**Equipo:**

Bubble Sort

**Integrantes:**

Juan Pablo Patiño Florez

Sebastian Mira Castro

**Ficha:**

2251848

**Instructor:**

Hugo Hernan Henao

**Análisis y Desarrollo de Sistemas de Información**

**SENA:**

Centro de Comercio y Turismo

**Fecha:**

12 Octubre del 2021

**Bubble Sort**

La Ordenación de burbuja (Bubble Sort) es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente elemento, intercambiándolos de posición si están en el orden equivocado.

**Código C# - Consola**

using System;

using System.IO;

using System.Diagnostics;

namespace Bubble\_Sort

{

class Program

{

static void Main(string[] args)

{

//Iniciando El Cronometro

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

//Solicitando el tamaño del vector

int TamanioVector;

Console.WriteLine("Hola, Fecha Y Hora De Inicio De La Prueba : " + DateTime.Now);

Console.WriteLine("Ingrese El Tamaño Del Vector");

TamanioVector = Convert.ToInt32(Console.ReadLine());

//Declaro el vector

int[] Lista = new int[TamanioVector];

//Generando Numeros Random

Random rnd = new Random();

//Añadiendo Los Valores A La Lista

for (int i = 0; i < Lista.Length; i++)

{

Lista[i] = rnd.Next(-1000, 1000);

}

//Imprimiendo La Lista(No Ordenada)

Console.WriteLine("Imprimiendo La Lista");

for (int i = 0; i < Lista.Length; i++)

{

//Console.WriteLine(Lista[i]);

TextWriter Vector = new StreamWriter(@"C:\Users\SENA\3D Objects\Bubble Sort\Bubble Sort\Bubble\" + "TotalDe" + TamanioVector + "Datos" + ".txt", true);

Vector.WriteLine(Lista[i]);

Vector.Close();

}

//Implementando Bubble Sort, para acomodar valores de Menor a Mayor.

int temp = 0;

for (int i = 0; i < Lista.Length; i++)

{

for (int j = 0; j < Lista.Length; j++)

{

if (Lista[i] < Lista[j])

{

temp = Lista[i];

Lista[i] = Lista[j];

Lista[j] = temp;

}

}

}

for (int i = 0; i < Lista.Length; i++)

{

Console.Write("({0}), ", Lista[i]);

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine();

//Finalizando El Cronometrado

stopwatch.Stop();

TimeSpan ts = stopwatch.Elapsed;

Console.WriteLine("El codigo se ha demorado en Ejecutarse Un lapso de " + ts);

}

}

}

**Total de líneas del código realizado en C#: 72**

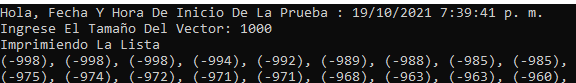
**Gráfica de Gantt**

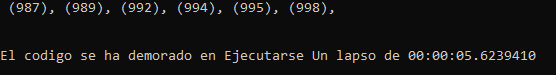
| **Actividades** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **21** | **24** | **25** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Investigar acerca del Algoritmo Bubble Sort. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Investigar e implementar un cronómetro en el Algoritmo.  Formato: HH:MM:SS:Milisegundos. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Investigar cómo almacenar datos en un Archivo txt. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Investigar cómo generar números aleatorios en un Algoritmo. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Mostrar los números aleatorios almacenados en un archivo txt. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Mostrar números ordenados con el Algoritmo de ordenamiento Bubble Sort. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Presupuesto Bubble Sort |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pruebas. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Tabla de pruebas**

