

Estudo de Caso Sobre Avaliação do Pensamento Computacional em Graduandos de Cursos de Computação

Álekiss Manço de Mélo¹

Departamento de Ciências Exatas (DCX) – Universidade Federal da Paraíba (UFPB)
Caixa Postal – 58.297-000 – Rio Tinto – PB – Brazil

alekiss.melo@dcx.ufpb.br

Abstract. *This article presents a survey carried out with freshmen and veteran students in the Licentiate Degree in Computer Science (LCC) and Bachelor of Information Systems (BSI) courses at the Federal University of Paraíba - Campus IV, with the aim of evaluating their Computational Thinking skills differ from freshman students to veterans. Data were collected through an instrument called Bebras, which was translated into Portuguese and made available to the sample participants remotely, as a result of the pandemic context (COVID-19) in which they were inserted. As a result, we find that there is no significant difference in stating that veterans' PC skills are superior to freshmen's.*

Resumo. *Este artigo apresenta uma pesquisa realizada com estudantes ingressantes e veteranos nos cursos de Licenciatura em Ciência da Computação (LCC) e Bacharelado em Sistemas de Informação (BSI) da Universidade Federal da Paraíba - Campus IV, com o objetivo de avaliar se as habilidades do Pensamento Computacional diferem dos alunos ingressantes para os veteranos. Os dados foram coletados através de um instrumento chamado Bebras, o qual foi traduzido para o português e disponibilizado aos participantes da amostra de forma remota, por consequência do contexto pandêmico (COVID-19) em que estávamos inseridos. Como resultado, encontramos que não há uma diferença significativa para afirmar que as habilidades de PC dos veteranos são superiores a dos ingressantes.*

1. Introdução

O Pensamento Computacional (PC), na visão de Wing (2006), é o processo de reconhecer aspectos da computação no mundo que nos rodeia e aplicar ferramentas e técnicas para entender e raciocinar sobre sistemas e processos naturais, sociais e artificiais. Wing (2006) também afirma que o PC deve ser uma habilidade básica a ser ensinada às crianças, assim como ler, escrever e aritmética.

Existem diferentes maneiras de estimular o PC. A programação baseada em blocos, conforme [Kelleher and Pausch 2005], procura abstrair a sintaxe de forma que os estudantes iniciantes mantenham seu foco apenas na semântica. Por isso, esse paradigma é um meio eficiente de atrair estudantes para o aprendizado do PC [Weintrop et al. 2016]. O ensino de robótica também pode ser um estimulante, pois faz parte das metodologias STEM/STEAM, que propõem a solução de problemas que fazem parte do dia a dia por meio da tecnologia.

Tão importante quanto estimular o PC é ter recursos para avaliar as habilidades desenvolvidas. Um dos problemas em aberto na área é que existem poucos instrumentos em português para avaliar habilidades do Pensamento Computacional.

O objetivo deste trabalho é avaliar se as habilidades de PC diferem dos alunos ingressantes

¹ Trabalho de conclusão de curso, sob orientação da professora Ana Liz Souto Oliveira, submetido ao Curso de Licenciatura em Ciência da Computação do Centro de Ciências Aplicadas e Educação (CCA) da Universidade Federal da Paraíba, como parte dos requisitos necessários para obtenção do grau de LICENCIADO EM CIÊNCIA DA COMPUTAÇÃO.

nos cursos de computação, para os alunos veteranos. Para atingir esse objetivo, um instrumento para avaliação de PC foi traduzido, adaptado para o português e disponibilizado aos alunos, para a coleta de dados.

O restante do trabalho está organizado da seguinte forma: Na Seção 2 é apresentada a fundamentação teórica utilizada, onde são discutidas as habilidades do Pensamento Computacional; Na Seção 3 são citados os trabalhos relacionados; Na Seção 4 são descritos os materiais e métodos, sendo eles: a amostra, o instrumento e a coleta de dados; Na Seção 5 são apresentados e discutidos os resultados e; Na Seção 6 são apresentadas as considerações finais.

2. Fundamentação Teórica

O PC é um processo cognitivo envolvendo raciocínio lógico, no qual os problemas são resolvidos e melhor compreendidos. Csizmadia et al [2015] dividiram esse processo em 5 habilidades: A capacidade de pensar algorítmicamente; A capacidade de pensar em termos de decomposição; A capacidade de pensar em generalizações, identificando e fazendo uso de padrões; A capacidade de pensar abstrações, escolhendo boas representações e; A capacidade de pensar em termos de avaliação.

O **raciocínio lógico** permite que os alunos entendam as coisas, analisando e verificando os fatos por meio do pensamento, de forma clara e precisa. Ele permite que os alunos utilizem seus próprios conhecimentos e modelos internos para fazer e verificar as previsões e tirar conclusões. É amplamente utilizado pelos alunos quando testam e depuram os algoritmos corretamente. O raciocínio lógico é a nova aplicação dos outros conceitos de Pensamento Computacional para resolver problemas.

O raciocínio lógico é a chave para permitir aos alunos refinarem seu código. Eles podem trabalhar com colegas para avaliar o código um do outro, isolar bugs e sugerir correções. Durante este processo, eles podem ter oportunidades para empregar abstração, avaliação e design algorítmico. O novo uso na correção de erros de código requer raciocínio lógico.

O **pensamento algorítmico** é uma forma de chegar a uma solução por meio de uma definição clara das etapas. Alguns problemas são únicos; eles são resolvidos, as soluções são aplicadas e então chega-se a uma resolução. O pensamento algorítmico precisa entrar em ação quando problemas semelhantes têm que ser resolvidos continuamente. Eles não precisam ser pensados de novo a cada tempo. É necessária uma solução que funcione sempre. Algoritmos de aprendizagem para fazer multiplicação ou divisão na escola são um exemplo. Se regras simples forem seguidas com precisão, por um computador ou uma pessoa, a solução para qualquer multiplicação pode ser encontrada. Uma vez que o algoritmo é compreendido, ele não precisa ser trabalhado assim que surja cada novo problema.

O pensamento algorítmico é a capacidade de pensar em termos de sequências e regras como forma de resolver problemas ou compreender situações. É uma habilidade fundamental que os alunos desenvolvem quando aprendem a criar programas em seu próprio computador.

A **decomposição** é uma maneira de pensar sobre os artefatos em termos de suas partes componentes. As peças podem em seguida ser compreendidas, resolvidas, desenvolvidas e avaliadas separadamente. Isso torna os problemas complexos mais fáceis de resolver, situações novas melhor compreendidas e sistemas grandes mais fáceis de projetar.

Por exemplo, fazer o café da manhã pode ser dividido ou decomposto em atividades separadas, como fazer torradas, fazer chá, ferver ovo, et cetera. Cada um deles, por sua vez, também pode ser dividido em um conjunto de etapas. Através da decomposição da tarefa original, cada parte pode ser desenvolvida e integrada posteriormente no processo. Considere desenvolver um jogo: diferentes pessoas podem projetar e criar os diferentes níveis de forma independente, desde que os

principais aspectos sejam previamente acordados. Um nível de arcade simples também pode ser decomposto em várias partes, como o movimento real de um personagem, rolando o plano de fundo e definindo as regras sobre como os personagens interagem.

A **generalização** está associada à identificação de padrões, semelhanças e conexões, e à exploração desses recursos. É uma maneira de resolver novos problemas rapidamente com base em soluções de problemas anteriores e construir a experiência. Fazer perguntas como “Isso é semelhante a um problema que já resolvi?” e “como é diferente?” são importantes aqui, assim como o processo de reconhecimento de padrões tanto nos dados, quanto nos processos/estratégias que estão sendo usados. Algoritmos que resolvem alguns problemas específicos podem ser adaptados para resolver toda uma classe de problemas semelhantes. Então, sempre que um problema dessa classe é encontrado, a solução pode ser aplicada.

A **abstração** torna os problemas ou sistemas mais fáceis de pensar. Abstração é o processo de fazer um artefato mais compreensível através da redução de detalhes desnecessários. Um exemplo clássico é o do mapa subterrâneo de Londres. Londres é um sistema altamente complexo. A representação de Londres de maneiras particulares (geralmente mapas ou fotos) auxilia diferentes usuários. O mapa do metrô de Londres é uma abstração altamente refinada com apenas informações suficientes para o viajante navegar na rede subterrânea sem a carga desnecessária de informações, como distância e posição geográfica exata. É uma representação que contém precisamente as informações necessárias para planejar uma rota de uma estação para outra - e não mais!

