



**Exemplos resolvidos no ensino das disciplinas iniciais de programação:
implementação de um domínio de conhecimento baseado em pares
exemplo-problema em um sistema tutor inteligente.**

Leandro Homma Nagano, PPGINF / UFPR, nagano_leandro@yahoo.com.br,
<https://orcid.org/0009-0001-7931-6651>

Andrey Ricardo Pimentel, PPGINF / UFPR, andrey@inf.ufpr.br,
<https://orcid.org/0000-0001-8565-3876>

Alexander Robert Kutzke, SEPT / UFPR, alexander@ufpr.br,
<https://orcid.org/0000-0001-9876-9640>

Resumo: Esse trabalho foi fundamentado pela Teoria da Carga Cognitiva, exemplos resolvidos, pares exemplo-problema, exibição de exemplos e problemas simultaneamente e sistema tutor inteligente. Para a exibição simultânea, considera-se o módulo do domínio, representado por uma relação entre o exemplo resolvido e um problema semelhante a ser resolvido. Os indícios de aprendizagem foram obtidos a partir da avaliação dos exemplos, da quantidade de tentativas realizadas pelo estudante até submeter a resposta correta e da aplicação de um questionário. Os experimentos realizados foram bem aceitos e indicam potencial para gerar impactos positivos durante o aprendizado dos estudantes. Tendo em vista os resultados obtidos, espera-se contribuir para o ensino das disciplinas iniciais de programação.

Palavras-chave: Pares exemplo-problema. Exemplos resolvidos. Ensino de programação. Sistema Tutor Inteligente.

**Worked examples in teaching initial programming subjects: implementation of a
knowledge domain based on example-problem pairs in an intelligent tutoring
system.**

Abstract: This work was based on Cognitive Load Theory, solved examples, example-problem pairs, simultaneous display of examples and problems, and an intelligent tutoring system. For simultaneous display, the domain module is considered, represented by a relationship between the solved example and a similar problem to be solved. Evidence of learning was obtained from the evaluation of the examples, the number of attempts made by the student until submitting the correct answer, and the application of a questionnaire. The experiments were well-received and indicate potential to generate positive impacts during student learning. Considering the results obtained, it is expected to contribute to the teaching of introductory programming courses.

Keywords: Example-problem pairs. Worked examples. Teaching programming. Intelligent Tutor System.

1. Introdução

No Mapa do Ensino Superior do Brasil divulgado em 2023 pelo Instituto Semesp, em 2021 a taxa de evasão dos cursos presenciais da área de Tecnologia da Informação (38,5 pontos percentuais) foi maior que a taxa de evasão dos demais cursos (30,7 pontos percentuais). No ensino a distância, essa diferença é historicamente menor.

A evasão dos estudantes nos cursos de computação, exposta pelo Instituto Semesp, representa uma grande preocupação, pois a carência de profissionais de



tecnologia da informação (TI) é imensa e a evasão tem consequências para toda a sociedade, como a redução do Índice de Desenvolvimento Humano (IDH) do país em virtude da escassez de mão-de-obra qualificada, que pode refletir em perda de produtividade da economia e de competitividade internacional (ALVIM, 2024).

Um dos motivos para essa taxa de evasão é que as disciplinas iniciais de programação são frequentemente indicadas pelos estudantes como as mais difíceis e possuem altas taxas de desistência. Estas disciplinas costumam ter altos índices de abandono e reprovação, sendo um dos gargalos existentes nos cursos de graduação, particularmente de computação, dificultando ou até mesmo impedindo a continuidade dos estudantes no curso (KUTZKE, 2015; ALVIM, 2024).

A programação de computadores é uma atividade que demanda uma carga cognitiva elevada, que se caracteriza pelo processo de desenvolvimento de um algoritmo (sequência de passos) através de uma linguagem de programação. O computador interpreta a linguagem e executa os passos programados. As linguagens de programação possuem sintaxe e semântica próprias (KUTZKE, 2015; ROBINS, 2003). Os programadores iniciantes conhecem a sintaxe e a semântica de instruções individuais, mas não sabem como combinar essas características em programas válidos. Mesmo quando sabem como resolver um problema manualmente, eles têm dificuldade em traduzir a solução manual para um programa de computador equivalente (WINSLOW, 1996).

Nesse contexto, uma forma de reduzir a evasão dos cursos da área de TI e a dificuldade encontrada nas disciplinas iniciais de programação seria reduzir a carga cognitiva empregada para o desenvolvimento de algoritmos. Conforme a Teoria da Carga Cognitiva, uma alternativa para alcançar esse objetivo é a utilização de exemplos resolvidos. Além disso, o uso de exemplos resolvidos pode contribuir para reduzir a dificuldade de traduzir a solução manual para um programa de computador equivalente.

