Gestión de Arreglos y Punteros en C++: Teoría, Ejemplos y Memoria"

Por Ana Roncal

Contenido

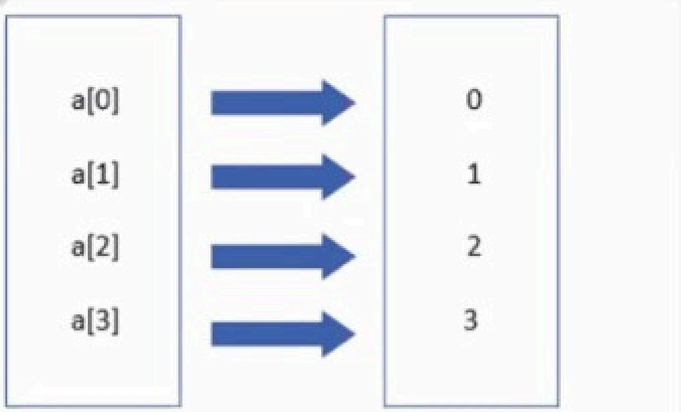
1 Arreglos unidimensionales

2 Arreglos bidimensionales

3 Punteros

1. Arreglos Unidimensionales

- **Definición:**
  - Un arreglo unidimensional es una estructura de datos que almacena una secuencia de elementos del mismo tipo, accesibles mediante un índice.
- **Declaración:**
  - `tipo nombreArreglo[tamaño];`
  - Ejemplo: `int numeros[5];`
- **Inicialización:**
  - `int numeros[5] = {1, 2, 3, 4, 5};`
  - `char letras[5] = {'a','b','c','d','e'};`
- **Gestión de Memoria:**
  - Se reserva un bloque contiguo de memoria para almacenar los elementos.
  - Cada elemento ocupa espacio en memoria según su tipo.



Ejemplo de Código - Arreglo unidimensional

```cpp
#include <iostream>
using namespace std;

int main() {
    int numeros[5] = {10, 20, 30, 40, 50};
    for(int i = 0; i < 5; i++) {
        cout << "Elemento en el índice " << i << ": " << numeros[i] << endl;
    }
    return 0;
}
```

Paso por Parámetros - Arreglo Unidimensional

Función:

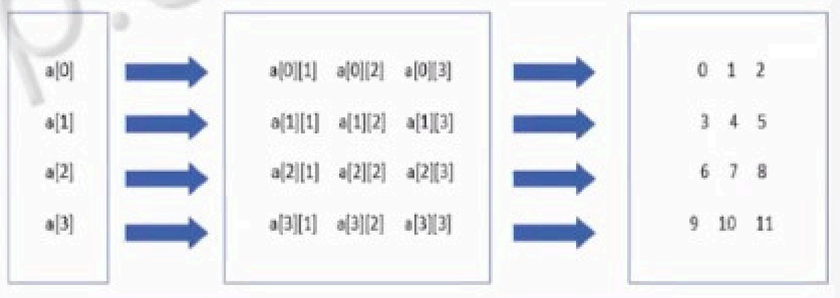
- `void imprimirArreglo(int arr[], int tamaño);`

Ejemplo de Uso:

```
void imprimirArreglo(int arr[], int tamaño) {
    for(int i = 0; i < tamaño; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
```

2. Arreglos Bidimensionales

- **Definición:**
  - Un arreglo bidimensional organiza elementos en filas y columnas con elementos del mismo tipo.
- **Declaración:**
  - `tipo nombreArreglo[filas][columnas];`
  - Ejemplo: `int matriz[3][3];`
- **Inicialización:**
  - `int matriz[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};`
- **Gestión de Memoria:**
  - La memoria se asigna contiguamente, organizada por filas.



Paso por Parámetros - Arreglo Bidimensional

Función:

- `void imprimirMatriz(int matriz[][3], int filas);`

Ejemplo de Uso:

```
void imprimirMatriz(int matriz[][3], int filas) {
    for(int i = 0; i < filas; i++) {
        for(int j = 0; j < 3; j++) {
            cout << matriz[i][j] << " ";
        }
        cout << endl;
    }
}
```

Cómo lo entendemos los programadores

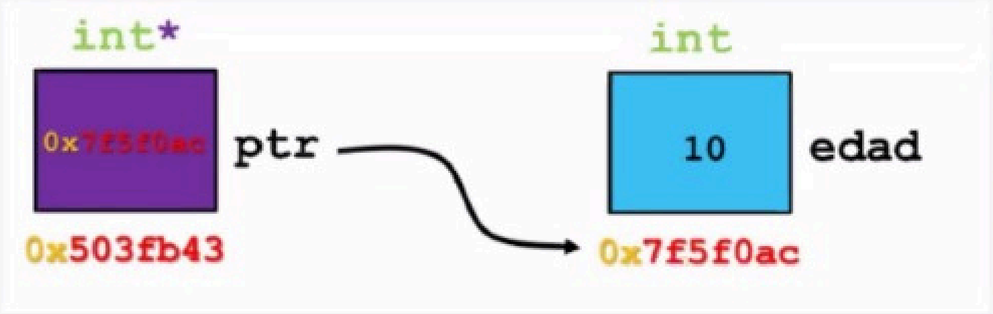
	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Cómo lo entiende la computadora