A habilidade em abstração está em escolher o detalhe certo para esconder, de forma que o problema se torne mais fácil, sem perder tudo o que é importante. Uma parte fundamental disso é escolher uma boa representação de um sistema. Diferentes representações tornam diferentes coisas fáceis de fazer.

Por exemplo, um programa de computador que joga xadrez é uma abstração. É um conjunto finito e preciso de regras realizadas cada vez que for a vez do computador. Está muito distante do análogo, emocional, tendencioso e processos mentais distraídos empreendidos por um jogador de xadrez humano. É uma abstração porque os detalhes desnecessários desses processos são removidos.

A **avaliação** é o modo de garantir o recurso mais adequado para o propósito de uma solução, seja um algoritmo, sistema ou processo. Várias propriedades de soluções precisam ser avaliadas. Elas estão corretas? São rápidas o suficiente? Usam recursos economicamente? São fáceis de usar? Promovem uma experiência apropriada? Compensações precisam ser feitas, já que raramente há uma única solução ideal para todas as situações. Há um foco específico e muitas vezes extremo na atenção aos detalhes na avaliação com base em Pensamento Computacional.

As interfaces de computador estão sendo desenvolvidas continuamente para atender às necessidades de diferentes usuários. Por exemplo, se um dispositivo médico for necessário para fornecer medicamentos automaticamente a um paciente, ele deve ser programável de forma livre de erros, rápida, simples e segura. A solução deve garantir que os enfermeiros sejam capazes de definir a dose facilmente sem cometer erros e que não seja frustrante para pacientes e enfermeiras usarem. No projeto proposto, haveria uma compensação a ser feita entre a velocidade de entrada dos números (eficiência) e prevenção de erros (eficácia e usabilidade). O projeto seria julgado na especificação proposta por médicos, reguladores e especialistas em design de dispositivos médicos (critérios) e as regras gerais relativas ao bom design (heurísticas). Critérios, heurísticas e necessidades do usuário permitem que os julgamentos sejam feitos de forma sistemática e rigorosamente.

3. Trabalhos Relacionados

Na linha do que se pretendeu desenvolver com o projeto aqui apresentado, foi utilizado como base o

trabalho de Mooney e Lockwood (2020), no qual eles acompanharam o primeiro ano de graduação de um grupo de estudantes. Usaram o teste Bebras para comparar as habilidades do Pensamento Computacional no início e no final desse primeiro ano, chegando à conclusão de que os alunos que estudaram matemática e programação tiveram um desempenho melhor que os outros.

Em Araujo et al, (2018) foi apresentado um processo para guiar a identificação de fatores em questões avaliativas de PC com apoio estatístico. Eles utilizaram a análise fatorial para nortear a exploração e confirmação de fatores do PC em um conjunto de questões. O instrumento utilizado por eles foram as respostas de alunos que fizeram o teste Bebras.

Em Araujo et al, (2015) foi apresentado a realização de um survey com profissionais de computação, para capturar a compreensão que eles têm do PC, chegando ao resultado que 64% dos profissionais pesquisados desconhecem o PC e sugerem que o termo e habilidades relevantes são pouco conhecidos e compreendidos, tanto na indústria, quanto na academia.

4. Materiais e Métodos

4.1 Participantes

A amostra foi composta por alunos ingressantes das turmas de Introdução à Programação e alunos veteranos das turmas de Linguagem de Programação e Programação Orientada à Objetos, nos cursos de licenciatura em Ciências da Computação (LCC) e bacharelado em Sistemas de Informação (BSI), do período 2021.1 (de acordo com o calendário acadêmico), da Universidade Federal da Paraíba - Campus IV.

4.2 Instrumento

Foi utilizado como referência o artigo de Mooney e Lockwood (2020), com a realização do processo de tradução do artigo, incluindo o questionário usado para a coleta de dados. O instrumento foi proposto em formato de formulário (disponível no link: <https://forms.gle/1e3iPppEeWm5zJkr5>), para facilitar a coleta de dados. Devido à pandemia de COVID-19, todo o contato com os participantes aconteceu de forma remota, usando as plataformas Google Meet e Discord.

As questões fazem parte de um instrumento chamado Bebras, criado por uma comunidade internacional de professores e pesquisadores originada na Lituânia, responsável por promover PC nas escolas. Ele tem como enredo uma família de castores que simulam várias situações do seu cotidiano, dando uma dinâmica mais lúdica ao questionário.

O questionário deste trabalho foi composto por 13 questões. Ele contém 11 questões objetivas e 2 questões abertas de resposta curta. O questionário também possui 12 questões, onde é utilizada como metodologia a escala likert, com questões apresentando uma afirmação auto descritiva, seguidas de opções de respostas, que são: concordo totalmente, concordo parcialmente, não concordo e nem discordo, discordo parcialmente e discordo plenamente.

4.3 Coleta de Dados

Após o processo de tradução, foi realizado um teste piloto com uma amostra menor de pessoas externas à pesquisa, com a finalidade de verificar se as perguntas estavam inteligíveis para a aplicação posterior aos participantes da amostra principal.

Posteriormente ao teste piloto, o instrumento foi aplicado inicialmente para as três turmas de Introdução à Programação do semestre letivo 2021.1 que compõem o grupo denominado ingressantes. Posteriormente a essa primeira coleta, o mesmo instrumento foi aplicado aos alunos da turma de Linguagem de Programação e das turmas de Programação Orientada à Objetos, que

compõem o grupo denominado veteranos. Na coleta de dados, o escopo do trabalho foi apresentado a todos em salas de aula online, onde posteriormente foi disponibilizado o link do questionário, levando em média 45 minutos para a sua resolução.

5. Resultados e Discussão

A coleta de dados contém uma amostra composta por 105 alunos, sendo eles 63 das disciplinas de Introdução à Programação, 21 das turmas de Linguagem de Programação e 21 das turmas de Programação Orientada à Objetos. 82,9% declararam ser do sexo masculino, 14,3% do sexo feminino e 2,9% preferiram não declarar. A média de idade foi de 20,5 anos e 69,5% dos estudantes residiam em pequenos centros urbanos.

Tendo como finalidade analisar se as habilidades do PC diferiam dos alunos ingressantes para os alunos veteranos, foi aplicado o instrumento Bebras, no qual foram obtidas algumas conclusões. No Gráfico 1 é possível observar o nível de acertos, em proporção, das questões objetivas do questionário. É observado que os alunos veteranos tiveram um nível maior de acertos em quase todas as questões, com exceção da questão 8 (oito) e 12 (doze), em relação aos alunos ingressantes, mas não houve diferença significativa, exceto apenas pela questão 14 (quatorze), onde o acerto dos veteranos foi mais de 20% superior ao dos ingressantes.

