

# Revisão Sistemática sobre Inventário de Conceitos aplicados em disciplinas introdutórias de cursos de computação

Inara Santana Ortiz<sup>1</sup>, Amaury Antônio Castro Junior<sup>1</sup>, Anderson Corrêa de Lima<sup>1</sup>

<sup>1</sup>Faculdade de Computação - Universidade Federal de Mato Grosso do Sul (UFMS)  
Caixa Postal 549 – 79.070-900 – Campo Grande – MS – Brazil

{inara.ortiz,amaury.junior, anderson.lima}@ufms.br

**Abstract.** This study carries out a Systematic Literature Review (SLR) on the use of Concept Inventories (CIs) in introductory computing subjects. CIs are valuable tools for identifying and evaluating misconceptions, i.e. students' mistaken understanding of certain fundamental concepts in the field, which significantly hinder the learning process. This RSL demonstrates the importance of CIs in computer science learning, by identifying gaps in students' knowledge and guiding teaching practices. The research also points to challenges such as constructing high-quality CI items and adapting them to different contexts. The results of this research contribute to the advancement of research in computing education, offering subsidies for the development of CIs, highlighting the need to expand the use of CIs to different disciplines, levels of education and different educational contexts.

**Resumo.** Este estudo realiza uma Revisão Sistemática da Literatura (RSL) sobre o uso de Inventários de Conceitos (ICs) em disciplinas introdutórias de computação. Os ICs são ferramentas valiosas para identificar e avaliar concepções errôneas, ou seja, a compreensão equivocada dos estudantes sobre determinados conceitos fundamentais da área, que dificultam significativamente o processo de aprendizagem. Essa RSL demonstra a importância dos ICs na aprendizagem em computação, ao identificar lacunas de conhecimento dos estudantes e orientar práticas pedagógicas. A pesquisa também aponta desafios como a construção de itens do IC de alta qualidade e sua adaptação a diferentes contextos. Os resultados dessa pesquisa contribuem para o avanço da pesquisa em educação em computação, oferecendo subsídios para o desenvolvimento de ICs, evidenciando a necessidade de expandir o uso de IC para diversas disciplinas, níveis de ensino e diferentes contextos educacionais.

## 1. Introdução

O desenvolvimento tecnológico acelerado e a crescente digitalização do mundo contemporâneo têm impulsionado mudanças na educação em computação. A Lei n.º 14.533/2023 [Brasil 2023], que estabelece a Política Nacional de Educação Digital, juntamente com a inclusão da computação na Base Nacional Comum Curricular (BNCC) [Brasil 2022], são exemplos de iniciativas que visam integrar competências digitais ao currículo escolar, promovendo a alfabetização digital desde os anos iniciais.

No entanto, um dos desafios no ensino de computação, é a dificuldade dos estudantes em compreender os conceitos abordados nas disciplinas introdutórias, que são

novos e exigem raciocínio lógico e abstrato diferente do utilizado em outras disciplinas [Duran et al. 2023, Fukao et al. 2023, Barbosa et al. 2011]. Segundo o “13º Mapa do Ensino Superior no Brasil”, publicado pelo Instituto Semesp em 2023, a taxa de evasão em cursos Tecnologia da Informação (TI) continua alta, refletindo dificuldades na retenção de estudantes [Semesp 2023].

Para superar desafios no ensino, é fundamental desenvolver abordagens acessíveis e eficazes, facilitando a compreensão de conceitos e auxiliando professores na adaptação de suas práticas. Metodologias inovadoras, como o uso de Inventários de Conceitos (ICs), têm se destacado, especialmente em disciplinas introdutórias de computação. Um IC é uma ferramenta de avaliação padronizada e projetada para medir a compreensão do estudante sobre os conceitos principais de um tópico [Goldman et al. 2010]. Pode ser usada como teste de diagnóstico, para identificar atividades de ensino e aprendizagem apropriadas, para avaliar o impacto da mudança de métodos de instrução na compreensão do estudante, para orientar os estudantes, para comparar métodos de instrução ou para avaliar o aprendizado geral e os efeitos instrucionais [Julie and Bruno 2019]. Na computação, ICs vêm ganhando destaque, especialmente em cursos introdutórios de programação, onde os estudantes precisam desenvolver uma compreensão sólida de conceitos-chave.

Este estudo realizou uma Revisão Sistemática da Literatura (RSL) sobre o uso e desenvolvimento de ICs em disciplinas introdutórias de cursos de computação, analisando os diferentes métodos utilizados para a criação e aplicação desses instrumentos, identificando as principais tendências e desafios na área. Para garantir a originalidade e relevância dos resultados, esta pesquisa não incluiu estudos secundários, como revisões sistemáticas ou mapeamentos sistemáticos da literatura. A exclusão dessas fontes se justifica pela necessidade de coletar evidências diretas e não sintetizadas sobre a aplicação dos ICs, permitindo uma análise mais detalhada e específica das metodologias, dificuldades e impactos observados nos estudos primários. Neste estudo, considera-se como disciplinas introdutórias de computação aquelas ofertadas nos primeiros anos da formação em computação, representando o primeiro contato dos estudantes com conceitos fundamentais da área. Essas disciplinas abrangem conteúdos essenciais, como lógica de programação, algoritmos, fundamentos de computação, pensamento computacional e introdução à ciência da computação. Além disso, considerou-se que essas disciplinas podem estar presentes em diferentes níveis de escolaridade, incluindo ensino básico, ensino médio e ensino superior, uma vez que a computação tem sido cada vez mais incorporada nos currículos escolares por meio de iniciativas educacionais e políticas públicas.

