle and monitoring instruments, with teacher-led mediation. Qualitative records and a Likert questionnaire (n=64) indicated a favorable perception of the proposal, with abstraction emerging as a recurring difficulty across four moments of the intervention, and collaboration and critical thinking with less recognition in students' self-perception.*

Resumo. *Este artigo relata a aplicação do framework SoPensa, voltado à integração do pensamento computacional e das soft skills no ensino introdutório de programação. A experiência foi conduzida em duas turmas do 1º ano do técnico integrado em Informática, com 70 estudantes, no semestre letivo de 2025. A intervenção percorreu quatro fases, combinando atividades desplugadas e plugadas, com mediação apoiada em um ciclo formativo e instrumentos de acompanhamento. Registros qualitativos e um questionário Likert (n=64) indicaram percepção favorável à proposta, com a abstração como dificuldade recorrente em quatro momentos da intervenção e a colaboração e o pensamento crítico com menor reconhecimento na autopercepção dos estudantes.*

1. Introdução

No ensino de Lógica de Programação, as dificuldades dos estudantes não se limitam ao uso da sintaxe ou ao domínio de comandos. Elas também envolvem interpretação de problemas, organização do raciocínio, abstração e tomada de decisão durante a construção de soluções [Silva and Moreira 2021]. Nesse cenário, o Pensamento Computacional (PC) oferece uma base para trabalhar processos de resolução de problemas antes e durante a codificação [Wing 2014].

Além dos processos cognitivos envolvidos na resolução de problemas, a aprendizagem de programação também envolve Soft Skills (SS), como

colaboração, comunicação, persistência, criatividade e pensamento crítico [Hudin 2023, Jalinus et al. 2024, Coelho et al. 2024]. A Sociedade Brasileira de Computação (SBC), ao definir os Grandes Desafios da Educação em Computação para 2025-2035, inclui o desenvolvimento e a avaliação de SS entre as cinco prioridades, além de observar que os currículos de referência reconhecem essas competências, mas as metodologias usuais não favorecem seu desenvolvimento [SBC 2025].

Apesar desse reconhecimento, propostas que reúnam PC e SS desde o ensino introdutório ainda são escassas [Maitz et al. 2022], com lacuna específica no ensino técnico de nível médio [Rege et al. 2025]. A partir desse problema, este relato apresenta a aplicação do SoPensa no ensino introdutório de programação.

O SoPensa é um framework de mediação docente para o ensino de programação baseado na integração entre PC e SS, estruturado em quatro fases sequenciais e iterativas: Diagnosticar e Organizar, Contextualizar e Ambientar, Desafiar e Mediar, Analisar e Consolidar. Nas Fases 2 e 3, a mediação docente apoia-se em um ciclo formativo de três etapas (Pensar, Agir e Refletir), acrescido da etapa Comunicar na Fase 3, que articula os componentes do PC e as competências de SS no trabalho com os estudantes. Sua fundamentação conceitual e sua avaliação com docentes são apresentadas em trabalho correlato.

Este artigo relata a aplicação do SoPensa em duas turmas do 1º ano dos cursos Técnico em Informática para Internet e Técnico em Informática com ênfase em Redes, no Instituto Federal do Acre, Campus Rio Branco, com 70 estudantes matriculados no semestre letivo de 2025. O objetivo é descrever a aplicação das quatro fases na disciplina de Lógica de Programação e discutir, a partir da experiência, os elementos que se mostraram efetivos, os ajustes incorporados ao longo da intervenção e as dificuldades que persistiram. O artigo está organizado da seguinte forma: a Seção 2 descreve o contexto e percurso da intervenção; a Seção 3 apresenta os resultados e a discussão por fase, seguidos das percepções discentes; a Seção 4 sintetiza as considerações finais.

2. Contexto e percurso da intervenção

Esta seção descreve o contexto da aplicação do SoPensa, os participantes, a estrutura da intervenção e os instrumentos utilizados para acompanhá-la.

2.1. Contexto, local e participantes

A intervenção ocorreu no primeiro semestre de 2025, no âmbito da disciplina anual de Lógica de Programação, cuja carga horária total é de 120 horas, das quais aproximadamente metade foi dedicada à intervenção. A disciplina integra a matriz do 1º ano dos cursos técnicos do eixo de Informação e Comunicação ofertados pelo Instituto Federal do Acre, Campus Rio Branco.

