



Instituto Federal do Espírito Santo – Campus Serra
Av. dos Sabiás – 330 – Morada de Laranjeiras – Serra – ES – 29166-630

Pós-graduação *stricto sensu*
Mestrado Profissional em
Computação Aplicada
Processo Seletivo de Alunos 2025-1
Caderno de Questões

Candidato(a): _____ CPF: _____

1 Instruções

Leia atentamente as instruções a seguir.

- 1) Você recebeu do fiscal o seguinte material:
 - (a) Este **caderno de questões**, com o enunciado das 30 (trinta) questões objetivas de múltipla escolha.
 - (b) O **cartão-resposta** destinado às respostas das questões objetivas formuladas nas provas.
- 2) Apenas o cartão-resposta será utilizado para correção. Nada que seja escrito no caderno de questões será utilizado ou considerado na correção da prova.
- 3) Após a conferência, o candidato deverá preencher o nome e o CPF, no espaço próprio do cartão-resposta, a caneta esferográfica na cor azul ou preta.
- 4) Para cada uma das questões objetivas são apresentadas 5 alternativas identificadas com as letras (a), (b), (c), (d), (e); só uma responde adequadamente à questão proposta. Você só deve assinalar **uma resposta**. A marcação em mais de uma alternativa anula a questão, mesmo que uma das respostas esteja correta.
- 5) Será eliminado do Processo Seletivo o candidato que:
 - (a) Abandonar o local de prova sem autorização;
 - (b) Ausentar-se do local de prova sem o acompanhamento do fiscal;
 - (c) Mantiver conduta incompatível com a condição de candidato ou for descortês com qualquer pessoa incumbida da realização do processo seletivo;
 - (d) For surpreendido, durante a realização da prova, em comunicação com outro candidato;
 - (e) Sair do local da prova sem entregar o cartão-resposta;
 - (f) Não permitir sua identificação;
 - (g) Não atender às determinações do edital 03/2025 e de seus atos complementares;

- (h) Utilizar, durante a realização das provas: chapéu, boné, livros, revistas, folhetos, impressos, anotações, calculadora ou quaisquer outros equipamentos eletrônicos de comunicação ou de consulta;
 - (i) Portar qualquer tipo de arma;
 - (j) Não assinar a lista de presença.
- 6) O candidato só poderá se ausentar do recinto da prova após 1 (uma) hora contada a partir do início da mesma.
- 7) Por motivos de segurança, o candidato só poderá levar o **caderno de questões**, depois de 2 (duas) horas contadas a partir de efetivo início da prova.
- 8) Recomenda-se que o candidato reserve os 20 (vinte) minutos finais para marcar seu cartão-resposta.
- 9) Quando terminar, entregue ao fiscal o cartão-resposta e assine a lista identificando que o cartão foi entregue.
- 10) O tempo disponível para esta prova de questões objetivas é de 3h00min (três horas), incluído o tempo para a marcação do seu cartão-resposta.

2 Questões de Inglês

Para responder as questões da prova de Inglês, considere o texto a seguir.

New training approach could help AI agents perform better in uncertain conditions

Sometimes, it might be better to train a robot in an environment that's different from the one where it will be deployed. A home robot trained to perform household tasks in a factory may fail to effectively scrub the sink or take out the trash when deployed in a user's kitchen, since this new environment differs from its training space. To avoid this, engineers often try to match the simulated training environment as closely as possible with the real world where the agent will be deployed. However, researchers from MIT and elsewhere have now found that, despite this conventional wisdom, sometimes training in a completely different environment yields a better-performing artificial intelligence agent.

Their results indicate that, in some situations, training a simulated AI agent in a world with less uncertainty, or "noise," enabled it to perform better than a competing AI agent trained in the same, noisy world they used to test both agents. The researchers call this unexpected phenomenon the indoor training effect. "If we learn to play tennis in an indoor environment where there is no noise, we might be able to more easily master different shots. Then, if we move to a noisier environment, like a windy tennis court, we could have a higher probability of playing tennis well than if we started learning in the windy environment," explains Serena Bono, a research assistant in the MIT Media Lab and lead author of a paper on the indoor training effect.