Portanto, o objetivo geral deste trabalho é investigar a utilização de pares exemplo-problema em um sistema tutor inteligente para o ensino das disciplinas iniciais de programação.

2. Fundamentação Teórica

Esta seção apresenta os fundamentos teóricos que foram utilizados para o desenvolvimento deste trabalho.

3.1. Teoria da Carga Cognitiva

Em um artigo de 1998 na *Educational Psychology Review*, Sweller *et al* (1998) descreveram a Teoria da Carga Cognitiva (TCC). Naquela época, discutiram os principais efeitos previstos pela teoria e revisaram estudos empíricos que forneceram suporte para esses efeitos. A teoria descreve a relação entre esforço mental e aprendizagem e foi projetada para orientar e auxiliar na apresentação de informações que incentivem as atividades do estudante na otimização do desempenho intelectual. De acordo com a teoria, deve-se incentivar atividades de aprendizagem que minimizem o processamento e/ou armazenamento que não seja pertinente para a aprendizagem, para evitar sobrecarregar a capacidade limitada da memória de trabalho (van MERRIENBOER, 2005; SWELLER *et al*, 1998; RENKL, 2004).

Na arquitetura cognitiva humana, temos uma memória de trabalho limitada, que lida com todas as atividades conscientes, e uma memória de longo prazo efetivamente ilimitada, que pode ser usada para armazenar esquemas de vários graus de automação (SWELLER *et al*, 1998).



A TCC assume uma memória de trabalho limitada que armazena cerca de sete elementos, mas opera com apenas de dois a quatro elementos. Ela é capaz de lidar com informações apenas por alguns segundos, sendo que a maior parte delas se perde após aproximadamente 20 segundos, a menos que sejam atualizadas por repetição. A teoria enfatiza que essas limitações de capacidade e duração da memória de trabalho só se aplicam a novas informações obtidas por meio da memória sensorial. A memória de trabalho não tem limitações conhecidas ao lidar com informações recuperadas da memória de longo prazo (VAN MERRIENBOER, 2005).

A teoria distingue três tipos de carga cognitiva: carga intrínseca, estranha (ou externa) e pertinente (ou relevante). A carga intrínseca depende da complexidade de uma tarefa, ou seja, do número de elementos de informação interativos que a tarefa contém. Cargas estranhas e pertinentes dependem da forma como a tarefa é projetada, com a carga estranha refletindo o esforço necessário para processar instruções mal projetadas e a carga pertinente refletindo o esforço que contribui para a construção de esquemas e para a aprendizagem (SWELLER *et al*, 1998).

A TCC se baseia em esquemas para reduzir a Carga Cognitiva. Presume-se que os esquemas permitem que muitos elementos sejam tratados como um único elemento na memória de trabalho e, como resultado, mais capacidade de memória de trabalho é liberada (SOUZA, 2010; SWELLER *et al*, 1998).

Segundo a Teoria dos Esquemas (CHI, 1982), o conhecimento é organizado na memória de longo prazo na forma de esquemas, cuja construção depende do processamento prévio das informações na memória de trabalho. Esses esquemas permitem reconhecer tipos de problemas, recuperar procedimentos de solução e explicam as diferenças de desempenho entre iniciantes e especialistas (SWELLER *et al*, 1998; MORENO, 2006).

A automação, obtida por meio da prática, é fundamental para a construção de esquemas, pois reduz o esforço consciente e a carga da memória de trabalho, liberando capacidade para a execução eficiente de tarefas familiares e para a aprendizagem de novas tarefas. A falta de automação compromete a eficiência do desempenho e pode impedir a execução e a aprendizagem de novas tarefas devido à limitação da memória de trabalho (SWELLER *et al*, 1998; VAN MERRIENBOER, 2005).

2.2. Exemplos resolvidos

Um efeito é uma demonstração experimental de que um procedimento instrucional baseado nos princípios da teoria da carga cognitiva facilita a aprendizagem ou a resolução de problemas em comparação com um procedimento mais tradicional. O uso de experimentos replicáveis, randomizados e controlados é essencial para este processo (SWELLER *et al*, 2011).

Sweller e Cooper demonstraram experimentalmente que alternar exemplos resolvidos com problemas a serem resolvidos é mais benéfico para a aprendizagem do que utilizar somente a resolução de problemas. Essa vantagem decorrente da utilização de exemplos resolvidos foi denominada Efeito do Exemplo Resolvido (SWELLER *et al*, 2011).

