



SECRETARIA DE ESTADO DA EDUCAÇÃO DO PIAUÍ - SEDUC-PI
ASSESSORIA TÉCNICA - UETEP - SEDUC-PI

Av. Pedro Freitas, S/N Centro Administrativo, Bloco D/F - Bairro São Pedro, Teresina-PI, <http://www.seduc.pi.gov.br>

Processo nº 00011.006304/2026-07

Teresina-PI, 30 de abril de 2026

EDITAL SEDUC-PI/GSE Nº: 16/2026

EDITAL DE PROCESSO SELETIVO DE ESTUDANTES E PROFESSORES PARA PARTICIPAÇÃO EM INTERCÂMBIO EDUCACIONAL INTERNACIONAL E NACIONAL PROGRAMA
 “DO PIAUÍ PARA O MUNDO” - EDIÇÃO 2026.

**PUBLICAÇÃO DO RESULTADO DA ANÁLISE DOS RECURSOS CONTRA A PROVA OBJETIVA - TECNOLOGIA DA INFORMAÇÃO (TI) E LÍNGUA INGLESA E
 EMPREENDEDORISMO E LÍNGUA INGLESA**

1) LÍNGUA INGLESA (PARA AMBAS AS PROVAS).

QUESTÕES	ARGUMENTO APRESENTADO	SITUAÇÃO	JUSTIFICATIVA
1 a 8	Ausência de comando expresso nas questões objetivas.	INDEFERIDO	<p>Sustentam os recorrentes que as questões 01 a 08 deveriam ser anuladas em razão da ausência de comando expresso. Contudo, da análise dos itens, verifica-se que se tratam de questões objetivas de compreensão leitora, compostas por texto-base e alternativas de resposta, em estrutura apta a permitir a identificação, pelo candidato, da alternativa correta a partir das informações veiculadas no excerto.</p> <p>Em provas objetivas de proficiência em leitura, a inteligibilidade do item decorre não apenas de enunciado verbal expresso, mas também da própria organização da questão, desde que suficiente para tornar inequívoca a tarefa interpretativa exigida. No caso, a estrutura apresentada permite compreender, sem ambiguidade relevante, que se exige do candidato a leitura do texto e a seleção da alternativa compatível com seu conteúdo.</p> <p>Não se verifica, portanto, ilegalidade manifesta ou vício material apto a comprometer a validade das questões. A alegação recursal aponta aspecto meramente formal, sem demonstração</p>

concreta de prejuízo à compreensão do item ou de violação à isonomia entre os candidatos.

Dessa forma, indefere-se o recurso, mantendo-se o gabarito e a validade das questões impugnadas.

2) EMPREENDEDORISMO

QUESTÃO	ARGUMENTO APRESENTADO	SITUAÇÃO	JUSTIFICATIVA
24	Trata-se de recurso interposto em face da Questão 24 da prova de Empreendedorismo, sob o argumento de que a questão incorreria em ambiguidade metodológica ao abordar a sequência de etapas inerentes às metodologias Lean Startup e Design Thinking, tornando plausível mais de uma alternativa como resposta correta.	INDEFERIDO	<p>Trata-se de recurso interposto pelo candidato em face da Questão 24 da prova de Empreendedorismo, sob o argumento de que a questão incorreria em ambiguidade metodológica ao abordar a sequência de etapas inerentes às metodologias Lean Startup e Design Thinking, tornando plausível mais de uma alternativa como resposta correta.</p> <p>Após análise detida dos argumentos apresentados, decide-se pelo não provimento do recurso, com fundamento nas razões a seguir expostas.</p> <p>I. Da clareza e objetividade do enunciado</p> <p>O enunciado da questão delimita com precisão o contexto metodológico avaliado, não ensejando interpretações divergentes que comprometam sua objetividade. A banca examinadora, ao elaborar o item, baseou-se em referencial teórico consolidado, amplamente adotado na literatura especializada em empreendedorismo e inovação, razão pela qual não se verifica qualquer vício de redação apto a gerar dúvida razoável no candidato devidamente preparado.</p> <p>II. Da sequência metodológica no Lean Startup</p> <p>O candidato argumenta que a construção do Produto Mínimo Viável (Minimum Viable Product - MVP) pode anteceder a etapa de validação de mercado, o que, em sua visão, tornaria admissível sequência distinta da contemplada no gabarito oficial. Contudo, tal argumento parte de uma compreensão equivocada da arquitetura lógica do método.</p> <p>Conforme proposto por Eric Ries na obra seminal The Lean Startup (2011), o ciclo de aprendizado validado (Build-Measure-Learn) pressupõe que a construção do MVP é, em si, uma resposta a hipóteses previamente formuladas sobre o problema e o segmento de clientes. Desse modo, etapas como a identificação do problema, a definição do público-alvo</p>

e a formulação de hipóteses constituem pré-condições lógicas e metodológicas à construção do MVP, e não etapas opcionais ou intercambiáveis. A inversão dessa ordem implicaria a construção de solução sem problema previamente validado, contradizendo os próprios fundamentos epistemológicos do método.