O estudo se organiza da seguinte forma: na Seção 2 Referencial teórico, alguns conceitos importantes para a compreensão do estudo são apresentados. Na Seção 3, é descrita a metodologia adotada para a RSL. Na Seção 4, são organizados e discutidos os dados relacionados à cada questão de pesquisa (QP). Por fim, as considerações finais são apresentadas na Seção 5.

## **2. Referencial teórico**

### **2.1. Inventário de Conceitos**

Um inventário de conceitos é uma ferramenta que consiste em questões de múltipla escolha, onde cada questão tem uma resposta correta e várias erradas. As erradas são os Distratores ou Concepção Errada, do inglês *Misconception*. Cada resposta errada do

questionário foi validada para que, se um estudante a escolher, possa ser determinado qual concepção errada o estudante tem [Lu and Krishnamurthi 2024]. A (Figura 1), adaptada de [Caceffo et al. 2018, Caceffo et al. 2017], apresenta um exemplo de questão de múltipla escolha elaborada para um IC composto por 27 itens, todos relacionados à linguagem C. A questão, apresentada à esquerda, solicita que o estudante complete uma função simples de adição. À direita, o gabarito detalha cada alternativa, indicando a correta e as demais, que representam concepções errôneas (*Misconception*) comuns sobre programação, as quais correspondem os erros mais frequentes cometidos pelos estudantes ao resolverem problemas de programação em C.

O código a seguir inclui uma função que adiciona cinco a um número. Ela é chamada em `x` na função principal:

```

1. int addFiveToNumber (int n) {
2.     int c = 0;
3.     // Insert code here
4.     return c;
5. }
6.
7. int main () {
8.     int x = 0;
9.     x = addFiveToNumber (x);
10.    return 0;
11. }
```

O código correto a ser inserido na linha 3 é:

- a) `scanf ("%d", &n);`
- b) `n = n + 5;`
- c) `c = n + 5;`
- d) `c = x + 5;`

#### Gabarito:

Choice	Misconception
a	A.1
b	A.2
c	Right Answer
d	A.3

#### Descrição das *misconception* identificadas:

- A.1 Definindo um valor de parâmetro por meio de uma fonte fora do escopo da função.
- A.2. Parâmetros passados por referência, e não por valor (ou seja, alterações feitas dentro de uma função seriam refletidas fora dela).
- A.3. Suposição de que os parâmetros podem ser acessados de fora do seu escopo, em qualquer lugar do programa.

**Figura 1. Adaptação de uma questão de um Inventário de Conceitos em linguagem C. Fonte: [Caceffo et al. 2018, Caceffo et al. 2017]**

Um IC não é um teste abrangente de tudo o que um estudante deve saber sobre um tópico, eles testam apenas conceitos críticos de um tópico. Se os estudantes demonstrarem compreensão desses conceitos críticos, então é razoável acreditar que eles entendem satisfatoriamente todos os outros conceitos do tópico [Herman et al. 2010].

Os ICs são algumas vezes administrados como avaliações pré e pós-teste, de modo a controlar o conhecimento no início do curso e fornecer uma visão clara do antes e depois da aprendizagem do estudante [Taylor et al. 2014]. Seu propósito é para melhorar a pedagogia em vez de avaliar os estudantes de forma somativa [Adams and Wieman 2011].

## 2.2. Confiabilidade e Validade

Ao desenvolver um IC, é fundamental garantir que ele seja válido e confiável, para assegurar sua qualidade. A validade se refere à capacidade do instrumento de medir aquilo que se propõe a medir, enquanto a confiabilidade se refere à consistência dos resultados obtidos [Ali et al. 2023].

É possível validar as perguntas que irão compor o IC por meio de entrevistas, com

especialistas e com os estudantes para garantir a interpretação das perguntas de forma consistente [Taylor et al. 2014].

Alguns métodos estatísticos são utilizados para esse fim, os mais utilizados são Teoria de Resposta ao Item e o Alfa de Cronbach. A Teoria de Resposta ao Item (TRI) pode ser usada no processo de Validação de IC, ela mede a dificuldade e a discriminação de uma questão. Sendo assim, para mensurar a dificuldade usa-se a porcentagem de participantes que conseguiram responder corretamente à questão. A discriminação mede o desempenho do estudante em uma determinada questão e sua predição para o desempenho no questionário em geral. [Parker et al. 2016].

O método mais comumente usado para medir a confiabilidade é o alfa de Cronbach, usado para medir a consistência de um questionário, ou seja, produz resultados semelhantes em diferentes aplicações. [Field 2024].

### **3. Condução do estudo**

A presente seção detalha o processo de RSL empregado para investigar o estado da arte dos Inventários de Conceitos.

#### **3.1. Objetivos e Questões de Pesquisa**

O objetivo principal desta pesquisa é realizar uma RSL para identificar e analisar estudos que empregam IC em disciplinas de Computação. Especificamente, busca-se identificar e analisar as diferentes abordagens metodológicas empregadas, os desafios enfrentados na implementação dos ICs e os resultados obtidos em termos de aprendizagem dos estudantes. Para tanto, as seguintes perguntas de pesquisa foram formuladas:

**QP<sub>1</sub>:** Quais tópicos e áreas da computação são abordados pelos inventários de conceitos existentes?

**QP<sub>2</sub>:** Quais são as principais metodologias utilizadas para desenvolver e validar inventários de conceitos em computação?