Os exemplos resolvidos facilitam a aquisição e aplicação de esquemas de resolução de problemas, exigindo menor carga da memória de trabalho do que a resolução direta de problemas, já que todos os elementos essenciais estão integrados no exemplo (SWELLER *et al*, 2011).

Um exemplo resolvido apresenta uma solução passo a passo para um problema ou tarefa. Trata-se de um problema totalmente solucionado que permite aos estudantes observar a sequência de etapas que conduzem à resposta final. Aprender a partir de



exemplos resolvidos (mais tarde também chamado de aprendizagem baseada em exemplos) é adequado especialmente para promover a aquisição inicial de habilidades cognitivas. Portanto, esse método de aprendizado geralmente é empregado quando os estudantes começam a estudar como resolver determinados tipos de problemas (REISSLEIN, 2006; RENKL, 2017). Muldner (2022) ressalta que “os exemplos são especialmente benéficos para os novatos, mas perdem sua eficácia à medida que os estudantes ganham experiência”.

Pares exemplo-problema são uma das formas mais comuns de apresentar exemplos resolvidos, onde cada exemplo é acompanhado de um problema semelhante ao exemplo para os estudantes resolverem. Os estudantes alternam entre estudar um exemplo resolvido e resolver um problema semelhante, alternando entre os dois através de vários ciclos (pares exemplo-problema) (SKUDDER, 2014; HUANG *et al*, 2015).

Aprender com exemplos e problemas simultaneamente libera os estudantes iniciantes da necessidade de considerar diferentes opções de solução. Eles podem então construir um esquema eficaz de resolução de problemas e colocá-lo em prática imediatamente no processo de resolução de problemas. De acordo com o princípio da contiguidade temporal (FIORELLA, 2021), usar exemplos e problemas simultaneamente pode ser mais eficaz do que usar exemplos e problemas separadamente, particularmente quando os estudantes precisam imitar exemplos resolvidos para resolver problemas atuais (HUANG *et al*, 2015).

2.3. Trabalhos relacionados

Nesta seção, destacamos os principais trabalhos encontrados na literatura que utilizaram exemplos e problemas simultaneamente.

Huang *et al* (2015) realizaram um experimento em papel na disciplina de cálculo com estudantes de graduação em matemática, com o objetivo de investigar o efeito de pares exemplo-problema quando eles são simultaneamente apresentados a um participante. O outro procedimento envolveu estudar um exemplo e então removê-lo e fazer com que o participante resolvesse um problema semelhante (exemplo-problema).

Lee *et al* (2015) realizaram um experimento em um sistema instrucional na disciplina de álgebra com estudantes de graduação em matemática, com o objetivo de investigar como a similaridade de exemplos afeta o mapeamento e a estratégia de solução que os estudantes adotam em resolução de problemas matemáticos. Em um experimento, um problema e um exemplo foram apresentados simultaneamente, e no outro experimento foram apresentados sequencialmente.

3. Procedimentos Metodológicos

O experimento foi realizado durante o segundo semestre de 2025, com estudantes da disciplina de Fundamentos de Programação de Computadores, do curso de Matemática da Universidade Federal do Paraná. Essa disciplina compõe a estrutura curricular do segundo semestre dos cursos da área de Ciências Exatas e Engenharias.

Para uma melhor organização e compreensão, esta seção está estruturada em duas subseções principais: 1) Relação com os pares exemplo-problema e 2) Execução das atividades.

3.1. Relação com pares exemplo-problema

A base para implementação do domínio de conhecimento baseado em pares exemplo-problema foi uma ferramenta já existente, denominada Farma-Alg (Sistema para a Promoção da Mediação do Erro no Ensino de Programação de Computadores), desenvolvida na tese de doutorado de Kutzke (2015).



A Farma-Alg é uma aplicação Web que tem o objetivo de auxiliar professores e estudantes no ensino de programação de computadores, visando promover a mediação do erro por meio da visualização e manipulação de registros de respostas e de suas relações.

Os objetos de aprendizagem criados pelos professores com a ferramenta Farma-Alg são exercícios de programação. As respostas são códigos-fonte nas linguagens C, Pascal ou Ruby. As linguagens que estarão disponíveis aos estudantes são definidas pelo professor no momento de criação do exercício. Através do sistema avaliador, a Farma-Alg é capaz, então, de compilar, executar e testar as respostas submetidas pelos estudantes.

O professor fornece padrões de entrada e saída, também chamados de casos de teste. Na tentativa de aferir a resposta do estudante, o sistema a compara contra tais padrões. Vários casos de teste podem ser definidos para um mesmo exercício. A resposta do estudante é testada para todos os casos definidos pelo professor, e, para cada um, tem seu resultado armazenado. Simultaneamente, a cada tentativa enviada por um estudante, a ferramenta armazena a resposta e todos os seus dados pertinentes (número da tentativa, saída de cada caso de teste etc.) e informa se a resposta está correta ou não.