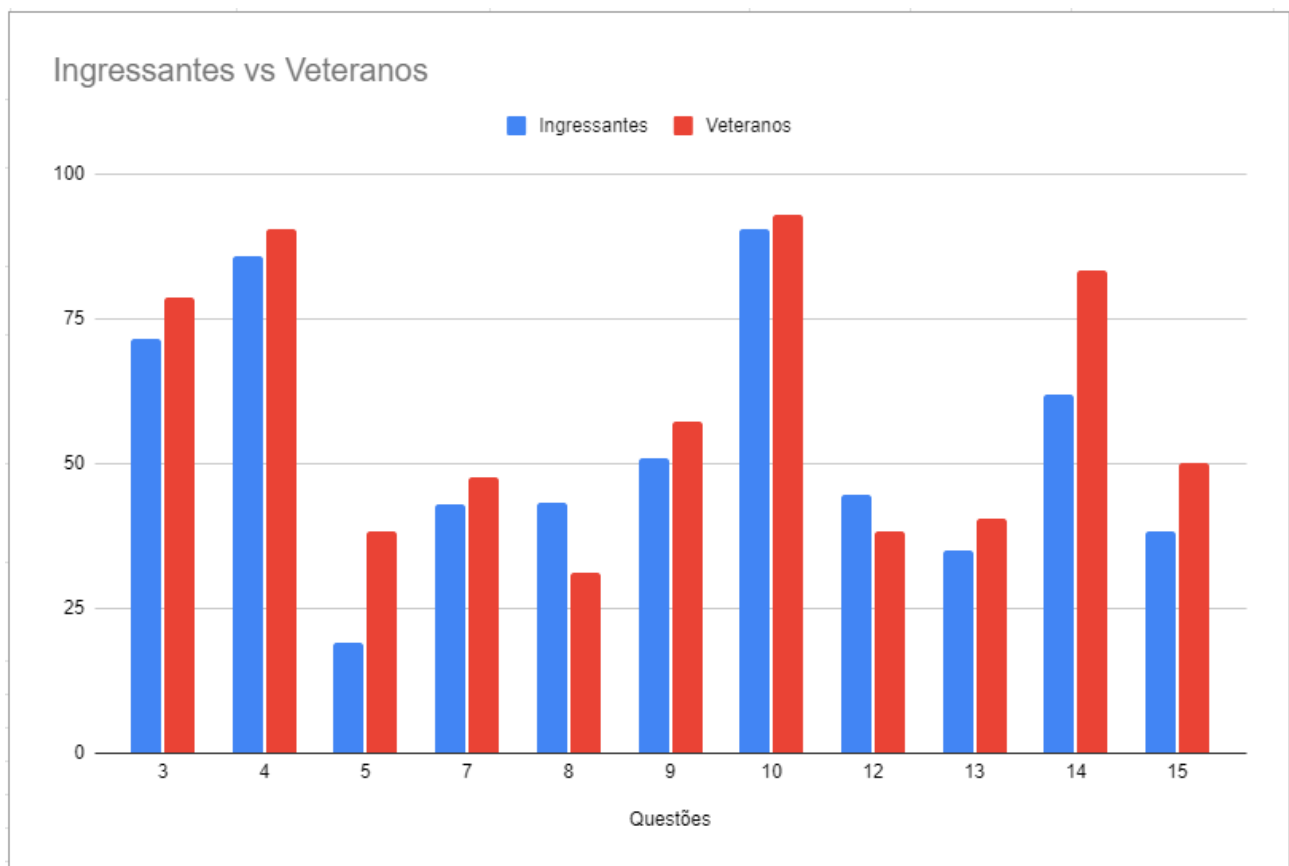


Gráfico 1. Acertos das questões de múltipla escolha ingressantes vs veteranos

Inicialmente, foi utilizado o teste de Shapiro-Wilk, com o objetivo de avaliar se a distribuição das amostras eram normais. Foi considerada como hipótese nula H_0 : Os dados seguem a distribuição normal e como hipótese alternativa H_a : Os dados não seguem a distribuição normal.

Foi obtido com o teste os seguintes valores para as amostras: Ingressantes p-valor = 0.528, P2 p-valor = 0.412, P3 p-valor = 0.530 e Veteranos p-valor = 0.821. Como todos os p-valores foram superiores a 0.05, a hipótese nula é aceita, o que quer dizer que as amostras seguem uma distribuição normal.

Para comparar as médias amostrais, foi usado o teste t, tendo como hipótese nula $H_0: \mu_{\text{Ingressantes}} = \mu_{\text{Veteranos}}$ e $H_a: \mu_{\text{Ingressantes}} \neq \mu_{\text{Veteranos}}$, como hipótese alternativa. As duas amostras são independentes, têm um nível de significância de 5%, e como está sendo verificado se as médias diferem, o teste é bicaudal.

O resultado do teste t foi um p-valor de 0,13. Como o p-valor foi maior que 0,05, que é o nível de teste, não podemos rejeitar a hipótese nula, ou seja, não é possível afirmar que as médias entre esses dois grupos diferem. Sendo assim, segundo o teste t, pode-se considerar que as diferenças entre os dois grupos não são estatisticamente significativas, considerando 5% de margem de erro.

Posteriormente foi dividida a amostra de veteranos em dois grupos distintos, chamados de P2, com os alunos das disciplinas de Linguagem de Programação e P3, composto pelos alunos das disciplinas de Programação Orientada à Objetos. O teste t também foi utilizado para comparar as médias dos dois grupos, com a hipótese nula $H_0: \mu_{P2} = \mu_{P3}$ e $H_a: \mu_{P2} \neq \mu_{P3}$, como hipótese alternativa. As duas amostras foram independentes, utilizando um nível de significância de 5%, o teste é bicaudal. O resultado foi um p-valor de 0,57. Dessa forma é aceita a hipótese nula, demonstrando que não existe diferença significativa entre os dois grupos, utilizando uma margem de erro de 5%.

No Gráfico 2 é possível verificar, de forma visual, que existe equiparidade entre a proporção de acertos dos dois grupos.

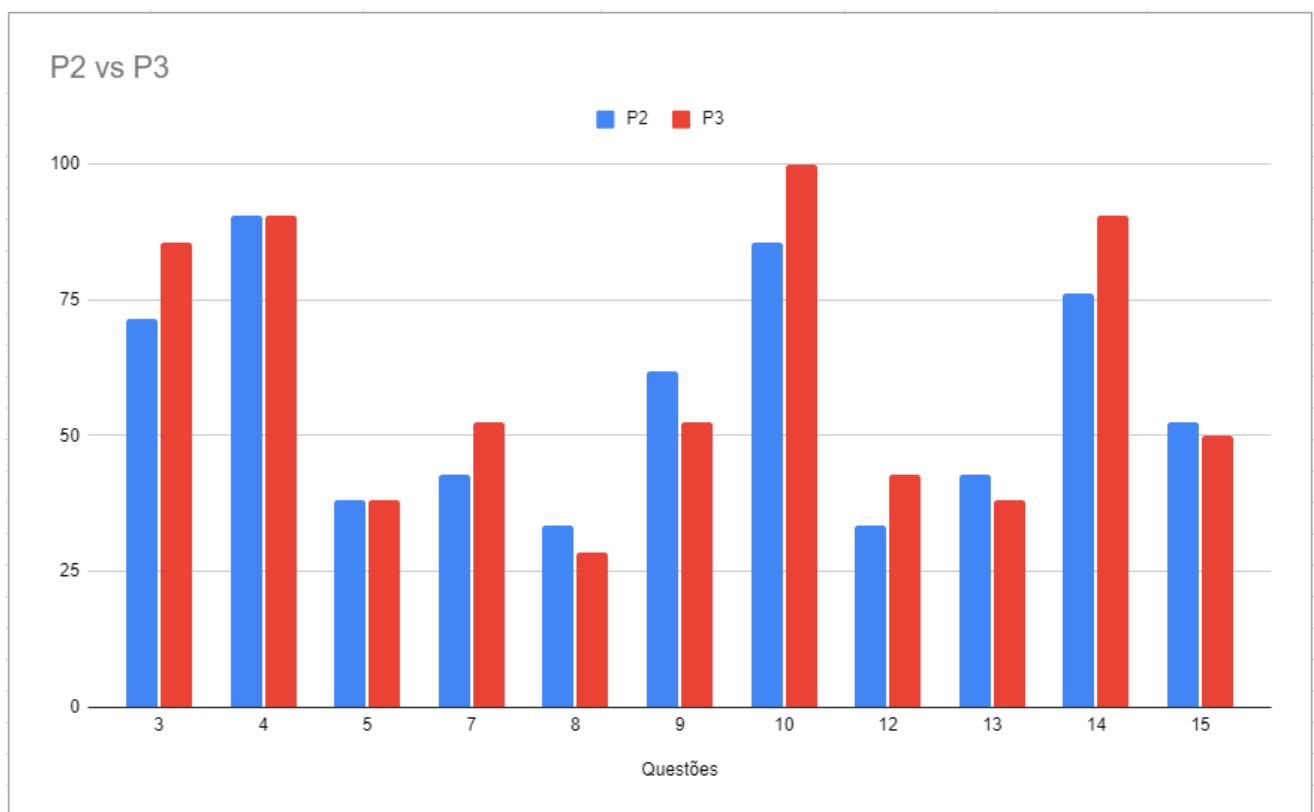


Gráfico 2. Acertos das questões de múltipla escolha P2 vs P3

Como parte do questionário, foi disponibilizado aos alunos uma pesquisa, para determinar a opinião deles a respeito da Ciência da Computação/Sistema de Informação. Foi usada a escala Likert com opções que variam de “Concordo totalmente” a “Discordo totalmente” para coletar as informações. Os dados do Gráfico 3 mostram as opiniões dos ingressantes. Um ponto que chama atenção é que 100% dos alunos concordam de forma total ou parcial, que ser capaz de resolver problemas é fundamental para a CC/SI.