**QP<sub>3</sub>:** Quais são os principais desafios e limitações identificados na literatura em relação ao uso de inventários de conceitos em computação?

**QP<sub>4</sub>:** Há evidências empíricas que demonstrem a melhoria do desempenho acadêmico de estudantes de computação em avaliações e exames como resultado direto da utilização de inventários de conceitos?

**QP<sub>5</sub>:** Quais são as perspectivas futuras para o uso de inventários de conceitos em pesquisa e prática em educação em computação?

#### **3.2. Definição da String de busca**

O processo de construção da string de busca foi realizado em etapas, seguindo as orientações de [Dermeval et al. 2020]. A partir das palavras-chave derivadas das questões de pesquisa, a seguinte *string* de busca foi utilizada nas bases de dados: "concept inventory"AND ("computer science")OR "computing education"OR "programming"OR "algorithms"OR "introductory programming courses").

A busca por estudos foi realizada em cinco bases de dados: ACM Digital Library, IEEE Digital Library, ISI Web of Science, SOL-SBC e Scopus. A escolha dessas bases se

**Tabela 1. Publicações encontradas, rejeitadas, duplicadas e aceitas por base de dados**

Base de Dados	Encontradas	Rejeitadas	Duplicadas	Aceitas
ACM Digital Library	181	133	19	29
IEEE Digital Library	30	11	11	8
ISI Web of Science	50	22	13	15
SOL-SBC	1	0	0	1
Scopus	104	41	44	19
<b>Total</b>	<b>366</b>	<b>207</b>	<b>87</b>	<b>72</b>

justifica por sua abrangência e relevância na área da Ciência da Computação. A inclusão da SOL-SBC visa ampliar a busca para estudos realizados no Brasil, enquanto as demais bases garantem a abrangência internacional da pesquisa.

### 3.3. Processo de seleção e coleta de dados

A busca nas bases de dados foi realizada em agosto de 2024 e 366 publicações foram encontradas. A ferramenta online Parsifal (<https://parsif.al>) foi utilizada para automatizar parte do processo de seleção, facilitando a identificação e a eliminação de estudos duplicados. Após a eliminação dos duplicados, os estudos restantes foram submetidos a uma primeira análise, com base na leitura dos resumos, foram aplicados os critérios de inclusão e exclusão, para filtrar os estudos que atendessem aos objetivos da pesquisa. A seleção dos estudos foi guiada pelos seguintes critérios de inclusão:

- Estudo que desenvolvem ou aplicam um IC e valida seus resultados, em disciplinas introdutórias em computação;
- Estudos que identificam *misconceptions* (concepções errôneas), em disciplinas introdutórias em computação;

Sendo os critérios de exclusão definidos como:

- Estudos duplicados;
- Estudo fora do escopo;
- Estudo redundante da mesma autoria;
- Estudos publicados antes de 2014

Seguindo este procedimento, estabelecido em [Dermeval et al. 2020] foi possível refinar o conjunto de estudos e avançar para as etapas subsequentes, como um total de 72 publicações aceitas conforme foram identificadas na Tabela 1.

### 3.4. Avaliação de qualidade

Após o processo de busca e seleção, foi realizada a leitura completa dos textos de cada estudo aceito, visando avaliar sua qualidade. As questões de qualidade foram desenvolvidas com base nas questões de pesquisa deste estudo, permitindo uma análise mais direcionada e alinhada aos objetivos da revisão. Durante a avaliação, cada estudo foi classificado em relação a cinco critérios de qualidade, com as respostas possíveis sendo: “Atende totalmente” (1.0), “Atende parcialmente” (0.5) e “Não atende” (0).

A pontuação final de cada estudo foi calculada com base na soma das respostas obtidas para esses critérios. Estudos que obtiveram uma pontuação inferior a 3.5 foram descartados, pois não atenderam adequadamente aos padrões estabelecidos para esta revisão. Com essa triagem, 22 estudos foram selecionados para a etapa de sumarização e síntese. Os critérios de qualidade utilizados na avaliação foram os seguintes:

