

CUATRIMESTRE 2°



ESTRUCTURA DE DATOS

Profesor: Valdez Hernandez Juanh
Augusto

Presentado por: Khevin Cruz Hernández

Matricula: 2331123331

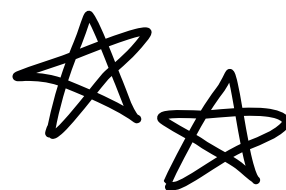
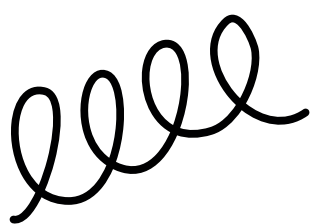
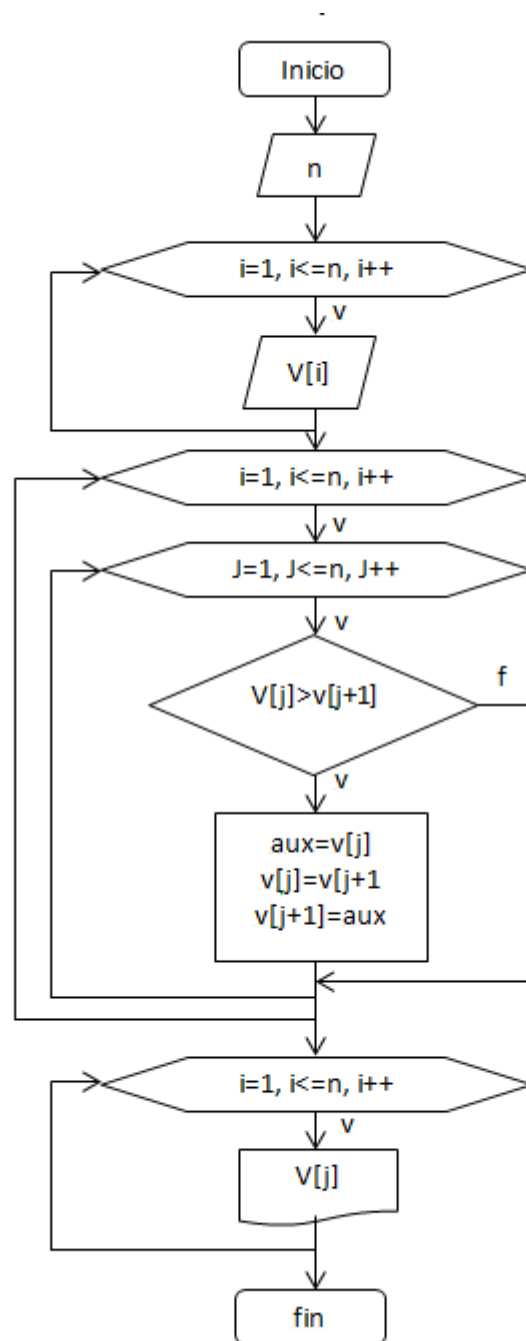
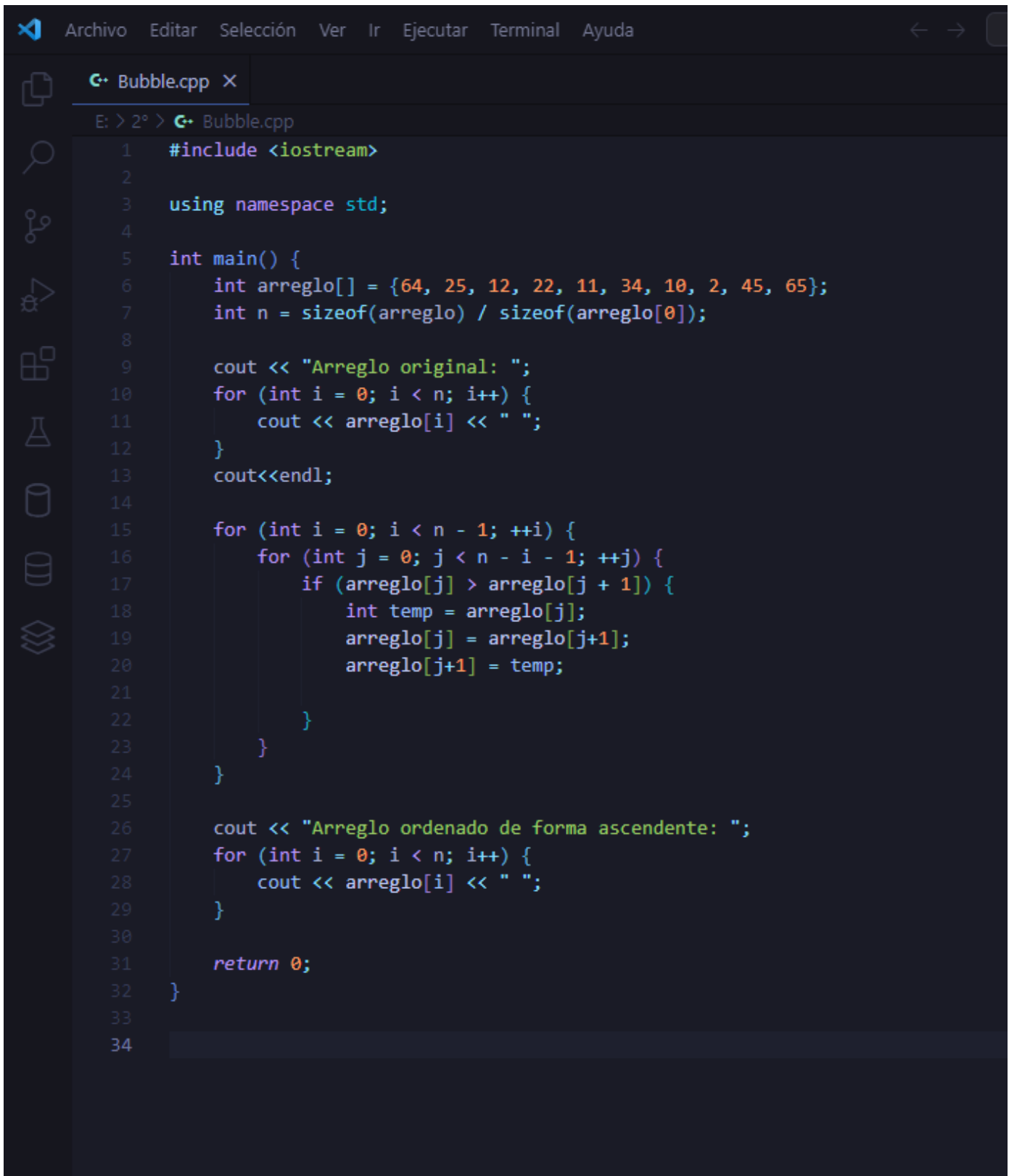