Para utilizar o conceito de exemplo resolvido, foram adicionados dois campos (passo-a-passo e código-fonte) no cadastro de questões. A Figura 1 demonstra o cadastro de questões e os campos adicionados (passo-a-passo e código-fonte) para utilizar o conceito de exemplo resolvido.

Título

Exemplo 1

Tipo de Questão

☒ Normal ☐ Recomendação

Nível de Dificuldade

☒ Fácil ☐ Médio ☐ Difícil ☐ Desafio ☐ Não classificado

Categoria

Desvios simples

Descrição

Elabore um programa que leia o tamanho de uma camisa e quando o tamanho for:

- "P", imprima "pequeno"
- "M", imprima "médio"
- "G", imprima "grande"

Passo a passo:

- 1) Criar uma variável para ler a idade.
- 2) Se o tamanho for igual a "P", imprimir "pequeno".
- 3) Se o tamanho for igual a "M", imprimir "médio".
- 4) Se o tamanho for igual a "G", imprimir "grande".

Código:

```
tamanho = input()

if tamanho == "P":
    print("pequeno")
elif tamanho == "M":
    print("médio")
elif tamanho == "G":
    print("grande")
```

Linguagem permitida para resposta

☐ Pascal ☐ C ☐ C++ ☒ Python

Salvar



Figura 1: Utilização do conceito de exemplo resolvido

Para implementar o domínio de conhecimento baseado em pares exemplo-problema, ao criar ou editar um exercício, um exemplo resolvido deverá ser relacionado com o problema a ser resolvido.

Com a implementação do domínio de conhecimento baseado em pares exemplo-problema, ao visualizar um problema para ser resolvido, o estudante poderá visualizar um exemplo resolvido simultaneamente, com o passo-a-passo e o código fonte correspondente à resolução do exemplo resolvido. Após a resolução do problema, o estudante deverá avaliar se o exemplo apresentado foi útil ou não. A Figura 2 apresenta o exemplo resolvido e o problema a ser resolvido simultaneamente.

Responder questão 'Problema 1'

Exemplo para resolver a questao:

Elabore um programa que leia o tamanho de uma camisa e quando o tamanho for:

- "P", imprima "pequeno"
- "M", imprima "médio"
- "G", imprima "grande"

Agora e sua vez:

Elabore um programa que leia o tamanho de uma camisa e quando o tamanho for:

- 38, imprima "pequeno"
- 40, imprima "médio"
- 42, imprima "grande"

Passo a passo para a resolução:

- 1) Criar uma variável para ler a idade.
- 2) Se o tamanho for igual a "P", imprimir "pequeno".
- 3) Se o tamanho for igual a "M", imprimir "médio".
- 4) Se o tamanho for igual a "G", imprimir "grande".

Código fonte:

```
tamanho = input()

if tamanho == "P":
    print("pequeno")
elif tamanho == "M":
    print("médio")
elif tamanho == "G":
    print("grande")
```

Sua Resposta

1

Linguagem permitida: **Python**

O exemplo foi útil para voce? **Sim** ☐ **Não** ☐

Submeter resposta

Figura 2: Exemplo resolvido e o problema a ser resolvido exibidos simultaneamente

3.2. Execução das atividades

Após a segunda aula do conteúdo de estruturas condicionais, os estudantes receberam uma lista de exercícios para serem resolvidos no dia da realização desse experimento. A lista era composta por 3 exercícios de programação de computadores para a linguagem Python, sobre o conteúdo estudado de estruturas condicionais.

Os estudantes foram agrupados de forma aleatória dando origem a dois grupos mistos. Para o presente experimento, o primeiro grupo foi nomeado como Grupo de controle (GC), para o qual foi disponibilizado a Farma-Alg sem a implementação dos pares exemplo-problema. O segundo grupo foi nomeado como Grupo experimental (GE), para o qual foi disponibilizado a Farma-Alg com a implementação dos pares exemplo-problema. O GC iniciou o experimento com 11 estudantes e o GE iniciou o experimento com 9 estudantes. Ambos os grupos receberam o mesmo conteúdo e a mesma lista de exercícios para ser resolvida.



Uma aula antes da realização do experimento, foi realizada uma conversa com estudantes, para apresentar o projeto, seu objetivo e suas justificativas. Nesse dia, foram distribuídos os Registros de Consentimento Livre e Informado, para que os participantes pudessem ler e entregar assinado até o experimento. Como dois participantes são menores de idade, eles assinaram o Registro de Assentimento Livre e Informado e seus pais assinaram o Registro de Consentimento Livre e Informado.