III. Da integração com o Design Thinking

No que concerne ao Design Thinking, metodologia desenvolvida pela IDEO e sistematizada pelo d.school de Stanford, a sequência canônica, Empatia → Definição → Ideação → Prototipagem → Teste, estabelece que a prototipagem somente ocorre após a compreensão profunda do usuário e a definição clara do problema (problem statement). A eventual flexibilidade iterativa admitida em estágios avançados do processo não autoriza a conclusão de que as etapas iniciais possam ser suprimidas ou reordenadas sem prejuízo metodológico.

A questão avalia, precisamente, o domínio dessa estrutura canônica, que é objeto de cobrança legítima em certames que aferem conhecimento técnico em empreendedorismo e inovação.

IV. Da ausência de dupla validade das alternativas

Para que se justifique a anulação de uma questão por multiplicidade de respostas corretas, é imprescindível que mais de uma alternativa seja igualmente defensável à luz do referencial técnico-científico adotado, e não apenas plausível sob interpretação particular ou leitura parcial da literatura. No presente caso, a alternativa apontada pelo gabarito oficial reflete com fidelidade a sequência metodológica prescrita pelos autores de referência, ao passo que a sequência proposta pelo candidato não encontra respaldo em fonte técnica reconhecida que sustente sua equivalência.

A mera possibilidade teórica de variações contextuais em aplicações práticas não é suficiente para infirmar a correção da alternativa oficial, tampouco para configurar ambiguidade invalidante.

3) TECNOLOGIA DA INFORMAÇÃO (TI)

QUESTÃO	ARGUMENTO APRESENTADO	SITUAÇÃO	JUSTIFICATIVA
---------	-----------------------	----------	---------------

01	O recurso alega conflito de precedência entre operadores aritméticos e lógicos.	INDEFERIDO	<p>A questão trata de entrada de dados com <code>split("#")</code>, tratamento de string com <code>strip()</code> e <code>title()</code>, conversão para inteiro e uso de <code>len()</code>.</p> <p>A entrada informada é expressamente ana clara # 7. Após o processamento, o nome torna-se Ana Clara, com 9 caracteres, incluindo o espaço interno. Assim, <code>7 + len("Ana Clara")</code> resulta em 16, produzindo a saída Ana Clara:16.</p> <p>Não há conflito de precedência nem múltiplas interpretações.</p>
02	O recurso menciona escopo de variável local e global.	INDEFERIDO	<p>A questão real não possui função, variável global ou alteração de escopo.</p> <p>O código avalia:</p> <pre>a = 4 b = 2.0 c = a / 2 == b</pre> <p>Como <code>a/2</code> resulta em 2.0, a comparação com <code>b</code> é verdadeira. Em seguida, <code>d = c + a</code> equivale a <code>True + 4</code>, resultando em 5, pois em Python <code>bool</code> é subtipo de <code>int</code>. Portanto, <code>type(d)</code> é <code><class 'int'></code>.</p> <p>O argumento do recurso não se aplica ao item.</p>
03	O recurso afirma haver operação entre string e inteiro sem conversão explícita.	INDEFERIDO	<p>A questão real trabalha apenas com atribuição, atualização numérica e atribuição múltipla.</p> <p>O código resulta em:</p> <pre>a = 10 b = 10 a = 15 c, a = b, a - b</pre> <p>Logo, <code>c = 10</code> e <code>a = 5</code>. A saída é 5 10 10.</p> <p>Não há string, coerção inválida ou <code>TypeError</code>.</p>
04	O recurso questiona a validade do identificador média, alegando que dependeria de conhecimento específico sobre Unicode.	INDEFERIDO	<p>Em Python 3, identificadores podem conter caracteres Unicode, desde que respeitem as regras sintáticas da linguagem.</p> <p>São válidos: <code>_servidores</code>, <code>total2026</code>, <code>classificacao_final</code> e <code>média</code>.</p> <p>São inválidos: <code>2fase</code>, por iniciar com número; <code>valor-total</code>, por conter hífen; e <code>for</code>, por ser palavra reservada.</p> <p>A questão avalia regras de identificadores em Python, conteúdo pertinente à prova de TI.</p>