**Tabela 2. Quadro Sumarização e Síntese do estudos selecionados.**

#	Titulo	Ano de publicação	Categoria do Estudo	Brasileiro	Disciplina	Nível acadêmico
1	[Rachmatullah et al. 2020]	2020	Criação	NÃO	Programação	E. fundamental
2	[Grover 2020]	2020	Criação	NÃO	Programação	E.fundamental
3	[Porter et al. 2014]	2014	Criação	NÃO	Todo o currículo de Ciência da Computação	Todos os níveis de Ciência da Computação
4	[Caceffo et al. 2019]	2019	Identificação	SIM	programação	Universitários
5	[Braz et al. 2021]	2021	Aplicação	SIM	programação	Universitários
6	[Julie and Bruno 2020]	2021	Identificação	NÃO	Programação	Universitários
7	[Julie et al. 2020b]	2020	Criação	SIM	Programação	Universitários
8	[Caceffo et al. 2016]	2016	Criação	SIM	Programação	Universitários
9	[Saarinen et al. 2019]	2019	Criação	NAO	Programação	Universitários
10	[Stephens-Martinez et al. 2017]	2017	Identificação	NÃO	Programação	Universitários
11	[Wiegand et al. 2016]	2016	Identificação	NÃO	Programação	Universitários
12	[Vahrenhold and Paul 2014]	2014	Identificação	NÃO	Algoritmos e estruturas de dados	Universitários
13	[Kennedy and Kraemer 2019]	2019	Identificação	NÃO	Programação	Universitários
14	[Kaczmarczyk et al. 2019]	2019	Identificação	NÃO	Programação	Universitários
15	[Lu and Krishnamurthi 2024]	2024	Identificação	NÃO	Programação	Universitários
16	[Wiegand et al. 2021]	2021	Identificação	NÃO	Programação	Universitários
17	[Dewji et al. 2015]	2015	Identificação	NÃO		Educadores
18	[Julie and Bruno 2019]	2019	Identificação	NAO	Programação	Universitários
19	[Ferrao et al. 2022]	2022	Aplicação	SIM	Programação	Universitários
20	[Parker et al. 2016]	2016	Aplicação	NÃO	Programação	Universtários
21	[Parker et al. 2023]	2023	Aplicação	NAO	Programação	Universtários
22	[Henry and Dumas 2020]	2020	Aplicação	NÃO	Programação	Universitários

1. O estudo aborda disciplinas na área da computação em cursos introdutórios?
2. A metodologia descrita no estudo atende aos critérios estabelecidos para o desenvolvimento e validação de inventários de conceitos em computação?
3. O artigo mostra evidências práticas da aplicação do IC no desempenho dos estudantes?
4. O estudo apresenta desafios e limitações em relação ao uso de inventários de conceitos em computação?
5. O estudo propõe perspectivas futuras ou novos desafios para o uso de inventários de conceitos na elaboração de ICs em computação?

### 3.5. Sumarização e Síntese

O quadro resumo apresentado a seguir, tabela 2, foi elaborado para identificar características chave para orientar o desenvolvimento RSL. Os estudos selecionados foram organizados em três grupos principais, com base em seus objetivos:

1. Estudos focados na criação de abordagens para identificação de *misconceptions*.
2. Estudos que, após a identificação das *misconceptions*, desenvolvem e validam um Inventário de Conceitos.
3. Estudos que aplicam um IC existente.

Para isso, foi criada a coluna “Categoria do Estudo”, que classifica os estudos de acordo com esses três objetivos.

Outro ponto relevante é verificar se algum dos estudos foi administrado no contexto brasileiro, dado que a perspectiva futura desta pesquisa é desenvolver um IC adaptado para a realidade brasileira. Além disso, foi criada a coluna “Disciplina”, para facilitar a visualização das áreas do conhecimento exploradas nos estudos analisados nesta revisão. Também foi incluída a coluna “Nível Acadêmico”, destacando o público-alvo. Por fim, o quadro também apresenta informações básicas de identificação, como o título do trabalho e o ano de publicação, para facilitar a consulta dos estudos revisados.

## 4. Análise e Discussão dos Resultados

### 4.1. QP1: Quais tópicos e áreas da computação são abordados pelos inventários de conceitos existentes?

Os inventários de conceitos (ICs) são amplamente utilizados em diversas áreas do conhecimento [Evans et al. 2003]. No campo da ciência da computação, há estudos significativos em disciplinas como lógica digital [Herman et al. 2010], Programação Orientada a Objetos [Julie et al. 2020a], Estruturas de Dados [Vahrenhold and Paul 2014, Porter et al. 2019], Segurança Cibernética [Sherman et al. 2020], Sistemas Operacionais [Webb and Taylor 2014], Engenharia de Software [Parker et al. 2016] e Fundamentos de Programação [Luxton-Reilly et al. 2018, Grover 2020].

A maioria dos ICs desenvolvidos foca em estudantes universitários, especialmente em cursos de ciência da computação e engenharia. Apesar de algumas adaptações para públicos mais jovens, como alunos do ensino básico [Grover 2020, Rachmatullah et al. 2020], o principal objetivo dos ICs continua sendo identificar dificuldades conceituais em cursos introdutórios no ensino superior.

### 4.2. QP2: Quais são as principais metodologias utilizadas para desenvolver e validar inventários de conceitos em computação?

A revisão da literatura identificou uma diversidade de metodologias para o desenvolvimento de Inventários de Conceitos (ICs), abrangendo abordagens tradicionais, baseadas na experiência de especialistas [Almstrum et al. 2006, Adams and Wieman 2011], e metodologias mais recentes que utilizam tecnologias modernas. Essas metodologias podem ser classificadas em três categorias principais: (1) identificação de *misconceptions*, (2) desenvolvimento e validação de ICs, e (3) aplicação de ICs já existentes.

Os métodos tradicionais são reconhecidos pelo rigor metodológico e validação estatística [Hamouda et al. 2017], mas demandam recursos significativos, como ciclos iterativos extensos e envolvimento de especialistas. Para superar essas limitações, abordagens alternativas têm sido exploradas, como o uso da metodologia *Evidence-Centered Design (ECD)* [Grover 2020, Rachmatullah et al. 2020], que combina fontes como avaliações prévias, revisões de currículos e contribuições de especialistas. Além disso, a abordagem de código aberto [Porter et al. 2014] facilita contribuições colaborativas e adaptações de ICs para diferentes disciplinas e contextos.

