

## CURSO DE FUNDAMENTOS DE JAVA

# CONSTRUCTORES EN JAVA



Ing. Ubaldo Acosta

Por el experto: Ing. Ubaldo Acosta



## CURSO DE FUNDAMENTOS DE JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

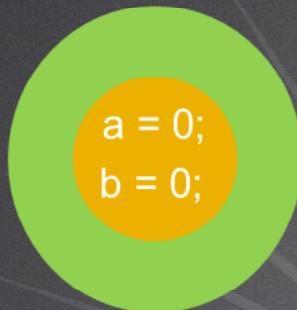
Hola, te saluda nuevamente Ubaldo Acosta. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de constructores en Java.

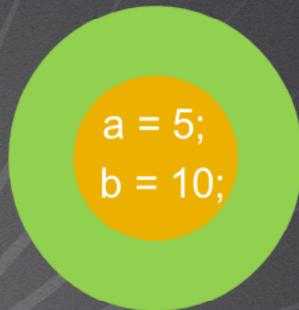
¿Estás listo? ¡Vamos!

# CONSTRUCTORES EN JAVA

Creación Objeto tipo Aritmetica  
Sin inicializar valores  
(uso del constructor vacio)



Creación Objeto tipo Aritmetica,  
inicializando valores  
(uso de constructor NO vacio)



`Aritmetica var1 = new Aritmetica();`

`Aritmetica var1 = new Aritmetica(5, 10);`

## CURSO DE FUNDAMENTOS DE JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Puede ser muy tedioso estar inicializando las variables de una clase cada vez que creamos un objeto, para ello Java permite agregar valores a los objetos desde el mismo momento de su creación. A este concepto se le conoce como Constructor, y básicamente es un método pero que nos permitirá crear un objeto con los valores que proporcionemos al momento de crear nuestro objeto.

Los constructores aunque son muy similares a los métodos, tienen ciertas diferencias y características, tales como:

- Estos métodos sólo se puede ejecutar al momento de la creación de un objeto, no es posible utilizarlos después.
- No devuelven ningún valor.
- El nombre del constructor es idéntico al nombre de la clase, así es como podemos identificar si una clase tiene constructores (siempre respetando las mayúsculas y minúsculas)
- Por defecto en Java, se crea un constructor sin argumentos, conocido como constructor vacío. Este constructor lo agrega en automático el compilador de Java a nuestra clase, sin embargo si nosotros definimos un constructor distinto al constructor vacío, es decir, con argumentos, entonces Java ya no agrega el constructor vacío y es nuestra responsabilidad agregar el constructor vacío a nuestra clase si fuera necesario.
- El constructor vacío es necesario para crear un objeto, recordemos la sintaxis básica general para crear un objeto.
  - `TipoClase objeto = new TipoClase();`
- Si observamos al final de la línea de código estamos abriendo y cerrando paréntesis, ese es precisamente el constructor vacío que el compilador agregó por nosotros siempre y cuando no agreguemos constructores con otros argumentos. Por lo que ahora ya podemos entender que después de la palabra `new` realmente lo que estamos colocando es el nombre del constructor que deseamos llamar, pudiendo tener argumentos o no.

En la figura podemos observar un ejemplo de nuestra clase Aritmetica, usando un constructor vacío y otro no vacío para crear nuestros objetos.

Supongamos que tenemos dos atributos en nuestra clase, de tipo entero, llamado `a` y `b`. En el primer caso estamos creando un objeto llamando al constructor vacío, por lo tanto los valores de nuestros atributos se inicializan con los valores por default según el tipo de cada atributo. Si es un tipo primitivo, se inicializan según su valor por default, y si son de tipo object su valor inicia es `null`.

Por otro lado creamos un objeto Aritmetica pero con valores iniciales desde el momento en que estamos creando el objeto respectivo. Estamos enviando los valores de 5 y 10 respectivamente, y por lo tanto podemos tomar estos valores para inicializar

los atributos de nuestras clases.

# EJEMPLO DE CONSTRUCTORES

## Forma general de un Constructor en Java:

```
//Forma general de un constructor en Java
NombreClase( arg1, arg2, etc...) {
    //Cuerpo del constructor de la clase
}
```

## Ejemplo de constructores en Java:

```
//Constructor sin argumentos
Aritmetica() {
    //Cuerpo del constructor
}

//Constructor con 2 argumentos
Aritmetica( int arg1 , int arg2) {
    //Cuerpo del constructor
}
```

Como hemos comentado, un constructor inicializa un objeto al momento de la creación, además es el encargado de reservar la memoria para la creación del objeto.

Para crear un constructor en una clase Java, la sintaxis general incluye el nombre de la clase, de esta forma podemos identificar si una clase tiene un constructor o no. Anterior a esto puede tener o no un modificador de acceso como public o private, estos modificadores los estudiaremos a detalle posteriormente. Después del nombre de la clase, similar a un método, puede tener o no argumentos.

Recordemos que Java crea el constructor sin argumentos de manera automática, sin embargo si hemos agregado un constructor con argumentos, entonces Java ya no agregaría este constructor por nosotros y es nuestra responsabilidad agregarlo si lo vamos a necesitar, de hecho por buena práctica deberíamos agregarlo siempre que definíamos un constructor con argumentos.

En Java cuando creamos dos métodos o más con el mismo nombre, a esta característica se le conoce como sobrecarga de métodos (overloading). Esto significa que podemos que al momento de crear un objeto tenemos dos o más opciones para crearlo.

Como observamos en el ejemplo, hemos agregado dos constructores a la clase Aritmetica, en el primer caso es el constructor vacío o sin argumentos. Este constructor puede tener opcionalmente un código al momento de crear el objeto, sin importar que no reciba argumentos.

En segundo lugar, tenemos un constructor con dos argumentos, y recibe dos argumentos de tipo entero, una vez que recibimos los argumentos, normalmente los utilizaremos para inicializar nuestros atributos de la clase. Esto es lo que pondremos en práctica en el siguiente ejercicio.

Por último cabe mencionar que un constructor a diferencia de un método, no tiene un valor de regreso, por ello no encontraremos la palabra return ni un tipo de retorno en la firma del constructor. Esto se debe a que lo que regresa un constructor es el objeto en sí mismo.

# EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Creación del proyecto Artimetica\_V2

**CURSO DE FUNDAMENTOS DE JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## CURSO ONLINE

# FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

## CURSO DE FUNDAMENTOS DE JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- |                          |                                          |
|--------------------------|------------------------------------------|
| ✓ Fundamentos de Java    | ✓ Hibernate Framework                    |
| ✓ Fundamentos de Java    | ✓ Spring Framework                       |
| ✓ Programación con Java  | ✓ JavaServer Faces                       |
| ✓ Java con JDBC          | ✓ Java EE (EJB, JPA y Web Services)      |
| ✓ HTML, CSS y JavaScript | ✓ JBoss Administration                   |
| ✓ Servlets y JSP's       | ✓ Android con Java                       |
| ✓ Struts Framework       | <u>Datos de Contacto:</u> ✓ HTML5 y CSS3 |

Sitio Web: [www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Email: [informes@globalmentoring.com.mx](mailto:informes@globalmentoring.com.mx)