The researchers studied this phenomenon by training AI agents to play Atari games, which they modified by adding some unpredictability. They were surprised to find that the indoor training effect consistently occurred across Atari games and game variations. They hope these results fuel additional research toward developing better training methods for AI agents. "This is an entirely new axis to think about. Rather than trying to match the training and testing environments, we may be able to construct simulated environments where an AI agent learns even better," adds co-author Spandan Madan, a graduate student at Harvard University.

Bono and Madan are joined on the paper by Ishaan Grover, an MIT graduate student; Mao Yasueda, a graduate student at Yale University; Cynthia Breazeal, professor of media arts and sciences and leader of the Personal Robotics Group in the MIT Media Lab; Hanspeter Pfister, the An Wang Professor of Computer Science at Harvard; and Gabriel Kreiman, a professor at Harvard Medical School. The research will be presented at the Association for the Advancement of Artificial Intelligence Conference.

Training troubles

The researchers set out to explore why reinforcement learning agents tend to have such dismal performance when tested on environments that differ from their training space. Reinforcement learning is a trial-and-error method in which the agent explores a training space and learns to take actions that maximize its reward. The team developed a technique to explicitly add a certain amount of noise to one element of the reinforcement learning problem called the transition function. The transition function defines the probability an agent will move from one state to another, based on the action it chooses.

If the agent is playing Pac-Man, a transition function might define the probability that ghosts on the game board will move up, down, left, or right. In standard reinforcement learning,

the AI would be trained and tested using the same transition function. The researchers added noise to the transition function with this conventional approach and, as expected, it hurt the agent's Pac-Man performance. But when the researchers trained the agent with a noise-free Pac-Man game, then tested it in an environment where they injected noise into the transition function, it performed better than an agent trained on the noisy game. "The rule of thumb is that you should try to capture the deployment condition's transition function as well as you can during training to get the most bang for your buck. We really tested this insight to death because we couldn't believe it ourselves," Madan says.

Injecting varying amounts of noise into the transition function let the researchers test many environments, but it didn't create realistic games. The more noise they injected into Pac-Man, the more likely ghosts would randomly teleport to different squares. To see if the indoor training effect occurred in normal Pac-Man games, they adjusted underlying probabilities so ghosts moved normally but were more likely to move up and down, rather than left and right. AI agents trained in noise-free environments still performed better in these realistic games. "It was not only due to the way we added noise to create ad hoc environments. This seems to be a property of the reinforcement learning problem. And that was even more surprising to see," Bono says.

Exploration explanations

When the researchers dug deeper in search of an explanation, they saw some correlations in how the AI agents explore the training space. When both AI agents explore mostly the same areas, the agent trained in the non-noisy environment performs better, perhaps because it is easier for the agent to learn the rules of the game without the interference of noise. If their exploration patterns are different, then the agent trained in the noisy environment tends to perform better. This might occur because the agent needs to understand patterns it can't learn in the noise-free environment.

"If I only learn to play tennis with my forehand in the non-noisy environment, but then in the noisy one I have to also play with my backhand, I won't play as well in the non-noisy environment," Bono explains. In the future, the researchers hope to explore how the indoor training effect might occur in more complex reinforcement learning environments, or with other techniques like computer vision and natural language processing. They also want to build training environments designed to leverage the indoor training effect, which could help AI agents perform better in uncertain environments.

Fonte: <https://news.mit.edu/2025/new-training-approach-could-help-ai-perform-better-0129>

1. According to the text, why might a home robot struggle when deployed in a real kitchen?
 - (a) It lacks the necessary sensors for household tasks
 - (b) It was trained in a factory environment, which differs from a real kitchen
 - (c) It has limited processing power for real-world applications
 - (d) It was not programmed to interact with humans
 - (e) It does not have access to updated data

2. What conventional approach do engineers use when training AI?
 - (a) They expose the AI to as many different environments as possible
 - (b) They avoid training AI in environments with uncertainty
 - (c) They ensure the training environment closely matches the deployment environment
 - (d) They focus solely on theoretical training before real-world applications
 - (e) They use only physical environments, avoiding simulations

