



**PROGRA\_AMANDO**  
holamundo.co



# Paso por valor

Se presenta cuando se envía a una función un dato primitivo, la firma de la función tiene parámetros con valores fijos y con estos datos que llegan a la función se espera que sean usados dentro de la función como parte del procedimiento para generar el resultado, NO SE BUSCA CAMBIAR EL VALOR QUE LLEGA.

El contenido de la variable que pasa por valor aunque sea cambiado, no afecta el valor que traía por fuera del contexto de la función, ya que el atributo se pasa como una copia del valor, no se pasa el contenedor que almacena la variable.



# Dato primitivo

Es un dato nativo, muy simple, básico y que tiene una sola tarea, almacenar un valor de un tipo específico, con una capacidad limitada dada por las características de su contenido.

Aunque es un valor básico, son bastante poderosos, porque fueron creados como parte del lenguaje de programación para usarse en problemas específicos y de esta forma usar mejor los recursos del sistema, siempre y cuando el programador se preocupe por usarlos bien.

Tienen una representación máxima y mínima en bits o bytes, la capacidad es usada por el compilador para reservar el espacio que el programa va a usar en tiempo de ejecución.



# Referencia

Es un indicador a una posición en la memoria, esta posición se conoce como dirección, que representa la ubicación donde esta guardado uno o más datos.

Por lo general el contenido de un variable que sea una referencia, puede variar sus requisitos de almacenamiento en tiempo de ejecución, un ejemplo claro de esto se presenta con los arrays, ya que, es posible crear un programa que pida al usuario la cantidad de elementos de un array y luego cree el array con esa cantidad, este comportamiento se conoce como uso dinámico de la memoria y hace que los datos que son referencias sean versátiles por su capacidad de cambio.

Además de los arrays, se han usado otros tipos de dato que usan referencias, como los String, Stringbuilders y las matrices.

# Paso por valor



main(...)

```
int a = 5;  
int b = 6;  
int c;  
c = sumar(a,b);  
imprimir(a);  
imprimir(b);
```

Contexto A

a:5, b:6 Nunca se cambio el valor  
porque se paso por copia

int sumar(int a, int b)

```
int ans = a + b;  
a = 7;  
b = 2;  
return ans;
```

Contexto B

Aunque a y b cambian,  
los valores del contexto A  
no se ven afectados



# Otros contextos

Al usar funciones se crea un contexto, pero incluso de la misma función se pueden crear otros contextos que permiten aislar el alcance de su contenido, entender este comportamiento es clave para evitar errores al acceder a una variable.

```
public void pruebaContexto()
{
```

```
    for(int i=0;i<10;i++)
```

```
    {
```

```
        System.out.println(i);
```

```
    }
```

```
    System.out.println(i);
```

Error, variable  
fuera de contexto

Contexto B

Contexto A



# Pre y Post incrementos

Es frecuente encontrarse con snippets (Bloques pequeños de código reutilizable) que usan como atributos, valores primitivos que usan los operadores pre-incremento o post-incremento, es importante saber el funcionamiento de cada uno, para evitar errores lógicos en la salida de un programa.

```
int i = 0, j=0;  
//Post incremento, incremente la variable en la siguiente línea  
j = incrementarVariable(i++);  
//Pre incremento, incrementa la variable en la misma línea  
j = incrementarVariable(++i);
```

La función **incrementarVariable** suma 1 a la parámetro de entrada y lo devuelve



# Errores frecuentes

Cambiar el valor de una variable que se pasa como copia a una función, esperando modificar un valor externo.

No usar el valor que llega como copia o reasignarlo al hacer una operación, dentro de la función.

Uso incorrecto de los contextos, lo cual ocasiona un error en tiempo de compilación truculento de solucionar.

Enviar un dato no-primitivo a una función, esto hace que el paso de la variable no sea por copia, por tanto tiene un comportamiento diferente en relación con la manipulación de la variable.





# Ejercicios

Realice un programa que:

Use un ciclo for anidado para imprimir una matriz, declare los índices dentro de la estructura del ciclo, ¿Cuál es el contexto de cada índice?

¿Qué pasa si se define un bloque de código entre llaves, sin usar alguna de las estructuras de programación vistas?

¿Es posible declarar una variable en una función, con el mismo nombre de uno de sus parámetros?