

## **Test de Autoevaluación**

### **1 ¿Qué son las estructuras selectivas?**

En lenguaje de programación, una estructura selectiva es un algoritmo utilizado para alterar el orden de operación de los diferentes comandos u opciones con los que esta cuenta, esto dependiendo del estado de la declaración booleana ("True" o "False") evaluada en la misma.

### **2 ¿Cuándo se ejecuta la sentencia de if?**

Cuando se necesita ejecutar una determinada acción en base a la posición en la que se encuentre su respectiva declaración booleana ("True" o "False").

### **3 ¿Qué colocamos entre paréntesis en if?**

La declaración booleana en base a la que la estructura "decidirá" si es pertinente ejecutar las líneas de comando que se encuentren dentro de la misma.

### **4 ¿Qué es una expresión relacional?**

Es una expresión lógica en la que se evalúa lo que representa una expresión o término matemático con respecto a otro, esto a través de un operador relacional.

### **5 ¿Cuáles son los operadores de una**

### **expresión relacional?**

Son los que evalúan la relación que guarda un término con respecto a otro. En el caso de C#, los operadores relacionales son: < (menor que), > (mayor que), <= (menor o igual que), >= (mayor o igual que), is (es), as (como), == (igual a) y != (diferente de).

### **6 ¿A qué valores posibles puede evaluar una**

### **expresión relacional o lógica?**

Una expresión relacional puede evaluar cualquier rango de valores del tipo numérico, mientras que una expresión lógica solamente es capaz de evaluar expresiones de tipo booleano.

## **7 ¿Qué es una expresión lógica?**

Se denomina así a una expresión que establece una serie de condiciones lógicas que se encargan de determinar si los datos evaluados dentro de la misma cumplen o no con dichas condiciones.

## **8 ¿Cuáles son los operadores de las expresiones lógicas?**

Son los que hacen la evaluación de los datos que se encuentran dentro de la expresión para determinar si esta es verdadera ("True") o falsa ("False"). En el caso de C#, estos son: && (Y), || (O) y ! (No).

## **9 ¿Qué operador solamente necesita un operando?**

El operador "!" (No).

## **10 ¿Cómo funciona el switch?**

Este sirve, de manera similar al If, mediante la evaluación de una expresión booleana como base a través de la cual modifica el orden de ejecución de los comandos inscritos dentro del mismo.

## **11 ¿Qué es la variable de comparación y cómo se coloca?**

La variable de comparación de switch es la que se evalúa dentro de la estructura para determinar si coincide con alguno de los "case" de la misma. Esta se coloca junto a la declaración de la estructura entre paréntesis.

La forma es la siguiente: switch (variable)

## **12 ¿Cómo definimos los casos y cómo usamos break?**

Un caso o "case" se define declarándolo junto a dos puntos, para luego describir dentro de este el bloque de código a ejecutarse en caso de ser "elegido". Luego, se utiliza "break;" para realizar el cierre.

Su forma es: case: (Bloque de código)

## **Test de Autoevaluación**

### **1 ¿Qué es un ciclo?**

Es una serie de instrucciones que se repiten tantas veces como su condicional lógica lo permita.

### **2 ¿Cuáles son las partes del ciclo for?**

El ciclo for cuenta con la declaración del ciclo, la declaración y/o inicialización de la variable itinerante (valor inicial), el rango de repetición del ciclo (valor final), el comportamiento del itinerante luego de cada repetición (contador) y el bloque de instrucciones o comandos del ciclo.

### **3 ¿Cómo colocamos el valor inicial de**

#### **conteo en un ciclo for?**

Esto se realiza mediante la declaración y/o inicialización de la variable itinerante del ciclo. Es también la primera de las condiciones del ciclo en ser declaradas dentro del mismo.

Su forma es: **for (int i = 1; valor final; contador)**

### **4 ¿Cómo colocamos el valor final de conteo**

#### **en un ciclo for?**

Mediante una expresión relacional entre el itinerante y el valor final hasta el que se pretende que el ciclo se extienda. Se escribe como la segunda condición que se declara.

Su forma es: **for (valor inicial; i <= 2; contador)**

### **5 ¿Cómo se lleva a cabo el incremento en**

#### **un ciclo for?**

El incremento, o bien, decremento del ciclo for, se lleva a cabo mediante la declaración del comportamiento del itinerante luego de finalizada cada repetición. Se escribe como la tercera y última condición declarada dentro del ciclo.

Su forma es: **for (valor inicial; valor final; i++)**

## **6 ¿Cómo funciona el ciclo do while?**

Este ciclo ejecuta un bloque de instrucciones para posteriormente evaluar, mediante su condicional, si es posible realizar una nueva repetición. Esto se realiza indefinidamente hasta que el valor de la condicional del ciclo sea "False".

## **7 ¿Por qué el ciclo while se lleva a cabo al menos una vez?**

Es posible que el ciclo While no se ejecute ni una sola vez, puesto que este primero evalúa un condicional, el cual puede ser "False"; al contrario del ciclo Do While, el cual sí se ejecutará al menos una vez antes de realizar la evaluación lógica.

## **8 ¿Se necesita punto y coma al finalizar el ciclo do while?**

Sí, es necesario finalizar la declaración del ciclo utilizando el punto y coma.

## **9 ¿Cómo funciona el ciclo while?**

Este evalúa una expresión condicional para determinar si es posible ejecutar, posteriormente, un bloque de instrucciones. Esta operación es realizada hasta que el valor de la condicional del ciclo sea "False".

## **10 ¿Cuántas veces se puede repetir el ciclo while?**

El ciclo While es capaz de repetirse infinitamente.

## **11 ¿Qué tipo de condición podemos colocar en el ciclo while?**

Podemos colocar una condición del tipo Bool.

## **12 ¿Se coloca un bloque de código en el ciclo while?**

Sí, posterior a la expresión condicional, este ciclo cuenta con un bloque de código a ser ejecutado en caso de que dicha expresión devuelva un valor "True".