3. What surprising finding did researchers from MIT discover?
 - (a) AI can only function effectively when trained in real-world settings
 - (b) AI struggles more in factory settings than in household environments
 - (c) Robots learn faster when exposed to extreme uncertainty
 - (d) AI agents cannot adapt to different training methods
 - (e) AI trained in different environments sometimes performs better

4. What does the word “uncertainty” mean in the context of AI training?
 - (a) Lack of clear objectives
 - (b) Presence of unpredictable elements
 - (c) Inability to perform calculations
 - (d) Errors in programming
 - (e) Over-reliance on human intervention

5. Why might training AI in a world with less noise be beneficial?
 - (a) It helps the AI learn basic skills before handling real-world uncertainty
 - (b) It prevents the AI from being overwhelmed by data
 - (c) It makes AI training faster and more cost-effective
 - (d) It allows engineers to control all variables precisely
 - (e) It ensures AI models remain simple and easy to manage

6. What does the phrase “despite this conventional wisdom” imply?
- (a) The results confirm what was already expected
 - (b) Engineers have always agreed on this approach
 - (c) The researchers did not trust their own findings
 - (d) The findings challenge commonly held beliefs
 - (e) Conventional wisdom remains the best approach
7. Which of the following best describes the tone of the passage?
- (a) Persuasive and argumentative
 - (b) Informal and humorous
 - (c) Emotional and dramatic
 - (d) Skeptical and dismissive
 - (e) Objective and analytical
8. What is the primary purpose of the passage?
- (a) To persuade readers to invest in AI technology
 - (b) To criticize existing AI research methods
 - (c) To inform readers about new AI training discoveries
 - (d) To provide a history of AI development
 - (e) To entertain readers with AI-related anecdotes
9. Based on the text, what could be a potential implication of this research?
- (a) AI will no longer require human oversight
 - (b) Engineers will stop using simulation-based training
 - (c) Robots will only be used in controlled environments
 - (d) AI training methods may need to be re-evaluated
 - (e) AI will never be able to adapt to real-world tasks
10. What analogy do the researchers use to explain the indoor training effect?
- (a) A student learning mathematics in a quiet library versus a noisy classroom
 - (b) A driver learning to navigate city streets before driving on a highway
 - (c) A tennis player practicing in a wind-free indoor court before playing in windy conditions
 - (d) A musician rehearsing with headphones before performing on stage
 - (e) A chef preparing meals in a test kitchen before working in a busy restaurant

3 Questões de Conhecimentos Específicos

11. Considere o seguinte trecho de código em Python 3:

```
def calcular(n):  
    total = 0  
    for i in range(1, n + 1):  
        if i % 2 == 0:  
            total -= i  
        else:  
            total += i  
    return total  
print(calcular(5))
```

Ao executar esse programa, qual será o valor impresso na tela?

- (a) -3 (b) 0 (c) 3 (d) 5 (e) 15

12. Considere o seguinte trecho de código em Python 3:

```
def menor_m(x):  
    m = 0  
    soma = 0  
    while soma < x:  
        m += 1  
        soma += m  
    return m  
print(menor_m(20))
```

O código acima tem como objetivo determinar o menor número inteiro positivo m para o qual a soma dos números de 1 até m seja maior ou igual a x . Qual é o valor impresso na tela ao executar este programa?

- (a) 4 (b) 5 (c) 6 (d) 7 (e) 8

13. Analise o seguinte trecho de código em Python 3:

```
def encontrar_perfeito():  
    n = 1  
    while True:  
        soma = 0  
        for i in range(1, n + 1):  
            if n % i == 0:  
                soma += i  
        if soma == 2 * n:  
            return n  
        n += 1  
print(encontrar_perfeito())
```

Qual é a saída impressa pelo programa?

