

# Prova de Seleção

**Programa de Pós-Graduação em Informática (PPGI)**  
**Processo seletivo de 2026/1**

---

## Instruções para a prova

- A prova contém **20 questões** objetivas com cinco alternativas cada. Para cada questão, **existe apenas uma alternativa correta**.
- O tempo total para a realização desta prova é de **2 horas**. Um aviso será dado 30 minutos antes do término.
- Você receberá um cartão de respostas oficial para marcar suas respostas. Preencha-o utilizando **caneta esferográfica preta ou azul**.
- Marque apenas **uma resposta por questão**. Caso haja marcações múltiplas, a questão será anulada.
- É **proibido** o uso de celulares, smartwatches ou qualquer outro dispositivo eletrônico durante a prova.
- Guarde **todos** os dispositivos **desligados** e fora do alcance (por exemplo, em sua mochila ou bolsa).
- Apenas canetas, lápis, borracha, documento de identidade e o material fornecido pela organização da prova são permitidos sobre a mesa.
- Caso tenha dúvidas durante a prova, levante a mão e aguarde a presença de um fiscal.
- Os fiscais não estão autorizados a esclarecer questões, apenas dúvidas relacionadas ao processo.
- Ao finalizar, entregue o caderno de questões e o cartão de respostas ao fiscal. Confira se preencheu corretamente todos os campos obrigatórios no cartão de resposta antes de entregá-lo. Lembre-se também de assinar a lista de entrega.
- Mantenha silêncio durante toda a prova para não atrapalhar os colegas.
- Qualquer tentativa de comunicação ou comportamento inadequado dentro, ou fora do local de prova poderá resultar na desclassificação.

**Boa prova!**

---

**Questão 1:** Considere que os enunciados a seguir são simultaneamente verdadeiros:

- I. Se Luc termina a graduação, então faz concurso para o magistério ou trabalha no banco Estrela do Norte;
- II. Se não é verdade que Luc não faz o concurso para o magistério e não trabalha no banco Estrela do Norte, então Luc reside na fazenda do seu pai;
- III. Se Luc trabalha no Haras, então participa do campeonato;
- IV. Não é verdade que, Luc não termina a graduação e não trabalha no Haras;
- V. Se Luc não treina com os cavalos, então não participa do campeonato.

Assinale qual das opções a seguir, representa o enunciado que é consequência lógica dos enunciados apresentados (de I a V):

- a) Luc reside na fazenda do seu pai
- b) Luc reside na fazenda do seu pai e não treina com os cavalos
- c) Se Luc não reside na fazenda do seu pai, então treina com os cavalos
- d) Luc participa do campeonato e termina a graduação
- e) Nenhuma das respostas anteriores

**Questão 2:** Considere que os enunciados a seguir são verdadeiros:

- Helena gosta de Liam.
- Quem gosta de Liam gosta de Gil.
- Helena gosta apenas de quem é educado.

Logo, podemos concluir logicamente que:

- a) Liam não é educado
- b) Gil não gosta de Liam
- c) Helena não gosta de Gil
- d) Gil é educado
- e) Nenhuma das respostas anteriores

**Questão 3:** Considere o enunciado:

“Nenhum ator gosta de todas as novelas”

Qual das opções a seguir representa o enunciado equivalente:

- a) Alguns atores gostam de pelo menos uma novela
- b) Todos os atores gostam de todas as novelas
- c) Para todo ator, então há ao menos uma novela da qual ele não gosta;
- d) Não é o caso que, para todo ator, há ao menos uma novela da qual ele gosta.
- e) Nenhuma das respostas anteriores

**Questão 4:** Dado o enunciado:

“Nem todo computador é eficiente.”

Assinale qual das alternativas apresenta a negação do enunciado dado:

- a) Existem computadores que não são eficientes.
- b) Nenhum computador é eficiente.
- c) Existem computadores que são eficientes.
- d) Nem todos os computadores não são eficientes.
- e) Não existem computadores que não sejam eficientes.

