

1. ¿Cuál de las estructuras iterativas existentes en C++ evalúa la condición luego de ejecutar la instrucción?

R= do .. while .

2. ¿Existe en C++ el ciclo Repetir? ¿Cuál es el equivalente y cómo funciona?

R= Si existe, se llama do .. while. Básicamente ejecuta todo un bloque de comando hasta que la condición dada sea falsa, y esta condición es evaluada luego de cada iteración.

3. ¿Una iteración realizada con while es equivalente a una iteración realizada con do .. while? Describa en qué se asemejan y en qué se diferencian.

R= si son equivalentes pero la única diferencia es que el while evalúa la condición al principio y el do..while al final, y el do..while se ejecuta al menos una vez.

4. Analice el siguiente ciclo, describa que ocurre con él en C++ y, si hay un error, indique cuál o cuáles errores se presentan y cómo debería corregirse para que el ciclo funcione:

```
for(i=1; i<=n; i--){  
cout << "El cuadrado de i es: " << i*i << endl;  
}
```

R= si n es mayor a 1 es un ciclo infinito, debido a que la condición de repetición es mientras que i<=n y siempre se le resta 1 a la i. para acomodarlo en ese caso sería sumarle 1 a la i o cambiar la condición a i>=n;

5. Analice el siguiente ciclo, describa que ocurre con él en C++ y, si hay un error, indique cuál o cuáles errores se presentan y cómo debería corregirse para que el ciclo funcione:

```
i=10;  
do {  
cout << "Suministre un valor para n" << endl;  
cin >> n;  
m = n / i;  
i--;  
} while (i > 0);
```

R= el ciclo no posee ningún error, se va a repetir 10 veces pidiendo al usuario que ingrese un número y le asigna a m ese número dividido por la i.

6. Los ciclos, al igual que los condicionales, trabajan evaluando expresiones que sólo utilizan los operadores de comparación = y  $\neq$

R= Falso, también aceptan operadores < y >. Debido a que son condiciones y esos conectores también devuelven un valor de verdad.