- (a) 4 (b) 6 (c) 8 (d) 12 (e) 28

14. Analise o seguinte trecho de código em Python 3:

```
def weighted_sum(lista, profundidade=1):
    total = 0
    for item in lista:
        if isinstance(item, list):
            total += weighted_sum(item, profundidade + 1)
        else:
            total += item * profundidade
    return total

print(weighted_sum([1, [2, [3, 4], 5], 6]))
```

O código implementa uma função recursiva que calcula a soma ponderada dos elementos de uma lista, onde cada número é multiplicado pela sua “profundidade” na estrutura aninhada. Qual é o valor impresso na tela ao executar este programa?

- (a) 36 (b) 38 (c) 42 (d) 44 (e) 46

15. Considere o seguinte trecho de código em Python 3:

```
def encontrar_numero(lista, num):
    for elemento in lista:
        if elemento == num:
            print("Número encontrado")
            break
    else:
        print("Número não encontrado")

encontrar_numero([2, 4, 6, 8], 5)
```

Qual será a saída deste programa?

- (a) Número encontrado
(b) Número não encontrado
(c) Nenhuma saída
(d) O programa gera um erro
(e) Número encontrado seguido de Número não encontrado

16. Considere o seguinte trecho de código em Python 3, que realiza uma busca em largura (BFS) em um grafo representado por um dicionário:


```
from collections import deque

def bfs(grafo, inicio):
    visitados = set()
    fila = deque([inicio])
    ordem = []
    while fila:
        vertice = fila.popleft()
        if vertice not in visitados:
            visitados.add(vertice)
            ordem.append(vertice)
            for vizinho in grafo[vertice]:
                if vizinho not in visitados:
                    fila.append(vizinho)
    return ordem
grafo = {
    'A': ['B', 'C'],
    'B': ['D', 'E'],
    'C': ['F'],
    'D': [],
    'E': ['F'],
    'F': []
}
print(bfs(grafo, 'A'))
```

Qual é a saída impressa pelo programa?

- (a) ['A', 'B', 'C', 'D', 'E', 'F']
- (b) ['A', 'C', 'B', 'F', 'D', 'E']
- (c) ['A', 'B', 'C', 'F', 'D', 'E']
- (d) ['A', 'C', 'B', 'D', 'E', 'F']
- (e) ['A', 'B', 'D', 'C', 'E', 'F']

17. Analise o seguinte trecho de código em Python 3:

```
def func(lista):
    pilha = []
    for x in lista:
        if not pilha or x >= pilha[-1]:
            pilha.append(x)
        else:
            while pilha and x < pilha[-1]:
                pilha.pop()
            pilha.append(x)
    return pilha
print(func([5, 3, 4, 8, 6, 7]))
```

Qual é a saída impressa pelo programa?

- (a) [5, 3, 4, 8, 6, 7]
- (b) [3, 4, 6, 7]
- (c) [5, 8, 6, 7]
- (d) [3, 4, 8]
- (e) [5, 8, 7]

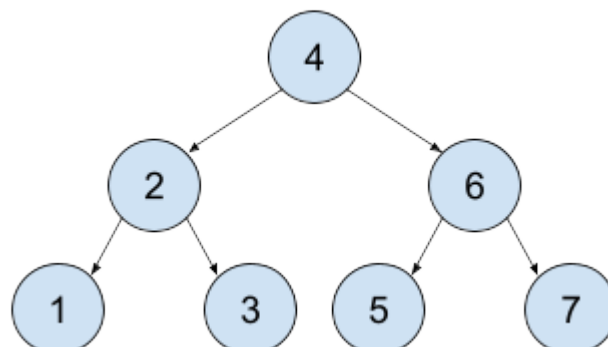
18. Considere o seguinte trecho de código em Python 3, que utiliza uma pilha para avaliar uma expressão pós-fixada (notação polonesa reversa):

```
def eval_postfix(expr):  
    stack = []  
    for token in expr.split():  
        if token.isdigit():  
            stack.append(int(token))  
        else:  
            b = stack.pop()  
            a = stack.pop()  
            if token == '+':  
                stack.append(a + b)  
            elif token == '-':  
                stack.append(a - b)  
            elif token == '*':  
                stack.append(a * b)  
            elif token == '/':  
                stack.append(a // b) # divisão inteira  
    return stack.pop()  
  
print(eval_postfix("5 1 2 + 4 * + 3 -"))
```

Qual é a saída impressa pelo programa?