Participaram duas turmas, uma do curso Técnico em Informática para Internet e outra do curso Técnico em Informática com ênfase em Redes, cada uma com 35 estudantes matriculados, totalizando 70 participantes entre 15 e 16 anos. As aulas foram conduzidas por dois professores responsáveis pelas turmas, com formação em Sistemas de Informação e Ciência da Computação, respectivamente, e oito e cinco anos de experiência no ensino de Lógica de Programação. O primeiro autor, responsável pela concepção do

framework, acompanhou as duas turmas com atuação de natureza técnica e operacional, mantendo diário reflexivo. A condução das aulas permaneceu sob responsabilidade dos professores. A aplicação paralela do SoPensa foi favorecida pelo fato de as turmas cursarem a disciplina introdutória de programação no mesmo período letivo, permitindo analisar registros de dois cursos com perfis distintos.

O estudo foi aprovado pelo Comitê de Ética em Pesquisa da instituição, sob CAAE 85642524.5.0000.0233. Os 64 estudantes que responderam ao questionário final foram anonimizados no formato A1 a A64, código utilizado quando suas respostas são citadas no texto. Antes da aplicação em sala, o framework foi revisado por dois professores experientes na disciplina de Lógica de Programação, com o objetivo de identificar ajustes necessários. Esses professores revisores não correspondem aos participantes da intervenção. A apreciação considerou a clareza das descrições, a coerência pedagógica entre os componentes e a viabilidade de aplicação em sala de aula. Com base nessa apreciação, foram incorporados à versão aplicada ajustes como a padronização da nomenclatura dos agrupamentos, a renomeação da Fase 3 para reforçar o papel mediador do professor, a explicitação do fluxo de retorno às Fases 2 e 3 de acordo com a lacuna identificada e a definição dos momentos de autoavaliação dos times.

2.2. Estrutura da intervenção e recursos

A aplicação seguiu as quatro fases do SoPensa apresentadas na Figura 1: Diagnosticar e Organizar (Fase 1), Contextualizar e Ambientar (Fase 2), Desafiar e Mediar (Fase 3) e Analisar e Consolidar (Fase 4). A mediação docente nas Fases 2 e 3 apoiou-se em um ciclo formativo composto pelas etapas Pensar, Agir e Refletir, ao qual se soma Comunicar na Fase 3, que mobiliza, de forma articulada, os quatro componentes do PC (decomposição, reconhecimento de padrões, abstração e algoritmos) e as quatro competências de SS (colaboração, comunicação, criatividade e pensamento crítico).

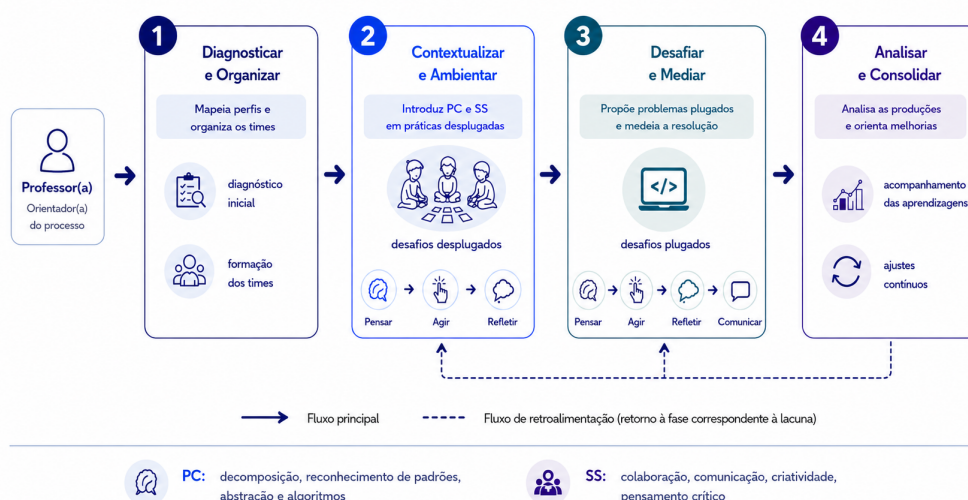


Figura 1. Estrutura do framework SoPensa

A Fase 1 mapeou o perfil inicial dos estudantes e organizou os times. Na Fase 2, de natureza desplugada e com foco na ambientação dos estudantes, o ciclo limitou-se a Pensar, Agir e Refletir, e a troca entre os estudantes ocorreu internamente ao time,

