

```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleGuilherme
{
    public class Funcoes
    {
        Random rand = new Random();
        Stopwatch stopWatch = new Stopwatch(); //declaração da contagem

        public string WorkFlow()
        {
            ExibirMenu(); //menu de opções
            string num = Console.ReadLine();
            int Numero = Convert.ToInt16(num);

            while (Numero <= 6) //Loop Software
            {
                switch (Numero) //Verfifica digito do usuário para realizar algum proceimento
                {
                    case 1:
                        InsertSort();
                        break;
                    case 2:
                        SelectionSort();
                        break;
                    case 3:
                        CombSort();
                        break;
                    case 4:
                        ShellSort();
                        break;
                    case 5:
                        GnomeSort();
                        break;
                    case 6:
                        break;
                    default:
                        break;
                }
                if (Numero != 6)
                {
                    Console.Clear(); //limpa tela
                    ExibirMenu();
                    num = Console.ReadLine();
                    Numero = Convert.ToInt16(num);
                }
                else
                {
                    Console.WriteLine("SAINDO...");
                    break;
                }
            }
            return num;
        }

        public void InsertSort()
        {
            Random rand = new Random();

            int[] vetor = new int[100]; //para criar vetor desordenado de 100 posições
            for (int i = 0; i < vetor.Length; i++)
            {
                vetor[i] = rand.Next(100);
            }
        }
    }
}

```

```

        Console.WriteLine("Insert Sort\n"); //nome do algoritmo
        Console.WriteLine("Vetor Original: ");
        Imprimir(vetor); //mostra pro usuário o vetor desordenado criado aleatoriamente

        int p = 0;
        int aux;

        IniciarTempo(); // inicio da contagem do tempo
        while (p < (vetor.Length - 1))
        {
            if (vetor[p] > vetor[p + 1])
            {
                aux = vetor[p];
                vetor[p] = vetor[p + 1];
                vetor[p + 1] = aux;
                if (p > 0)
                {
                    p -= 2;
                }
            }
            p++;
        }
        PararTempo(); // fim da contagem do tempo

        Console.WriteLine();
        Console.WriteLine("Vetor Ordenado: ");
        Imprimir(vetor); //mostra pro usuário o vetor ordenado
        MostrarTempoDecorrido(); //mostra o tempo de execução
        Console.WriteLine("\n\nTecle 'ENTER' para Próximo.");
        Console.ReadLine();
    }

    public void ShellSort()
    {
        int[] vetor = new int[100]; //para criar vetor desordenado de 100 posições
        for (int i = 0; i < vetor.Length; i++)
            vetor[i] = rand.Next(100);

        Console.WriteLine("Shell Sort\n"); //nome do algoritmo
        Console.WriteLine("Vetor Original: ");
        Imprimir(vetor); //mostra pro usuário o vetor desordenado criado aleatoriamente

        int h = 1;
        int n = vetor.Length;

        IniciarTempo(); // inicio da contagem do tempo
        while (h < n)
            h = h * 3 + 1;

        h = h / 3;
        int c, j;
        while (h > 0)
        {
            for (int i = h; i < n; i++)
            {
                c = vetor[i];
                j = i;
                while (j >= h && vetor[j - h] > c)
                {
                    vetor[j] = vetor[j - h];
                    j = j - h;
                }
                vetor[j] = c;
            }
            h = h / 2;
        }
        PararTempo(); // fim da contagem do tempo
    }

