**[Clases y tipos genéricos en Java](https://jonsegador.com/2012/10/clases-y-tipos-genericos-en-java/)**

En [Java](http://jonsegador.com/tag/java/), cuando definimos una nueva clase,**debemos conocer el tipo de dato** con el que trabajaremos. Si queremos realizar una operación específica dentro de esta nueva clase, **sea cual sea el tipo de datos** que va a recibir, podemos hacer uso de los **tipos genéricos**. Este tipo genérico asumirá el tipo de dato que realmente le pasaremos a la clase.

Mejor con un ejemplo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **class** ClaseGenerica<T> {  T obj;    **public** ClaseGenerica(T o) {  obj = o;  }    **public** **void** classType() {  System.out.println("El tipo de T es " + obj.getClass().getName());  }  }    **public** **class** MainClass {  **public** **static** **void** main(String args[]) {  *// Creamos una instancia de ClaseGenerica para Integer.*  ClaseGenerica<Integer> intObj = **new** ClaseGenerica<Integer>(88);  intObj.classType();    *// Creamos una instancia de ClaseGenerica para String.*  ClaseGenerica<String> strObj = **new** ClaseGenerica<String>("Test");  strObj.classType();    }  } |

Unos apuntes:

* T es el tipo genérico que será reemplazado por un tipo real.
* T es el nombre que damos al parámetro genérico.
* Este nombre se sustituirá por el tipo real que se le pasará a la clase.

El resultado será el siguiente:

|  |  |
| --- | --- |
| 1  2 | El tipo de T es java.lang.Integer  El tipo de T es java.lang.String |

Hay que tener en cuenta que **los generics de java solo funcionan con objetos**. El código siguiente nos mostrará un error:

|  |  |
| --- | --- |
| 1 | ClaseGenerica<int> myOb = **new** ClaseGenerica<int>(53); *// Error, can't use primitive type* |

## Existen una serie de convenciones para nombrar a los genéricos:

* E – Element (usado bastante por Java Collections Framework)
* K – Key (Llave, usado en mapas)
* N – Number (para números)
* T – Type (Representa un tipo, es decir, una clase)
* V – Value (representa el valor, también se usa en mapas)
* S,U,V etc. – usado para representar otros tipos.

**Las clases y tipos genéricos** nos serán de gran utilidad para programar ciertas funcionalidades en nuestras aplicaciones para [Android](http://jonsegador.com/category/android/).