O experimento teve a duração aproximada de 2 horas. Nos primeiros 15 minutos, foi realizada uma ambientação com os computadores do laboratório. Nos 15 minutos posteriores, foi resolvido um problema prático com todos os estudantes, para que pudessem se ambientar com a ferramenta Farma-Alg. Esse problema não foi contabilizado nas estatísticas. O tempo restante foi dedicado para os participantes resolverem a lista de exercícios. A Tabela 1 apresenta a lista de exercícios, com os exemplos e os problemas a serem resolvidos.

Tabela 1: Lista de exercícios, com os exemplos e os problemas a serem resolvidos.

Exemplo 1: Elabore um programa que leia o tamanho de uma camisa e quando o tamanho for: → "P", imprima "pequeno". → "M", imprima "médio". → "G", imprima "grande".	Problema 1: Elabore um programa que leia o tamanho de uma camisa e quando o tamanho for: → 38, imprima "pequeno". → 40, imprima "médio". → 42, imprima "grande".
Exemplo 2: Elabore um programa que leia a idade e imprima: → "Maior", se a idade for maior ou igual que 18. → "Menor", se a idade for menor que 18.	Problema 2: No Brasil, o voto e o alistamento eleitoral são obrigatórios para maiores de 17 anos e são facultativos para analfabetos, maiores de 70 anos e pessoas com idade entre 16 e 17 anos. Elabore um programa que leia a idade e imprima: → "obrigatório", são obrigatórios para maiores de 17 anos e menores de 71 anos. → "facultativo": maiores de 70 anos e pessoas com idade entre 16 e 17 anos.
Exemplo 3: Em relação aos lados, um triângulo pode ser classificado como equilátero, isósceles ou escaleno. Elabore um programa que leia os três lados de um triângulo e imprima: → "equilátero": Todos os três lados possuem a mesma medida. → "isósceles": Possui dois lados com a mesma medida. → "escaleno": Todos os três lados possuem medidas diferentes.	Problema 3: Os triângulos podem ser classificados em três tipos com base nos seus ângulos internos: acutângulo, retângulo e obtusângulo. Elabore um programa que leia os três ângulos de um triângulo e imprima: → "acutângulo": Todos os ângulos internos são menores que 90 graus (ângulos agudos). → "retângulo": Um dos ângulos internos é igual a 90 graus (ângulo reto). → "obtusângulo": Um dos ângulos internos é maior que 90 graus (ângulo obtuso).

4. Resultados e Discussões

Foram utilizados três instrumentos para avaliar o resultado do experimento: avaliação dos exemplos, quantidade de tentativas e aplicação de um questionário.

4.1 Avaliação dos exemplos

Com o intuito de verificar se o exemplo fornecido foi útil para a resolução do problema, os estudantes que participaram do GE puderam avaliar se o exemplo fornecido foi útil para a resolução do problema. Dos 9 estudantes que fizeram parte do GE, 9 avaliaram se os exemplos exibidos foram úteis para a resolução dos problemas. A Tabela 2 apresenta a avaliação realizada pelos estudantes para os problemas 1, 2 e 3 da lista de exercícios:

Tabela 2: Avaliação dos exemplos



Estudante	Problema 1	Problema 2	Problema 3
Estudante GE 1	Não	Não	Não
Estudante GE 2	Não	Não	Não
Estudante GE 3	Sim	Sim	Sim
Estudante GE 4	Sim	Sim	Sim
Estudante GE 5	Sim	Sim	Sim
Estudante GE 6	Sim	Sim	Sim
Estudante GE 7	Não	Sim	Sim
Estudante GE 8	Sim	Sim	Sim
Estudante GE 9	Sim	Sim	Sim
Média	66,67	77,78	77,78

Em suma, 66,67% dos estudantes avaliaram que o exemplo cadastrado para o problema 1 foi útil para a resolução do problema 1, 77,78% dos estudantes avaliaram o exemplo cadastrado para o problema 2 foi útil para a resolução do problema 2 e 77,78% dos estudantes avaliaram o exemplo cadastrado para o problema 3 foi útil para a resolução do problema 3. Em média, 74,07% dos estudantes avaliaram que os exemplos exibidos foram úteis para a resolução dos problemas.

4.2 Quantidade de tentativas

Com o intuito de verificar se a utilização de pares exemplo-problema auxiliou o estudante, foi analisada a quantidade de tentativas de cada estudante até a resolução correta (ou não) do problema.