- (a) 12
- (b) 13
- (c) 14
- (d) 15
- (e) 16

19. Qual das opções a seguir contém a sequência correta de inserção que gera a árvore binária de busca representada na figura a seguir?



- (a) 1, 2, 3, 4, 5, 6, 7
- (b) 7, 6, 5, 4, 3, 2, 1
- (c) 1, 3, 5, 7, 2, 6, 4
- (d) 4, 2, 6, 1, 3, 5, 7
- (e) 1, 2, 3, 5, 6, 7, 4

20. Analise o seguinte trecho de código em Python 3, que utiliza o módulo `heapq` para construir uma fila de prioridade contendo tuplas:

```
import heapq

def processar():
    heap = []
    dados = [(3, 'C'), (1, 'A'), (2, 'B'), (1, 'D'), (2, 'E')]
    for item in dados:
        heapq.heappush(heap, item)
    saida = []
    while heap:
        saida.append(heapq.heappop(heap))
    return saida
print(processar())
```

Qual é a saída impressa pelo programa?

- (a) [(1, 'A'), (1, 'D'), (2, 'B'), (2, 'E'), (3, 'C')]
- (b) [(1, 'D'), (1, 'A'), (2, 'B'), (2, 'E'), (3, 'C')]
- (c) [(3, 'C'), (2, 'B'), (2, 'E'), (1, 'A'), (1, 'D')]
- (d) [(1, 'A'), (2, 'B'), (1, 'D'), (2, 'E'), (3, 'C')]
- (e) [(1, 'A'), (2, 'B'), (2, 'E'), (3, 'C'), (1, 'D')]

21. Sobre algoritmos de busca, considere o seguinte código e assinale a alternativa correta:

```
def search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

- (a) Utiliza a estratégia dividir para conquistar, reduzindo o espaço de busca pela metade a cada iteração.
- (b) Implementa a Busca Linear, pois percorre os elementos um por um.
- (c) Funciona em listas não ordenadas sem necessidade de pré-processamento.
- (d) Tem complexidade de tempo $O(n)$ no pior caso.
- (e) Não funciona para listas de strings, apenas para números inteiros.

22. Analise o seguinte trecho de código em Python 3:

```
def busca_binaria(lista, chave):  
    esquerda = 0  
    direita = len(lista) - 1  
    while esquerda <= direita:  
        meio = (esquerda + direita) // 2  
        if lista[meio] == chave:  
            return meio  
        elif lista[meio] < chave:  
            esquerda = meio + 1  
        else:  
            direita = meio - 1  
    return -1  
  
dados = [15, 3, 9, 8, 5, 2]  
dados.sort()  
print(dados)  
pos = busca_binaria(dados, 8)  
print(pos)
```

Qual é a saída impressa pelo programa?

- (a) [2, 3, 5, 8, 9, 15]
3
- (b) [2, 3, 5, 8, 9, 15]
2
- (c) [15, 3, 9, 8, 5, 2]
3
- (d) [2, 3, 5, 8, 9, 15]
-1
- (e) [2, 3, 5, 8, 9, 15]
4

23. Qual dos seguintes algoritmos de busca é mais eficiente, em média, para encontrar um elemento em um conjunto ordenado de dados?

- (a) Busca Linear
- (b) Busca Binária
- (c) Busca Exponencial
- (d) Busca em Largura (BFS)
- (e) Busca por Tabela Hash

24. Sobre algoritmos de ordenação, qual das alternativas abaixo NÃO é um algoritmo de ordenação por comparação?

- (a) QuickSort
- (b) MergeSort
- (c) BubbleSort
- (d) Counting Sort
- (e) HeapSort

25. Qual das opções abaixo contém a implementação correta do algoritmo Insertion Sort em Python?

(a)

```
def insertion_sort(arr):  
    for i in range(len(arr) - 1):  
        key = arr[i]  
        j = i  
        while j > 0 and arr[j - 1] > key:  
            arr[j] = arr[j - 1]  
            j -= 1  
        arr[j] = key  
    return arr
```

(b)

