

Tipos de datos Java

Los tipos de datos Java son la base de la manipulación de datos en la programación Java. Definen el tamaño y el tipo de valores que pueden almacenarse en una variable. Java es un lenguaje estáticamente tipado, lo que significa que cada variable debe declararse con un tipo de datos antes de su uso. Los tipos de datos Java se clasifican en dos grupos principales: tipos de datos primitivos y tipos de datos de referencia.

Tipos de datos primitivos

Los tipos de datos primitivos son los tipos de datos más básicos disponibles en Java. Hay ocho tipos de datos primitivos, cada uno de los cuales tiene una finalidad específica:

1. byte:

- Talla: 8-bit
- Intervalo: de -128 a 127
- Utilización: Almacenamiento eficiente en memoria en grandes matrices.

```
byte b = 100;
```

[Powered By](#)

2. corto:

- Talla: 16-bit
- Intervalo: de -32.768 a 32.767
- Utilización: Adecuado para ahorrar memoria en grandes matrices.

```
short s = 10000;
```

[Powered By](#)

3. int:

- Talla: 32 bits
- Intervalo: -2^{31} a $2^{31}-1$
- Utilización: Opción por defecto para valores enteros.

```
int i = 100000;
```

Powered By

4. **largo:**

- Talla: 64 bits
- Intervalo: $-2^{63} \text{ a } 2^{63}-1$
- Utilización: Para valores enteros grandes.

```
long l = 100000L;
```

Powered By

5. **flotar:**

- Talla: 32 bits
- Utilización: Para números fraccionarios, con precisión única.

```
float f = 234.5f;
```

Powered By

6. **doble:**

- Talla: 64 bits
- Utilización: Para números fraccionarios, con doble precisión.

```
double d = 123.4;
```

Powered By

7. **booleana:**

- Valores: true o false
- Utilización: Para banderas y condiciones simples.

```
boolean flag = true;
```

Powered By

8. **char:**

- Talla: 16-bit
- Alcance: 0 a 65.535 (caracteres Unicode)
- Utilización: Para almacenar caracteres.

```
char c = 'A';
```

Variables y tipos de datos en Java: tipos simples, clases y tipos envoltorio o wrapper

Publicado por [campusMVP](#) el 17 de julio de 2018 a las 08:00

No es lo mismo tener que almacenar en la memoria de un ordenador un número entero que uno decimal. Aunque al final todo son ceros y unos dentro de la memoria de una máquina, es la forma de interpretarlos lo que marca la diferencia, tanto al almacenarlos como al recuperarlos. Este el motivo por el que los lenguajes de programación cuentan con el concepto de tipos de datos: se trata de distintas maneras de interpretar esos "ceros y unos" en función de ciertas configuraciones que establecen el espacio utilizado así como la representación aplicada para codificar y descodificar esa información. En este artículo te presentamos los diferentes tipos de datos que tiene Java, sus usos y restricciones, etc.

[De la arquitectura al despliegue: aprende a crear aplicaciones web desde cero con !\[\]\(cbe80b694ebd74fcfe136a095b608235_img.jpg\) Java y !\[\]\(27df6be88af07602ea392719b144fe7f_img.jpg\) Spring.](#)

Todo programa de ordenador persigue ofrecer una funcionalidad determinada para la que, por regla general, **necesitará almacenar y manipular información**. Dicha información, que son los datos sobre los que operaremos, deben **almacenarse temporalmente en la memoria del ordenador**. Para poder almacenar y recuperar fácilmente información en la memoria de un ordenador los lenguajes de programación ofrecen el concepto de **variables**, que no son más que **nombres que "apuntan" a una determinada parte de la memoria** y que el lenguaje utiliza para escribir y leer en esta de manera controlada.

Si te interesa aprender bien los fundamentos de esta plataforma, no pierdas de vista nuestro [curso de Java](#)

El acceso a esta información se puede mejorar dependiendo del tipo de información que almacenemos. Por ejemplo, no es lo mismo tener la necesidad de manejar números, que letras que conjuntos de datos. Y dentro de éstos no es igual tener que almacenar un número entero que uno decimal. **Aunque al final todo son ceros y unos dentro de la memoria de nuestro ordenador, es la forma de interpretarlos lo que marca la diferencia**, tanto al almacenarlos como al recuperarlos.

Este el motivo por el que los lenguajes de programación cuentan con el concepto de **tipos de datos**: se trata de distintas maneras de interpretar esos "ceros y unos" en

función de ciertas configuraciones que establecen el espacio utilizado así como la representación aplicada para codificar y decodificar esa información.

Cada **tipo de datos** se identifica por un **nombre** y es capaz de almacenar una determinada clase de información así como un rango de valores concreto.

Ahora que ya tenemos claro los conceptos de variable y de tipo de datos, vamos a ver cómo se traduce esto a la práctica en uno de los lenguajes más populares: [Java](#).



Foto por [Farzad Nazifi](#) en [Unsplash](#)

Tipos de datos primitivos en Java

Java cuenta con un pequeño conjunto de tipos de **datos primitivos**. Podríamos considerarlos **fundamentales**, ya que la mayor parte de los demás tipos, los tipos estructurados o complejos, son composiciones a partir de estos más básicos. Estos tipos de datos primitivos sirven para gestionar los tipos de información más básicos, como números de diversas clases o datos de tipo verdadero/falso (también conocidos como "valores booleanos" o simplemente "booleanos").

De estos tipos primitivos, ocho en total, seis de ellos están destinados a facilitar el trabajo con números. Podemos agruparlos en **dos categorías**: tipos numéricos enteros y tipos numéricos en punto flotante. Los primeros permiten operar exclusivamente con números enteros, sin parte decimal, mientras que el segundo grupo contempla también números racionales o con parte decimal.