



Comprobación de lectura – Ciclos en C#

Nombre Jimena Hernández Revisa Fátima López

Punteo 65

Instrucciones: Responder de manera clara y concisa cada enunciado.

- 1 ¿Cuál es la diferencia principal entre las instrucciones for y foreach en C#?
- 2 ¿Cómo se puede salir anticipadamente de un bucle for o while en C#?
- 3 ¿Qué sucede si la condición en un bucle while nunca se evalúa como false?
- 4 Explique el propósito de la instrucción continue dentro de un bucle en C#.
- 5 ¿Cuál es la diferencia entre las instrucciones do y while en C#?
- 6 ¿Cómo se puede iterar sobre una colección personalizada utilizando foreach en C#?
- 7 ¿Qué es un iterador en C# y cómo se implementa?
- 8 ¿Qué sucede si se modifica la colección sobre la que se está iterando con un bucle foreach?
- 9 ¿Es posible anidar múltiples bucles for en C#? Proporcione un ejemplo.
- 10 ¿Cómo se puede utilizar un bucle for para iterar sobre una colección que no tiene un índice basado en enteros?
- 11 ¿Qué consideraciones de rendimiento se deben tener en cuenta al elegir entre for y foreach?
- 12 ¿Cómo se puede implementar una iteración personalizada en una clase en C#?

- 1.) El **for** ejecuta el cuerpo mientras se evalúa como true y el **foreach** enumera los elementos de la colección y trabaja para cada elemento de la colección. ✓
- 2.) Mediante **break**; ✓
- 3.) No podrá ejecutarse, porque el **while** debe de ser true para ejecutarse.
Error en tiempo de ejecución de bucle infinito
- 4.) La función de paso a paso a la siguiente iteración. 1/2 X
Falló: Omite resto de código
- 5.) **Do** = El **do** se ejecuta 1 o más veces
While = El **while** se ejecuta 0 o más veces. ✓
- 6.) ~~foreach (int number in numbers)~~ **foreach (int numbers in elements)** X
Console.WriteLine ("elementos") 1/2
- 7.) Es un bloque que indica que se repite n veces → **++i, ++i, i--, --i**. ✓
Es el bloque que ejecuta una repetición toda vez se ejecute la condición.

8.) Este no podrá ejecutarse, ya que el foreach lee y ejecuta cada elemento establecido, entonces no podrá ejecutarse. ✓
en la colección Error específico → Invalid operation

9.) Sí.

```
for (int i = 0; i < 5; i++) {  
    for (int j = 0; j < 10; j++) {  
    }  
}
```

 X

10.) Con la palabra clave = VAR X Enumerator o foreach

```
for (var = 0; var < 3; var++)
```

11.) El foreach ayuda a trabajar en una colección de datos entonces este es más específico. X 1/2
El for es más eficiente en cuanto memoria, el foreach es más seguro por su legibilidad.

12.) X Ejemplos en el chat de Teamj