```
def insertion_sort(arr):  
    for i in range(1, len(arr)):  
        key = arr[i]  
        j = i - 1  
        while j >= 0 and arr[j] > key:  
            arr[j + 1] = arr[j]  
            j -= 1  
        arr[j + 1] = key  
    return arr
```

(c)

```
def insertion_sort(arr):  
    for i in range(len(arr)):  
        j = i - 1  
        key = arr[j]  
        while j >= 0 and key > arr[j]:  
            arr[j + 1] = arr[j]  
            j -= 1  
        arr[j] = key  
    return arr
```

(d)

```
def insertion_sort(arr):  
    for i in range(len(arr)):  
        key = arr[i]  
        j = i - 1  
        while j >= 0 and arr[j] < key:  
            arr[j + 1] = arr[j]  
            j -= 1  
        arr[j + 1] = key  
    return arr
```

(e)

```
def insertion_sort(arr):  
    for i in range(1, len(arr)):  
        key = arr[i]  
        j = i  
        while j > 0 and arr[j] > key:  
            arr[j] = arr[j - 1]  
            j -= 1  
        arr[j] = key  
    return arr
```

26. Assinale a alternativa que corresponde à **negação** lógica da afirmação:

“Não há bolsas de estudo disponíveis e todo candidato deve aguardar.”

- (a) Há bolsas de estudo disponíveis ou alguns candidatos não devem aguardar.
- (b) Há bolsas de estudo disponíveis e alguns candidatos não devem aguardar.
- (c) Todas as bolsas de estudo estão disponíveis e todos candidatos não devem aguardar.
- (d) Todas as bolsas de estudo estão disponíveis ou todos candidatos não devem aguardar.
- (e) Há bolsas de estudo indisponíveis e todo candidato não deve aguardar.

27. Sabendo-se que:

Todos os pesquisadores são curiosos.

Todos os curiosos são estudiosos.

é necessariamente verdade que:

- (a) Todos os estudiosos são curiosos
- (b) Todos que não são pesquisadores não são curiosos.
- (c) Todos os curiosos são pesquisadores
- (d) Todos que não são estudiosos não são pesquisadores
- (e) Alguns estudiosos não são curiosos.

28. A proposição “Se o modelo não é adequado então os dados não são confiáveis” equivale a:

- (a) O modelo não é adequado e os dados são confiáveis
- (b) O modelo não é adequado ou os dados não são confiáveis
- (c) Se os dados são confiáveis então o modelo é adequado
- (d) Se o modelo é adequado, então os dados são confiáveis
- (e) O modelo não é adequado se e somente se os dados não são confiáveis

29. Sabendo-se que valor lógico da proposição $p \wedge q \rightarrow r$ é falso, pode-se afirmar que é necessariamente verdadeiro o valor lógico da proposição:

- (a) $p \wedge q \wedge r$
- (b) $p \wedge q \leftrightarrow \sim r$
- (c) $\sim (p \wedge q) \vee r$
- (d) $\sim r \rightarrow \sim p \vee \sim q$
- (e) $p \rightarrow q \wedge r$

30. Sejam p , q e r proposições. Lembrando que, para qualquer q e r , vale a equivalência

$$q \rightarrow r \equiv \sim q \vee r.$$

Considere as fórmulas:

$$F_1 : (p \wedge q) \rightarrow r,$$

$$F_2 : p \rightarrow (q \rightarrow r).$$

Qual das alternativas abaixo está correta?

- (a) $F_1 \equiv F_2$, pois ambas são equivalentes a $\sim p \vee \sim q \vee r$.
- (b) $F_1 \not\equiv F_2$, pois F_1 é equivalente a $\sim p \vee \sim q \vee r$, enquanto F_2 é equivalente a $\sim p \vee (\sim q \wedge r)$.
- (c) F_1 implica F_2 , mas a recíproca não é válida.
- (d) F_2 implica F_1 , mas a recíproca não é válida.
- (e) Nenhuma das anteriores.