Estudos recentes também exploraram a utilização de tecnologias avançadas. Stephens et al. [Stephens-Martinez et al. 2017] analisaram grandes volumes de dados de respostas abertas e erros em atividades de programação, identificando padrões mais precisos de *misconceptions*. Saarinen et al. [Saarinen et al. 2019] desenvolveram a ferramenta *Quizius*, que utiliza aprendizado de máquina para analisar respostas de estudantes e identificar *misconceptions* em larga escala, enquanto Lu et al. [Lu and Krishnamurthi 2024] combinaram *Quizius* com o modelo SMoL (*Standard Model of Languages*), criando o tutor interativo SMoL Tutor, que fornece feedback personalizado.

Outro exemplo de inovação é a reutilização de *misconceptions* previamente documentados para criar novos ICs. Caceffo et al. [Caceffo et al. 2019] demonstraram que muitos equívocos são independentes da linguagem de programação, ao desenvolver um IC para Java com base em estudos de C e Python. Estudos como os de Parker et al.

[Parker et al. 2016] e Braz et al. [Braz et al. 2021] validaram ICs em novos contextos e idiomas, destacando sua versatilidade. Ferrão et al. [Ferrao et al. 2022] aplicaram um IC existente [Caceffo et al. 2018] para medir rapidamente a compreensão conceitual de estudantes em um curso intensivo de linguagem C.

A validação é um aspecto crítico no desenvolvimento de ICs, com práticas comuns incluindo testes estatísticos, como o *alfa de Cronbach* e a Teoria da Resposta ao Item (TRI) [Rachmatullah et al. 2020, Braz et al. 2021], além de avaliações por especialistas e testes pilotos [Grover 2020, Caceffo et al. 2019]. Essas estratégias asseguram a confiabilidade e a precisão dos instrumentos, enquanto abordagens inovadoras contribuem para tornar a pesquisa na área mais acessível e abrangente, especialmente em contextos que exigem agilidade ou maior escala.

#### **4.3. QP3: Quais são os principais desafios e limitações identificados na literatura em relação ao uso de inventários de conceitos em computação?**

1. **Seleção e Validação dos Itens do IC:** A elaboração de questões que representem com precisão os conceitos fundamentais de uma disciplina é um desafio crítico na construção de ICs. Distratores baseados em erros comuns dos estudantes são ferramentas eficazes para capturar concepções errôneas, como destacado por [Xie et al. 2019], embora possam não abranger todo o conhecimento conceitual dos estudantes, demandando análises detalhadas e processos trabalhosos.
2. **Generalização vs. Especificidade:** A tensão entre a generalização e a especificidade é um aspecto importante no desenvolvimento de ICs. Estudos como [Caceffo et al. 2016] apontam para a utilidade de ICs independentes de linguagens de programação, enquanto outros, como [Braz et al. 2021], destacam que questões genéricas podem ignorar dificuldades específicas, como a interpretação de pseudocódigo, limitando a eficácia do IC em capturar concepções errôneas ligadas a linguagens particulares.
3. **Aplicabilidade em Diferentes Contextos Educacionais:** A adaptação de ICs para diferentes níveis e contextos educacionais é um desafio recorrente. Embora sejam amplamente utilizados em cursos universitários, as diferenças culturais, curriculares e de conhecimento prévio entre os estudantes podem limitar a aplicabilidade do IC, conforme discutido por [Julie and Bruno 2020]. Essas variações podem exigir adaptações que, por sua vez, comprometem a validade do instrumento.
4. **Manutenção:** A manutenção dos ICs é crucial para garantir sua relevância frente às mudanças em disciplinas e currículos. Estudos como [Porter et al. 2014] destacam a necessidade de atualização contínua dos ICs, especialmente diante da evolução de linguagens de programação e tecnologias, para evitar que os instrumentos fiquem obsoletos.
5. **Limitações do Pré-Teste:** ICs em outras áreas, como Física, são amplamente utilizados em pré-testes e pós-testes para medir o progresso dos estudantes ao longo de um curso [Hestenes et al. 1992]. Contudo, no contexto de ciência da computação, essa prática enfrenta desafios devido à ausência de conceitos preexistentes entre estudantes. Entretanto, com a crescente inserção de tópicos de computação no currículo das séries iniciais, como conceitos básicos de programação, o uso de ICs como pré-testes em universidades torna-se uma possibilidade concreta e relevante.

#### **4.4. QP4: Há evidências empíricas que demonstrem a melhoria do desempenho acadêmico de estudantes de computação em avaliações e exames como resultado direto da utilização de inventários de conceitos?**

Os estudos analisados nesta RSL demonstram uma lacuna entre a criação e validação desses instrumentos e sua aplicação sistemática em sala de aula. O foco principal dos estudos é o desenvolvimento e refinamento dos ICs, com a coleta de dados dos estudantes servindo primariamente para validar os itens e identificar conceitos errôneos. A aplicação dos ICs como ponto de partida para intervenções pedagógicas mais específicas, como a criação de materiais didáticos personalizados ou a adaptação das estratégias de ensino, é menos explorada.