Diagrama de flujo



Código



The image shows a code editor window with a dark theme. The title bar at the top contains the menu items: Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, and Ayuda. Below the title bar, there is a tab labeled "Bubble.cpp" with a close button. The editor area shows the following C++ code:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int arreglo[] = {64, 25, 12, 22, 11, 34, 10, 2, 45, 65};
7      int n = sizeof(arreglo) / sizeof(arreglo[0]);
8
9      cout << "Arreglo original: ";
10     for (int i = 0; i < n; i++) {
11         cout << arreglo[i] << " ";
12     }
13     cout<<endl;
14
15     for (int i = 0; i < n - 1; ++i) {
16         for (int j = 0; j < n - i - 1; ++j) {
17             if (arreglo[j] > arreglo[j + 1]) {
18                 int temp = arreglo[j];
19                 arreglo[j] = arreglo[j+1];
20                 arreglo[j+1] = temp;
21             }
22         }
23     }
24
25     cout << "Arreglo ordenado de forma ascendente: ";
26     for (int i = 0; i < n; i++) {
27         cout << arreglo[i] << " ";
28     }
29
30     return 0;
31 }
32
33
34
```

Implementación

Para poner en marcha el código se busca un entorno de desarrollo integrado (IDE) que admita C++, se crea un archivo nuevo y copia el código del algoritmo de ordenamiento de burbuja en el archivo. Una vez teniendo el código se compila y si no hay errores de compilación, se creará un archivo ejecutable, deberíamos ver la salida del programa, que incluirá el arreglo original y el arreglo ordenado.

Siguiendo estos pasos, podemos poner en marcha el código, probar su ejecución y realizar ediciones según sea necesario. Es importante tener en cuenta que el proceso puede variar ligeramente dependiendo del entorno de desarrollo y del sistema operativo que se este utilizando, en este caso se uso el lenguaje de programación C++ en el entorno de desarrollo; Microsoft Visual Studio.



Resultados

El resultado arrojado del algoritmo es un arreglo ordenado de forma ascendente usando el método de ordenamiento de burbuja:

```
Arreglo original: 64 25 12 22 11 34 10 2 45 65
```

Después de aplicar el algoritmo de ordenamiento de burbuja, el arreglo se ha ordenado:

```
Arreglo ordenado de forma ascendente: 2 10 11 12 22 25 34 45 64 65
```

Esto muestra la entrada original y el arreglo resultante después de aplicar el algoritmo de ordenamiento de burbuja.

Conclusión

El algoritmo de ordenamiento de burbuja es un método sencillo pero ineficiente para ordenar elementos en un arreglo. A medida que el tamaño del arreglo aumenta, el tiempo de ejecución del algoritmo también aumenta significativamente.

Aunque el algoritmo de burbuja es fácil de entender e implementar, su rendimiento se ve superado por otros algoritmos de ordenamiento más eficientes, como el ordenamiento rápido o el ordenamiento por mezcla, que tienen complejidades temporales más bajas y, por lo tanto, son más adecuados para aplicaciones en tiempo real o con grandes conjuntos de datos.

En resumen, aunque el algoritmo de ordenamiento de burbuja puede ser útil para pequeñas cantidades de datos o para fines educativos, su ineficiencia lo hace poco práctico para aplicaciones en producción donde se requiere un rendimiento óptimo en el procesamiento de grandes conjuntos de datos.