

Trabajo sobre datos primitivos

Programación I

Juan Emauel Ulloa Castro

203M

**UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHIA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
FACULTAD DE INGENIERÍA**

2025

TABLA DE CONTENIDO

Contenido

INTRODUCCIÓN	3
OBJETIVOS	4
DESARROLLO.....	5
CONCLUSIONES	9
REFERENCIAS	10

INTRODUCCIÓN

El presente trabajo tiene como propósito explicar de manera clara, practica y detallada los distintos tipos de datos primitivos en Java. Estos son una base fundamental para manejar la información simple dentro del lenguaje, ya que son predefinidos por el sistema y el lenguaje de programación en si, además que se almacenan en la memoria directamente haciéndolos rápidos, eficaces y fáciles de manipular. Es vital para cualquier programador la comprensión de estos datos porque son el primer paso para la construcción de programas complejos y que llevan mucha estructura.

OBJETIVOS

1. identificar los distintos tipos de datos primitivos en java, reconociendo sus características principales, su rango de valores y las situaciones en las que se recomienda utilizar
2. Explicar la función y propósito de cada dato, complementándolo mediante ejemplos prácticos
3. Analizar la importancia de los datos en la eficiencia y consumo de memoria en algún programa
4. Aplicar los conocimientos adquiridos en ejemplos prácticos para aprender y entender la manipulación de variables

DESARROLLO

Antes de empezar a hablar sobre cada uno de los datos primitivos, veremos una tabla sobre el tamaño el tipo, su tamaño en bits y sus valores correspondientes.

Tipo	Tamaño (Bits)	Valor Mínimo	Valor Máximo
Byte	8	-128	127
Short	16	-32.768	32.767
Int	32	-2.147.483.648	2.147.483.647
Long	64	-9.223.372.036.854.775.808	9.223.372.036.854.775.807
Float	32	1.4×10^{-45} (aprox.)	3.4028235×10^{38} (aprox.)
Double	64	4.9×10^{-324} (aprox.)	$1.7976931348623157 \times 10^{308}$ (aprox.)
Char	16	0(\u0000)	65.535(\uffff)

Además de esto está el boolean pero este es de un Byte y funciona con true y false.

1. Datos primitivos enteros

- **Byte:** El dato byte pertenece a los datos primitivos enteros que se caracteriza por ser el más pequeño de todos en cuanto su capacidad de almacenamiento. Ocupa solo 8 bits que por así decirlo son 1 byte de memoria, lo cual lo vuelve una opción eficiente en cuando se requiere usar números de rangos reducidos. Su bajo rango de valores hace que sea adecuado para presentar información simple como edades o valores de control con códigos pequeños que a diferencia de otros enteros como el int o long lo vuelve consciente de situaciones en las que puede aplicarse evitando lo tan llamado desbordamiento. Su valor por defecto es 0. El

uso de este dato tiene alta relevancia en proyectos donde se tiene pequeña información numérica y digamos al usar byte en vez de int se consigue un ahorro de memoria donde puede ser útil en arrays de alto tamaño o en procesamiento de binarios donde cada bit cuenta.

- **Short:** el dato short también es otro dato numérico primitivo ubicado por encima de byte en capacidad porque ocupa 16 bits lo que son 2 bytes de memoria y puede presentar rangos de valores mínimos y máximos más amplios que byte. El uso de short es útil en situaciones donde es necesario manejar cantidades mayores que las permitidas por el byte, pero también ahorra memoria con respecto a int. Pero al igual que byte es importante tener en cuenta ese rango para evitar los desbordamientos. Este tipo de dato se encuentra en diversos sectores donde el almacenamiento es vital como en juegos o en sistemas embebidos. pero en la práctica se prefiere el int por su rango más amplio y que muchas operaciones matemáticas en Java se hacen automáticamente a int.
- **Int:** es el dato por excelencia y siendo el más usado dentro de los primitivos numéricos enteros. Este ocupa 32 bits (4 bytes de memoria) y tiene un valor mínimo y valor máximo amplio, que se adapta a la mayoría de las necesidades en programación lo que lo convierte en el estándar en comparación con byte o short, int tiene más comodidad para el programador pues permite trabajar con números grandes sin preocuparse por límites y es conveniente en java ya que garantiza más seguridad en los cálculos. su valor por defecto es 0 y su clase envolvente es Integer. El uso de int es común en aplicaciones que necesiten un rango considerable como el manejo de índices de estructuras como las listas y los mapas y en cálculos matemáticos que no requieren precisión decimal y lo hace la preferida por la mayoría de los programas que a menudo se escriben dentro de Java.
- **Long:** el tipo long es un numero con mayor capacidad ya que tiene 64 bits y 8 bytes de memoria pero que, aunque consume más memoria su uso es necesario si hay aplicaciones que requieren manejar cantidades grandes como en cálculos financieros o mediciones de tiempo en milisegundos o cuando se trabaja en operaciones matemáticas que exceden la capacidad del int que era de 32 bits. si

valor por defecto es 0L y su clase envolvente es Long. en comparación con el anterior ósea int, el long brinda mucho más rango, pero consume el doble. por esta razón solo se usa cuando sea realmente necesario usarlo para no desperdiciar recursos.