A RSL, ao priorizar o objetivo da pesquisa, pode ter direcionado os estudos para um modelo em que os estudantes são vistos principalmente como participantes de experimentos, com o objetivo de validar instrumentos como os ICs. Essa abordagem pode ter reduzido o número de estudos que tratam os estudantes como beneficiários diretos de intervenções pedagógicas, com foco em acompanhar a evolução de seu aprendizado ao longo do tempo. É importante ressaltar que a afirmação de que os ICs podem ser utilizados como ferramenta de intervenção pedagógica é amplamente aceita na literatura, e alguns estudos destacam que seus resultados podem auxiliar a interversão no ensino dos estudantes [Grover 2020, Julie and Bruno 2020, Stephens-Martinez et al. 2017, Ferrao et al. 2022].

#### **4.5. QP5: Quais são as perspectivas futuras para o uso de inventários de conceitos em pesquisa e prática em educação em computação?**

1. **Ampliar a coleta de dados:** Apesar das diversas estratégias existentes para identificar e avaliar *misperceptions* dos estudantes em computação, a área ainda apresenta potencial significativo de desenvolvimento. Dados obtidos a partir de questionários, entrevistas e atividades em sala têm mostrado eficácia em identificar tanto *misperceptions* conhecidos quanto novos, conforme evidenciado em estudos como [Stephens-Martinez et al. 2017, Vahrenhold and Paul 2014, Saarinen et al. 2019].
2. **ICs além da identificação de *misperceptions*:** Inventários de Conceitos (ICs) não apenas identificam dificuldades específicas, mas também permitem investigar os processos de aprendizagem. Estudos como [Rachmatullah et al. 2020] enfatizam a definição do pensamento computacional, enquanto [Kennedy and Kraemer 2019] observaram padrões de pensamento e estratégias de resolução de problemas durante a codificação. Além disso, trabalhos como [Henry and Dumas 2020, Julie and Bruno 2019, Julie and Bruno 2020] destacam o uso dos ICs para identificar perfis de estudantes e apoiar estratégias de ensino personalizadas. Por exemplo, [Julie and Bruno 2020] utilizou um IC para ajudar professores a compreender modelos mentais de estudantes iniciantes em programação.
3. **Adaptação e reutilização de ICs:** A aplicação de ICs em diferentes contextos é essencial para garantir sua validade e confiabilidade. Estudos como [Braz et al. 2021, Ferrao et al. 2022, Parker et al. 2016, Parker et al. 2023] enfatizam a importância da validação contínua e adaptação dos ICs para diferentes

- públicos. [Parker et al. 2023], por exemplo, refinou a aplicação diagnóstica ao explorar subescalas conceituais de um IC, demonstrando sua flexibilidade e eficácia.
- 4. **Contexto da educação básica:** Há uma lacuna significativa no uso de ICs na educação básica, com a maioria dos instrumentos concentrados no nível de graduação. Estudos como [Rachmatullah et al. 2020, Grover 2020] destacam a importância de desenvolver ICs para identificar *misperceptions* precocemente, fortalecendo a formação em ciência da computação desde os primeiros anos escolares.
  - 5. **Inovação e novas tecnologias:** Tecnologias inovadoras estão transformando o desenvolvimento e uso de ICs. Por exemplo, [Saarinen et al. 2019] relatam que o uso do sistema Quizius acelerou a geração de instrumentos, enquanto [Wiegand et al. 2021] utilizaram sistemas de tutoria inteligente e algoritmos como o DECA para classificar erros. Além disso, [Julie et al. 2020b] criaram um IC tangível em que os estudantes constroem soluções com blocos físicos e recebem *feedback* por dispositivos móveis, promovendo uma abordagem intuitiva e motivadora.
  - 6. **Fomentar a utilização dos ICs:** Incentivar o uso de ICs na prática pedagógica é fundamental para aprimorar o ensino e a aprendizagem em computação. Estudos como [Dewji et al. 2015] ressaltam a importância do engajamento dos educadores, exemplificando com um formato inovador de “quiz show” para apresentar ICs de forma lúdica e colaborativa.

## 5. Considerações finais

A Revisão Sistemática da Literatura (RSL) destaca a relevância dos Inventários de Conceitos (ICs) como ferramentas cruciais para diagnosticar e corrigir *misperceptions* em disciplinas introdutórias de computação. Os estudos analisados foram classificados em três grupos: (1) focados na criação de abordagens para identificar *misperceptions*, (2) que desenvolvem e validam um IC após a identificação dos *misperceptions*, e (3) que aplicam um IC existente. Essa divisão reflete a diversidade da pesquisa sobre ICs.

A revisão evidenciou uma variedade de metodologias utilizadas na criação e validação de ICs, embora não haja uma metodologia padronizada. As etapas essenciais para o desenvolvimento de ICs incluem a seleção de conceitos por meio de coleta de dados, identificação de *misperceptions*, elaboração das questões e a garantia de validade e confiabilidade. Além disso, a pesquisa evoluiu, incorporando abordagens inovadoras como análise de dados avançada, modelagem cognitiva e uso de tecnologias para otimizar o processo.

Ao analisar perspectivas futuras, observa-se que, apesar da predominância de ICs aplicados em disciplinas de programação, há potencial para expandir o uso desses instrumentos para áreas emergentes, como Ciência de Dados, Robótica e Sistemas Distribuídos. Fortalecendo essas áreas, os ICs podem se tornar ferramentas ainda mais eficazes para diagnosticar e corrigir dificuldades de aprendizagem, contribuindo para a melhoria do ensino e aumento da retenção de estudantes nas áreas de computação.