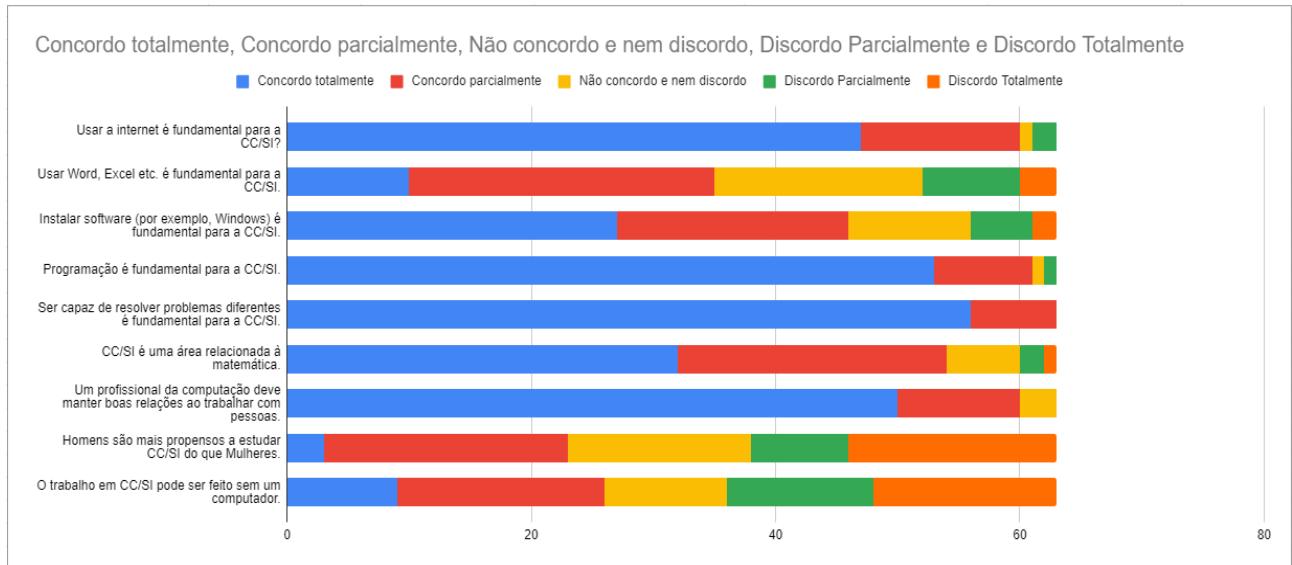


Gráfico 3. Opinião dos estudantes ingressantes sobre a Ciência da Computação (CC)/Sistema de Informação (SI)

No Gráfico 4, que mostra as opiniões dos veteranos, um ponto a se observar é que apenas 1 aluno concorda de forma total, que usar Word, Excel, etc é fundamental para a CC/SI. Também é possível ser visto no Gráfico 3 e no Gráfico 4, que as opiniões dos alunos a respeito do trabalho em CC/SI poder ser feito sem um computador, são bem divididas.

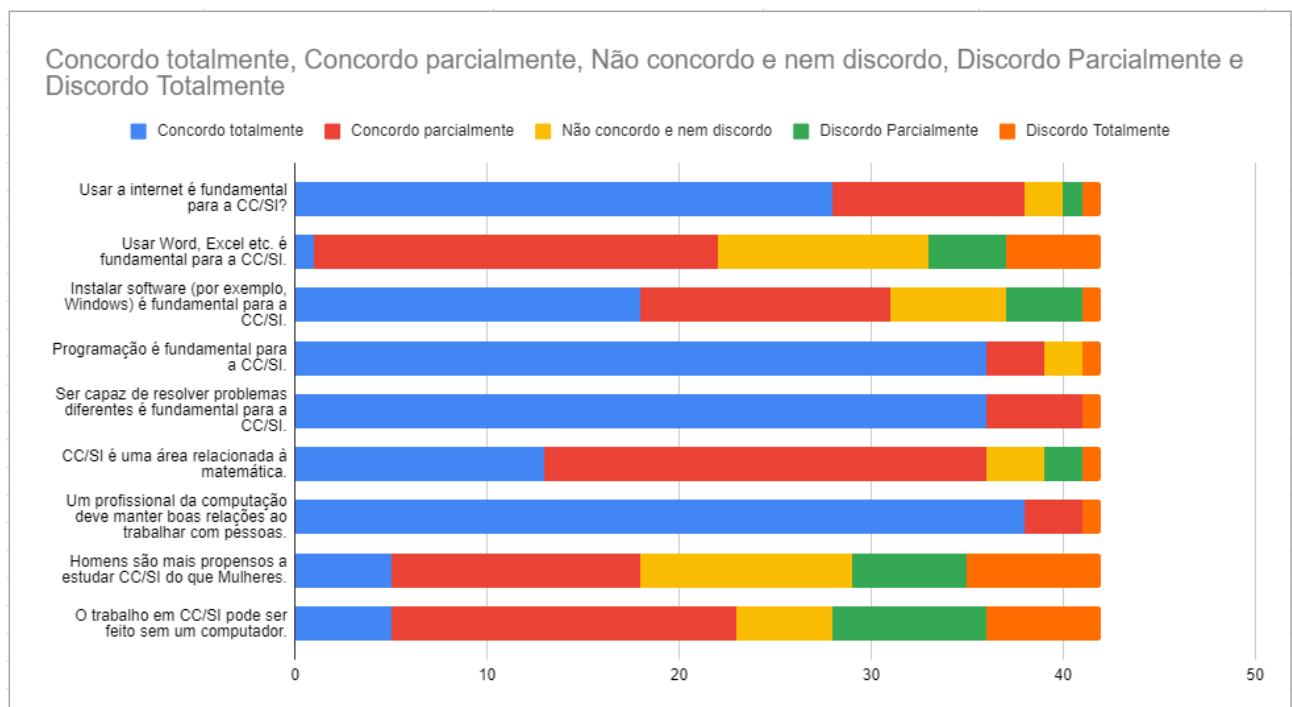


Gráfico 4. Opinião dos estudantes veteranos sobre a Ciência da Computação (CC)/Sistema de Informação (SI)

Nos Gráficos 3 e 4 também é possível observar uma imparcialidade sobre homens serem mais propensos a estudar CC/SI do que mulheres, o que não é visto na realidade, como constatado no Gráfico 5: 82,9% dos participantes declararam ser do sexo masculino. Isso pode ocorrer devido a uma série de fatores, como o maior incentivo aos meninos com jogos eletrônicos e computadores, enquanto as meninas são mais incentivadas a brincar com outros objetos que envolvam cuidados, como bonecas e bichos de pelúcia, como cita Miriam Vasco: “Num mundo tradicionalmente masculino, quase 100% da gestão das áreas relacionadas a TI é provida por homens. Mas podemos pensar em questões como: qual é a dinâmica na maioria dos lares? Os meninos são privilegiados desde cedo com os games, com os computadores e todo tipo de brinquedo eletrônico, e ainda contam com o incentivo do pai ou até mesmo da mãe, para isso”. Existem atualmente algumas políticas de incentivo às mulheres no meio da tecnologia, vemos uma crescente gradual, mas que ainda está longe de ser o ideal.