sob a forma de discussão dos resultados, sem apresentação pública à turma. Na Fase 3 (plugada), quando as atividades passaram a exigir maior autonomia técnica, acrescentou-se a etapa Comunicar, com apresentação das soluções, relato do processo e justificativa das escolhas técnicas. A Fase 4 encerrou o percurso com a análise das produções dos times e o feedback de retomada. As fases ocorreram em sequência, mantendo os times da Fase 1 e a rotação de papéis a cada aula.

Os recursos materiais envolveram os laboratórios de informática da instituição, materiais desplugados preparados previamente pela equipe de pesquisa, entre eles as caixas de Missões Lógicas e a demarcação do percurso do Robô HidroLógico, o ambiente Scratch para programação visual e o ambiente Online Python para programação textual. Os estudantes utilizaram os computadores do laboratório durante as aulas para acessar os ambientes de programação e desenvolver as atividades plugadas.

2.3. Instrumentos de acompanhamento e análise

Os instrumentos foram planejados para acompanhar a intervenção em momentos diferentes. Antes da aplicação, um questionário diagnóstico com onze itens mapeou o contato prévio dos estudantes com programação e seus perfis em PC e SS. Durante a aplicação, os professores utilizaram rubricas qualitativas, fichas de observação e fichas de autoavaliação dos times. As autoavaliações foram preenchidas pelos estudantes ao final das Fases 2 e 3 e retomadas na Fase 4 durante o feedback. O pesquisador, por sua vez, manteve um diário reflexivo com registros sobre as interações e a mediação, utilizado na análise qualitativa da experiência.

Os três instrumentos do framework, somados ao diário reflexivo do pesquisador, foram revisitados durante a intervenção e na etapa de redação dos resultados. Ao término da intervenção, foi aplicado um questionário Likert de cinco pontos, organizado em cinco eixos: contribuição geral, utilidade das atividades, desenvolvimento de PC, desenvolvimento de SS e recomendação de uso. As respostas foram apresentadas por distribuição percentual, em barras horizontais empilhadas.

3. Resultados e Discussão

Esta seção apresenta os resultados da aplicação do SoPensa nas duas turmas, destacando decisões pedagógicas, ajustes e pontos críticos em cada fase do framework.

3.1. Fase 1 - Diagnosticar e Organizar

A Fase 1 teve por objetivo levantar o perfil inicial dos estudantes em PC e SS, bem como organizar times com perfis equilibrados para as fases seguintes. Na primeira semana de abril, os 70 estudantes das duas turmas responderam ao questionário diagnóstico, composto por onze itens que cobriram o contato prévio com programação e a autoavaliação em quatro componentes do PC (decomposição, reconhecimento de padrões, abstração e algoritmos) e quatro competências de SS (colaboração, criatividade, pensamento crítico e comunicação), além de informações sobre o perfil da turma.

Sobre o contato prévio com programação, apenas cinco estudantes de Informática para Internet e quatro de Redes relataram alguma experiência, quase sempre em atividades de robótica educacional. Esses estudantes foram distribuídos entre os times, quando possível, para evitar a concentração desse conhecimento em uma única equipe.

As respostas foram classificadas sem caráter avaliativo individual, apenas como apoio à leitura do perfil das turmas. Com base no diagnóstico, foram organizados, em cada turma, sete times de cinco integrantes, balanceados pelas respostas codificadas como adequado (1) ou em desenvolvimento (0). A tabulação e a execução do algoritmo de agrupamento couberam ao pesquisador, e a composição resultante foi validada pelos professores antes do prosseguimento. Os estudantes foram ordenados pela média individual de SS, usando a média de PC como critério de desempate, e distribuídos pelo algoritmo *round-robin* para evitar perfis semelhantes em um mesmo time, com trocas pontuais entre times quando havia desequilíbrio nas médias gerais. O resultado foram times equilibrados nas duas turmas, com variação máxima de 0,05 entre as médias gerais em Informática para Internet e de 0,025 em Informática em Redes, preservando a heterogeneidade interna e garantindo justiça entre as equipes.

