**UNIDAD II**

Para comenzar a programar podemos comenzar hablando sobre los objetos que vamos a utilizar para almacenar nuestros valores y poder trabajar con ellos en conjunto.

Lo que estaremos utilizando son las llamadas variables, en donde se almacenarán nuestros datos en memoria, cabe recalcar que durante la ejecución de nuestro programa, estas pueden llegar a cambiar sus valores.

La recomendación para identificar nuestras variables siempre será la misma:

* **No pueden ser extremadamente cortas**

Esto debido a que podemos perder, qué datos estamos metiendo en esa variable y comenzaríamos a perder tiempo buscando qué quiere decir o a qué nos referíamos al momento de declarar esa variable.

* **No pueden ser extremadamente largas**

Ya que al momento de estar utilizando esa variable, terminaríamos con un código muy abultado visualmente.

* **Comenzar por letras, sin signos de puntuación o espacios**

Dado que el lenguaje nativo de la programación se origina en base al inglés de manera estándar y tomando en cuenta que en ese lenguaje, no existen los signos de puntuación por ejemplo, podemos llegar a experimentar errores al momento de avanzar con nuestro código

Al igual que los caracteres especiales los cuales pueden estar relacionados a una lengua en específico en la cual la computadora no los comprenda y por ende, errores.

Por esto y otras razones, es mejor evitarnos este dolor de cabeza y ser concisos y coherentes con nuestras variables.

Los tipos de datos o valores que vamos a estar usando, pertenecen a un conjunto universal muy extenso, por lo que al momento de declarar estos en nuestro programa, es importante diferenciarlos por tipos de variables

Ya sean los siguientes:

* **Númericos**
  + Enteros
  + Racionales

Esto dependiendo del lenguaje y qué distinciones haga el mismo.

* **String**

Texto en su mayoría de ocasiones o cadena de caracteres, también puede estar representado por otro nombre acorde al lenguaje respectivamente.

Una vez declarados nuestros datos, podemos usarlos para realizar, operaciones entre ellos, comparaciones, validaciones, conjuntos, etc.

Las comparaciones y validaciones pueden ir agarradas de las manos ya que en base al resultado de estas, podemos abrirle paso a un proceso o a otro.

Entre las comparaciones que podemos llevar a cabo encontramos:

|  |  |
| --- | --- |
| **Comparación** | **Operador** |
| Igual a | == |
| No es igual a | != |
| Mayor que | > |
| Mayor o igual que | >= |
| Menor que | < |
| Menor o igual que | <= |
| Verdadero | true |
| Falso | false |

Nótese que los últimos 2, son operadores lógicos que indican uno u otro valor solamente.

Y estos vienen ligados a las operaciones lógicas AND / OR / NOT, con las que realizamos comparaciones para retornar valores verdaderos o falsos.

Acorde a las comparaciones que realicemos, podemos llevar a cabo distintas condiciones que resultan en procesos específicos

Existen distintas formas de llevar a cabo condiciones:

* **If**

Podemos decir que **si**(lo que declaremos) se cumple(true), llevará a cabo una serie de instrucciones específica.

* **Else**

Con **If**, si se entiende que al cumplirse su condición, llevará a cabo sus instrucciones y con **else**, pasa lo contrario, si no se llega a cumplir la primera condición, podemos seguir cubriendo los casos que se vayan dando con más instrucciones.

Con estas dos condiciones, podemos solventar una serie de variantes para nuestras validaciones hasta que se decida cerrar el proceso.

Existen también bucles en los cuales podemos repetir una serie de instrucciones una cantidad de veces y para representar o llevar un conteo de las veces que se va a repetir, se declara una variable para uso del bucle, con el fin de que vaya aumentando una vez se termine el proceso o la serie de instrucciones que coloquemos dentro, para declararlo utilizamos **For**

En For controlamos la cantidad de veces que se va a repetir el bucle, pero también podemos decir que mientras no se cumpla una función o se respete un rango específico, el bucle se seguirá repitiendo y esto lo logramos mediante el uso de **While**.

**UNIDAD II – Análisis**

Podemos encontrar comparaciones en distintas situaciones como en los parques temáticos en donde para subirnos a una de las atracciones, hay un requerimiento de altura o peso, entonces si nuestra altura es igual o mayor a la solicitada, podremos subirnos al juego.

Igual para requisitos en procesos bancarios, alguien sin récord crediticio no puede optar a un préstamo.

Las clasificaciones de películas o videojuegos restringen su uso o compra para personas menores de edad.

Las condiciones vistas en esta unidad de igual forma pueden ser implementadas de igual forma en nuestra rutina diaria colocando como ejemplo los horarios laborales, hay quienes marcan asistencia al momento de ingresar y salir de sus trabajos y no hay motivo de sanción siempre y cuando se respeten dichos horarios, así pues; se puede decir que durante la semana, **si** entramos en los horarios establecidos, nuestra asistencia se encontrará intacta, de lo contrario se nos puede llegar a sancionar.

De igual forma los bucles están presentes en muchas circunstancias, como en las becas en donde mientras nuestro promedio se mantenga en un valor establecido, podremos seguir optando por el beneficio.