

**Complete lo que se le solicita en ambas páginas.**

Estructuras condicionales o de selección, y estructuras de repetición						
Estructura	Describa el uso más adecuado	¿Permite el uso de múltiples variables?	¿Permite el uso de múltiples condiciones?	¿Depende de otra estructura? Indique	De un ejemplo propio del uso que le ha dado a la estructura (por ejemplo: valide X escenario)	Califique de 1 a 10 la estructura según su consideración.
IF	Si una condición se cumple, ejecutar: { }	SI	Si, con And(&&) o Or (  )	No	If (Variable = 1) { Console.WriteLine("Enero"); }	8. Es una estructura simple y sencilla de utilizar, pero eso logra que para programas más complejos el código tenga que extenderse.
ELSE	Si la estructura if no se cumple, ejecutar: { }	Si	No, ya que solo evalúa el caso de que IF sea Falso.	Depende de IF	If (Variable >= 0) { C.W("positivo"); } Else { C.W("Negativo"); }	6. Depende de if e incluso no es necesario terminar la estructura con else.
ELSE IF	Si la estructura if no se cumple, evaluar: (----); Y ejecutar: { }	Si	SI	Depende de If	If (Variable > 0) { C.W("positivo"); } Else if (Variable == 0) { C.W("Neutro"); }	7. Estructura simple y fácil de utilizar, la estructura es clara y no hay limite de cuantos se pueden poner. Pero siempre depende de if
SWITCH	Para la variable se establecen los casos posibles y el código a ejecutar cuando se cumpla algún caso.	no	Si, pero solo para una misma variable. Y en	No	switch (diaDeLaSemana) { case "1":	8. Es más simplificado que if, pero no

			los casos diferentes.		<pre>Console.WriteLine("Hoy es lunes"); break; case "2": Console.WriteLine("Hoy es martes"); break;</pre>	permite múltiples variables.
WHILE	Es un bucle que mientras se cumpla una condición, ejecutará: { }	Si	Si, al igual que if	No	<pre>int contador = 1;  while (contador &lt;= 10) { if (contador % 2 == 0) { C.W("El numero es par");</pre>	8. Muy útil y flexible para validar el cumplimiento de ciertos criterios.
DO-WHILE	Siempre se ejecuta al menos una vez, independientemente de si se cumple la condición	Si	Si	No	<pre>string nombre; do { Console.Write("Ingresa tu nombre: "); nombre = Console.ReadLine(); } while (nombre == ""); Console.WriteLine("Hola: " + nombre);</pre>	9. Bastante elegante. Permite multiples variables y condiciones.
FOR	Repite tantas veces un bloque de código según se le indique	No	Si	No	<pre>for (int i = 0; i &lt; 10; i++) { Console.WriteLine("El valor de i es: " + i); }</pre>	8. Igualmente, simple de usar y puede tener usos complejos.

Responda o complemente la siguiente información			
¿Cómo se ha sentido en el curso hasta el momento?	¿Ha sido agradable su experiencia usando Visual Studio y C#? Describa	¿Considera que hay algo que debe repasar en el curso? Tome en cuenta los repasos hechos en teoría hasta la semana 7	Si la respuesta anterior es sí, ¿Por qué motivo considera que debe repasar ese o esos temas?

Bastante bien, muy explicativo todo y es interesante el uso de analogías	Si, tanto el Visual como el lenguaje son bastante intuitivos.	Tal vez ver las diferentes situaciones, posibilidades y combinaciones entre todas las estructuras, ya que han sido pocas las que hemos combinado.	Para ampliar nuestro conocimiento y estar preparado para cualquier situación.
--	---	---	---