

1. Con sus propias palabras explique las diferencias entre switch e if

Previo a nombrar la diferencia más grande es importante aclarar que ambas tienen algo en común y es que son estructuras de control; nos ayudan a construir el flujo de nuestras tareas. El switch es una condicional súper potente que compara un valor y lo pasa por diferentes casos hasta que alguno de estos coincide y allí se ejecute una instrucción. Además, es importante a la hora de definir varias posibilidades en las que podría tornarse nuestro programa, permitiendo al programador y al software la elección de que haga algo en diversos casos de manera más óptima, allí radica una de las diferencias principales, la agilidad que esta permite cuando tenemos muchísimos casos; el switch va acompañado de un break y un default, que define la condición si no se cumplen ninguna de las antes desarrolladas. El if por lo general se utiliza en condicionales con menos posibilidades a diferencia del switch. Por ejemplo, si algo sucede, se ejecutaría un bloque de código, sino ejecutaría otro bloque diferente; además de que solo tiene en cuenta condiciones falsas o verdaderas. Es importante aclarar que el if por lo general siempre irá acompañado de un else if.

2. Con sus propias palabras explique las diferentes estructuras repetitivas

**For:** Esta estructura siempre irá acompañada de un índice para decirle a nuestro programa cuántas veces hará qué tarea; es decir, se define desde la misma línea de código del for la cantidad de veces, llámese “n” ingresada por el usuario o el número que el programador requiera.

**While:** La estructura de while lo que pretende es decirle a nuestro programa que haga algo si se cumple una condición específica que el programador define allí; esto puede ocurrir n cantidad de veces o ninguna; es decir, la cantidad de veces que esto ocurre es la cantidad de veces que se cumple la condición.

**Do While:** Esta estructura es una variación del while, pero esta primero hace una ejecución de la instrucción (es decir siempre se ejecuta a diferencia del while) y luego hay una validación si la condicional definida se cumple o no.

3. Con sus propias palabras defina y dé un ejemplo de los siguientes términos:
  - a. **Array:** Estos también son conocidos como arreglos, y se podrían definir como varios elementos del mismo tipo; aquí entrarían las matrices y vectores; es decir los unidimensionales y los bidimensionales.
  - b. **Vector:** Es una fila de contenedores, los cuales los podemos definir como vacíos e ir llenándolos por medio de nuestro código o definirlos desde un inicio con diferentes tipos de variables (int, string, etc); es importante definir que cada uno de ellos tiene una posición asociada, estos siempre arrancan desde la posición 0 en adelante; es decir, un vector podría ser un tren, donde cada vagón representa

un contenedor: 0 1 2 3 4. Estos son siempre unidimensionales y se deben definir a diferencia de las variables con cuántas posiciones contarán.

- c. **Matriz:** La matriz es hermana del vector, ya que también está hecha de contenedores para almacenar lo que deseemos, sin embargo, las matrices son multidimensionales y se pueden mover en un plano cartesiano con diferente información. Un ejemplo en el cual se comprende una matriz es el juego de buscaminas; el cual está desarrollado a partir de casillas que podemos ir explorando en todas las direcciones (horizontal, vertical y diagonalmente)
- d. **Variable:** Es un contenedor en el cual se puede almacenar lo que desee; yo lo comprendo como una canasta en la cual puedo meter diferentes frutas cuando yo desee. Estos contienen en sí valores. Estos valores pueden ir cambiando a lo largo del tiempo, es decir la variable: “*entero*” puede contener un 3, luego un 5, luego un 8, luego el resultado de una operación matemática, etc.
- e. **Constante:** Son contenedores que no pueden variar su valor. En el mismo ejemplo de las canasta, si por ejemplo guardo una manzana, ya no podré guardar otra cosa en esa misma canasta; es decir, si necesito guardar diferentes productos, necesito diferentes contenedores.
- f. **Operadores Lógicos:** Los operadores lógicos nos ayudan a comparar valores numéricos o alfanuméricos; como bien lo dice el nombre, los resultados siempre serán lógicos, es decir será algo falso o verdadero. Y son *AND*, *OR* y *NOT*
- g. **Operadores Matemáticos:** Estos nos sirven para realizar operaciones entre números, constantes o variables; a diferencia de los lógicos el resultado siempre será numérico; es por eso que no podemos utilizar operadores matemáticos en un string, a menos que hagamos una conversión del string a un número. Algunos de los operadores matemáticos más comunes y usados en programación son: *Suma*, *Resta*, *Multipliación*, *División*, *Mod*.