```

```

        Console.WriteLine();
        Console.WriteLine("Vetor Ordenado: ");
        Imprimir(vetor); //mostra pro usuário o vetor ordenado
        MostarTempoDecorrido();
        Console.WriteLine("\n\nTecle 'ENTER' para Próximo."); //opção pro usuário
        Console.ReadLine();
    }

    public void CombSort()
    {
        int[] vetor = new int[100]; //para criar vetor desordenado de 100 posições
        for (int i = 0; i < vetor.Length; i++)
            vetor[i] = rand.Next(100);

        Console.WriteLine("CombSort\n"); //nome do algoritmo
        Console.WriteLine("Vetor Original: ");
        Imprimir(vetor); //mostra pro usuário o vetor desordenado criado aleatoriamente

        int gap = vetor.Length;
        bool swapped = true;

        IniciarTempo(); // início da contagem do tempo
        while (gap > 1 || swapped)
        {
            if (gap > 1)
                gap = (int)(gap / 1.247330950103979);

            int i = 0;
            swapped = false;
            while (i + gap < vetor.Length)
            {
                if (vetor[i].CompareTo(vetor[i + gap]) > 0)
                {
                    int t = vetor[i];
                    vetor[i] = vetor[i + gap];
                    vetor[i + gap] = t;
                    swapped = true;
                }
                i++;
            }
        }
        PararTempo(); // fim da contagem do tempo

        Console.WriteLine();
        Console.WriteLine("Vetor Ordenado: ");
        Imprimir(vetor); //mostra pro usuário o vetor desordenado criado aleatoriamente
        MostarTempoDecorrido(); //mostra o tempo de execução
        Console.WriteLine("\n\nTecle 'ENTER' para Próximo.");
        Console.ReadLine();
    }

    public void SelectionSort()
    {
        int[] vetor = new int[100]; //para criar vetor desordenado de 100 posições
        for (int i = 0; i < vetor.Length; i++)
            vetor[i] = rand.Next(100);

        Console.WriteLine("Selection Sort\n"); //nome do algoritmo
        Console.WriteLine("Vetor Original: ");
        Imprimir(vetor); //mostra pro usuário o vetor desordenado criado aleatoriamente

        int min, aux;

        IniciarTempo(); //início da contagem do tempo
        for (int i = 0; i < vetor.Length - 1; i++)
        {
            min = i;

            for (int j = i + 1; j < vetor.Length; j++)

```

```

        if (vetor[j] < vetor[min])
            min = j;

    if (min != i)
    {
        aux = vetor[min];
        vetor[min] = vetor[i];
        vetor[i] = aux;
    }
}
PararTempo(); //fim da contagem do tempo

Console.WriteLine();
Console.WriteLine("Vetor Ordenado: ");
Imprimir(vetor);
MostarTempoDecorrido(); //mostra pro usuário o vetor desordenado criado aleatoriamente
Console.WriteLine("\n\nTecle 'ENTER' para Próximo.");
Console.ReadLine();
}

public void GnomeSort()
{
    int[] vetor = new int[100]; //para criar vetor desordenado de 100 posições
    for (int i = 0; i < vetor.Length; i++)
        vetor[i] = rand.Next(100);

    Console.WriteLine("Gnome Sort\n"); //nome do algoritmo
    Console.WriteLine("Vetor Original: ");
    Imprimir(vetor); //mostra pro usuário o vetor desordenado criado aleatoriamente

    int min, aux;

    IniciarTempo(); //inicio da contagem do tempo
    for (int i = 0; i < vetor.Length - 1; i++)
    {
        min = i;

        for (int j = i + 1; j < vetor.Length; j++)
            if (vetor[j] < vetor[min])
                min = j;

        if (min != i)
        {
            aux = vetor[min];
            vetor[min] = vetor[i];
            vetor[i] = aux;
        }
    }
    PararTempo(); //fim da contagem do tempo

    Console.WriteLine();
    Console.WriteLine("Vetor Ordenado: ");
    Imprimir(vetor); //mostra pro usuário o vetor ordenado
    MostarTempoDecorrido(); //mostra o tempo de execução
    Console.WriteLine("\n\nTecle 'ENTER' para Próximo.");
    Console.ReadLine();
}

void Imprimir(int[] vetor)
{
    for (int i = 0; i < vetor.Length; i++)
        Console.Write(vetor[i].ToString() + ",");
    Console.WriteLine();
}

public void ExibirMenu() //exibe menu de opções
{
    Console.WriteLine("-----MENU-----");
}

```

```

        Console.WriteLine("Escolha uma das opções abaixo:");
        Console.WriteLine("1 - Inser Sort;");
        Console.WriteLine("2 - Selection Sort;");
        Console.WriteLine("3 - Comb Sort;");
        Console.WriteLine("4 - Shell Sort;");
        Console.WriteLine("5 - Gnome Sort;");
        Console.WriteLine("6 - Sair;");

        Console.Write("\n\nSUA OPÇÃO: ");
    }

    public void IniciarTempo()
    {
        stopwatch.Reset();//zerar tempo
        stopwatch.Start();//iniciar cronômetro
    }

    public void PararTempo()
    {
        stopwatch.Stop();//parar cronômetro
    }

    public void MostarTempoDecorrido()
    {
        Console.WriteLine();
        Console.Write("TEMPO:      " + stopwatch.Elapsed); //mostra tempo de execução
    }
}
}

```