05	O recurso menciona inconsistência na indexação de arrays ou listas.	INDEFERIDO	<p>A questão real não trata de listas, arrays ou índice [3]. O item avalia precedência de operadores aritméticos.</p> <p>A expressão é:</p> $18 + 5 * 2 ** 3 // 4 - 7 \% 4$ <p>Primeiro calcula-se a potência: $2 ** 3 = 8$. Depois multiplicação, divisão inteira e módulo: $5 * 8 // 4 = 10$ e $7 \% 4 = 3$. Por fim: $18 + 10 - 3 = 25$.</p> <p>O argumento do recurso não corresponde à questão.</p>
06	O recurso menciona passagem de parâmetros por referência, objetos mutáveis e métodos.	INDEFERIDO	<p>A questão real não possui função, método ou passagem de parâmetro.</p> <p>O código compara 7, 7.0 e "7". Em Python, $7 == 7.0$ é verdadeiro, 7 is 7.0 é falso, pois identidade de objeto não é igualdade de valor, e $7 == \text{int}("7")$ é verdadeiro.</p> <p>A saída correta é True False True.</p> <p>O argumento recursal é estranho ao conteúdo do item.</p>
07	O recurso alega ambiguidade entre divisão inteira e ponto flutuante, mencionando Python 2 e Python 3.	INDEFERIDO	<p>A questão real não utiliza divisão. O item trata de operadores lógicos e expressão booleana.</p> <p>Com idade = 17, acompanhado = True e autorizacao = False, a expressão resulta em True, pois o adolescente está acompanhado e a condição de bloqueio não se configura.</p> <p>Não há qualquer dependência de versão relacionada ao operador /.</p>
08	O recurso sustenta erro por divisão por zero.	INDEFERIDO	<p>A questão real não possui divisão nem laço de repetição. Trata-se de estrutura condicional com if, elif e else.</p> <p>Com nota = 6.8, frequencia = 74 e recurso = True, as duas primeiras condições são falsas por causa da frequência. A terceira condição é verdadeira porque recurso é verdadeiro e a frequência é maior ou igual a 70. A saída é Análise especial.</p> <p>O argumento de exceção por divisão por zero não se aplica.</p>
09	O recurso menciona curto-circuito em expressões lógicas.	INDEFERIDO	<p>A questão real trata de estruturas condicionais aninhadas, envolvendo protocolo, assinatura e pagamento.</p> <p>Com protocolo_regular = True, documento_assinado = False e taxa_paga = True, o fluxo entra no bloco do protocolo regular, segue para o caso de documento não assinado e, como a taxa está paga, imprime Aguardando assinatura.</p>

			A questão possui resposta objetiva. A alegação apresentada não invalida o item.
10	O recurso fala sobre ordenação de dicionários.	INDEFERIDO	<p>A questão real trata de match/case em Python 3.10+, expressamente indicado no enunciado.</p> <p>Para status = ("financeiro", 2), o caso aplicável é:</p> <p>case ("financeiro", x) if x > 0:</p> <p>Assim, x = 2 e a saída é 20.</p> <p>Não há dicionário, ordenação ou omissão de versão, pois o enunciado informa Python 3.10+.</p>
11	O recurso afirma que o algoritmo tenta realizar append em uma tupla.	INDEFERIDO	<p>A questão real não usa tuplas nem append. O item trabalha com laços for aninhados e acumulação numérica.</p> <p>A soma dos lotes é:</p> <p>Setor 1: 1 + 2 = 3</p> <p>Setor 2: 2 + 3 = 5</p> <p>Setor 3: 3 + 4 = 7</p> <p>Total: 3 + 5 + 7 = 15.</p> <p>O argumento do recurso não corresponde ao código da questão.</p>
12	O recurso menciona acesso a atributo privado em orientação a objetos.	INDEFERIDO	<p>A questão real não possui classe, objeto ou atributo privado.</p> <p>O código faz atualizações sucessivas na variável x:</p> <p>x = 7</p> <p>x += 3, resultando em 10</p> <p>x *= x - 4, resultando em 10 * 6 = 60</p> <p>x //= 2, resultando em 30</p> <p>A saída é 30.</p> <p>Não há qualquer vício relacionado a orientação a objetos.</p>
13	O recurso sustenta cenário de concorrência, múltiplas threads e condição de corrida.	INDEFERIDO	<p>A questão real não utiliza threads, concorrência ou sincronização.</p> <p>O item avalia lista, substituição por fatia, remoção com pop() e inserção com append().</p> <p>A lista evolui assim:</p> <p>[3, 6, 9, 12]</p> <p>Após fila[1:3] = [7, 8, 9]: [3, 7, 8, 9, 12]</p>