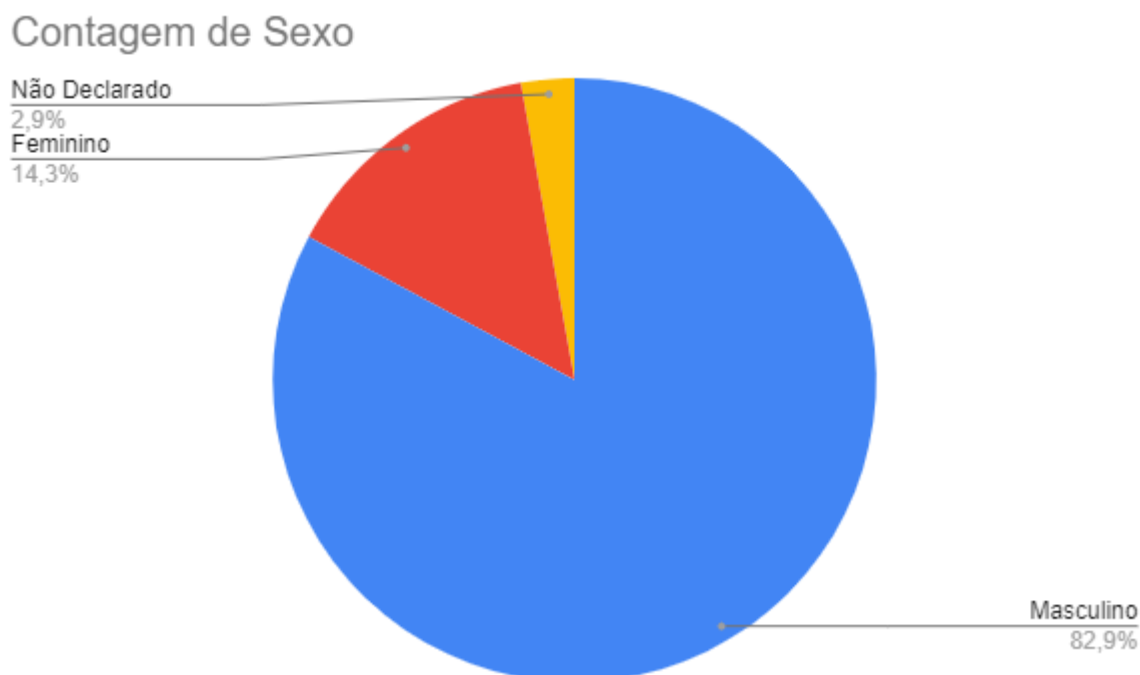


Gráfico 4. Comparativo do número de estudantes do sexo masculino e feminino

6. Considerações Finais

Neste trabalho, foi realizada uma pesquisa para entender se existe uma diferença nas habilidades do Pensamento Computacional dos alunos ingressantes nos cursos estudados, para os alunos veteranos. Os resultados apontaram que não há uma diferença significativa para afirmar que as habilidades de

PC dos veteranos são superiores a dos ingressantes. Também foi analisado se existia essa diferença dentro do grupo de veteranos, sendo ele dividido em dois subgrupos: o P2, composto pelos alunos de Linguagem de Programação, e o P3, composto pelos alunos de Programação Orientada à Objetos - que é uma disciplina cursada posteriormente à de Linguagem de Programação -, chegando à mesma conclusão.

Como trabalhos futuros, planejamos realizar nova coleta de dados com estudantes de Computação de outras universidades. Um outro estudo poderia ser conduzido com alunos de cursos na área de matemática e engenharia, para comparar os resultados. Também seria interessante investigar quais outros aspectos qualitativos podem ter levado a esse resultado.

Referências

- Mooney, A. and Lockwood, J. (2020) “The Analysis of a Novel Computational Thinking Test in a First Year Undergraduate Computer Science Course”. Aishe-j, 2020. Disponível em: <https://ojs.aishe.org/index.php/aishe-j/article/view/420>. Acesso em: 26 set. 2021.
- Csizmadia, A. et al (2015) “Computational thinking a guide for teachers”. Computing at school, 2015. Disponível em: Computing at School. Acesso em: 26 set. 2021.
- Howarth, L. et al (2014) “UK Bebras Computational Thinking Challenge”. Disponível em: Welcome - UK Bebras. Acesso em: 26 set. 2021.
- Araujo, A. et al (2015) “Pensamento Computacional sob a visão dos profissionais da computação: uma discussão sobre conceitos e habilidades”. In Anais dos VI Workshops do CBIE.
- Wing, J. 2006. Computational thinking. Commun. ACM, 49, p.3335.
- Araujo, A. et al (2018) “Como Identificar Habilidades do Pensamento Computacional? Um Estudo Empregando Análise Fatorial”. In Anais dos VII Workshops do CBIE.
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, 37, 88-137.
- Weintrop, D. et al (2016). Defining Computational Thinking for Mathematics and Science Classrooms. Journal of Science Education and Technology, 25, 127-147.

Apêndices

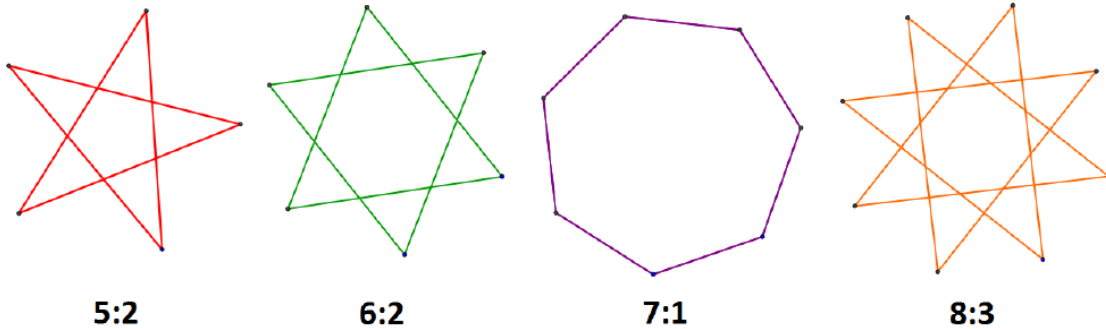
Anexo 1

Desenhando estrelas

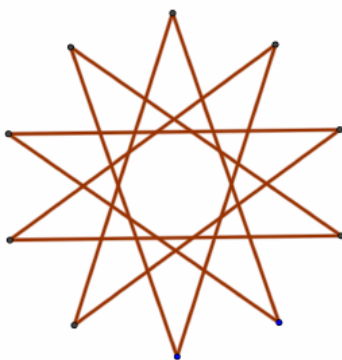
Manuella adora desenhar estrelas. Ela criou um sistema para marcar suas estrelas de acordo com a sua forma. Ela usa dois números:

- O primeiro número refere-se à quantidade de pontos da estrela.
- O segundo número refere-se a um valor que indica se há uma linha reta, a partir de um ponto, para o ponto mais próximo (nesse caso o valor é 1), para o segundo ponto mais próximo (nesse caso o valor é 2), etc.

Abaixo estão quatro exemplos do sistema de marcação de Manuella:



Questão: Como Manuella deve marcar a estrela a seguir?

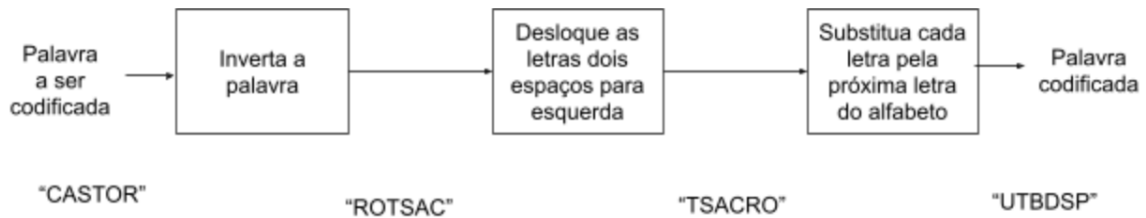


- a) 9:3
- b) 9:4
- c) 10:4
- d) 10:5

Resposta: Letra C

Você não irá descobrir

Os castores Cajú e Castanha enviaram mensagens um ao outro com a seguinte sequência de transformações em cada palavra.



Por exemplo, a palavra "CASTOR" foi transformada em "UTBDSP".
Castanha recebeu a seguinte mensagem "BWBDPM" de Cajú.

Questão: O que danado Cajú queria dizer?

- a) GAIATO b) CASUAL c) CAVALO d) PEIXES

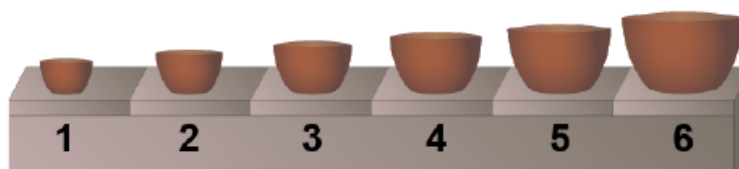
Resposta: Letra C

Fábrica de potes

Uma fábrica produz conjuntos de 6 potes de tamanhos diferentes. Uma longa esteira transportadora move os potes um por um, da esquerda para a direita.

O conjunto de 6 potes é colocado sobre a esteira transportadora em uma ordem aleatória.

