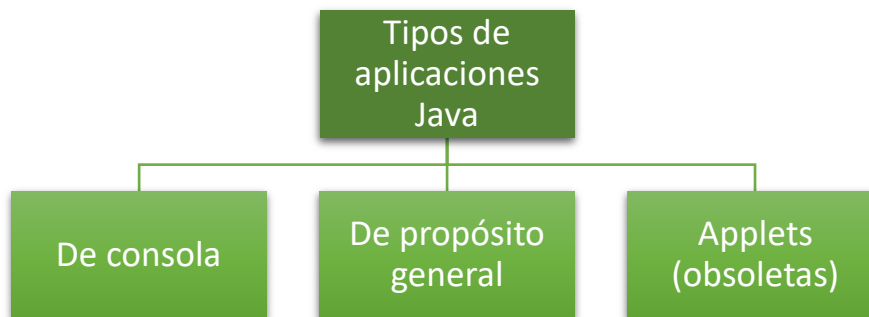




## TIPOS DE APLICACIONES, DATOS Y VARIABLES

### Tipos de aplicaciones en Java



### Tipos de datos

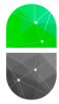
Java es un lenguaje de comprobación estricta de tipos. Todas las variables tienen que tener un tipo declarado.

En Java hay ocho tipos primitivos de datos. Cuatro son enteros, dos son en coma flotante (decimales), uno de tipo carácter y uno de tipo boolean.

- **Enteros**

Los tipos de datos enteros se utilizan para valores numéricos que no tiene parte fraccionaria. Permite negativos. Java tiene cuatro tipos de enteros que aparecen en la siguiente tabla:

<i>Tipo</i>	<i>Requisitos de almacenamiento</i>	<i>Rango (inclusive)</i>
<b>int</b>	4 bytes	Desde -2.147.483.648 hasta 2.147.483.647
<b>short</b>	2 bytes	Desde -32.768 hasta 32.767
<b>long</b>	8 bytes	Desde -9.223.372.036.854.775.808 hasta 9.223.372.036.854.775.807
<b>byte</b>	1 byte	Desde -128 hasta 127



## Introducción

En la mayoría de los casos en los que nos podamos encontrar, el tipo **int** es el más práctico. Si se necesita expresar los granos de arena de una playa utilizaremos **long**. **byte** y **short** están destinados a aplicaciones especializadas.

Estos intervalos en los tipos de datos no dependen nunca de la máquina en la que se ejecuta el código al contrario de lo que sucede con otros lenguajes de programación.

- **Coma flotante (decimales)**

Los tipos con coma flotante son los números con parte fraccionaria. Existen dos tipos que aparecen en la siguiente tabla:

Tipo	Requisitos de almacenamiento	Alcance
<b>float</b>	4 bytes	Aprox +- 3,40282347E+38F (6-7 cifras decimales)
<b>double</b>	8 bytes	Aprox +-1,797693134E+308 (15 cifras decimales)

Los de tipo **double** deben su nombre a que tienen el doble de precisión que los de tipo **float**.

El tipo que utilizaremos en la mayoría de las ocasiones será el **double**, debido a que la escasa precisión del **float** será insuficiente en muchas ocasiones.

- ✓ **Nota:** al dividir un número por 0, o al intentar hallar la raíz cuadrada de un número negativo, el resultado será NaN ( Not a Number ).

- **Tipo char**

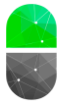
En Java el tipo char describe una unidad de código con la codificación UTF-16.

Se recomienda no utilizar el tipo char en los programas salvo que se vaya a manipular unidades de código UTF-16. Las cosas van siempre bien cuando se utilizan las cadenas, que se verán más adelante en este manual.

- **Tipo boolean**

El tipo boolean tiene dos posibles valores, false y true. Se utiliza para evaluar condiciones lógicas.

No se pueden hacer conversiones entre valores enteros y boolean.



## Variables

Una variable es un nombre que se asocia a un espacio en la memoria del ordenador, en el que se guarda el valor asignado a dicha variable. Hay varios tipos de variables que requieren distintas cantidades de memoria para guardar datos.

Todas las variables han de declararse antes de usarlas, la declaración consiste en una sentencia en la que figura el tipo de dato y el nombre que asignamos a la variable. Una vez declarada se le podrá asignar valores.

En Java, todas las variables tienen un tipo. Las variables se declaran en Java poniendo primero el tipo seguido del nombre de la variable:

**double** sueldo; **int** días de vacaciones; **long** población terrestre; **boolean** hecho;

El punto y coma al final de cada declaración es necesario.

Los nombres de las variables tienen que empezar por una letra y deben ser una sucesión formada por letras y cifras. La longitud del nombre de una variable es ilimitada.

No se pueden utilizar palabras reservadas para nombrar variables.

Se pueden declarar variables en una misma línea:

```
int i, j;
```

En los nombres se distingue entre mayúscula y minúscula. Esta característica propia del lenguaje Java y otros, es lo que se denomina **Case Sensitive**.

El lenguaje Java utiliza el conjunto de caracteres Unicode, que incluye no solamente el conjunto ASCII sino también caracteres específicos de la mayoría de los alfabetos. Así, podemos declarar una variable que contenga la letra ñ: **int año=1999**;

- **Iniciación de variables**

Tras declarar una variable, es preciso darle valor inicial mediante una sentencia de asignación.

