



APRENDERAPROGRAMAR.COM

CLASES CON DOS O MÁS  
CONSTRUCTORES EN JAVA.  
SOBRECARGA DE  
CONSTRUCTORES O  
MÉTODOS. (CU00640B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

**Resumen:** Entrega nº40 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

## CLASES CON DOS O MÁS CONSTRUCTORES. SOBRECARGA DE CONSTRUCTORES O MÉTODOS.

En este apartado vamos a ver cómo una clase en Java puede tener más de un constructor y a entender qué implicaciones y significado tiene esto. Escribiremos el código de una clase y lo compilaremos para ir analizando en base a este código cómo se generan clases con varios constructores y el significado del concepto de sobrecarga de constructores o métodos.



Escribe y compila el siguiente código:

```
//Ejemplo de clase con dos constructores y un método
public class Persona {
    private String nombre;
    private int edad;

    public Persona (String nombrePersona) { //CONSTRUCTOR 1
        nombre = nombrePersona;
        edad = 0; }

    public Persona () { //CONSTRUCTOR2
        nombre = "";
        edad = 0; }

    public String getNombre () { return nombre; } //Cierre del método
} //Cierre de la clase
```

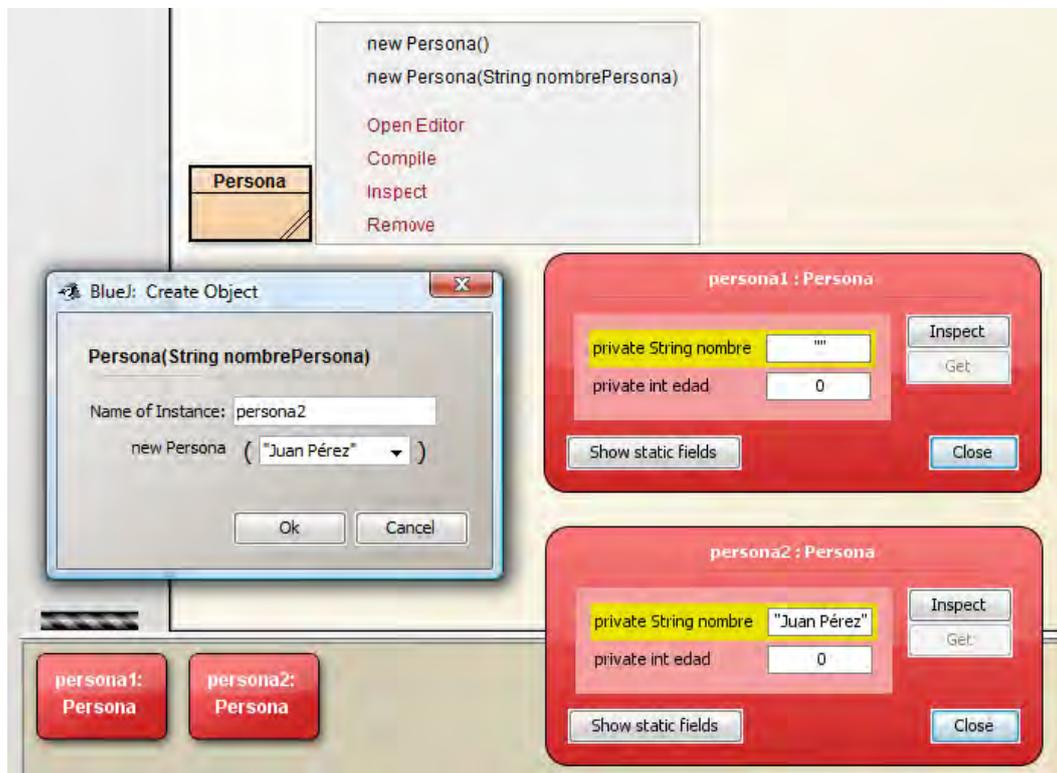
Hemos definido una clase, denominada *Persona*, que nos permite crear objetos de tipo *Persona*. Todo objeto de tipo *Persona* estará definido por dos campos: *nombre* (tipo *String*) y *edad* (tipo entero), y admitirá un método: *getNombre()*. Al realizar la invocación *nombreDelObjeto.getNombre()* obtendremos el atributo *nombre* del objeto.

La clase tiene dos constructores. ¿Qué significado tiene esto? Pues que **podremos crear personas de dos maneras distintas**:

- Personas que se creen con el constructor 1: habrá de indicarse, además del nombre del objeto, el parámetro que transmite el nombre de la persona.
- Personas que se creen con el constructor 2: no requieren parámetros para su creación y se inicializan a unos valores por defecto (nombre cadena vacía y edad cero).

Cuando más de un constructor o método tienen el mismo nombre pero distintos parámetros decimos que el constructor o método está **sobrecargado**. La sobrecarga de constructores o métodos permite llevar a cabo una tarea de distintas maneras (por ejemplo crear un objeto *Persona* con un nombre ya establecido o crearlo sin nombre establecido).

Pulsa sobre el icono de la clase y elige la opción *new Persona()* para crear un objeto. Seguidamente, pulsa de nuevo sobre el icono de la clase y con botón derecho elige la opción correspondiente al otro constructor disponible *new Persona(String nombrePersona)*. Introduce un nombre como parámetro, por ejemplo "Juan Pérez". A continuación, utiliza el inspector de objetos para comprobar cuál es el estado de cada objeto.



La existencia de dos constructores se ha visto reflejada en que disponemos de más de una opción de *new Persona* para crear objetos. Según la opción que elijamos, el objeto *Persona* se creará de una forma u otra. Esto nos lleva a la conclusión de que cada constructor define una forma de crear objetos.

## EJERCICIO

Define una clase *Profesor* considerando los siguientes atributos de clase: nombre (*String*), apellidos (*String*), edad (*int*), casado (*boolean*), especialista (*boolean*). Define un constructor que reciba los parámetros necesarios para la inicialización y otro constructor que no reciba parámetros. Crea los métodos para poder establecer y obtener los valores de los atributos. Compila el código para comprobar que no presenta errores, crea un objeto usando un constructor y luego otro objeto usando el otro constructor. Comprueba que se inicializan correctamente consultando el valor de sus atributos después de haber creado los objetos. Para comprobar si es correcta tu solución puedes consultar en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

**Próxima entrega:** CU00641B

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=68&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188)