**Questão 5:** Considere as seguintes premissas como verdadeiras:

- 1.  $p \rightarrow (q \vee r)$
- 2.  $\sim r \rightarrow \sim s$
- 3.  $s \vee \sim t$
- 4.  $(\sim t \vee u) \rightarrow \sim q$

Deseja-se acrescentar uma quinta premissa, também considerada verdadeira, e propor uma conclusão, de modo que o argumento formado pelas premissas 1, 2, 3, 4, 5 seja dedutivamente válido.

Assinale a alternativa cujo acréscimo (premissa + conclusão) torna o argumento logicamente válido.

- a. Premissa 5:  $q$   
Conclusão:  $p$
- b. Premissa 5:  $\sim r$   
Conclusão:  $\sim p$
- c. Premissa 5:  $\sim p$   
Conclusão:  $s$
- d. Premissa 5:  $t$   
Conclusão:  $q$
- e. Nenhuma das respostas anteriores.

**Questão 6:** Considere uma pilha que opera com números naturais (1, 2, 3, etc) e com operadores pós-fixados. Os números são simplesmente empilhados. Entre os operadores, existem:

- **add** (adição) e **mul** (multiplicação), que são binários e utilizam os dois elementos no topo da pilha, substituindo-os pelo resultado da operação;
- **pop**, que é unário e remove o elemento do topo da pilha.

Sabendo que o topo da pilha corresponde ao lado direito da notação [ ... ]. Considere a sequência de operandos e operadores:

```
2 3 mul 2 add 1 pop 1 pop 3 add 2 mul 3 4 add 3 4 pop mul add
```

Qual é o conteúdo final da pilha após processar a sequência acima?

- a) [ 46 ]
- b) [ ] (pilha vazia)
- c) [ 42 ]
- d) [ 43 ]
- e) [ 6 18 ]

**Questão 7:** Em relação aos mecanismos avançados da Programação Orientada a Objetos (POO), analise as afirmativas a seguir:

- I. Em linguagens que suportam herança múltipla, como C++, o problema do diamante (*diamond inheritance*) ocorre quando uma classe herda de duas classes que possuem uma mesma classe-base, podendo gerar ambiguidade na resolução dos membros herdados.
- II. O princípio de substituição de Liskov (LSP) estabelece que uma classe derivada deve poder substituir sua classe base sem alterar o comportamento correto do programa, garantindo que métodos sobrescritos não restrinjam pré-condições nem ampliem pós-condições.
- III. O conceito de coesão está relacionado ao grau de dependência entre módulos, indicando que módulos altamente coesos tendem a exigir forte acoplamento para manter consistência comportamental.
- IV. No polimorfismo paramétrico, como ocorre com templates em C++ ou *generics* em Java, a verificação de tipos é sempre feita apenas em tempo de execução, permitindo maior flexibilidade e reduzindo erros de compilação.

Está correto o que se afirma **apenas** em:

- a) I e II
- b) I, II e III
- c) II e IV
- d) I e III
- e) III e IV

**Questão 8:** Considere as afirmativas abaixo sobre estrutura de dados:

- I. Em uma lista duplamente encadeada, cada nó armazena referências tanto para o próximo quanto para o nó anterior, permitindo percorrer a lista em ambas as direções.
- II. Uma das principais aplicações práticas de uma Fila é o controle de chamadas de funções recursivas em um sistema operacional, onde a última função chamada deve ser a primeira a ser finalizada.
- III. Em uma fila padrão, tanto as operações de inserção quanto as de remoção podem ser realizadas em qualquer extremidade da estrutura sem alterar seu funcionamento correto.
- IV. Em uma pilha, é possível acessar diretamente qualquer elemento pelo seu índice em tempo constante, da mesma forma que ocorre em um array.

É correto afirmar que:

- a) Somente a II e IV estão corretas
- b) Somente a IV está correta
- c) Somente a I está correta
- d) Somente a II está correta
- e) Todas estão incorretas

**Questão 9:** Considere as afirmativas abaixo sobre tabelas hash e suas características de desempenho:

- I. A busca em uma tabela hash possui complexidade média  $O(1)$  quando a função de hash distribui bem os elementos.
- II. No pior caso, a busca em uma tabela hash pode ter complexidade  $O(n)$  devido a colisões.
- III. O uso de endereçamento aberto garante que o pior caso da busca nunca ultrapasse  $O(\log n)$ .
- IV. Em uma tabela hash, aumentar o fator de carga (*load factor*) sempre melhora o desempenho da busca.