			<p>Após fila.pop(-2): [3, 7, 8, 12] Após append(fila[0] + fila[-1]): [3, 7, 8, 12, 15]</p> <p>A questão é determinística e possui alternativa correta.</p>
14	O recurso menciona garbage collection e destruição imediata de objeto.	INDEFERIDO	<p>A questão real trata de lista aninhada criada por multiplicação, com compartilhamento inicial de referência.</p> <p>Em:</p> <p>matriz = [[1, 2]] * 2</p> <p>as duas posições inicialmente apontam para a mesma lista interna. Ao executar matriz[0][1] = 9, ambas refletem a alteração. Em seguida, matriz[1] = [3, 4] substitui a segunda linha por uma nova lista.</p> <p>O resultado final é [[1, 9], [3, 4]].</p> <p>Não há cálculo de memória nem dependência de garbage collector.</p>
15	O recurso alega recursividade infinita.	INDEFERIDO	<p>O recurso alega recursividade infinita. A questão real não possui função recursiva. Trata-se de laço for com break.</p> <p>A execução acumula:</p> <p>n = 3, total 3</p> <p>n = 4, total 7</p> <p>n = 5, total 12</p> <p>n = 6, total 18</p> <p>Como 18 > 15, o laço é interrompido. A saída é 6 18.</p> <p>O argumento de stack overflow é inaplicável.</p>
16	O recurso fala em sobrecarga de operadores entre objetos de classes distintas.	INDEFERIDO	<p>A questão avalia operação aritmética entre booleanos em Python.</p> <p>len("2026") == 4 resulta em True. Logo:</p> <p>resultado = True + False</p> <p>Em Python, True equivale numericamente a 1 e False a 0 em operações aritméticas. O resultado é 1, cujo tipo é <class 'int'>.</p> <p>A questão avalia tipagem e comportamento de bool em Python, conteúdo pertinente à prova.</p>
17	O recurso menciona comparação de ponto flutuante e padrão IEEE 754.	INDEFERIDO	<p>A questão real não realiza comparação de float. O item avalia precedência de operadores aritméticos.</p> <p>A expressão é:</p>

			$27 // 4 + 2 * 3 ** 2 - 5 \% 3$ <p>Tem-se:</p> $27 // 4 = 6$ $3 ** 2 = 9$ $2 * 9 = 18$ $5 \% 3 = 2$ <p>Resultado: $6 + 18 - 2 = 22$.</p> <p>O argumento do recurso não se aplica ao item.</p>
18	O recurso afirma que o código utiliza palavra reservada como variável.	INDEFERIDO	<p>A questão real não utiliza class, return ou qualquer palavra reservada como identificador.</p> <p>O código compara valores e identidade:</p> <p>$x == y$ resulta em True, pois $5 == 5.0$;</p> <p>$x != \text{int}(z)$ resulta em False, pois $\text{int}("5")$ é 5;</p> <p>$x \text{ is not } y$ resulta em True, pois 5 e 5.0 não são o mesmo objeto.</p> <p>A saída é True False True.</p> <p>Não há erro de sintaxe.</p>
19	O recurso menciona atribuição múltipla e ordem de avaliação em cadeia.	INDEFERIDO	<p>A questão real não possui atribuição múltipla. Ela avalia operadores lógicos.</p> <p>Com <code>servidor_ativo = False</code>, <code>backup_ok = True</code> e <code>janela_manutencao = False</code>:</p> <p><code>(servidor_ativo or backup_ok)</code> resulta em True;</p> <p><code>not janela_manutencao</code> resulta em True;</p> <p><code>(servidor_ativo or not servidor_ativo)</code> também resulta em True.</p> <p>Portanto, a expressão completa resulta em True.</p> <p>A questão é objetiva e determinística.</p>
20	O recurso menciona erro de sintaxe em f-strings e mistura de padrões de formatação.	INDEFERIDO	<p>A questão real não contém f-string, .format() ou interpolação de strings. O item trata de estrutura condicional.</p> <p>Com <code>nota = 6.4</code>, <code>frequencia = 72</code> e <code>recurso_deferido = True</code>, as duas primeiras condições são falsas. A terceira é verdadeira:</p> <p><code>recurso_deferido and nota >= 6 and frequencia >= 70</code></p> <p>Logo, a saída é Classificado por recurso.</p>

