Fundamentos de Programación 101

By Ernie

Ernesto José Canales Guillén

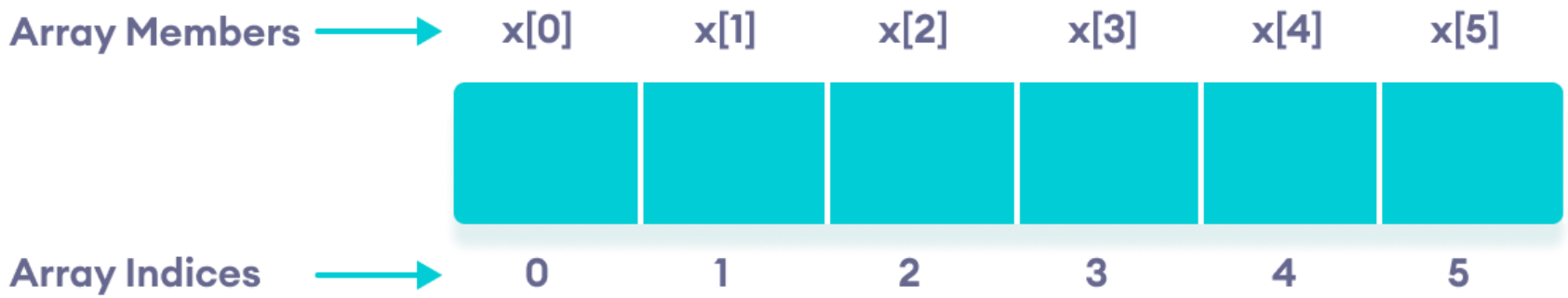
Círculos de estudio UCA

Ciclo Virtual 01/2021

Arreglos en C++

Arrays (Areglos)

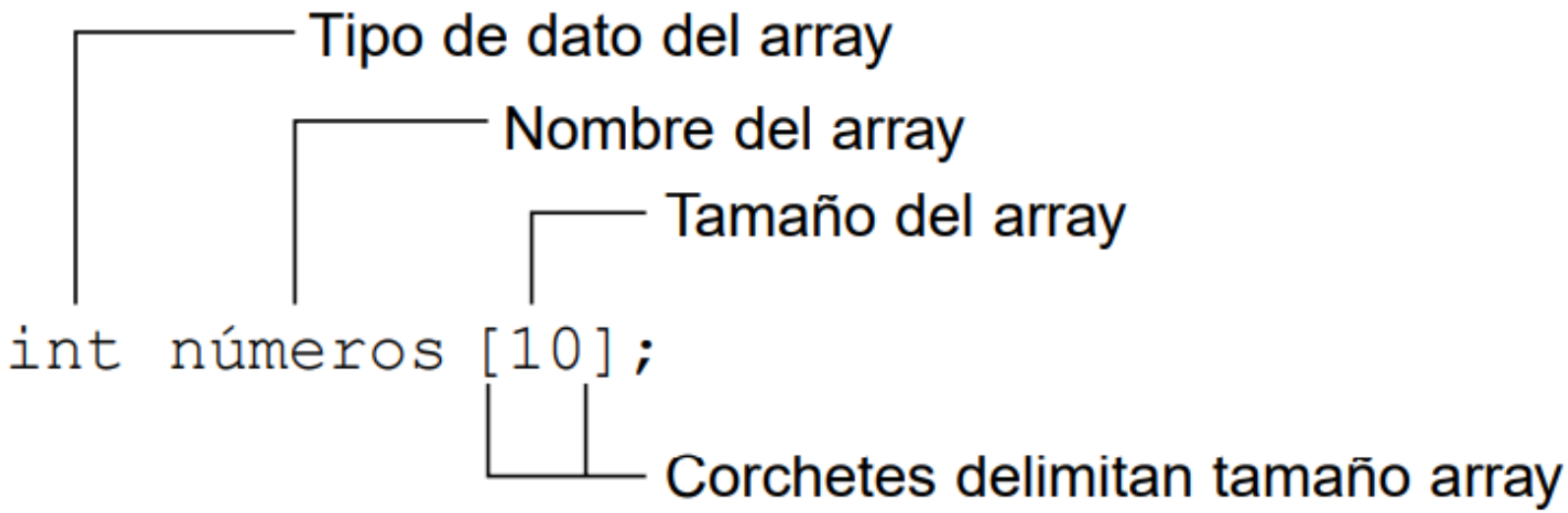
Un array o arreglo (lista o tabla) es una secuencia de objetos del mismo tipo. Los objetos se llaman elementos del array y se numeran consecutivamente 0, 1, 2, 3... El tipo de elementos almacenados en el array puede ser cualquier tipo de dato de C++, incluyendo estructuras definidas por el usuario.



