

## **PAGINA 110**

1. Las estructuras selectivas son aquellas que mandan la ejecución de algún programa hacia un conjunto de sentencias o condiciones.
2. La sentencia if es un sino si una acción se cumple el programa hará algo y si esta no se cumple el programa hará otra cosa
3. Entre paréntesis en el if colocamos la condición que deseamos para que algo se cumpla o no
4. Una expresión relacional son símbolos logicos booleanos que se utilizan para comparar dos valores entre ellos
5. Los operadores de una expresión relacional son: <, >, <=, >=, ==, !=, <>
6. a dos valores: true y false.
7. Una expresión relacional, lógica o booleana es aquella que solo tiene un valor verdadero y otro falso
8. Los operadores que se utilizan para las expresiones lógicas son: and, or y negación.
9. La negación es el operador que necesita solo un operando.
10. El switch funciona de la misma manera en que hagamos if anidados, pero es mas resumido.
11. La variable de comparación se utiliza en el switch, es la que se compara con la variable que pide al usuario para comparar, y si estas son iguales entra al apartado que deseamos y se coloca, pidiéndola primero y comparando con lo que el usuario ingresó, y así este evalúa cada caso.
12. Los casos se definen poniendo el número del caso, un punto y coma y dentro de eso se coloca lo que hará el programa al entrar al caso y se cierra con un "break;" para que se sepa que ya terminó el caso.

## **PAGINA 146:**

1. Un ciclo es una acción que se ejecutara repetidas veces cada vez que se cumpla un x instrucción.
2. Las tres partes del ciclo for son: inicio del ciclo, condiciones del ciclo y el incremento o decremento de la variable.
3. El valor inicial en un ciclo for se lleva a cabo por medio de una asignación normal, declarando una variable con el valor que deseamos, por ejemplo: i=0.

4. El valor final de un ciclo for se coloca por medio de un límite de conteo, ejemplo=  
`i<10`.
5. El incremento en un ciclo for se lleva a cabo poniendo el `++` a la par de la variable que deseamos incrementar, ejemplo: `i++`.
6. El ciclo do-while funciona de la misma forma en que funciona el ciclo while pero con la diferencia que en el do while primero hará la instrucción y después revisará la condición.
7. el ciclo while no se lleva a cabo una vez por lo menos, es el ciclo do while debido a que antes de evaluar la condición hace la instrucción.
8. Si se necesita punto y coma después del do while.
9. El ciclo while funciona casi igual que el ciclo for, se coloca la condición entre paréntesis y cuando esta se cumpla hará las instrucciones que están dentro de esta y se repetirá hasta que la condición deje de cumplirse.
10. El ciclo while se puede ejecutar todas las veces que la condición se cumpla.
11. En el ciclo while se pueden colocar condiciones comparativas, por ejemplo:  
`while(a==b)`
12. Si en el ciclo while se coloca un bloque de instrucciones.