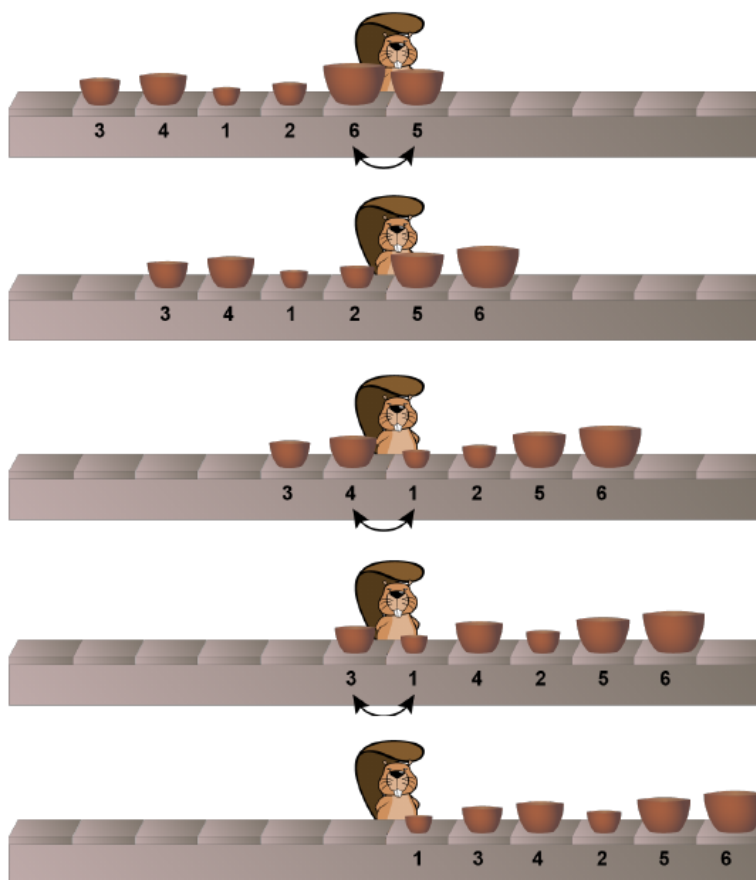
Antes de embalar os potes, eles precisam ser ordenados da forma como aparece na figura abaixo:



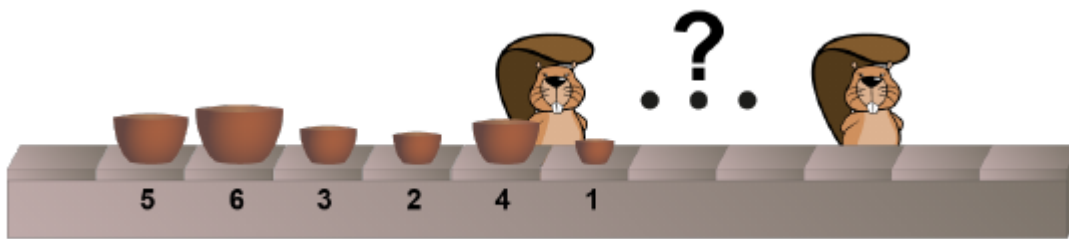
Para ajudar na ordenação, a fábrica coloca os trabalhadores ao longo da esteira transportadora.

Quando um conjunto de potes passa por um trabalhador, ele deve trocar quaisquer dois potes vizinhos que estejam na ordem errada. O trabalhador vai continuar fazendo isso até que o conjunto de 6 potes tenha passado completamente pela esteira.

Veja como a ordem dos potes muda ao passar pelo trabalhador:



Questão: Quantos trabalhadores devem ser colocados ao longo da esteira transportadora para ordenar o conjunto de potes a seguir?



- a) 3
- b) 4
- c) 5
- d) 6

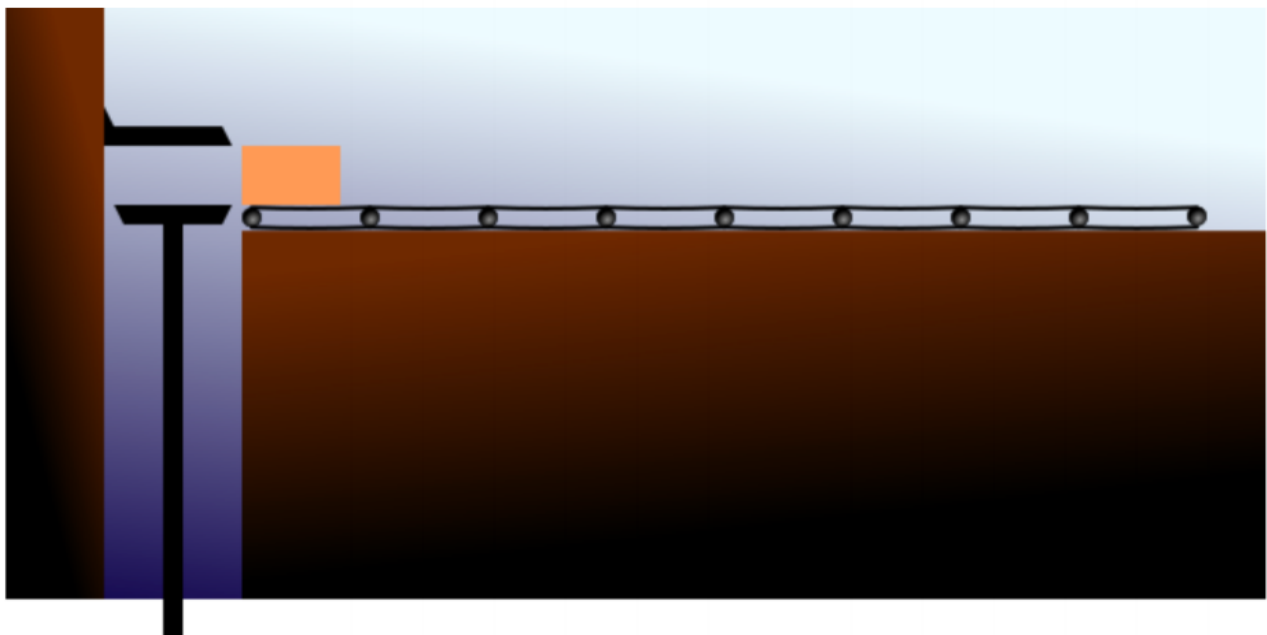
Resposta: Letra B

O Computador Empilha

O Computador Empilha é carregado com caixas provenientes de uma esteira transportadora. As caixas são marcadas com um Número ou um Operador (+, -, * ou /).

O computador é carregado até que a caixa superior seja uma caixa marcada com um operador. Esse operador é então usado nas duas caixas anteriores a ele. As três caixas são reunidas em uma única caixa e marcada com o resultado do cálculo.

No Computador Empilha, os números e operadores são inseridos de uma maneira diferente de uma calculadora normal.



Exemplos:

$2 + 3$ devem entrar como $2\ 3\ +$

$10 - 2$ devem entrar como $10\ 2\ -$

$5 * 2 + 3$ devem entrar como $5\ 2\ *\ 3\ +$

$5 + 2 * 3$ devem entrar como $5\ 2\ 3\ *\ +$

$(8 - 2) * (3 + 4)$ devem entrar como $8\ 2\ -\ 3\ 4\ +\ *$

Questão: Como deve ser a entrada de caixas para o Computador Empilha calcular $4*(8+3)-2$?

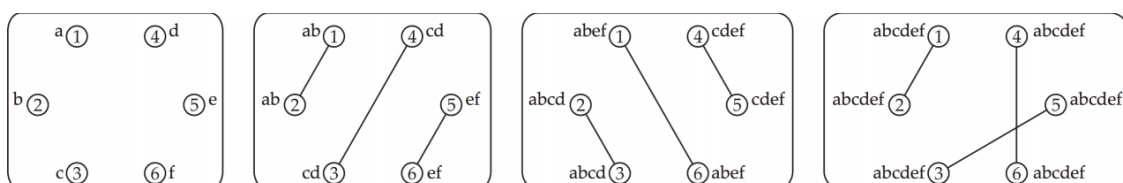
Resposta: $4\ 8\ 3\ +\ *\ 2\ -$

Espiões

Toda sexta-feira, seis espiões trocam toda a informação que coletaram durante a semana. Um espião nunca pode ser visto com mais de um espião ao mesmo tempo. Assim, eles precisam ter várias rodadas de reuniões nas quais eles se encontram aos pares e partilham todas as informações que possuem até aquele momento.

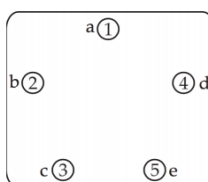
O grupo de 6 espiões precisa apenas de três rodadas de reuniões para distribuir todos os segredos:

Antes das reuniões, cada espião possui uma única informação. (O espião 1 sabe "a", o espião 2 sabe 'b', etc.). Na primeira rodada, os espiões 1 e 2 se reúnem e trocam informações para que agora ambos saibam 'ab'. Os diagramas mostram quais espiões se encontraram em cada rodada com uma linha. Ele também mostra que pedaços de informação todos eles possuem. Após três rodadas de reuniões, todas as informações foram distribuídas.