Cada item del array se denomina elemento. Los elementos de un array se numeran, consecutivamente 0, 1, 2, 3... Estos números se denominan valores índice o subíndice del array.

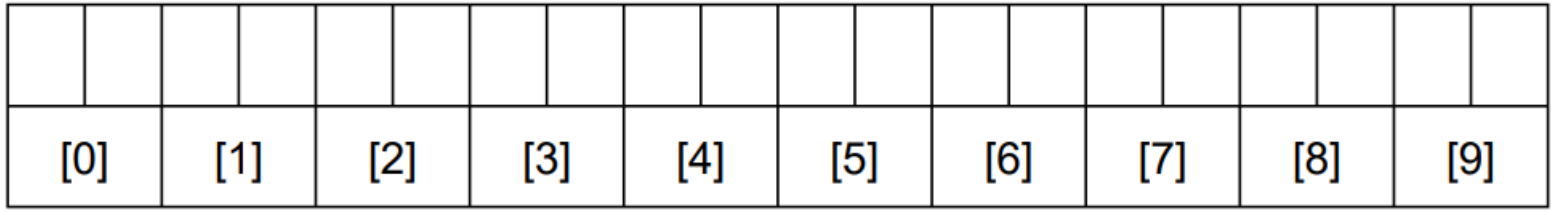
Declaración de un array

```
tipo nombreArray[numeroDeElementos];
```



Declaración de un array

Esta declaración hace que el compilador reserve espacio suficiente para contener diez valores enteros. En C++ los enteros ocupan, normalmente, 2 bytes, de modo que un array de diez enteros ocupa 20 bytes de memoria. La Figura muestra el esquema de un array de diez elementos; cada elemento puede tener su propio valor.



Acceso a los elementos de un array

Gran parte de la utilidad de un array proviene del hecho que se pueda acceder a los elementos de dicho array de modo individual. El método para acceder a un elemento es utilizar un subíndice o un índice.

```
nombreArray[n]
```

número del elemento dentro del array

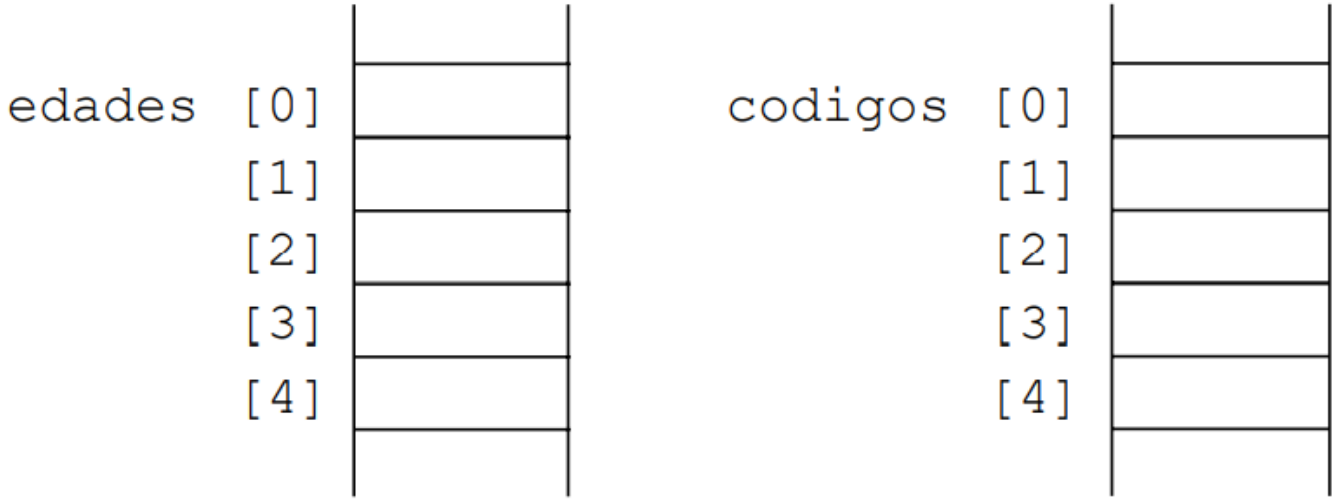
En los programas se pueden referenciar elementos utilizando fórmulas para los subíndices. Mientras que el subíndice puede evaluar a un entero, se puede utilizar una constante, una variable o una expresión para el subíndice.

```
edad[4]
ventas[total + 5]
bonos[mes]
salario[mes[i] * 5]
```