É correto afirmar que:

- a) Somente a afirmativa I e II estão corretas
- b) Somente a afirmativa I e III estão corretas
- c) Somente as afirmativas I, II e IV estão corretas
- d) Somente as afirmativas II e IV estão corretas
- e) Todas as afirmativas estão corretas

**Questão 10:** Analise o seguinte trecho de código escrito em C:

```
int valor;

int soma (int x, int *y) {
    int z = x + *y
    return z
}

int main () {
    valor = 10;
    int a = 20;
    soma (valor, &a);
    return 0;
}
```

Avalie as seguintes afirmativas:

- I. A variável `valor` é alocada na memória *heap* e possui escopo global.
- II. A variável `z` é alocada na Pilha (*stack*) e será liberada da memória assim que a função `soma` encerrar a sua execução.
- III. Na função `soma`, um parâmetro é passado por cópia e outro por referência de memória.

É correto afirmar que:

- a) Todas as afirmativas estão corretas
- b) Nenhuma das afirmativas estão corretas
- c) Somente as afirmativas I e III estão corretas
- d) Somente as afirmativas II e III estão corretas
- e) Somente as afirmativas I e II estão corretas

**Questão 11:** Considere as afirmativas abaixo sobre os principais conceitos de Programação Orientada a Objetos (POO):

- I. **Herança** permite que uma classe derive atributos e métodos de outra classe, promovendo reutilização de código.
- II. **Encapsulamento** consiste em ocultar detalhes internos de uma classe, permitindo acesso controlado aos seus atributos e métodos.
- III. **Polimorfismo** permite que um mesmo método possa se comportar de maneiras diferentes, dependendo do objeto que o invoca.
- IV. **Abstração** corresponde ao ato de expor apenas os detalhes essenciais de um objeto, escondendo características irrelevantes.
- V. Em POO, **uma classe pode herdar de múltiplas classes simultaneamente**, independentemente da linguagem utilizada.

Em relação às afirmações:

- a) Todas estão corretas

- b) Todas estão incorretas
- c) Só existe uma afirmação incorreta
- d) Só existe uma afirmação correta
- e) Existem três afirmações corretas e duas incorretas

**Questão 12:** Considere o grafo não orientado  $G = (V, E)$ , cujas arestas são:

- A – B
- A – D
- B – C
- D – E

Deseja-se aplicar um algoritmo de Busca em Largura (BFS) para explorar sistematicamente os vértices deste grafo. Considere que, sempre que houver mais de um vértice adjacente disponível para expansão, o algoritmo deve selecionar os vizinhos em ordem alfabética. Com base nessas condições, a sequência correta de visita dos vértices realizada pelo algoritmo BFS é:

- a) A, B, C, D, E
- b) A, D, E, B, C
- c) A, B, D, C, E
- d) A, C, B, D, E
- e) A, B, D, E, C

**Questão 13:** Sobre as propriedades dos algoritmos de ordenação a seguir, assinale a alternativa correta.

É correto afirmar que:

- a) O Bubble Sort é eficiente para grandes volumes de dados porque sempre executa em tempo  $O(n)$
- b) O Selection Sort é um algoritmo in-place e tem complexidade  $O(n^2)$  no pior caso
- c) O Merge Sort é um algoritmo in-place, pois não utiliza memória auxiliar adicional.
- d) O Insertion Sort possui complexidade  $O(n^3)$  no pior caso e não é in-place.
- e) O Heap Sort é estável e possui complexidade  $O(n)$  no pior caso.

**Questão 14:** Considere os seguintes trechos de pseudocódigo:

1	para i de 1 até n: imprimir(i)
2	para i de 1 até n: para j de 1 até n: imprimir(i, j)
3	imprimir("Olá")
4	para i de 1 até n: k = n enquanto k > 1: k = k / 2

Qual a alternativa que indica corretamente a complexidade de tempo de cada um?

