# Tipos Enum

El tipo **enum** definido en java a partir de la versión 5.0, permite definir un conjunto de posibles valores o estados, que luego podremos utilizar donde queramos. Estos valores constituyen constantes que pueden ser utilizadas donde se necesiten.

Permiten definir un tipo con un conjunto finito de elementos de un tipo determinado (este conjunto de elementos) todo el resto de elementos no pertenecerá al tipo definido. Por ejemplo, podemos entender a un boolean como un tipo Enum de dos valores (true, false).

El nuevo tipo definido se caracterizará por la comprobación de tipos que realizará el compilador, únicamente existen una determinada cantidad de instancias válidas del conjunto, que se extiende de la clase java.lang.Enum y que implementa == o equals() y compareTo().

Los ejemplos más comunes de enum incluyen direcciones (valores de Norte, Sur, Este y Oeste) y los días de la semana.

Debido a que son constantes, los nombres de un tipo enum de los campos son en mayúsculas.

Ejemplo de un código con utilización de enum, creado en una clase tipo java.lang.Enum:



El resultado de ejecutar el código anterior es:



Ejemplo de un Enum dentro de una clase, siendo accedido desde otra clase:



El resultado de ejecutar el código anterior es:



Un tipo enum no puede ser declarado dentro de un método. Todos los miembros de un enum son del mismo tipo que el enum al que pertenecen.

En un enum se pueden declarar constructores y métodos declarando el parámetro del constructor en cada uno de sus miembros. Los constructores de un enum no pueden ser invocados directamente.

En el siguiente código se observa un enum con constructor y métodos asignados:



Además, cada enum tiene dos métodos estáticos implícitos que permiten acceder a los diferentes valores:

* Cafe.values(): devuelve una tabla de Cafe con todos los valores posibles
* Cafe.valueOf (String): devuelve el Cafe del nombre pasado por parámetro (por ejemplo, Cafe.valueOf("MEDIANO") devolverá Café.MEDIANO).

Los diferentes valores del enum son constantes estáticas y públicas y son accesibles directamente como los campos public static (por ejemplo con Café.MEDIANO).

Por lo tanto, los enums pueden ser usados directamente en un switch.