

Universidad Rafael Urdaneta  
Estructuras dinámicas de datos  
Sección "A" - 2022 B

### Asignación dinámica de arreglos

Hasta el momento se ha utilizado arreglos de tipo estático, por ejemplo, se crea un arreglo llamado `numeros[]` y obligatoriamente se le tenía que pasar el número de elementos, si se le ponían 10 elementos habíamos creado un arreglo que podría ser de cualquier tipo: entero, flotante, entre otros; y le habíamos puesto el número estático de elementos, este número no podría cambiarse, es decir, siempre el arreglo iba a tener 10 elementos `numeros[10]`.

La desventaja de esto es que si solamente se utilizaban 5 elementos, tendríamos memoria que no se utilizaría, esta memoria se desperdiciaba y eso no es una buena práctica de programación.

Para que esto no suceda, se utilizan los arreglos dinámicos mediante la asignación dinámica de arreglos, con estos arreglos sólo se utiliza el espacio en memoria necesario para cada arreglo, para ello, se utilizan dos operadores muy importantes, los cuales son:

- **New**: este operador se encarga de reservar el número de bytes solicitados para la declaración, es decir, el `new` nos va a ayudar a reservar el espacio de memoria requerido para no desperdiciar memoria para el arreglo que se crea.





• Delete: Este se encarga de liberar un bloque de bytes reservados con anterioridad, es decir, vamos a liberar la memoria que se había utilizado o que se había creado/reservado con el operador new. Para esto, debemos digitar el delete y el nombre de el arreglo creado antes de finalizar el programa.

• Ejercicio propuesto: Pedir al usuario N números, almacenarlos en un arreglo dinámico, posteriormente ordenar los números utilizando el método burbuja en orden ascendente y mostrarlos en pantalla.

// Programa en C++

#include <iostream>

#include <conio.h>

#include <stdlib.h> // Librería que incluye el new y delete

using namespace std;

// Prototipos de Funciones

void Pedir\_datos();

void ordenar\_numeros(int\*, int);

void Mostrar\_arreglo\_ordenado(int\*, int);

int numero\_elementos;

int \*elementos;

int main(){

    Pedir\_datos();

    ordenar\_numeros(elementos, numero\_elementos);

    Mostrar\_arreglo\_ordenado(elementos, numero\_elementos);

delete[] elementos; // Liberar la memoria utilizada para el arreglo dinámico

getch();

return 0;

}

// Función utilizada para pedirle los datos al usuario

void Pedir\_datos(){

cout << "Introduce el numero de elementos: ";

cin >> numero\_elementos;

elementos = new int[numero\_elementos]; // Crear el arreglo

for (int i = 0; i < numero\_elementos; i++){

cout << "Introduce un numero[" << i << "]: ";

cin >> \*(elementos + i); // elementos[i]

}

}

// Ordenamiento por método burbuja con punteros

void Ordenar\_numeros(int \*elementos, int numero\_elementos){

int aux;

for (int i = 0; i < numero\_elementos; i++){

for (int j = 0; j < numero\_elementos - 1; j++){

if (\*(elementos + j) > \*(elementos + j + 1)){

aux = \*(elementos + j);

\*(elementos + j) = \*(elementos + j + 1);

\*(elementos + j + 1) = aux;

}

}

}

}





```
//Función para mostrar el arreglo ordenado
void Mostrar_arreglo_ordenado(int *elementos, int numero_elementos){
    cout << "\n\n Mostrando el Arreglo ordenado: ";
    for(int i=0; i<numero_elementos; i++){
        cout << *(elementos + i) << " ";
    }
}
```