

Eliminado: <object>

PROMETEO

# Test Programación Unidad 3

## PREGUNTAS

### Pregunta 1

¿Cuál es la función principal de una estructura if en Java?

- A) Repetir un bloque de código indefinidamente
- B) Evaluar una expresión booleana y ejecutar código solo si es verdadera
- C) Saltar a una etiqueta específica en el programa
- D) Crear una nueva variable con un valor condicional

### Pregunta 2

¿Cuál es la diferencia fundamental entre if-else e if simple?

- A) if-else es más rápido que if simple
- B) if-else solo funciona con números
- C) if-else proporciona una alternativa para cuando la condición es falsa
- D) if-else puede evaluar múltiples condiciones simultáneamente

### Pregunta 3

¿Cuándo es más apropiado usar una cadena if-else-if en lugar de múltiples if independientes?

- A) Cuando necesitas evaluar múltiples condiciones mutuamente excluyentes de forma ordenada
- B) Cuando quieres que todas las condiciones se evalúen siempre
- C) Cuando necesitas mejor rendimiento de compilación
- D) Cuando trabajas con valores de punto flotante

### Pregunta 4

¿En qué situación es más recomendable usar switch en lugar de if-else?

- A) Cuando necesitas comparar una variable contra múltiples valores específicos y discretos

Eliminado: <object>

## PROMETO

- B)** Cuando tienes solo dos opciones
- C)** Cuando trabajas con datos de texto muy largos
- D)** Cuando necesitas rendimiento máximo sin importar legibilidad

### Pregunta 5

¿Cuál es la principal ventaja de las switch expressions introducidas en Java 14?

- A)** Permiten eliminar la necesidad de break statements
- B)** Son mucho más rápidas que switch tradicionales
- C)** Funcionan solo con números enteros
- D)** Permiten más de 100 casos simultáneamente

### Pregunta 6

Si ejecutas el siguiente código:

```
int i = 5;
while(i < 3) {
    System.out.println(i);
    i++;
}
```

¿Cuántas veces se ejecutará el bloque de código?

- A)** 0 veces
- B)** 2 veces
- C)** 5 veces
- D)** Infinitamente

### Pregunta 7

¿Cuál es la característica principal que distingue do-while de while?

- A)** do-while es más rápido
- B)** do-while garantiza que el código se ejecute al menos una vez
- C)** do-while no necesita llaves {} para envolver el código
- D)** do-while no puede tener incrementos

### Pregunta 8

¿Cuál es el contexto ideal para usar un bucle for en lugar de while?

Eliminado: <object>

## PROMETO

- A)** Cuando no sabes cuántas iteraciones necesitas
- B)** Cuando conoces el número exacto de iteraciones o trabajas con rangos específicos
- C)** Cuando necesitas leer un archivo línea por línea
- D)** Cuando quieres garantizar al menos una ejecución

### Pregunta 9

¿Cuál es la ventaja principal del bucle for-each (introducido en Java 5) sobre un for tradicional?

- A)** Es significativamente más rápido
- B)** Simplifica la iteración sobre colecciones y evita errores como IndexOutOfBoundsException
- C)** Permite modificar elementos de la colección durante la iteración
- D)** Solo funciona con arrays

### Pregunta 10

¿Cuál es la recomendación general de buenas prácticas para limitar la profundidad de anidamiento en estructuras de control?

- A)** No hay límite, cuanto más anidamiento mejor
- B)** Máximo 1 nivel de anidamiento
- C)** No más de 3-4 niveles, extrayendo lógica a métodos separados cuando sea necesario
- D)** Máximo 10 niveles de anidamiento

## HOJA DE RESPUESTAS

Pregunta	Respuesta Correcta	Justificación
1	<b>B</b>	if evalúa una condición booleana y ejecuta código solo si es verdadera
2	<b>C</b>	if-else proporciona una alternativa (else) cuando la condición es falsa

Eliminado: <object>

## PROMETO

3	<b>A</b>	if-else-if evalúa múltiples condiciones mutuamente excluyentes de forma ordenada
4	<b>A</b>	switch es óptimo para comparar contra múltiples valores específicos y discretos
5	<b>A</b>	Las switch expressions eliminan la necesidad de break, haciendo código más seguro y conciso
6	<b>A</b>	Como i=5 no es menor que 3, la condición es falsa desde el inicio, nunca se ejecuta
7	<b>B</b>	do-while garantiza al menos una ejecución antes de evaluar la condición
8	<b>B</b>	for es ideal cuando conoces el número exacto de iteraciones o trabajas con rangos
9	<b>B</b>	for-each simplifica iteración sobre colecciones y evita errores de índice comunes
10	<b>C</b>	La recomendación es máximo 3-4 niveles de anidamiento para mantener legibilidad