Almacenamiento en memoria de los arrays

Los elementos de los arrays se almacenan en bloques contiguos.

```
int edades[5];
char codigos[5];
```



El tamaño de los arrays (sizeof)

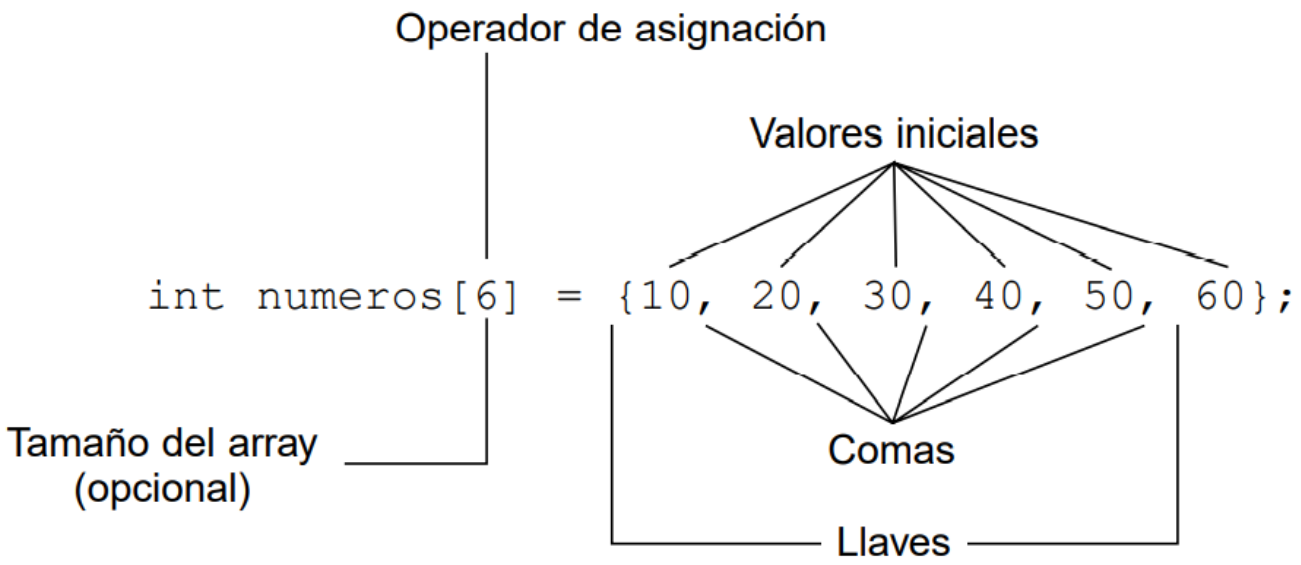
La función sizeof() devuelve el número de bytes necesarios para contener su argumento. Si se usa sizeof() para solicitar el tamaño de un array, esta función devuelve el número de bytes reservados para el array completo.

```
n = sizeof(edades);
```

Inicialización (iniciación) de un array

Se deben asignar valores a los elementos del array antes de utilizarlos, tal como se asignan valores a variables.

```
precios[0] = 10;
precios[1] = 20;
precios[3] = 30;
precios[4] = 40;
...
```



