# ESCUELA POLITÉCNICA NACIONL

**ESFOT**

* **Curso:** Gr1
* **Tarea:**  4
* **Nombres y apellidos:** Erick Omar Vargas Albán
* **Fecha de asignación:** 13/11/2018 **Fecha de entrega:** 20 /11/2018
* **Tarea:** Deber
* **Implementación método de ordenamiento burbuja**
* **Resolución**

**¿Qué es el método burbuja?**

Sencillo algoritmo que funciona revisando cada elemento de la lista intercambiándolos de posición si están en orden equivocado, revisa la lista tantas veces hasta verificar que todos los elementos estén en la correcta posición.

**Método Implementado en compilador C++**

**//Implemetacion algorimo burbuja**

**#include <iostream>**

**#include <conio.h>**

**#include <ctime> //libreria para analizar el tiempo de ejecucion**

**using namespace std;**

**int main(){**

**int numeros []={4,1,3,2,5};//arreglo a ordenar**

**int i, j, aux;// variables para bucles i,j ... aux va a hacer el numero que vaya intercambiándose**

**//algoritmo metodo burbuja**

**for(i=0; i<=5; i++){ //tome la cantidad de elementos**

**for (j=0; j<5; j++){//arregle la cantidad de elementos**

**if(numeros[j]>numeros[j+1]){//numero actual es numero siguiente en la condicion**

**aux=numeros[j];//si cumple se hace el intercambio al número actual**

**numeros [j]=numeros[j+1];//numero actual va a se rigual a numero siguiente**

**numeros[j+1]=aux;//numero siguiente va a hacer igual al espacio que ocupa**

**}**

**}**

**}**

**cout<<"ordenamento burbuja"<<endl;**

**for(i=0; i<5; i++){//bucle para la impresion del ordenamiento**

**cout<<numeros[i]<<" ";**

**}**

**cout<<endl;**

**unsigned t0, t1; //Variables para reducir el tiempo de ejecucion**

**t0=clock();**

**t1 = clock();**

**float time = (float(t1-t0)/CLOCKS\_PER\_SEC);//CLOCKS\_PER\_SEC nos permite traducir el tiempo a segundos**

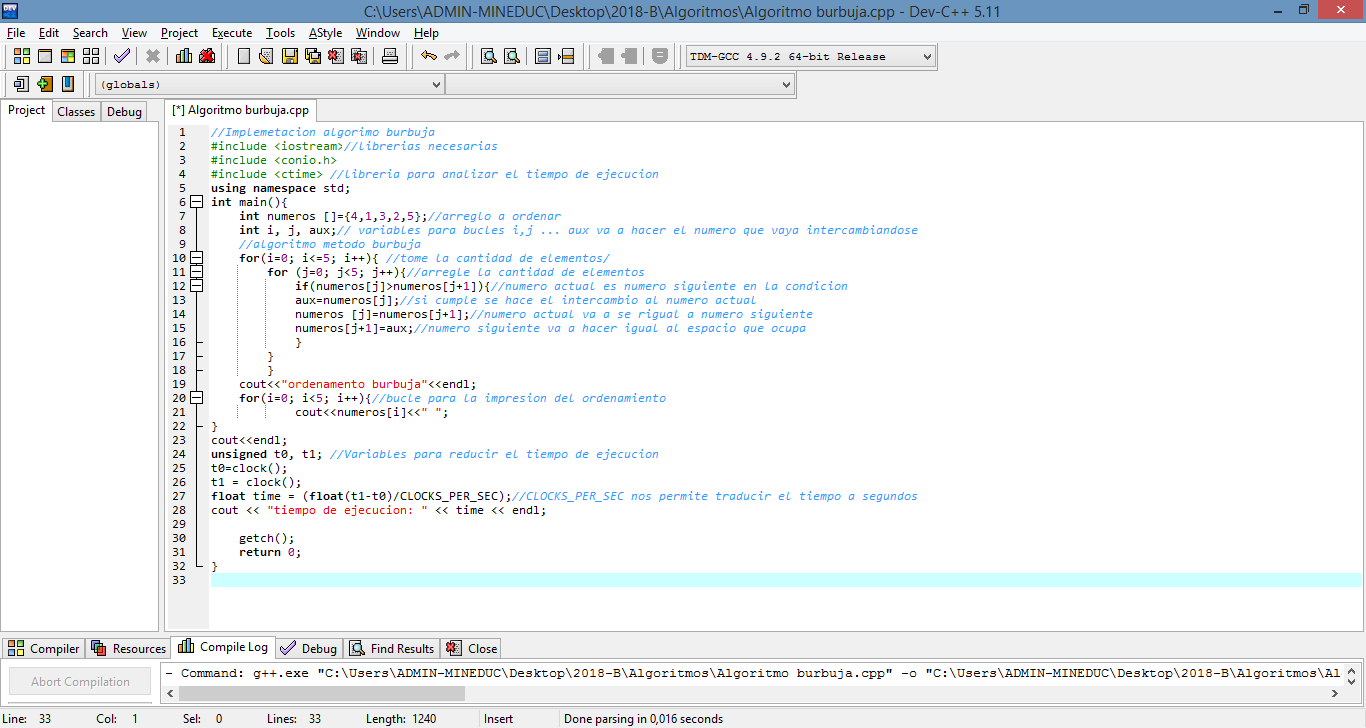
**cout << "tiempo de ejecucion: " << time << endl;**

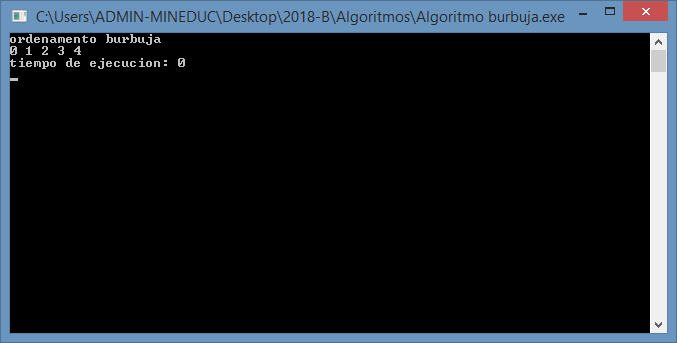
**getch();**

**return 0;**

**}**

**Resultados**





**Conclusiones**

\_EL código funciona de forma eficaz y ha sido reducido al máximo para su ejecución

\_En el compilador c++ no hay una librería que presente en milisegundo el tiempo de ejecución pero a un valor de intervalos el código se ejecuta entre 2 a 3 milisegundos

\_En este algoritmo las comparaciones se repiten una y otra vez tratar por más que se reduzca el código sigue cumpliendo el mismo objetivo

**Recomendaciones**

\_Ahorramos líneas de código en declarar desde un inicio las variables

\_Almacenar el número que cambia en una variable auxiliar esta permitirá más fácil un cambio entre elementos