

## TIPO ENUMERADO EN JAVA

Un tipo enumerado restringe los posibles valores que puede tomar una variable.

### Sintaxis:

```
modificador de acceso enum nombreTipoEnumerado { ELEMENTO1, ELEMENTO2, ELEMENTO3, ..., ELEMENTOn };
```

### Ejm:

```
public enum Demarcación {PORTERO, DEFENSA, CENTROCAMPISTA, DELANTERO}
```

Dentro de las llaves se declaran los valores de los que consta el tipo enumerado. Por convención sus nombres se escriben en letras mayúsculas para recordarnos que son valores fijos (que en cierto modo podemos ver como constantes).

Realmente un enumerado (o Enum) es una clase "especial" (en eclipse debemos ir dentro del menú por New Class para crear la clase enumerado), que limita la creación de objetos a los especificados explícitamente en la implementación de la clase.

La clase enumerado es una clase abstracta, lo que significa que la creación de objetos no se realiza con new, se realiza declarando una variable del tipo enumerado (nombreTipoEnumerado nombreVariable), y asignándole un elemento concreto especificando el nombreDelTipoEnumerado.ELEMENTO.

### Ejm:

```
Demarcacion jugador= Demarcacion.DELANTERO; //Instancia de un enumerado de la clase Demarcación. Si se intentara poner, demarcacion.ENTRENADOR, me daría un error.
```

No se pueden crear más objetos del tipo enumerado que los especificados en su declaración.

Puesto que el tipo enumerado es una clase, dispone de una serie de métodos. Algunos de los métodos más utilizados son:

- **name()**: Devuelve un String con el nombre de la constante del tipo enumerado.
- **toString()**: Devuelve un String con el nombre de la constante del tipo enumerado.
- **ordinal()**: Devuelve un entero con la posición del enumerado según está declarado.
- **values()**: Devuelve un array que contiene todos los enumerados. Se usa en combinación con for-each.

**Ejm:**

```
Demarcacion delantero = Demarcacion.DELANTERO;
```

```
Demarcacion defensa = Demarcacion.DEFENSA;
```

*// Devuelve un String con el nombre de la constante*

```
System.out.println("delantero.name()= "+delantero.name());
```

```
System.out.println("defensa.toString()= "+defensa.toString());
```

*//Devuelve un entero con la posición de la constante según está declarada.*

```
System.out.println("delantero.ordinal()= "+delantero.ordinal());
```

*// Recorre todas las constantes de la enumeración*

```
for(Demarcacion d: Demarcacion.values()){
```

```
    System.out.println(d.toString()+" - ");
```

```
}
```

**Salida:**

```
delantero.name()= DELANTERO
```

```
defensa.toString()= DEFENSA
```

```
delantero.ordinal()= 3
```

**Ejm:**

```
public enum TipoDeMadera { ROBLE, CAOBA, NOGAL, CEREZO, BOJ };
```

```
public class TestEnum {
```

```
    public static void main (String[ ] args) {
```

```
        TipoDeMadera maderaUsuario=TipoDeMadera.ROBLE;
```

```
        System.out.println ("La madera elegida por el usuario es " +  
        maderaUsuario.toString().toLowerCase());
```

```
        System.out.println ("¿Es la madera elegida por el usuario caoba?  
        Resultado: " + (maderaUsuario==TipoDeMadera.CAOBA) );
```

```
        System.out.println ("¿Es la madera elegida por el usuario roble?  
        Resultado: " + (maderaUsuario==TipoDeMadera.ROBLE) );
```

```
    }
```

```
} //Cierre de la clase
```

**Salida:**

La madera elegida por el usuario es roble

¿Es la madera elegida por el usuario caoba?  
Resultado: false

¿Es la madera elegida por el usuario roble?  
Resultado: true