Dos 11 estudantes que fizeram parte do GC, 10 conseguiram resolver corretamente o problema 1, 9 conseguiram resolver corretamente o problema 2 e 8 conseguiram resolver corretamente o problema 3. A quantidade média de tentativas foi de 2,70 tentativas para o problema 1, 3,22 tentativas para o problema 2 e 3,88 tentativas para o problema 3. Um estudante não conseguiu resolver nenhum problema e um estudante conseguiu resolver somente um problema. A Tabela 3 apresenta a quantidade de tentativas de cada estudante do GC até a resolução correta de cada problema.

Tabela 3: Quantidade de Tentativas - GC

Estudante	Problema 1	Problema 2	Problema 3
Estudante GC 1	2	3	3
Estudante GC 2	8	-	-
Estudante GC 3	3	3	-
Estudante GC 4	2	3	5
Estudante GC 5	1	1	1
Estudante GC 6	2	2	9
Estudante GC 7	1	3	5
Estudante GC 8	2	4	3
Estudante GC 9	5	7	2
Estudante GC 10	-	-	-
Estudante GC 11	1	3	3
Média	2,70	3,22	3,88

Dos 9 estudantes que fizeram parte do GE, 9 conseguiram resolver corretamente o exercício 1, 8 conseguiram resolver corretamente o exercício 2 e 8 conseguiram resolver corretamente o exercício 3. A quantidade média de tentativas foi de 3,67 tentativas para o exercício 1, 9,13 tentativas para o exercício 2 e 2,50 tentativas para o exercício 3. A Tabela 4 apresenta a quantidade de tentativas de cada estudante do GE até a resolução correta de cada problema:



Tabela 4: Quantidade de tentativas - GE

Estudante	Exercício 1	Exercício 2	Exercício 3
Estudante GE 1	10	30	8
Estudante GE 2	1	1	1
Estudante GE 3	1	21	1
Estudante GE 4	10	2	4
Estudante GE 5	1	2	3
Estudante GE 6	3	-	-
Estudante GE 7	2	10	1
Estudante GE 8	1	1	1
Estudante GE 9	4	6	1
Média	3,67	9,13	2,50

A Tabela 5 apresenta a quantidade de tentativas de cada estudante do GE até a resolução correta de cada problema e que avaliou que o exemplo fornecido foi útil para a resolução dos problemas.

Tabela 5: Quantidade de Tentativas - GE (que avaliou que o exemplo fornecido foi útil para a resolução dos problemas)

Estudante	Problema 1	Problema 2	Problema 3
Estudante GE 1	-	-	-
Estudante GE 2	-	-	-
Estudante GE 3	1	21	1
Estudante GE 4	10	2	4
Estudante GE 5	1	2	3
Estudante GE 6	3	-	-
Estudante GE 7	-	10	1
Estudante GE 8	1	1	1
Estudante GE 9	4	6	1
Média	3,33	7,00	1,83

A Tabela 6 apresenta um comparativo entre quantidade média de tentativas dos estudantes do GC e do GE que avaliou que o exemplo fornecido foi útil até a resolução correta de cada problema.

Tabela 6: Quantidade média de tentativas - GC x GE (que avaliou que o exemplo fornecido foi útil para a resolução dos problemas)

Problema	GC	GE	Percentual
Problema 1	2,7	3,33	+23,33
Problema 2	3,22	7,00	+117,39
Problema 3	3,88	1,83	-52,83
Média	3,26	4,05	+24,23

Em suma, pudemos verificar que: a quantidade média de tentativas dos estudantes do GE que avaliou que o exemplo fornecido foi útil para a resolução dos problema até a resolução correta de cada problema foi 24,23% superior que a quantidade média de tentativas dos estudantes do GC até a resolução correta de cada problema. Na questão 2 encontramos o pior desempenho do GE: quantidade média de tentativas dos estudantes do GE que avaliou que o exemplo fornecido foi útil para a resolução dos problema até a resolução correta de cada problema foi 117,39% superior



que a quantidade média de tentativas dos estudantes do GC até a resolução correta de cada problema. Na questão 3 encontramos o melhor desempenho do GE: a quantidade média de tentativas dos estudantes do GE que avaliou que o exemplo fornecido foi útil para a resolução dos problema até a resolução correta de cada problema foi 52,83% inferior que a quantidade média de tentativas dos estudantes do GC até a resolução correta de cada problema.