Os resultados por dimensão revelaram dois perfis de turma distintos, apresentados na Tabela 1. O reconhecimento de padrões foi o componente de maior adequação em ambas as turmas, com percentuais iguais ou superiores a 80%, e a abstração foi a de menor adequação nas duas, abaixo de 30%, resultado que dialoga com estudos sobre a dificuldade de abstração entre iniciantes de programação [Qian and Lehman 2017, Cheah 2020]. As demais dimensões diferenciaram os perfis das turmas, uma vez que Informática para Internet se destacou em pensamento crítico, enquanto a turma de Redes se destacou em colaboração.

Tabela 1. Adequação inicial por dimensão e turma (%)

Dimensão avaliada	Informática para Internet	Informática em Redes
PC		
Reconhecimento de padrões	85,7	80,0
Decomposição	54,3	51,4
Algoritmos	51,4	42,9
Abstração	25,7	28,6
SS		
Colaboração	62,9	82,9
Pensamento crítico	71,4	40,0
Criatividade	51,4	34,3
Comunicação	40,0	42,9

A Fase 1 não se limitou à coleta de dados. O contato prévio reduzido em programação reforçou a necessidade de iniciar a Fase 2 por atividades desplugadas, a baixa adequação em abstração antecipou um ponto crítico das fases seguintes, e os dois perfis de turma orientaram ajustes diferentes de mediação em cada contexto, o que fez do diagnóstico a base operacional para o planejamento das Fases 2 e 3.

3.2. Fase 2 - Contextualizar e Ambientar

A Fase 2 teve por objetivo introduzir os fundamentos de PC e SS por meio de atividades dialogadas e vivências desplugadas, preparando os estudantes para os desafios técnicos da fase seguinte. A opção por uma fase inicial sem tecnologia atendeu ao perfil das turmas observado no diagnóstico, já que os estudantes, com pouca experiência prévia em programação, precisavam de uma base comum antes de lidar com código.

A fase começou com duas aulas dialogadas, cada uma dedicada a dois componentes do PC, conduzidas a partir do exemplo cotidiano da construção de uma casa. A

decomposição foi trabalhada pela divisão do processo de construção em etapas menores, do projeto ao acabamento; o reconhecimento de padrões partiu da mesma casa em versões de cores diferentes, indicando que a estrutura se mantinha apesar das variações; a abstração foi explorada pela pergunta sobre o que de fato precisaria aparecer para que alguém reconhecesse uma casa, acompanhada por uma sequência de imagens com simplificação progressiva, da casa detalhada à forma essencial; e os algoritmos foram introduzidos pela proposta de ensinar um robô a construí-la, organizando os passos até chegar a uma sequência em Português. As SS foram apresentadas em uma aula expositiva ao final do ciclo dialogado, com exemplos do cotidiano profissional do programador.

Ao final das aulas, foram propostas duas atividades. A primeira foi o *Robô HidroLógico*, atividade desplugada em que cada time conduzia um copo de água do ponto inicial até o professor por meio de instruções escritas pelos próprios estudantes, com um colega no papel de robô-executor, que cumpria os comandos sem improvisar. Cada erro de percurso obrigava o time a refazer o algoritmo. A atividade foi planejada para mobilizar os quatro componentes do PC e as competências de SS em um mesmo exercício. O percurso incluía obstáculos demarcados no chão, os quais deveriam ser contornados pelo robô-executor.

Os registros em diário das observações revelaram um contraste entre as turmas, em Informática para Internet houve mais agilidade e iniciativa na execução, com estratégias variadas para superar os obstáculos; em Redes, os estudantes dedicaram mais tempo à discussão e à validação dos comandos, além de demonstrarem mais dificuldade para resolver os problemas propostos e coordenar as decisões dentro do time. Embora o diagnóstico inicial tenha indicado melhor resultado da turma de Redes em colaboração, a observação prática revelou dificuldade dos times na hora de colaborar. Por isso, o professor da turma de Redes ajustou a mediação, dedicando mais tempo ao planejamento e à organização das atividades coletivas.