- a) 1:  $O(n)$  | 2:  $O(n^2)$  | 3:  $O(1)$  | 4:  $O(n \log n)$
- b) 1:  $O(n)$  | 2:  $O(n^2)$  | 3:  $O(1)$  | 4:  $O(\log n)$
- c) 1:  $O(n^2)$  | 2:  $O(n)$  | 3:  $O(1)$  | 4:  $O(n \log n)$
- d) 1:  $O(n)$  | 2:  $O(n)$  | 3:  $O(n^2)$  | 4:  $O(\log n)$
- e) 1:  $O(n)$  | 2:  $O(n)$  | 3:  $O(n)$  | 4:  $O(n)$

**Questão 15:** Considere o trecho de código a seguir, escrito em linguagem C:

```
int partition(Item *a, int lo, int hi) {
    int i = lo, j = hi+1;
    Item aux, v = a[lo];
    while(1) {
        while (less(a[++i], v))
            if (i == hi) break;
        while (less(v, a[--j]))
            if (j == lo) break;
        if (i >= j) break;
        swap(&a[i], &a[j]);
    }
    swap(&a[lo], &a[j]);
    return j;
}
```

Considere que:

- A função `less(x, y)` retorna verdadeiro se `x` for menor que `y`;
- A função `swap(x, y)` realiza a troca dos valores de duas variáveis do tipo `Item`.

Analise as alternativas a seguir:

- I. A função `partition` é referente a partição do algoritmo *Quick Sort*, em que a cada execução, o pivô é escolhido (`a[lo]`), todos os elementos

menores e maiores são reposicionados no vetor e o pivô é colocado na posição final correta na ordenação.

- II. A função `partition` tem comportamento linear, varrendo todo o vetor de entrada `a` cada vez que é executada.
- III. Nesta implementação, a função `partition` continua as varreduras da esquerda para a direita (variável `i`), e da direita para a esquerda (variável `j`) enquanto as chaves são iguais a `a[lo]`.

É correto afirmar que:

- a) Todas as afirmativas estão corretas
- b) Nenhuma das afirmativas estão corretas
- c) Somente a afirmativa I e II estão corretas
- d) Somente as afirmativas II e III estão corretas
- e) Somente a alternativa I está correta

**Questão 16:** Considere as seguintes afirmações sobre paradigmas de programação:

- I. Linguagens orientadas a objetos normalmente utilizam conceitos como classes, objetos, herança e encapsulamento.
- II. Na programação funcional, é comum evitar estados mutáveis e preferir funções que não produzem efeitos colaterais.
- III. A programação procedural é uma forma de programação imperativa, na qual o foco está em procedimentos que operam sobre dados.
- IV. Linguagens multiparadigma permitem ao desenvolvedor adotar mais de um estilo de programação dentro da mesma linguagem.

Qual alternativa apresenta apenas afirmações corretas?

- a) Apenas as afirmações I e III
- b) Apenas as afirmações II, III e IV
- c) Apenas as afirmações I, II e IV
- d) Todas as afirmações
- e) Nenhuma das afirmações

**Questão 17:** Considere o trecho de código a seguir escrito em C:

```
#include <stdio.h>

struct P {
    int x;
    int y;
};

int main() {
    struct P v[3] = { {1, 2}, {3, 4}, {5, 6} };
    int soma = 0;

    for (int i = 0; i < 3; i++)
        soma += v[i].x + v[i].y;
}
```

```

soma += v[i].x + v[2 - i].y;

printf("%d", soma);
return 0;
}

```

Quase será a saída produzida por este programa?