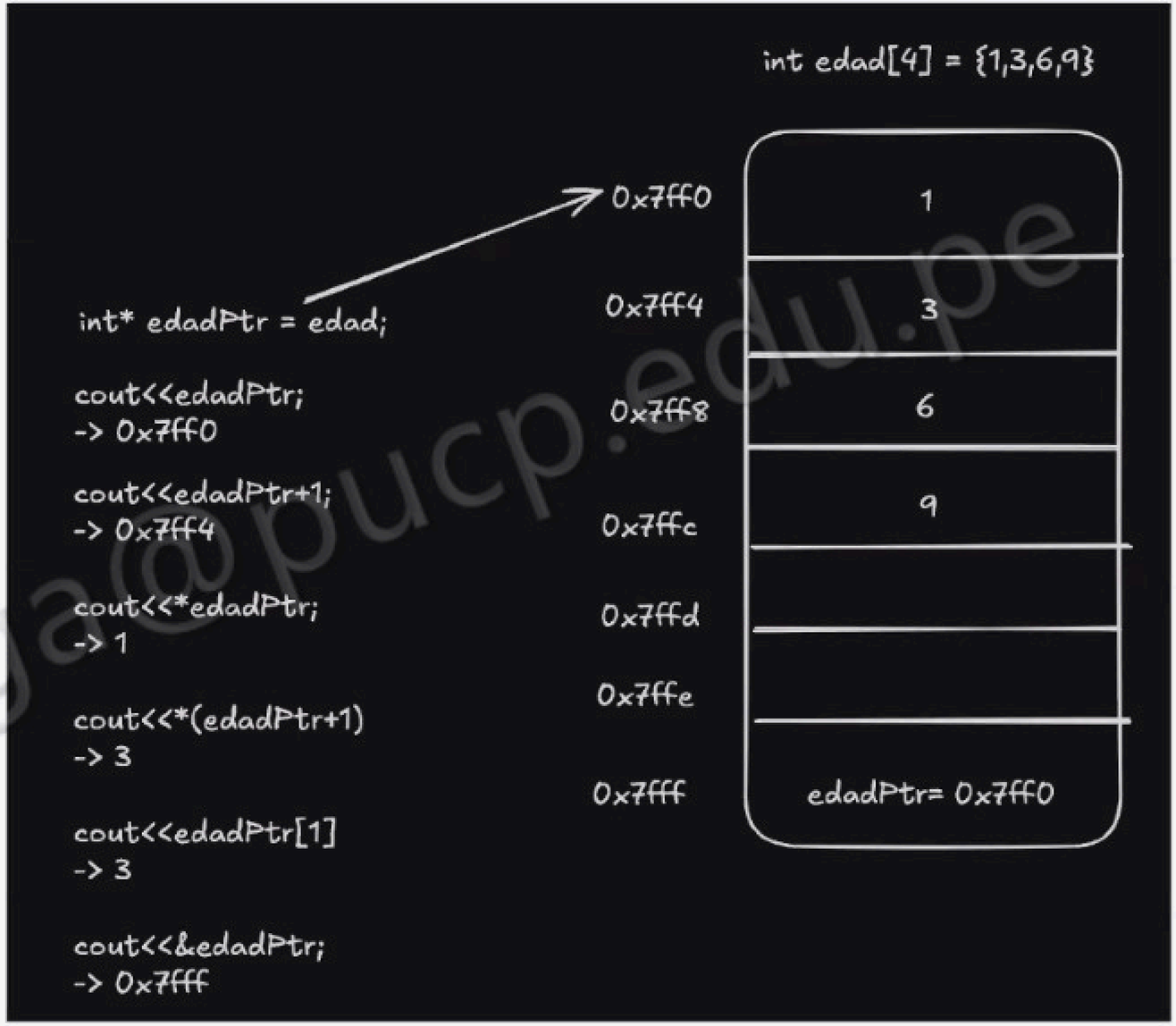
0	1	2	3	4	5	6	7	8
0x3+0	0x3+1	0x3+2	1x3+0	1x3+1	1x3+2	2x3+0	2x3+1	2x3+2
1	2	3	4	5	6	7	8	9

3. Punteros

- **Definición:**
  - Un puntero es una variable que almacena la dirección de memoria de otra variable.
- **Declaración:**
  - `tipo *nombrePuntero;`
  - Ejemplo: `int* p;`
- **Inicialización y Uso:**
  - `int x = 10;`
  - `int* p = &x;`



¿Cómo funciona las variables y la memoria?



Ejercicios Prácticos

Ejercicio 1:

- Crear un programa que lea un arreglo unidimensional de 10 enteros, los ordene e imprima.

Ejercicio 2:

- Escribir un programa que llene una matriz de 3x3 con números y calcule la suma de la diagonal principal.

Ejercicio 3:

- Implementar un programa que use punteros para intercambiar los valores de dos variables.