

## Sesión 3

### 1. ¿Qué diferencia hay entre while y for?:

- a. El bucle for puede no llegar a ejecutarse nunca pero el while siempre se ejecuta al menos una vez.
- b. El bucle for se ejecuta un número determinado de veces y el while un número indeterminado de veces.
- c. El bucle for no puede convertirse en un bucle while pero sí al contrario.
- d. El bucle while permite su inicialización pero el bucle for no.

### 2. De acuerdo a la sintaxis del bucle while:

```
while (condición) {  
    sentencias  
}
```

¿qué es falso en relación con la condición?

- a. La condición es una variable booleana.
- b. La condición sólo se evalúa al principio de la ejecución del bucle.
- c. Si la condición es verdadera, se ejecuta el bloque de sentencias, y se vuelve al principio del bucle.
- d. Si la condición es falsa, no se ejecuta el bloque de sentencias.

### 3. De acuerdo a la sintaxis del bucle do-while:

```
do{  
    sentencias  
} while (condición)
```

¿Cuál es la afirmación falsa?

- a. Si condición == true, entonces el bucle se sigue ejecutando.
- b. Aunque se cumpla condición == false, el bucle se llega a ejecutar alguna vez.
- c. Si condición == false, el bucle no se llega a ejecutar nunca.
- d. Ninguna de las anteriores es falsa.

**4. Transforma este bucle for en su equivalente bucle while.**

```
for( a=0,b=0; a < 7; a++,b+=2 )
```

**5. De acuerdo a la sintaxis del bucle for**

```
for (inicialización ; condición ; actualización) {  
    sentencias  
}
```

¿Cuál es la afirmación falsa?

- a. La inicialización se realiza cada vez antes de entrar al bucle.
- b. La condición se comprueba cada vez antes de entrar al bucle.
- c. La actualización se realiza siempre al terminar de ejecutar la iteración.
- d. La inicialización, condición y actualización son elementos "opcionales".

**6. ¿Cuál de estas características de las sentencias "break" y "continue" es "falsa"?**

- a. La sentencia break permite salirnos de un bucle que está en ejecución, sin embargo, "continue sigue ejecutando las siguientes iteraciones del bucle.
- b. La sentecias break y continue pueden usarse en tanto en sentencias de repetición (bucles) como de bifurcación (switch).
- c. Después de ejecutarse la sentencia continue en un bucle, se vuelve a evaluar la condición del bucle.
- d. La sentencia break termina la ejecución de una sentencia de bifurcación (switch).

**7. ¿ Cuántas veces se ejecuta el cuerpo del siguiente bucle ?**

```
int i = 0;  
while (true) {  
    i ++;  
    if (i <10)  
        continue;  
    i ++;
```

```
if (i== 10)  
    break;  
}
```

- a. 10.
- b. El bucle se ejecuta permanentemente (no sale).
- c. 0.
- d. 11.