No se puede utilizar una variable hasta que no se le haya dado un valor inicial. Esta sentencia daría error:

```
int diasdevacaciones;
```

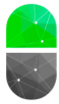
```
System.out.print(diasdevacaciones);
```

Para asignar valor a una variable declarada previamente, se pone el nombre de la variable a la izquierda, después un signo = y a continuación alguna expresión de Java que tenga el valor adecuado a su derecha. **int diasdevacaciones; diasdevacaciones=14;**

Se puede declarar y dar valor inicial en la misma línea. Por ejemplo: **int diasdevacaciones=14;**

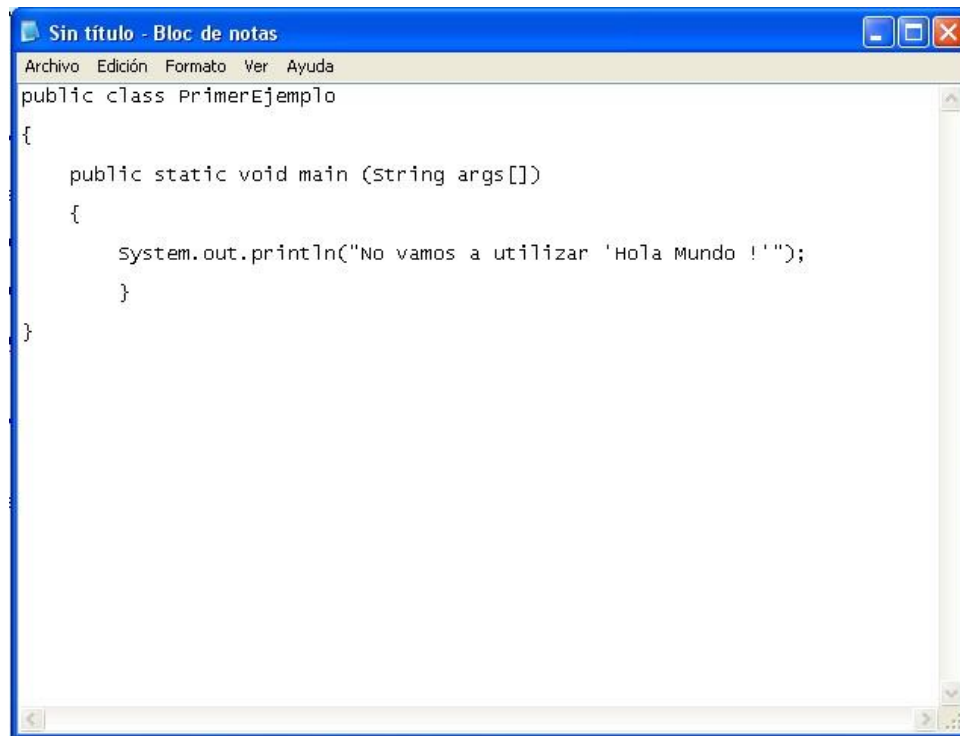
En Java se pueden poner declaraciones en cualquier lugar del código.

Es aconsejable declarar variables tan cerca de donde se van a utilizar como sea posible.



## Un programa sencillo en Java

A continuación, vamos a exponer un ejemplo del programa más sencillo que se puede construir en Java. Este programa se limitará a mostrar un mensaje en la consola:



Conviene familiarizarse con las estructuras de la imagen anterior. Sus partes aparecerán de forma reiterativa en todos los programas de Java.

Java respeta las mayúsculas y las minúsculas. Es lo que se denomina **Case Sensitive**.

Examinado las diferentes partes del código nos encontramos con la palabra reservada **public**. *Public* es un modificador de acceso. Estos modificadores controlan el nivel de acceso que tienen a este código otras partes del programa.

La palabra reservada **class** nos recuerda que todo lo que hay en un programa Java está dentro de una clase. Las clases son los bloques de construcción mediante los cuales se construyen todas las aplicaciones de Java.

Después de la palabra reservada **class** está el **nombre de la clase**. Los nombres de las clases deben empezar por una letra y a continuación puede haber cualquier combinación de letras y dígitos. La longitud del nombre no está limitada. Se suele establecer que la primera letra del nombre de la clase sea mayúscula, aunque no es necesario para el funcionamiento del programa.

No se pueden utilizar palabras reservadas para nombrar las clases de Java.

Cuando se utiliza Java NombreClase para ejecutar un programa compilado, la máquina virtual de Java empieza la ejecución en el código que se encuentra dentro del **método main** de la clase indicada.

Es preciso tener un método main en el código.



Las **llaves** existentes en el código delimitan las partes o bloques del programa. Los métodos deben empezar por una llave abierta { y terminar por una cerrada }

Toda aplicación debe tener un método main que se declara de la siguiente manera:

```
public class NombreClase {  
  
    public static void main(String args[])  
  
    {  
  
        Sentencias del programa  
  
    }  
  
}
```

En el fragmento:

```
{  
  
System.out.println("No vamos a utilizar „Hola mundo ¡“ “");  
  
}
```

Las llaves marcan el principio y el final del método. Este método solo contiene una declaración o sentencia. **En Java toda declaración o sentencia debe acabar con un punto y coma.**