## Agradecimentos

Os autores agradecem o apoio e o suporte institucional da Universidade Federal de Mato Grosso do Sul (UFMS) para o desenvolvimento deste estudo.

## Referências

- Adams, W. K. and Wieman, C. E. (2011). Development and validation of instruments to measure learning of expert-like thinking. *International journal of science education*, 33(9):1289–1312.
- Ali, M., Ghosh, S., Rao, P., Dhegaskar, R., Jawort, S., Medler, A., Shi, M., and Dasgupta, S. (2023). Taking stock of concept inventories in computing education: A systematic literature review. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*, pages 397–415.
- Almstrum, V. L., Henderson, P. B., Harvey, V., Heeren, C., Marion, W., Riedesel, C., Soh, L.-K., and Tew, A. E. (2006). Concept inventories in computer science for the topic discrete mathematics. In *Working group reports on ITiCSE on Innovation and technology in computer science education*, pages 132–145.
- Barbosa, L. S., Fernandes, T. C., and Campos, A. M. (2011). Takkou: Uma ferramenta proposta ao ensino de algoritmos. In *XVIII Workshop sobre Educação em Computação (WEI 2011)*.
- Brasil (2022). Base nacional comum curricular (bncc) de computação. Anexo ao Parecer CNE/CEB nº 2/2022. Acesso em: 2 nov. 2024.
- Brasil (2023). Lei nº 14.533, de 11 de janeiro de 2023. Institui a Política Nacional de Educação Digital e dá outras providências. Acesso em: 2 nov. 2024.
- Braz, A. C. R., Carvalho, L. S., Oliveira, E. H., Oliveira, D. B., Bittencourt, R. A., Santana, B. L., and Pereira, F. D. (2021). Tradução e validação de um inventário de conceitos sobre programação introdutória. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, pages 1253–1264. SBC.
- Caceffo, R., de França, B., Gama, G., Benatti, R., Aparecida, T., Caldas, T., and Azevedo, R. (2017). An antipattern documentation about misconceptions related to an introductory programming course in c. In *Technical Report 17-15*, page 42. Institute of Computing, University of Campinas.
- Caceffo, R., Frank-Bolton, P., Souza, R., and Azevedo, R. (2019). Identifying and validating java misconceptions toward a cs1 concept inventory. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, pages 23–29.
- Caceffo, R., Gama, G., Benatti, R., Aparecida, T., Caldas, T., and Azevedo, R. (2018). A concept inventory for cs1 introductory programming courses in c. *Institute of Computing, University of Campinas, Tech. Rep. IC-18-06*, page 107.
- Caceffo, R., Wolfman, S., Booth, K. S., and Azevedo, R. (2016). Developing a computer science concept inventory for introductory programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 364–369.
- Derméval, D., Coelho, J. A. d. M., and BITTENCOURT, I. I. (2020). Mapeamento sistemático e revisão sistemática da literatura em informática na educação. *JÁ-QUES, Patrícia Augustin; SIQUEIRA, Sean; BITTENCOURT, Ig; PIMENTEL, Mariano.(Org.) Metodologia de Pesquisa Científica em Informática na Educação: Abordagem Quantitativa. Porto Alegre: SBC.*

- Dewji, N., Wolfman, S. A., Herman, G. L., Porter, L., Taylor, C., and Vahrenhold, J. (2015). The cs concept inventory quiz show. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 669–670.
- Duran, R., Bim, S. A., Gimenes, I., Ribeiro, L., and Correia, R. C. M. (2023). Potential factors for retention and intent to drop-out in brazilian computing programs. *ACM Transactions on Computing Education*, 23(3):1–33.
- Evans, D., Gray, G. L., Krause, S., Martin, J., Midkiff, C., Notaros, B. M., Pavelich, M., Rancour, D., Reed-Rhoads, T., Steif, P., et al. (2003). Progress on concept inventory assessment tools. In *33rd Annual Frontiers in Education, 2003. FIE 2003.*, volume 1, pages T4G–1. IEEE.
- Ferrao, R. C., Montagner, I. D. S., Caceffo, R., and Azevedo, R. (2022). How much c can students learn in one week? experiences teaching c in advanced cs courses. In *2022 IEEE frontiers in education conference (FIE)*, pages 1–8. IEEE.
- Field, A. (2024). *Discovering statistics using IBM SPSS statistics*. Sage publications limited.
- Fukao, A. T., Colanzi, T. E., Martimiano, L. A., and Feltrim, V. D. (2023). Estudo sobre evasão nos cursos de computação da universidade estadual de maringá. In *Anais do III Simpósio Brasileiro de Educação em Computação*, pages 86–96. SBC.
- Goldman, K., Gross, P., Heeren, C., Herman, G. L., Kaczmarczyk, L., Loui, M. C., and Zilles, C. (2010). Setting the scope of concept inventories for introductory computing subjects. *ACM Transactions on Computing Education (TOCE)*, 10(2):1–29.
- Grover, S. (2020). Designing an assessment for introductory programming concepts in middle school computer science. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 678–684.
- Hamouda, S., Edwards, S. H., Elmongui, H. G., Ernst, J. V., and Shaffer, C. A. (2017). A basic recursion concept inventory. *Computer Science Education*, 27(2):121–148.
- Henry, J. and Dumas, B. (2020). Developing an assessment to profile students based on their understanding of the variable programming concept. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 33–39.
- Herman, G. L., Loui, M. C., and Zilles, C. (2010). Creating the digital logic concept inventory. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 102–106.
- Hestenes, D., Wells, M., Swackhamer, G., et al. (1992). Force concept inventory. *The physics teacher*, 30(3):141–158.
- Julie, H. and Bruno, D. (2019). Towards the identification of profiles based on the understanding of programming concepts: the case of the variable. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.
- Julie, H. and Bruno, D. (2020). Approach to develop a concept inventory informing teachers of novice programmers' mental models. In *2020 IEEE frontiers in education conference (FIE)*, pages 1–9. IEEE.