A segunda atividade desplugada foi o jogo *Caixas de Missões Lógicas*, voltado ao reforço da lógica proposicional. Após o trabalho com os componentes do PC, esse conteúdo inicial da ementa foi retomado em uma dinâmica prática, com foco em proposições, conectivos lógicos e construção de tabelas-verdade. Foram confeccionadas sete caixas, uma por time, cada uma contendo as mesmas quatro missões contextualizadas com o cotidiano: *Alavanca da Porta*, *O Cofre Silencioso*, *Alerta no Laboratório* e *Código da Mente*. Cada time abria uma missão por vez e só avançava após a validação da solução pelo professor, em cerca de duas aulas consecutivas de 45 minutos, com premiação simbólica para o time que resolvesse mais desafios. As observações registradas em diário indicaram que a decomposição e o reconhecimento de padrões apareceram com mais frequência nas resoluções, enquanto a abstração já se mostrava difícil nesta etapa, com parte dos times encontrando obstáculos para selecionar as informações essenciais antes de montar as tabelas-verdade no papel.

3.3. Fase 3 - Desafiar e Mediar

Na Fase 3, o professor propôs aos estudantes desafios plugados de complexidade crescente, com a aplicação em código dos conceitos previstos na ementa. Os conteúdos trabalhados envolveram tipos de dados, variáveis e constantes, entrada e saída de dados, algoritmos e operadores aritméticos, relacionais e lógicos, estes últimos aplicando a lógica

proposicional já introduzida na Fase 2. A fase foi composta por duas atividades plugadas sequenciais, conduzidas primeiro no ambiente Scratch, com programação por blocos visuais, e em seguida na linguagem Python, com programação textual. A progressão do visual para o textual buscou aproximar gradualmente os estudantes da resolução de problemas em programação [Resnick et al. 2009].

Nesta fase, a etapa Comunicar do ciclo formativo passou a ser formalizada por meio da apresentação das soluções em sala de aula, com relato do processo e justificativa das escolhas técnicas. Os times foram mantidos como na Fase 1, sem solicitações de troca de integrante. Essa estabilidade, somada à maior desenvoltura dos estudantes nas interações, sustentou as apresentações públicas das soluções em sala de aula.

A primeira atividade foi conduzida no Scratch, com a proposta *A Travessia do Rio Açaí*. O personagem principal deveria atravessar o rio, coletar itens e evitar obstáculos até alcançar a meta, mobilizando variáveis de pontuação, controle de tempo e condições de vitória e derrota. Os registros em diário indicaram algoritmos mais organizados do que nas atividades desplugadas da Fase 2 e reconhecimento de padrões frequente na reutilização de blocos semelhantes. Foi a atividade de maior engajamento da fase, com comunicação intensa e colaboração frequente entre os times. O apoio visual dos blocos, que mantém parte da estrutura lógica à vista do estudante, parece ter favorecido a criatividade das soluções, como sugerem os registros em rubrica e diário.

A segunda atividade foi conduzida em Python, no ambiente Online Python, após aulas introdutórias sobre operadores, variáveis e estruturas condicionais. Foram propostos dois desafios contextualizados, sintetizados na Tabela 2, com parâmetros que exigiam o uso conjunto de operadores aritméticos, relacionais e lógicos. As observações registraram obstáculos na construção dos algoritmos, com vários times produzindo códigos que funcionavam apenas em parte e demandas frequentes de mediação durante a atividade.

O contraste entre as turmas, antecipado no diagnóstico, apareceu de forma concreta no fechamento da etapa, pois, antes da retomada prevista para a Fase 4, um time de Informática para Internet e três times de Redes ainda não haviam concluído o desafio. Também ficou mais visível a diferença de ritmo entre estudantes de um mesmo time, com alguns avançando mais rapidamente que os colegas. Na turma de Redes, três estudantes permaneceram retraídos ao longo das dinâmicas, reforçando a necessidade de mediação individualizada e de ações específicas para estimular a comunicação e o engajamento.