Quanto à lista de exercícios, o problema 1 forneceu um exemplo (utilizando condicional simples) com enunciados e resoluções mais próximos do problema proposto (também utilizando condicional simples). O problema 2 apresentou um nível de complexidade maior (utilizando condicionais compostas), enquanto forneceu um exemplo mais simples (utilizando condicionais simples). O problema 3 forneceu um exemplo (utilizando condicional simples e composta) com enunciados e resoluções mais próximos do problema proposto (também utilizando condicional simples e composta).

Uma possível justificativa para a diferença na quantidade de tentativas até a resolução correta do problema 2 é a diferença de complexidade apresentada entre o exemplo e o problema. Tanto que podemos constatar que a quantidade de tentativas até a resolução correta do problema 3 foi menor para o GE.

4.3 Questionário

Dos 9 estudantes que fizeram parte do GE, 7 responderam o questionário. Para medir o grau de concordância dos participantes foi utilizada a escala de Likert em um nível de 1 até 5 para representar a concordância. As seguintes afirmações foram utilizadas para analisar o perfil e avaliar a utilização dos pares exemplo-problema:

- 1) Posso conhecimento prévio de programação (anterior à disciplina).
- 2) Encontro dificuldade nas disciplinas iniciais de programação.
- 3) Prestei atenção nos exemplos exibidos na Farma-Alg e me senti motivado pela possibilidade de resolver um problema.
- 4) A apresentação de exemplos auxiliou meu aprendizado.
- 5) Dentre os elementos e estratégias da utilização de exemplos, o passo a passo apresentado no exemplo me ajudou a resolver o problema.
- 6) Dentre os elementos e estratégias da utilização de exemplos, o código fonte apresentado no exemplo me ajudou a resolver o problema

A Tabela 6 mostra um resumo da avaliação da utilização dos pares exemplo-problema, feita pelos integrantes do grupo experimental de acordo com o conhecimento prévio de programação e de acordo com a dificuldade nas disciplinas iniciais de programação.

Tabela 6 - Resumo da avaliação da utilização dos pares exemplo-problema, de acordo com o conhecimento prévio de programação e de acordo com a dificuldade nas disciplinas iniciais de programação.

1	2	3	4	5	6
Concordo totalmente	Discordo totalmente	Indiferente	Indiferente	Indiferente	Indiferente
Concordo totalmente	Discordo totalmente	Discordo totalmente	Discordo totalmente	Discordo totalmente	Discordo totalmente
Concordo totalmente	Discordo totalmente	Concordo totalmente	Concordo totalmente	Indiferente	Concordo parcialmente
Concordo totalmente	Discordo parcialmente	Discordo parcialmente	Discordo parcialmente	Discordo parcialmente	Discordo parcialmente
Indiferente	Discordo parcialmente	Concordo parcialmente	Concordo parcialmente	Concordo parcialmente	Indiferente
Discordo totalmente	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
Discordo totalmente	Concordo parcialmente	Concordo parcialmente	Concordo parcialmente	Concordo parcialmente	Concordo parcialmente

Nas respostas ao questionário, como indicadas na Figura 6, pode-se ver uma tendência à concordância nas perguntas 3, 4, 5 e 6. Isto indica uma sensação, por parte



dos estudantes, de que a aplicação do par exemplo-problema ajudou na resolução dos problemas.

Em suma, os resultados obtidos através da aplicação desse questionários aos estudantes nos permitiu notar uma tendência de que a utilização de exemplos auxiliou no aprendizado dos estudantes, principalmente os que declararam não possuir conhecimento prévio em programação e os que declararam encontrar maior dificuldade nas disciplinas iniciais de programação, através da concordância plena nas perguntas 3, 4, 5 e 6, conforme ilustrado pelas duas últimas linhas da Tabela 6.

5. Considerações Finais

Conhecendo os referenciais teóricos: ensino da programação, Sistemas Tutores Inteligentes, Teoria da Carga Cognitiva, exemplos resolvidos, pares exemplo-problema e exibição de exemplos e problemas simultaneamente, surgiu a proposta de juntar esses temas para criar um domínio de conhecimento baseado em pares exemplo-problema e incorporá-lo em um Sistema Tutor Inteligente, a Farma-Alg. O domínio de conhecimento atinge o objetivo de exibir o exemplo resolvido e o problema a ser resolvido simultaneamente.

Experimentos realizados utilizando pares exemplo-problema e exibição de exemplos e problemas simultaneamente foram feitos com estudantes da disciplina de Fundamentos de Programação de Computadores, do curso de Matemática da Universidade Federal do Paraná. Os estudantes avaliaram se o exemplo apresentado foi útil para a resolução do problema e responderam um questionário sobre a utilização dos exemplos na Farma-Alg. Também foi analisada a quantidade de tentativas de cada estudante até conseguir a resolução correta do problema, informação que é gravada na Farma-Alg. Os resultados demonstraram que a estratégia é promissora, sendo que os exemplos foram avaliados positivamente na Farma-Alg.