Omitir tamaño del array

No es necesario especificar el tamaño de la matriz. Pero si no lo hace, solo será tan grande como los elementos que se inserten en él:

```
string cars[] = {"Volvo", "BMW", "Ford"}; // size of array is always 3
```

Esto está completamente bien. Sin embargo, el problema surge si desea espacio adicional para elementos futuros. Entonces tienes que sobrescribir los valores existentes:

```
string cars[] = {"Volvo", "BMW", "Ford"};

string cars[] = {"Volvo", "BMW", "Ford", "Mazda", "Tesla"};
```

Arrays de caracteres y cadenas de texto

“Las cadenas se deben almacenar en arrays de caracteres, pero no todos los arrays de caracteres contienen cadenas”

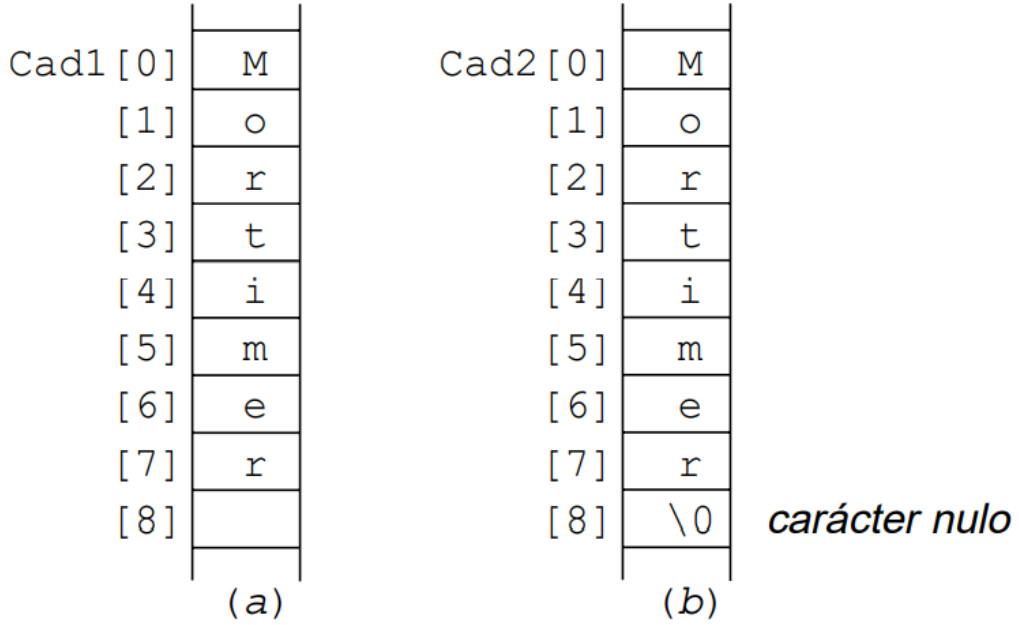


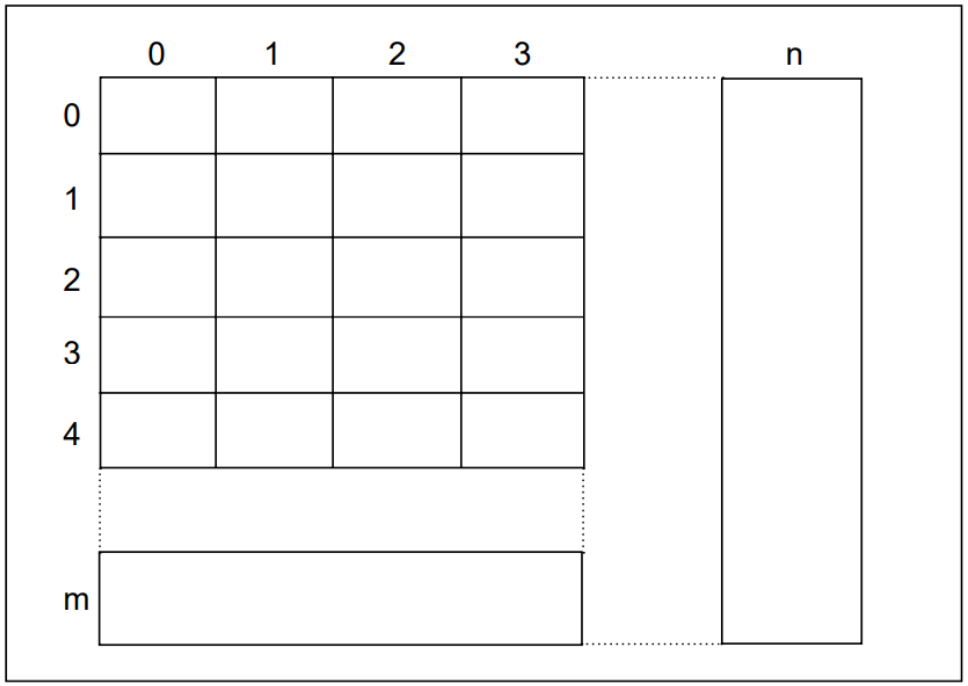
Figura 7.7. (a) Array de caracteres; (b) cadena.