- Julie, H., Bruno, D., Patrick, H., and Tony, L. (2020a). Object-oriented programming: Diagnosis understanding by identifying and describing novice perceptions. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE.
- Julie, H., Tom, M., Antoine, C., Benoit, V., and Bruno, D. (2020b). A tangible-augmented concept inventory to identify novices' misconceptions in programming. In *2020 15th international conference on computer science & education (ICCSE)*, pages 370–374. IEEE.
- Kaczmarczyk, L. C., Petrick, E. R., East, J. P., and Herman, G. L. (2019). The first of the top ten sigcse symposium research papers of the last 50 years: Identifying student misconceptions of programming. *ACM Inroads*, 10(2):65–69.
- Kennedy, C. and Kraemer, E. T. (2019). Qualitative observations of student reasoning: Coding in the wild. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, pages 224–230.
- Lu, K.-C. and Krishnamurthi, S. (2024). Identifying and correcting programming language behavior misconceptions. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1):334–361.
- Luxton-Reilly, A., Becker, B. A., Cao, Y., McDermott, R., Mirolo, C., Mühling, A., Petersen, A., Sanders, K., Simon, and Whalley, J. (2018). Developing assessments to determine mastery of programming fundamentals. In *Proceedings of the 2017 ITiCSE Conference on Working Group Reports*, pages 47–69.
- Parker, M. C., Davidson, M. J., Kao, Y. S., Margulieux, L. E., Tidler, Z. R., and Vahrenhold, J. (2023). Toward cs1 content subscales: A mixed-methods analysis of an introductory computing assessment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, pages 1–13.
- Parker, M. C., Guzdial, M., and Engleman, S. (2016). Replication, validation, and use of a language independent cs1 knowledge assessment. In *Proceedings of the 2016 ACM conference on international computing education research*, pages 93–101.
- Porter, L., Taylor, C., and Webb, K. C. (2014). Leveraging open source principles for flexible concept inventory development. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 243–248.
- Porter, L., Zingaro, D., Liao, S. N., Taylor, C., Webb, K. C., Lee, C., and Clancy, M. (2019). Bdsi: A validated concept inventory for basic data structures. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pages 111–119.
- Rachmatullah, A., Akram, B., Boulden, D., Mott, B., Boyer, K., Lester, J., and Wiebe, E. (2020). Development and validation of the middle grades computer science concept inventory (mg-csci) assessment. *EURASIA Journal of Mathematics, Science and Technology Education*, 16(5):em1841.
- Saarinen, S., Krishnamurthi, S., Fisler, K., and Tunnell Wilson, P. (2019). Harnessing the wisdom of the classes: Classsourcing and machine learning for assessment instrument generation. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 606–612.

- Semesp, I. (2023). 13º mapa do ensino superior no brasil 2023. Acesso em: 2 nov. 2024.
- Sherman, A. T., Herman, G. L., Oliva, L., Peterson, P. A., Golaszewski, E., Poulsen, S., Scheponik, T., and Gorti, A. (2020). Experiences and lessons learned creating and validating concept inventories for cybersecurity. In *National Cyber Summit*, pages 3–34. Springer.
- Stephens-Martinez, K., Ju, A., Parashar, K., Ongowarsito, R., Jain, N., Venkat, S., and Fox, A. (2017). Taking advantage of scale by analyzing frequent constructed-response, code tracing wrong answers. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 56–64.
- Taylor, C., Zingaro, D., Porter, L., Webb, K. C., Lee, C. B., and Clancy, M. (2014). Computer science concept inventories: past and future. *Computer Science Education*, 24(4):253–276.
- Vahrenhold, J. and Paul, W. (2014). Developing and validating test items for first-year computer science courses. *Computer science education*, 24(4):304–333.
- Webb, K. C. and Taylor, C. (2014). Developing a pre-and post-course concept inventory to gauge operating systems learning. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 103–108.
- Wiegand, R. P., Bucci, A., Kumar, A. N., Albert, J., and Gaspar, A. (2021). Identifying informatively easy and informatively hard concepts. *ACM Transactions on Computing Education (TOCE)*, 22(1):1–28.
- Wiegand, R. P., Bucci, A., Kumar, A. N., Albert, J. L., and Gaspar, A. (2016). A data-driven analysis of informatively hard concepts in introductory programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 370–375.
- Xie, B., Davidson, M. J., Li, M., and Ko, A. J. (2019). An item response theory evaluation of a language-independent cs1 knowledge assessment. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 699–705.