- a) 18
- b) 21
- c) 24
- d) 27
- e) 30

**Questão 18:** Considere a equação a seguir:

$$y_i = f \left( \sum_{j=0}^{n-1} W[j] \cdot X[i][j] + b \right), \quad \text{onde } f(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

Considerando que  $X \in R^{m \times n}$ ,  $W \in R^n$  e  $y \in R^m$  deseja-se implementar a função:

```
void forward(int m, int n, double X[m][n], double W[n], double b, int y[m]);
```

Qual das alternativas a seguir contém a implementação correta dessa função?

a)

```

void forward(int m, int n, double X[m][n], double W[n], double b, int y[m])
{
    for (int i = 0; i < m; i++)
        y[i] = (X[i][0] * W[0] > 0);
}

```

b)

```

void forward(int m, int n, double X[m][n], double W[n], double b, int y[m])
{
    for (int i = 0; i < m; i++) {
        double z = 0;
        for (int j = 0; j < n; j++)
            z += X[i][j] + W[j];
        y[i] = (z >= 0);
    }
}

```

c)

```

void forward(int m, int n, double X[m][n], double W[n], double b, int y[m])
{
    for (int i = 0; i < m; i++) {

```

```

        double z = b;
        for (int j = 0; j < n; j++)
            z += W[j] * X[i][j];
        y[i] = (z >= 0 ? 1 : 0);
    }
}

```

d)

```

void forward(int m, int n, double X[m][n], double W[n], double b, int y[m])
{
    for (int j = 0; j < n; j++) {
        double z = b + W[j];
        y[j] = (z >= 0);
    }
}

```

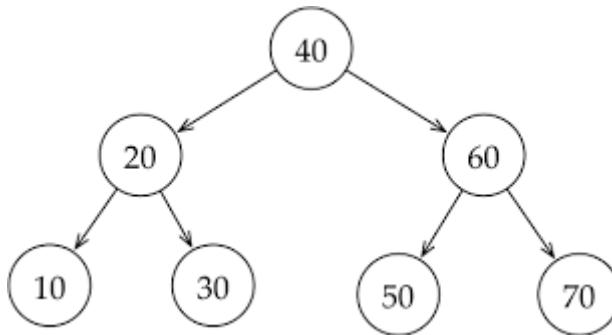
e)

```

void forward(int m, int n, double X[m][n], double W[n], double b, int y[m])
{
    for (int i = 0; i < m; i++)
        y[i] = (b >= 0);
}

```

**Questão 19:** Considere a estrutura de dados ilustrada abaixo e as afirmações:



- I. A estrutura trata de uma árvore B, estrutura clássica de estruturas de dados que tem grau  $m$ , e, portanto, no máximo  $m-1$  chaves e  $m$  filhos, com  $m \geq 1$ .
- II. O tempo de execução de operações de busca e inserção, nesta estrutura com configuração balanceada, é proporcional a sua altura -  $O(\log_2 n)$ , com  $n$  sendo o número de elementos da estrutura.
- III. Esta estrutura de dados pode se degenerar, tendo comportamento linear quando todas as inserções acontecem de forma ordenada, crescente ou decrescente.

É correto afirmar que:

- a) Todas as afirmativas estão corretas
- b) Somente II e III estão corretas

- c) Somente a afirmativa III está correta
- d) Somente a afirmativa I está correta
- e) Nenhuma das afirmações está correta

**Questão 20:** Considere o seguinte trecho de pseudocódigo que representa uma relação de herança entre classes:

```
Classe Animal:  
    atributo nome  
    método falar()
```

```
Classe Cachorro herda de Animal:  
    método falar() sobrescreve o método da classe Animal
```

```
Classe Gato herda de Animal:  
    método falar() sobrescreve o método da classe Animal
```

Considere as afirmações a seguir:

- I. Objetos das classes Cachorro e Gato podem acessar os atributos definidos em Animal
- II. Um objeto da classe Animal pode ser usado onde se espera um objeto do tipo Cachorro, pois Animal contém todas as características de Cachorro
- III. O método falar() de Cachorro e Gato substitui a versão original definida em Animal
- IV. A herança permite reutilizar código, pois Cachorro e Gato não precisam redefinir o atributo nome
- V. A herança implica que Animal é uma subclasse de Cachorro e Gato

Qual alternativa descreve corretamente as afirmações verdadeiras?

- a) Apenas I, III e IV
- b) Apenas II e V
- c) Apenas I e V
- d) Apenas I, II e IV
- e) Apenas III e V