Depois de um incidente internacional, um espião parou de assistir às reuniões. Qual é o número mínimo de rodadas necessárias para que os cinco espiões restantes troquem todas as informações?

Resposta: Letra D.



Fogos de artifício

Dois castores vivem em tocas separadas por uma grande floresta.

Eles decidiram enviar mensagens entre si, lançando fogos de artifício.

Cada mensagem é uma sequência de palavras, embora os castores só conheçam cinco palavras diferentes.

Os castores podem lançar dois tipos de fogos de artifício, um após o outro, e conhecem os seguintes códigos:

Word	Code
Tronco	
Árvore	
Pedra	
Rio	
Comida	

Por exemplo, para mandar a mensagem (bastante estranha) "Comida, Tronco, Comida", o castor deveria lançar:



Questão: Quantos significados diferentes a mensagem a seguir possui?



- a) 1 b) 2 c) 3 d) 4

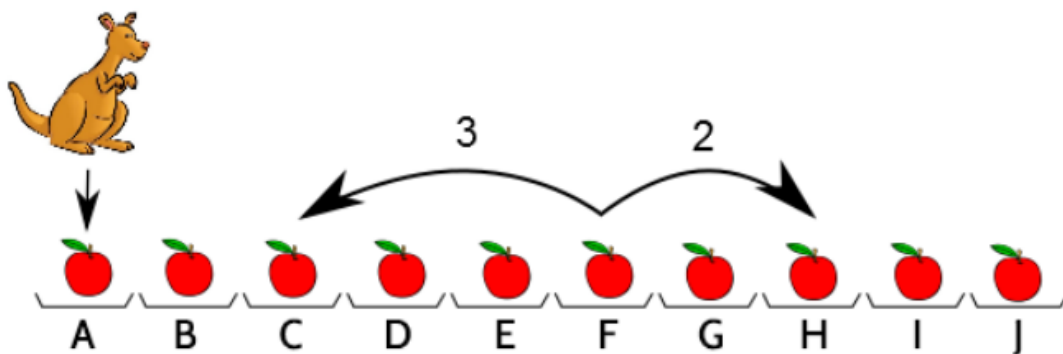
Resposta: Letra D

Canguru

Existem 10 pratos em fileira. Há uma maçã em cada prato. Jack, o canguru, adora pular. Primeiro, ele salta sobre o prato mais à esquerda com a letra A, recolhendo a maçã.

Após isso, em cada salto, ou ele salta para frente, pulando dois pratos, ou ele salta para trás, pulando três pratos.

(Um exemplo dos dois possíveis saltos nos pratos é mostrado com as setas na figura abaixo.)



Jack só pode saltar em pratos que ainda tenham maçã.

Se ele pular em cima de um prato, ele recolhe a maçã para ele.

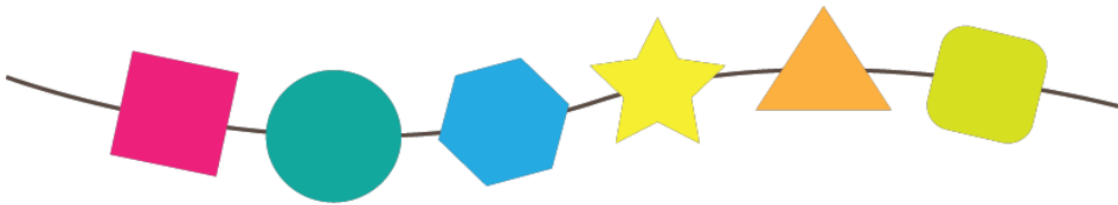
Questão: Se Jack coletar todas as 10 maçãs, qual maçã ele coletará por último?

A B C D E F G H I J

Resposta: Prato i.

Pulseira

Josefina quebrou sua pulseira favorita. A pulseira quebrada agora tem essa sequência:



Questão: Qual das quatro pulseiras a seguir mostra como era a pulseira quando estava inteira?



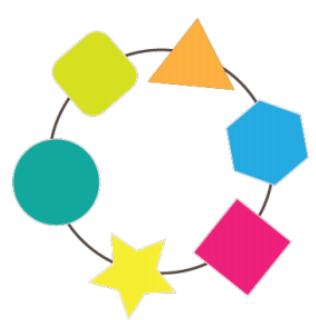
A



B



C



D

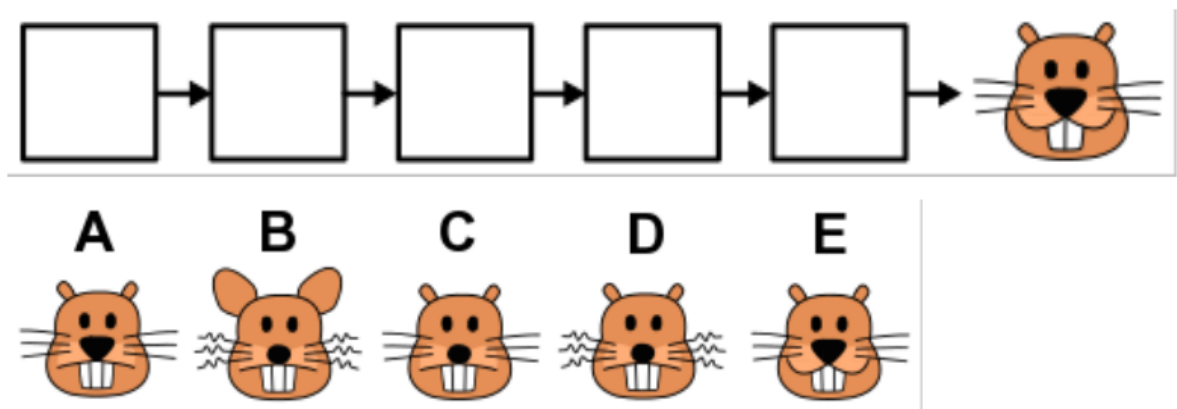
Resposta: Letra B

Animação

Joca está planejando uma animação de um rosto feita a partir de uma sequência de fotos.

Para fazer a animação funcionar sem problemas, apenas uma característica do rosto deve mudar de uma imagem para a outra.

Infelizmente, as fotos se misturaram. Agora Joca deve encontrar a ordem correta novamente. Felizmente, ele sabe a última foto.



Questão: Coloque as imagens na ordem correta para que o Joca faça a animação correta, mudando apenas uma característica do rosto.

Resposta: BDCAE

Competição Animal



Os castores e cães fizeram uma competição. No total, nove animais participaram. Os nove participantes tiveram as seguintes pontuações: 1, 2, 2, 3, 4, 5, 5, 6, 7.

Nenhum cão pontuou mais do que qualquer castor.

Um cachorro empatou com um castor.

Havia também dois outros cães que empataram um com o outro.

Questão: Quantos cães participaram da competição?

- a) 2
- b) 3
- c) 5
- d) 6
- e) 7

Resposta: Letra d.

Cruzando o País

Três castores muito velozes irão competir em uma corrida para cruzar o país.

O Ciço ultrapassará um castor ao correr morro acima.



A Zefa ultrapassará um castor ao correr ladeira abaixo.

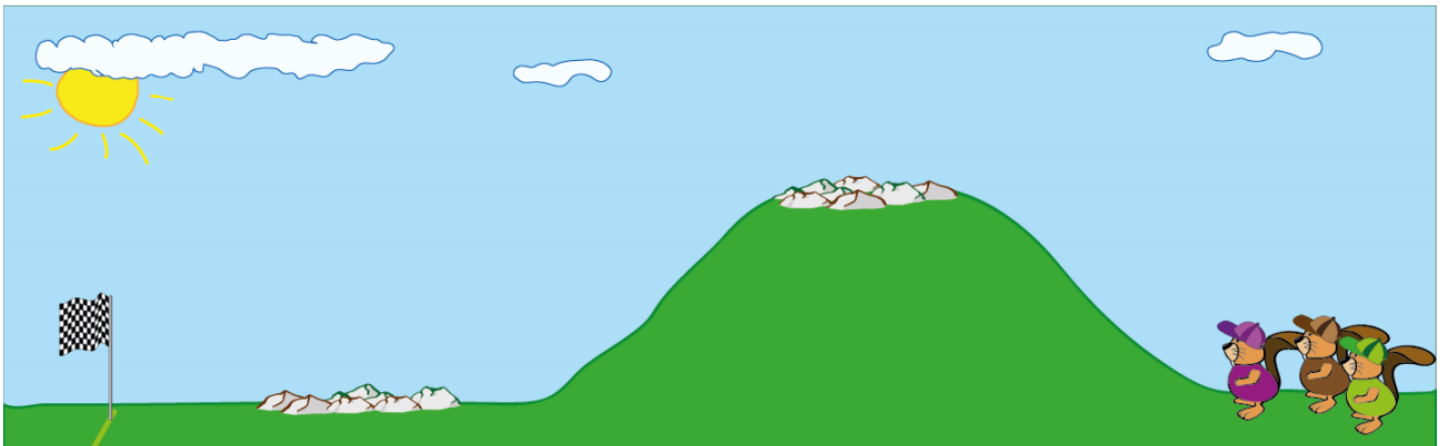


A Maria ultrapassará um castor ao correr sobre as rochas.