Em média, 74,07 por cento dos estudantes do grupo experimental avaliaram que os exemplos exibidos foram úteis para a resolução dos problemas. E na resolução do problema 3 da lista de exercícios encontramos o melhor desempenho do grupo experimental: a quantidade média de tentativas dos estudantes do grupo experimental até a resolução correta do problema e que avaliaram que o exemplo fornecido foi útil para a resolução dos problema foi 52,83 por cento inferior que a quantidade média de tentativas dos estudantes do grupo de controle até a resolução correta do problema.

As respostas ao questionário indicam uma sensação, por parte dos estudantes, de que a aplicação do par exemplo-problema ajudou na resolução dos problemas, principalmente os que declararam não possuir conhecimento prévio em programação e os que declararam encontrar maior dificuldade nas disciplinas iniciais de programação.

Tendo em vista os resultados obtidos, espera-se contribuir para o ensino das disciplinas iniciais de programação, utilizando um domínio de conhecimento baseado em pares exemplo-problema.

Referências

- ALVIM, I. V.; BITTENCOURT, R. A.; DURAN, R. S. Evasão nos cursos de graduação em computação no Brasil. In: Simpósio Brasileiro de Educação em Computação, 38, 2024.
- CHI, M. T. H., GLASER, R., & REES, E. Expertise in problem solving. In: R. J. Sternberg (Ed.), *Advances in the psychology of human intelligence* (vol. 1, pp. 7-76), 1982.



de SOUZA, N. P. C. Teoria da carga cognitiva: origem, desenvolvimento e diretrizes aplicáveis ao processo ensino-aprendizagem. 2010. Dissertação (Mestrado em Educação em Ciências e Matemáticas) - Universidade Federal do Pará, Belém, 2010.

FIORELLA, L.; MAYER, R. (2021). Principles for reducing extraneous processing in multimedia learning: coherence, signaling, redundancy, spatial contiguity, and temporal contiguity principles. Cambridge Handbooks in Psychology. Cambridge University Press; 2021.

HUANG, Y.; LIN, K.; YU, X.; HUNG, J. C. (2015). Comparison of different approaches on example-based learning for novice and proficient learners. Hum. Cent. Comput. Inf. Sci. 5, 29, 2015.

INSTITUTO SEMESP. Mapa do Ensino Superior no Brasil. 13ª edição, 2023.

KUTZKE, A. Informática educacional e a mediação do erro na educação: um estudo teórico-crítico e uma proposta de instrumento computacional. Tese (Doutorado em Informática) - Universidade Federal do Paraná, Curitiba, 2015.

LEE, H. S., BETTS, S. e ANDERSON, J. R. (2015). Not taking the easy road: When similarity hurts learning. Mem Cogn, (43):939–952.

MORENO, R. (2006). When worked examples don't work: Is cognitive load theory at an impasse? Learning and Instruction, 16(2), 2006.

MULDNER, K.; JENNINGS, J.; CHIARELLI, V. A review of worked examples in programming activities. ACM Trans. Comput. Educ. 23(1), 2022.

REISSLEIN, J.; ATKINSON, R. K.; SEELING, P.; REISSLEIN, M. Encountering the expertise reversal effect with a computer-based environment on electrical circuit analysis. Learning and Instruction, v.16, 2006.

RENKL, A.; ATKINSON, R. K.; GROBE, C. S. How fading worked solution steps works - a cognitive load perspective. Instructional Science, 32, 2004.

RENKL, A. Learning from worked-examples in mathematics: students relate procedures to principles. ZDM Mathematics Education, 49, 2014.

ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: A review and discussion. Computer science education, 13(2), 2003.

SWELLER, J.; AYRES, P.; KALYUGA, S. Cognitive Load Theory: Explorations in the Learning Sciences, Instructional Systems and Performance Technologies. LLC, 2011.

SWELLER, J.; van MERRIENBOER, J. J. G.; PAAS, F. G. W. C. Cognitive architecture and instructional design. Educational Psychology Review, 10(3), 1998.

SKUDDER, B.; LUXTON-REILLY, A. Worked examples in computer science. In: Proceedings of the Sixteenth Australasian Computing Education Conference (ACE2014), v.148. Auckland, 2014.

van MERRIENBOER, J. J. G.; SWELLER, J. Cognitive load theory and complex learning: Recent developments and future directions. Educational Psychology Review, 17(2), 2005.

WINSLOW, L. E. Programming pedagogy - a psychological overview. SIGCSE Bulletin, 28(3), 1996.