

## Referencias en C++

En el lenguaje C++ existen dos implementaciones de referencia: los apuntadores y las referencias. Ambas implementaciones tienen en común que son variables que contienen una dirección de memoria; no obstante, las referencias deben inicializarse en el momento en que se crean, es decir, que no pueden ser inicializadas a un valor arbitrario. Adicionalmente, las referencias no pueden almacenar un valor NULL (a diferencia de los apuntadores que sí pueden hacerlo) y tampoco pueden reasignarse a otra variable.

Así, las referencias se pueden ver como una forma de darle un alias a una variable que está previamente definida, y se busca, mediante este alias, realizarle modificaciones a dicha variable.

A continuación, se muestra un ejemplo del uso de referencias:

```
void intercambio(int &a, int &b){  
    int temp = a;  
    a = b;  
    b = temp;  
}
```

**Figura 1: Ejemplo de referencias en C++**

Por otro lado, se cuenta con una “keyword” denominada *const*, la cual permite indicar algo que no se puede cambiar; esto se puede utilizar para indicar que no cambie una variable, punteros, referencias, clases, instancias de una clase y/o métodos de una clase.

Dicha palabra clave se caracteriza por tomarse en cuenta al momento de la compilación y por tener una representación física en la memoria RAM del computador, de tal manera que la declaración que tenga un *const* va a ser incrustado directamente en las instrucciones del procesador y así se logra evitar una asignación de valor a esta posición mediante el uso de apuntadores.

A continuación, se muestra una serie de ejemplos en los que se utiliza la sentencia *const*:

- `const int a = 90` (No se puede modificar el valor de la variable)
- `const int* a = 90` (No se puede modificar el contenido del apuntador)
- `int* const a = 90` (Se puede modificar el contenido del apuntador, pero no se puede reasignar el apuntador para que apunte a algo más)

Igualmente, *const* se puede utilizar dentro de un método `get` de una clase, de tal manera que no se puede modificar un atributo de la instancia de dicha clase.

## Bibliografía

- Coding or not. 2022. ¿Cuándo se utiliza «paso por referencia» y cuándo «paso por valor»? Tomado de: <https://codingornot.com/cuando-se-utiliza-paso-por-referencia-y-cuando-paso-por-valor>
- forCode. 2022. C++ Punteros V: Referencias. Tomado de: <https://www.forcode.es/lenguaje/net-categoria/c-punteros-v->

referencias/#:~:text=Las%20referencias%20son%20variables%20que,los%20operadores%20unarios%20\*%20%C3%B3%20%26.

- Microsoft Docs. 2022. *Referencias C++*. Tomado de: <https://docs.microsoft.com/es-es/cpp/cpp/references-cpp?view=msvc-170>