

# CURSO DE PROGRAMACIÓN CON JAVA

# BLOQUES DE CÓDIGO EN JAVA



Ing. Ubaldo Acosta

Por el experto: Ing. Ubaldo Acosta



## CURSO DE PROGRAMACIÓN CON 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 bloques de código en Java.

¿Estás listo? ¡Vamos!

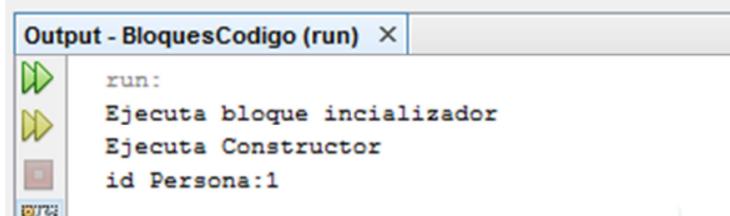
# BLOQUES DE CÓDIGO EN JAVA

## Bloques de código inicializadores:

```

1 public class Persona {
2     private final int idPersona;
3     private static int contadorPersonas;
4     {
5         System.out.println("Ejecuta bloque inicializador");
6         idPersona = ++contadorPersonas;
7     }
8
9     Persona() {
10        System.out.println("Ejecuta Constructor");
11    }
12
13    public int getIdPersona() {
14        return idPersona;
15    }
16 }
```

Bloque anónimo inicializador. Se ejecuta antes que el constructor



Un bloque de código en Java básicamente es cualquier parte de código que inicia con llaves y termina con llaves {}.

Cualquier contenido dentro de estas llaves se conoce como un bloque de código. Sin embargo, existen bloques de código anónimo, y a diferencia de los bloques que hemos utilizado por ejemplo para crear una clase o un método, los bloques anónimos nos permiten comúnmente inicializar variables de la clase. Estos bloques llamados inicializadores pueden iniciar dos tipos de variables, ya sea de instancia o de tipo static.

Para declarar un bloque de tipo anónimo e inicializar variables de instancia podemos ver el ejemplo en el código mostrado.

Podemos observar cómo de las líneas 4 a 7 se ha declarado un bloque de inicialización anónimo, cuyo objetivo es que este código se ejecute antes de la ejecución de nuestro constructor, de hecho este bloque de código se copia a cada constructor y se ejecuta antes de la ejecución de cada constructor.

Por ello en la salida de código (output), podemos observar que primero se ve en consola el mensaje "Ejecuta bloque inicializador" y posteriormente se observar el mensaje "Ejecuta Constructor". Y vemos también que se ha asignado correctamente el valor del idPersona con ayuda de la variable estática contadorPersonas.

Recordemos que en primer lugar se crean las variables estáticas, y posteriormente las variables dinámicas o atributos de clase, es por ello que en la línea 6 es posible utilizar la variable estática.

Con esto hemos visto cómo utilizar un bloque de código anónimo, el cual utilizamos para inicializar algunos elementos de nuestras clases. Este tipo bloques de código los veremos en algunas ocasiones en los clases Java, y ya podemos entender cómo funciona este tipo de bloques de código.

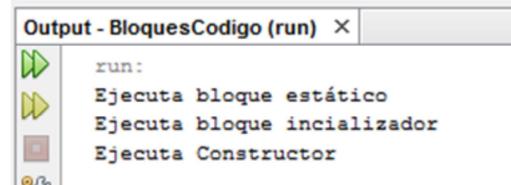
# BLOQUES DE CÓDIGO ESTÁTICOS EN JAVA

## Bloques de código inicializadores estáticos:

```

1 public class Persona {
2     private final int idPersona;
3     private static int contadorPersonas;
4
5     static {
6         System.out.println("Ejecuta bloque estático");
7         contadorPersonas=10;
8     }
9
10    {
11        System.out.println("Ejecuta bloque inicializador");
12        idPersona = ++contadorPersonas;
13    }
14
15    Persona() {
16        System.out.println("Ejecuta Constructor");
17    }
18 }
```

Bloque estático anónimo inicializador. Se ejecuta antes que el bloque NO estático



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

Ahora veremos cómo inicializar las variables estáticas de nuestro código por medio de un bloque de código anónimo y estático.

Similar a como vimos en la lámina anterior, es posible utilizar un bloque de código anónimo para realizar tareas de inicialización de variables estáticas. Para ello declaramos un bloque de código con la única diferencia que debemos anteponer la palabra static y posteriormente abrir y cerrar las llaves {}.

Dentro de este bloque de código podemos agregar código estático, es decir, variables estáticas.

Como observamos en el código observamos que podemos cambiar el valor de la variable estática contadorPersonas (línea 7), ya que no siempre tendremos disponible el valor que deseamos asignar al momento de declarar nuestras variables, y en este tipo de bloques de código es que podemos agregar una lógica más compleja para inicializar nuestras variables estáticas en Java.

Y aquí podemos observar y concluir lo mismo que estudiamos en la lección de contexto estático. Podemos ver en la salida (output) que primero se ejecuta el bloque estático, y posteriormente los bloques no estáticos, es decir el bloque inicializador y finalmente el constructor de la clase.

Con esto podemos entender cómo se van combinando los bloques de código para ir inicializando y construyendo nuestras clases Java, desde variables, bloques de código de inicialización y constructores. Con esto ya tenemos un mayor entendimiento de la construcción de nuestras clases.

Vamos a ver un ejemplo del uso de bloques de código.

# EJERCICIO CURSO PROGRAMACIÓN CON JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Manejo de Bloques de Código en Java

**CURSO DE PROGRAMACIÓN CON JAVA**

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

## CURSO ONLINE

# PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

### CURSO DE PROGRAMACIÓN CON 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:

- |                          |  |
|--------------------------|--|
| ✓ Programación con 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)