Las cadenas se señalan incluyendo un carácter al final de la cadena: el carácter nulo (\0), cuyo valor en el código ASCII es 0.

Arrays multidimensionales

Arrays multidimensionales

- Los arrays vistos anteriormente se conocen como arrays unidimensionales y se caracterizan por tener un solo subíndice. Estos arrays se conocen también por el término listas.
- Los arrays multidimensionales son aquellos que tienen más de una dimensión y, en consecuencia, más de un índice. Los arrays más usuales son los de dos dimensiones, conocidos también por el nombre de tablas o matrices.



Sintaxis para la declaración de un array de dos dimensiones

<tipo de datoElemento> <nombre array> [<NúmeroDeFilas<] [<NúmeroDeColumnas>]

Algunos ejemplos de declaración de tablas son:

```
char Pantalla[25][80];
int puestos[6][8];
int equipos[4][30];
int matriz[4][2];
```

Arrays de más de dos dimensiones

Arrays de más de dos dimensiones

- C++ proporciona la posibilidad de almacenar varias dimensiones, aunque raramente los datos del mundo real requieren más de dos o tres dimensiones. El medio más fácil de dibujar un array de tres dimensiones es imaginar un cubo.

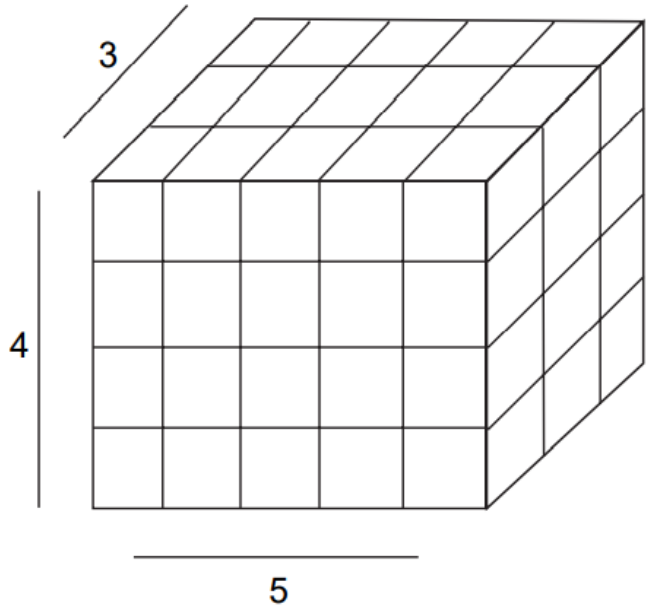


Figura 7.11. Un array de tres dimensiones (4 × 5 × 3).

```
int equipos[3][15][10];
```

Una aplicación práctica

El array libro tiene tres dimensiones [PAGINAS] [LINEAS] [COLUMNAS], que definen el tamaño del array. El tipo de datos del array es char, ya que los elementos son caracteres.

- ¿Cómo se puede acceder a la información del libro? El método más fácil es mediante bucles anidados.

Bibliografía

- L. J. Aguilar, Programación en C++. Algoritmos, estructuras de datos y objetos, Aravaca (Madrid): McGRAW-HILL, 2006.

- D. Malik, C++ Programming: From Problem Analysis to Program Design, Boston, MA: Cengage Learning, 2003.