			O argumento recursal é alheio à questão.
21	O recurso menciona deep copy e shallow copy.	INDEFERIDO	<p>A questão real não trata de cópia de listas nem objetos internos. O item avalia laços aninhados.</p> <p>O cálculo é:</p> <p>Para $i = 1$: $1*1 + 1*2 = 3$ Para $i = 2$: $2*1 + 2*2 + 2*3 = 12$ Para $i = 3$: $3*1 + 3*2 + 3*3 + 3*4 = 30$</p> <p>Total: $3 + 12 + 30 = 45$.</p> <p>Não há cópia rasa ou profunda no código.</p>
22	O recurso fala em operadores bitwise << e >>.	INDEFERIDO	<p>A questão real não contém operadores bitwise. Ela avalia entrada de dados, split(), strip(), fatiamento de string e divisão inteira.</p> <p>Com a entrada SETOR-145 ; 11:</p> <p>dados[0].split("-")[-1] obtém 145; [-2:] extrai 45; int(dados[1]) // 3 resulta em 3.</p> <p>A saída é 45-3.</p> <p>A alegação do recurso não corresponde ao conteúdo da questão.</p>
23	O recurso menciona lazy evaluation, generator e estado de iterador.	INDEFERIDO	<p>A questão real não possui generator. O item utiliza range, continue, soma acumulada e break.</p> <p>A soma considera os números de 2 a 10, ignorando múltiplos de 3:</p> <p>2, soma 2; 4, soma 6; 5, soma 11; 7, soma 18; 8, soma 26.</p> <p>Como a soma ultrapassa 20, o laço é interrompido. A saída é 26.</p> <p>Não há avaliação preguiçosa ou iterador previamente consumido.</p>
24	O recurso menciona herança múltipla e MRO.	INDEFERIDO	<p>A questão não possui classes, herança ou resolução de métodos. Ela avalia operações com lista: atribuição por índice, pop, insert, del e fatiamento.</p> <p>A lista inicial é [2, 4, 6, 8, 10].</p>

			<p>Após <code>fila[2] = fila[0] + fila[-1]</code>, fica [2, 4, 12, 8, 10]. Após <code>fila.insert(1, fila.pop(3))</code>, o valor 8 é removido do índice 3 e inserido no índice 1, resultando em [2, 8, 4, 12, 10]. Após <code>del fila[-2]</code>, remove-se 12, ficando [2, 8, 4, 10]. Por fim, <code>fila[1:]</code> resulta em [8, 4, 10].</p> <p>O item tem resposta única e não envolve MRO.</p>
25	O recurso menciona encoding de arquivo e leitura em ASCII.	INDEFERIDO	<p>A questão real não trabalha com arquivos. O item trata de matriz implementada por lista de listas.</p> <p>A matriz inicial é: [[0, 1, 2], [3, 4, 5]]</p> <p>A instrução: <code>matriz[0][2] = matriz[1][0] + matriz[0][1]</code> altera <code>matriz[0][2]</code> para $3 + 1 = 4$.</p> <p>Depois: <code>matriz[1].append(matriz[0][2] - matriz[1][1])</code> acrescenta $4 - 4 = 0$ à segunda linha.</p> <p>O cálculo final é: <code>matriz[0][2] + matriz[1][2] + matriz[1][3] = 4 + 5 + 0 = 9</code>.</p> <p>Não há arquivo, ASCII, acentuação ou incompatibilidade de encoding.</p>

DECISÃO

Dessa forma, indefere-se os recursos, mantendo-se o gabarito e a validade das questões impugnadas.

(Assinatura Eletrônica)

Rodrigo Torres de Araújo Lima

Secretário de Estado da Educação do Piauí



Documento assinado eletronicamente por **RODRIGO TORRES DE ARAUJO LIMA - Matr.443549-4, Secretário de Estado da Educação**, em 30/04/2026, às 20:06, conforme horário oficial de Brasília, com fundamento no Cap. III, Art. 14 do [Decreto Estadual nº 18.142, de 28 de fevereiro de 2019](#).



A autenticidade deste documento pode ser conferida no site https://sei.pi.gov.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0023866763** e o código CRC **823EDA8D**.

Processo SEI: 00011.006304/2026-07

Documento SEI: 0023866763

Criado por brendacastro@seduc.pi.gov.br, versão 10 por brendacastro@seduc.pi.gov.br em 30/04/2026 18:51:03.