O terreno é mostrado na foto: subida, seguida de algumas pedras, descida e depois mais algumas pedras.

A Zefa começa na primeira posição, seguida pelo Ciço e pela Maria.



Questão: Em que ordem os castores terminarão a corrida?

- a) Zefa, Ciço, Maria
- b) Ciço, Zefa, Maria
- c) Ciço, Maria, Zefa
- d) Maria, Zefa, Ciço

RESPOSTA: LETRA C

Jogue os Dados

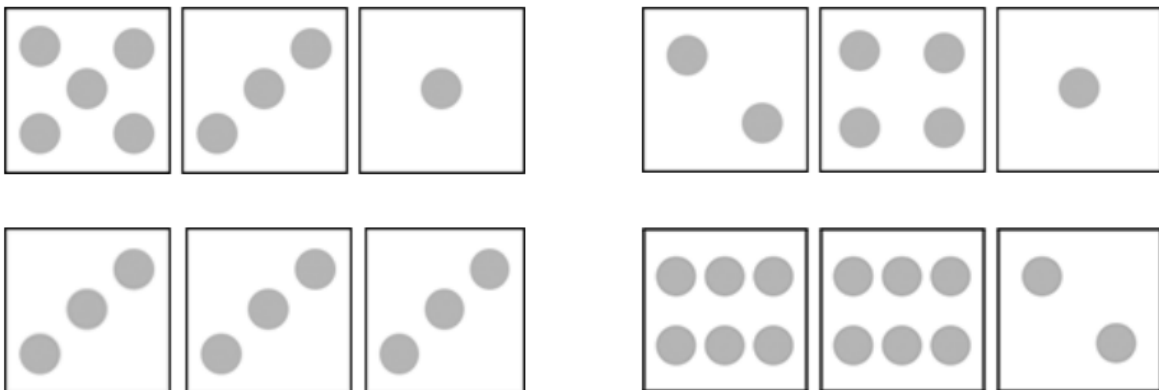
Depois da escola, os jovens castores costumam brincar juntos.

Para evitar discussões sobre onde jogar, eles jogam um dado normal de seis lados.

A decisão é encontrada de acordo com esta regra:

1	SE	o primeiro lance é maior que o segundo lance
2	ENTÃO	vamos brincar na mata
3	SENÃO	
4	SE	o terceiro lance é menor que o primeiro lance
5	ENTÃO	vamos jogar no rio
6	SENÃO	vamos jogar no campo

Questão: Qual sequência de arremessos vai mandar os jovens castores para o campo?



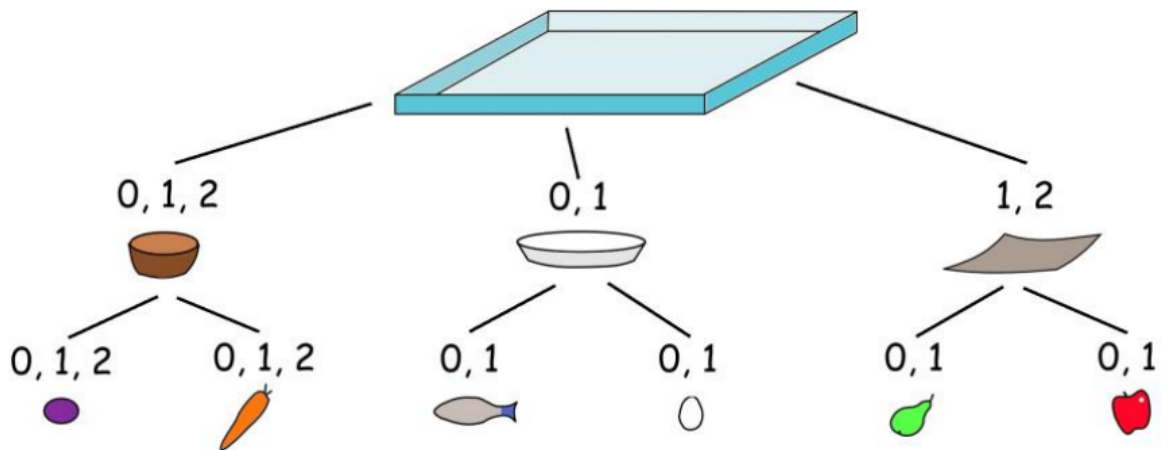
Resposta: Letra B.

Almoço de Castor

Huum, o que levar para o almoço hoje?

O refeitório dá instruções sobre como escolher um almoço de castor.

Isso é mostrado como um diagrama:



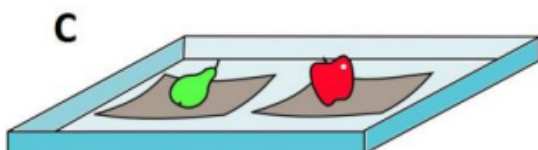
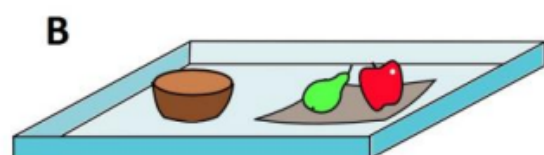
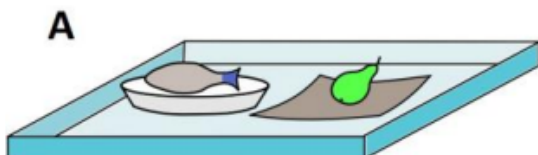
Abaixo da bandeja, você vê diferentes tipos de recipientes para alimentos.

Os números indicam quantos recipientes desse tipo podem ser adicionados a uma bandeja.

Cada recipiente só pode ter alimentos colocados nele que são mostrados abaixo dele.

Os números indicam quantos alimentos deste tipo podem ser adicionados aos recipientes.

Questão: Qual dos seguintes almoços não é um almoço de castor adequado?



Resposta: Letra D.

Anexo 2

Questionário Artigo

Responda as questões abaixo com os seus conhecimentos a respeito da Ciência da Computação/Sistema da Informação.

1. A Ciência da Computação/Sistema da Informação é interessante para você?
2. A Ciência da Computação/Sistema da Informação é desafiante?
3. Usar a internet é fundamental para a Ciência da Computação/Sistema da Informação.
4. Usar Word, Excel etc. é fundamental para a Ciência da Computação/Sistema da Informação.
5. Instalar software (por exemplo, Windows, iTunes) é fundamental para a Ciência da Computação/Sistema da Informação.
6. A programação é fundamental para a Ciência da Computação/Sistema da Informação.
7. Ser capaz de resolver problemas diferentes é fundamental para a Ciência da Computação/Sistema da Informação.
8. Ciência da Computação/Sistema da Informação é uma área relacionada à matemática.
9. Um Cientista da Computação deve ser bom trabalhando com pessoas.
10. Homens são mais propensos a estudar Ciência da Computação/Sistema da Informação do que Mulheres.
11. O trabalho em Ciência da Computação/Sistema da Informação pode ser feito sem um computador.
12. Você já ouviu falar no termo Pensamento Computacional?

Anexo 3

Questionário de Dados Pessoais

- Nome
- Sexo
- Idade
- Cidade onde mora

Qual era a sua situação antes de iniciar o curso?

- Nunca tinha programado e não sabia nada sobre programação
- Nunca tinha programado, mas tinha noções de programação
- Já tinha feito curso (presencial ou online) de programação
- Já sabia programar, mas não em Python
- Já sabia programar, inclusive em Python