Tabela 2. Problemas propostos em Python na atividade plugada da Fase 3

Desafio	Descrição do problema	Regras ou instruções de resolução
1 - A Porta Secreta	Criar um sistema de segurança que libere o acesso apenas se a senha for 1234 e o usuário tiver autorização, representada por 1 para sim e 0 para não.	O programa solicita a senha e o status de autorização, verifica as duas condições e exibe “Acesso liberado” quando ambas são verdadeiras, ou “Acesso negado” caso contrário.
2 - Corrida contra o Relógio	Desenvolver um programa que verifique se o estudante consegue pegar o ônibus escolar com base na hora atual, no tempo até o ponto e no horário de saída.	O programa calcula o horário de chegada, compara com o horário de saída do ônibus (com tolerância de 5 minutos) e informa se o estudante conseguiu ou perdeu o transporte.

A análise das duas atividades indica o papel da abstração nesta fase. No ambiente

visual, os blocos sustentam parte do raciocínio e mantêm a estrutura da solução à vista do estudante; no código textual, esse apoio desaparece e o estudante precisa formular com maior autonomia a estrutura do problema antes de escrevê-la. Foi nessa transição que as dificuldades se concentraram, o que sugere que a abstração demanda mediação docente reforçada justamente na passagem entre os dois ambientes. Já a comunicação e a colaboração mostraram-se mais presentes do que nas atividades desplugadas, favorecidas pela formalização da etapa Comunicar e pela natureza das propostas. Esses aspectos reaparecem nas percepções discentes da Subseção 3.5.

3.4. Fase 4 - Analisar e Consolidar

A Fase 4 encerrou o percurso do framework com a análise das produções dos times e a reorientação das aprendizagens. Encerrada a implementação das atividades plugadas, os professores examinaram as entregas e, a partir dessa leitura, conduziram um momento de feedback e retomada. A leitura da situação de cada time apoiou-se em três instrumentos já descritos na Subseção 2.3. As rubricas qualitativas permitiram identificar os componentes do PC e as competências de SS em que cada time avançou ou apresentou dificuldades ao longo das Fases 2 e 3. A ficha de observação trouxe registros sobre a clareza das atividades, o envolvimento dos estudantes e a mediação em sala. A autoavaliação acrescentou a visão interna dos próprios times sobre como haviam se organizado. Esses registros permitiram que o professor compreendesse a situação de aprendizagem de cada time antes da devolução das atividades.

Com base nesse diagnóstico, os professores corrigiram as produções e identificaram fragilidades nas entregas. Os times cujas produções precisavam de ajuste foram reconduzidos à Fase 2, quando a lacuna estava nos fundamentos trabalhados nas atividades desplugadas, ou à Fase 3, quando dizia respeito à construção do código, e os times que voltaram melhoraram suas entregas. A abstração reapareceu com função distinta em relação às fases anteriores; se antes havia sido o ponto mais difícil de mobilizar, agora motivou a maior parte das devoluções para revisão, sobretudo nos códigos em que os times não haviam separado com clareza a estrutura essencial do problema. O ciclo de feedback e retomada permitiu a esses estudantes elaborar o que a primeira tentativa não havia consolidado.

A autoavaliação dos times acrescentou um dado que o olhar docente sozinho não alcançaria. Vários times reconheceram que a participação dos integrantes nem sempre ocorreu de forma equilibrada e que a comunicação interna poderia ter sido mais clara. Esse reconhecimento é coerente com a diferença de ritmo observada na Fase 3 e ajuda a qualificar o desenvolvimento das SS, já que compor um time, por si só, não garante colaboração distribuída, e a identificação dessa lacuna pelos próprios times configura um exercício de pensamento crítico sobre a prática coletiva. A Fase 4 também levou os professores a identificar ajustes para o próprio framework. Foram apontados como prioritários a ampliação do tempo destinado às atividades, a simplificação de fichas e rubricas para uso durante a aula e a automatização da formação dos times por meio de um software.

3.5. Percepções discentes no questionário Likert pós-intervenção

O questionário Likert foi aplicado ao final da intervenção e respondido por 64 dos 70 estudantes matriculados, taxa de 91,4%. Os itens foram organizados em cinco eixos: contribuição geral do framework, utilidade das atividades, desenvolvimento percebido dos

componentes de PC, desenvolvimento percebido das competências de SS e recomendação de uso.