2. Datos primitivos de tipo de punto flotante

- **Float:** El tipo de dato float pertenece a los primitivos de punto flotante es decir los que permiten usar decimales, este tipo usa 32 bits o lo mismo que 4 bytes de memoria siguiendo un estándar IEEE 754 lo que le da una precisión de 6 a 7 cifras decimales. Aunque usa números con decimales su rango y precisión son menores a comparación de double, por eso se usa principalmente cuando se requiere almacenar valores con decimales, pero sin tanta exactitud. Su valor por defecto es 0.0f y su clase envolvente es Float, su sufijo es que al final de un número se ponga la f para que no lo detecte e interprete como un double (ejemplo: float n= 3.1415f;). Sus usos frecuentes son aplicaciones graficas como coordenadas o cálculos científicos y hasta en juegos.
- **Double:** el tipo de dato double es el principal de punto flotante en Java y se usa para trabajar con números decimales y muy preciso. Ocupa 64 bits (8 bytes de memoria) y al igual que float de estándar tiene IEEE 754, pero con el doble de capacidad. Esto permite representar números con 15 o 16 cifras decimales que lo vuelve el dato más usado para operaciones matemáticas con decimales por lo general. Su valor por defecto es 0.0d y su clase envolvente es Double. a menudo se usa en programas financieros o estadísticos en representación de medidas físicas muy grandes o muy muy pequeñas pero puede tener complicaciones porque al almacenar en binario puede dar resultados inesperados y al igual que vimos en el long es muy pesado y es mejor usar float si se quiere ahorrar espacio y no se necesita tanta precisión.

3. Dato primitivo de tipo carácter

- **Char:** el tipo de dato char a diferencia de los tipos enteros o punto flotante, su propósito es almacenar un solo factor Unicode, como una letra, un numero o un

símbolo. Este ocupa 16 bits o 2 bytes de memoria y gracias al Unicode en vez de ASCII no solo maneja caracteres en inglés sino también en símbolos especiales o también con letras con acentos o alfabeto griego, árabe y otros idiomas. Su valor por defecto es `\u0000` que equivale a carácter nulo y su clase envolvente es `Character`. Se usa para representar letras, o símbolos además cadenas de texto o codificaciones, Unicode y otros.

4. Dato primitivo lógico

- **Boolean:** que tipo es el boolean es el más sencillo de los primitivos, pero también de los más importantes su función es dar un `true` y `false` respectivamente, este no maneja números ni caracteres solo sirve para expresar condiciones lógicas lo que lo convierte en la base de estructuras. por defecto viene `false` y su clase envolvente es `Boolean`. Sirven para controlar flujos de ejecución en condiciones, representar estados o evaluar expresiones lógicas en bucles. No se puede asignar un número a boolean directamente ya que en Java a diferencia de otros lenguajes solo admite `true` o `false` y usa mucho en combinación con operadores como `&&` (And lógico) o `!` (Not lógico) y otros.

CONCLUSIONES

Los tipos de datos primitivos son la base para representar y manipular la información de manera eficiente al programar en Java y cada uno tiene propósitos como los enteros en manejar números sin decimales, los flotantes que trabajan con decimales de distinta precisión, char que facilita el uso de caracteres y el boolean aporta la lógica para la toma de decisiones en programas lo que hace que en conjunto sean rápidos, de bajo consumo o de consumo organizado y una estructura clara y son muy fundamentales para comprender como Java procesa la información

REFERENCIAS

manual web. (2025, agosto 30). *Tipos de Datos Primitivos en Java*. Manual Web. Disponible en:

<https://www.manualweb.net/java/tipos-datos-primitivos-java/>

Rodríguez, A. *Tipos de datos Java. Tipos primitivos (int, boolean...) y objeto (String, array o arreglo...) Variables (CU00621B)*. Aprenderaprogramar.com.

Disponible en:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=419:tipos-de-datos-java-tipos-primitivos-int-boolean-y-objeto-string-array-o-arreglo-variables-cu00621b&catid=68&Itemid=188

Todo sobre los tipos de datos primitivos en Java. (2025, enero 7). IONOS Digital Guide. Disponible en:

<https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/primitivos-de-java/>

Barragán, A. (2023, octubre 2). Introducción a Java: Datos y variables. *Openwebinars.net*. disponible en:

<https://openwebinars.net/blog/introduccion-a-java-datos-y-variables/>