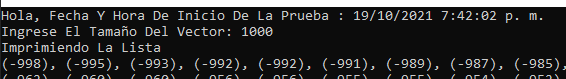
| Cantidad de números. | Resultado 1 | Resultado 2 |
| --- | --- | --- |
| 1.000 | El código se ha demorado en Ejecutarse Un lapso de **00:00:05.6239410.** | El código se ha demorado en Ejecutarse Un lapso de **00:00:02.9598616.** |
| 5.000 | El código se ha demorado en Ejecutarse Un lapso de **00:00:06.2248492.** | El código se ha demorado en Ejecutarse Un lapso de **00:00:05.4893410.** |
| 10.000 | El código se ha demorado en Ejecutarse Un lapso de **00:00:09.3258703.** | El código se ha demorado en Ejecutarse Un lapso de **00:00:07.9396571.** |
| 20.000 | El código se ha demorado en Ejecutarse Un lapso de **00:00:14.2902369.** | El código se ha demorado en Ejecutarse Un lapso de **00:00:13.3749033.** |
| 50.000 | El código se ha demorado en Ejecutarse Un lapso de **00:00:36.3242542.** | El código se ha demorado en Ejecutarse Un lapso de **00:00:35.1971173.** |
| 100.000 | El código se ha demorado en Ejecutarse Un lapso de **00:01:36.4014927.** | El código se ha demorado en Ejecutarse Un lapso de **00:01:37.0511855.** |

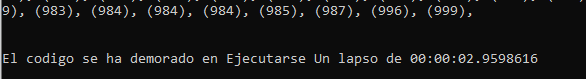
**Número 1.000 - Resultado 1:**



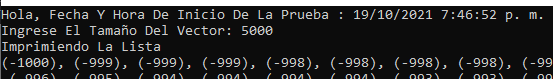


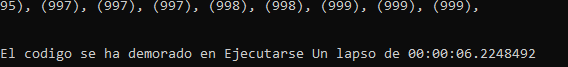
**Número 1.000 - Resultado 2:**



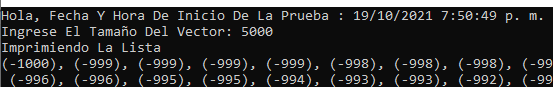


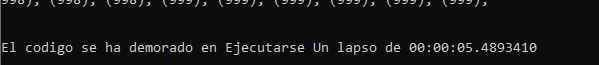
**Número 5.000 - Resultado 1:**

****

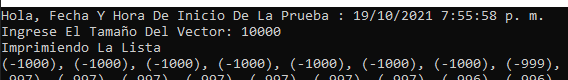
****

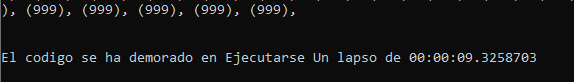
**Número 5.000 - Resultado 2:**

****

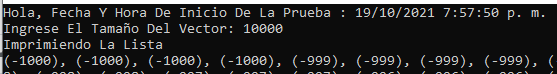
****

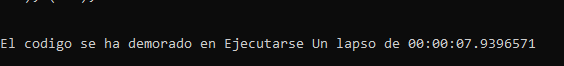
**Número 10.000 - Resultado 1:**



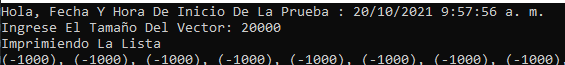


**Número 10.000 - Resultado 2:**



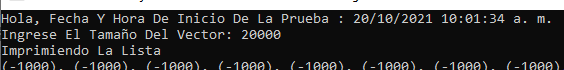


**Número 20.000 - Resultado 1:**



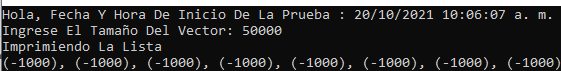


**Número 20.000 - Resultado 2:**



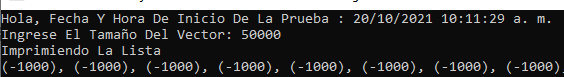


**Número 50.000 - Resultado 1:**

****

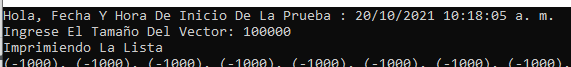
****

**Número 50.000 - Resultado 2:**

****

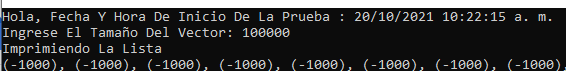
****

**Número 100.000 - Resultado 1:**

****

****

**Número 100.000 - Resultado 2:**

****

****