As respostas indicaram percepção favorável à proposta. A contribuição do framework para o aprendizado foi reconhecida por 67,2% dos respondentes, a utilidade das atividades alcançou 71,9% e a recomendação de uso para outras turmas atingiu 70,3%. Ainda no eixo de contribuição geral, dois itens complementares completam o quadro. O percentual mais discreto refere-se à percepção de aulas mais interessantes, com 57,8%, enquanto o item reverso, formulado de maneira negativa ('Algumas atividades do SoPensa não acrescentaram muito ao meu aprendizado'), registrou 81,2% de discordância, valor favorável após a inversão da escala, o que acompanha a tendência positiva dos demais itens do bloco. Os resultados referentes aos componentes do PC e às SS são apresentados na Figura 2 e discutidos a seguir.

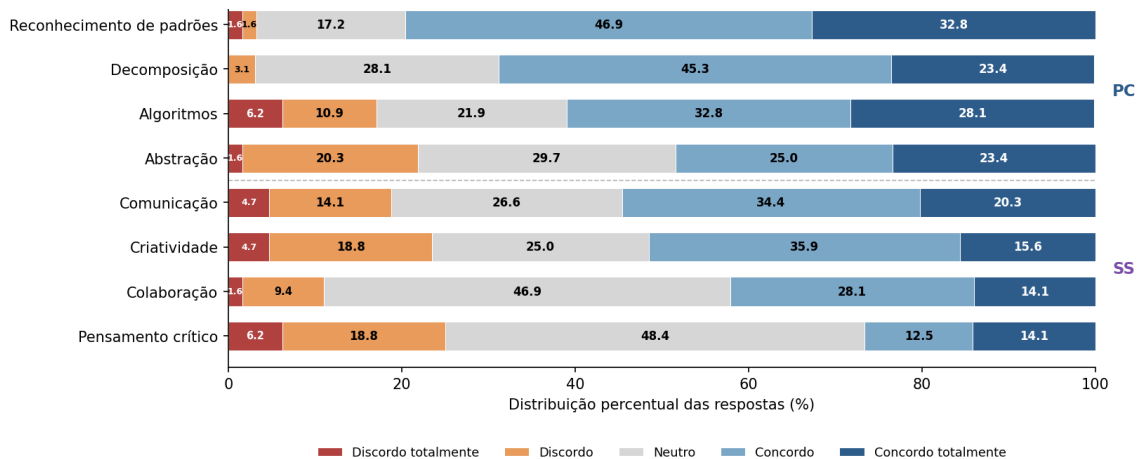


Figura 2. Percepção dos estudantes sobre os componentes do PC e as SS (n=64)

Sobre os componentes de PC, observou-se um padrão alinhado ao diagnóstico inicial. O reconhecimento de padrões foi o componente de maior concordância, com 79,7%, seguido por decomposição, com 68,7%, e algoritmos, com 60,9%. A abstração apresentou o menor índice, com 48,4% de concordância e 21,9% de discordância, o maior percentual entre os componentes. Esse resultado completa um padrão que atravessa toda a intervenção, iniciado no diagnóstico, quando a abstração apareceu como dificuldade, com 25,7% e 28,6% de adequação nas duas turmas, persistiu nas atividades das Fases 2 e 3, motivou devoluções para revisão na Fase 4 e reaparece na autopercepção dos estudantes. Uma resposta aberta mostra esse padrão: A26 declarou se sair bem na resolução de problemas, mas reconheceu dificuldade em “pensar fora da caixa”, referência direta ao que a abstração exige. A recorrência desse padrão em quatro momentos distintos indica que a abstração demanda atenção nas disciplinas introdutórias de programação [Cheah 2020].

Nas SS, a comunicação obteve a maior concordância, com 54,7%, seguida pela criatividade, com 51,5%. A colaboração registrou 42,2% de concordância e 46,9% de neutralidade; o pensamento crítico apresentou a menor concordância, 26,6%, e o maior percentual de neutralidade do questionário, 48,4%. Foram os dois itens com maior concentração de respostas neutras. Duas respostas abertas ajudam a entender o resultado da colaboração: A58 relatou ter feito parte das atividades sozinho, por não conseguir trabalhar adequadamente em equipe, e A62 registrou que o trabalho fluía quando havia

proximidade com os colegas, mas que, em outros momentos, o time “travou” por falta dessa proximidade.

O contraste entre PC e SS pede interpretação cuidadosa. Embora o framework proponha o desenvolvimento conjunto das duas dimensões, a percepção discente foi mais favorável ao PC do que às SS, e, dentro das SS, colaboração e pensamento crítico apresentaram menor reconhecimento pelos estudantes. A experiência indica que inserir as SS no planejamento das atividades não basta para que os estudantes reconheçam seu desenvolvimento, pois competências interpessoais parecem exigir uma mediação intencional, que sinalize ao estudante quais competências de SS estão sendo desenvolvidas.

O elevado percentual de neutralidade, mais do que de discordância, pode sugerir menos uma rejeição à proposta e mais uma dificuldade dos estudantes em reconhecer essas habilidades no próprio percurso. Essa leitura é coerente com a natureza menos observável de competências como colaboração e pensamento crítico, bem como com a duração relativamente curta de um semestre letivo.

4. Considerações Finais

Este relato descreveu a aplicação do framework SoPensa na disciplina de Lógica de Programação do 1º ano do técnico integrado, em duas turmas do Instituto Federal do Acre, com 70 estudantes, no semestre letivo de 2025. A análise dos registros qualitativos e das respostas ao questionário Likert (n=64) indicou dois aspectos da experiência. O primeiro foi a abstração como dificuldade persistente, observada em quatro momentos distintos, no diagnóstico inicial, nas atividades das Fases 2 e 3, nas devoluções para revisão da Fase 4 e na autopercepção dos estudantes. O segundo foi um limite da integração pretendida, pois a percepção discente foi mais favorável ao PC do que às SS. A inserção das SS no planejamento não foi suficiente para que os estudantes reconhecessem seu desenvolvimento, o que indica a necessidade de mediação que torne explícito o que está sendo trabalhado.

Entre as limitações, destacam-se o duplo papel do pesquisador, autor do framework e observador da intervenção, e o possível viés decorrente dessa condição. Esse risco foi reduzido pela condução das aulas por dois professores que não participaram da elaboração do framework e pela manutenção de registros em diário reflexivo. Também constituem limitações a aplicação em uma única instituição, com duas turmas, durante um semestre e sem grupo de controle, bem como o tratamento apenas descritivo do questionário, que impede atribuir os resultados exclusivamente ao framework.

Como continuidade, os próprios docentes sugeriram, durante a intervenção, a ampliação do tempo destinado às atividades, a simplificação dos instrumentos de acompanhamento e a automatização da formação dos times por meio de um sistema de apoio. A experiência indica que o trabalho com os componentes do PC e as competências de SS é viável no ensino introdutório de programação no técnico integrado, desde que apoiado em mediação docente intencional e em instrumentos que tornem visíveis, ao longo do processo, tanto os componentes do PC quanto as competências de SS. A fundamentação conceitual do framework e sua avaliação por docentes são apresentadas em trabalho correlato.

Declaração de Uso de Inteligência Artificial

Os autores utilizaram o modelo de linguagem Claude (Anthropic) apenas para revisão gramatical e ortográfica. Nenhuma seção foi integralmente gerada por inteligência artificial. A concepção do estudo, o delineamento metodológico, a coleta e a análise dos dados, a interpretação dos resultados e o conteúdo final são de responsabilidade dos autores.

Referências

- 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.
- Coelho, M., Araújo, A. A., Freire, S., and Paixao, M. (2024). Um estudo comparativo entre a visão de líderes e liderados sobre a importância de soft skills em desenvolvimento de software. In *Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES)*, pages 130–140. SBC.
- 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).
- 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. GI.
- Qian, Y. and Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–24.
- 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.
- SBC (2025). *Grandes Desafios da Educação em Computação 2025-2035: Resumo Executivo*. Sociedade Brasileira de Computação (SBC), Porto Alegre.
- Silva, F. and Moreira, I. (2021). Análise das dificuldades na aprendizagem de programação no curso de análise e desenvolvimento de sistemas do ifrn/pau dos ferros. In *Anais do XIV Encontro Unificado de Computação do Piauí e XI Simpósio de Sistemas de Informação*, pages 41–48, Porto Alegre, RS, Brasil. SBC.
- Wing, J. M. (2014). Computational thinking benefits society. *40th anniversary blog of social issues in computing*, 2014:26.