

Laboratorio Grupal Práctica 03

Curso de Algoritmia y Programación

Prof. Luis E. Garreta U.
Universidad Autonoma de Occidente – Cali
Depto. Operaciones y Sistemas
Facultad de Ingeniería

30 de octubre de 2018

La siguiente práctica se puede desarrollar en grupos y es preparatoria para la parte práctica del tercer corte. OJO: Si la hace en grupos asegúrese que entiendan los dos lo que están haciendo ya que la práctica del examen es individual y con tiempo fijo.

1. Problema:

Se debe implementar en visual C# una aplicación para leer los datos de N jugadores, ordenar sus resultados e imprimir dos reportes: uno después de leer los datos y el otro después de ordenarlos, así:

- El algoritmo principal (Figura 1) primero lee el número de personas que van a jugar (Figura 2), y después lee sus nombres y sus puntajes, los cuales los guarda en dos arreglos uno para nombres y otro para puntajes, respectivamente. Crea un reporte con estos datos y lo imprime.
- Después se llama al subproceso externo **ordenamientoBurbuja** que toma como entrada los dos arreglos (ya llenos) y su tamaño N , los ordena de menor a mayor de acuerdo al valor del puntaje y así mismo se ordena el arreglo de nombres. Crea un reporte con estos datos y lo imprime. El proceso externo de **ordenamientoBurbuja** ya existe, es decir solo debe llamarlo correctamente (no implementarlo) y después en visual C# usar la librería de funciones que se le entregará (ver más adelante).
- De código le entregamos la forma principal con dos eventos: *iniciar* y *salir*. En el evento *iniciar* se llama a una función **principal** que es donde se implementa todo el código del proceso principal. Como ayuda para esta ocasión se le entrega el código del principal y solo falta implementar las funciones descritas en los flujogramas.
- La librería de funciones que se le va a entregar contiene solo la función **ordenamientoBurbuja**. Las otras funciones que aparecen llamadas en el proceso principal las debe implementar usted de forma exacta a como aparecen llamadas en el proceso principal.
- La entrada de datos se hace con `inputBox` y la salida se hace con `messageBox`. Revise el código principal para que vea como se utilizan y así los utilice en el subproceso de *leerDatos* donde se hace entrada del nombre y el puntaje.

2. Diagramas de Flujo Subprocesos

Figura1 Proceso Principal

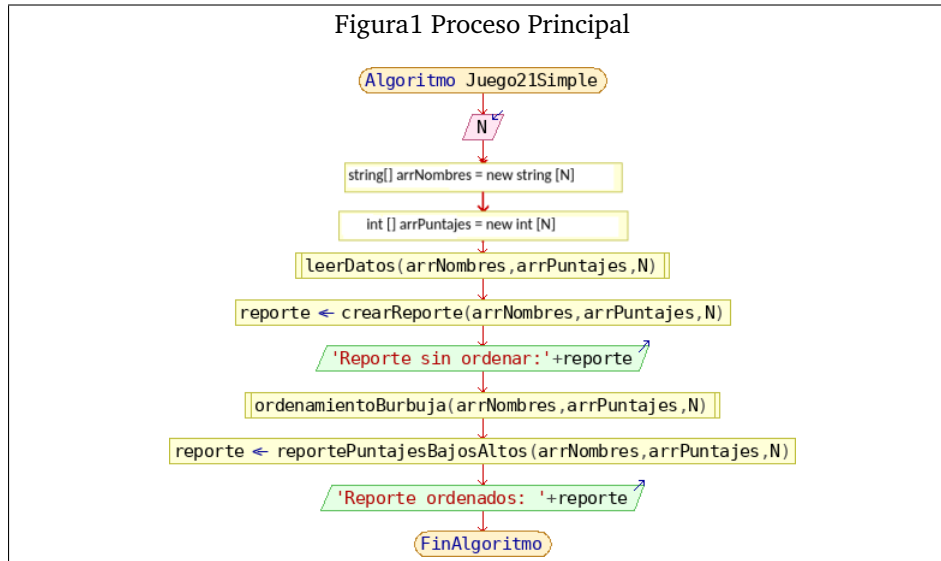


Figura2 Subproceso leerDatos

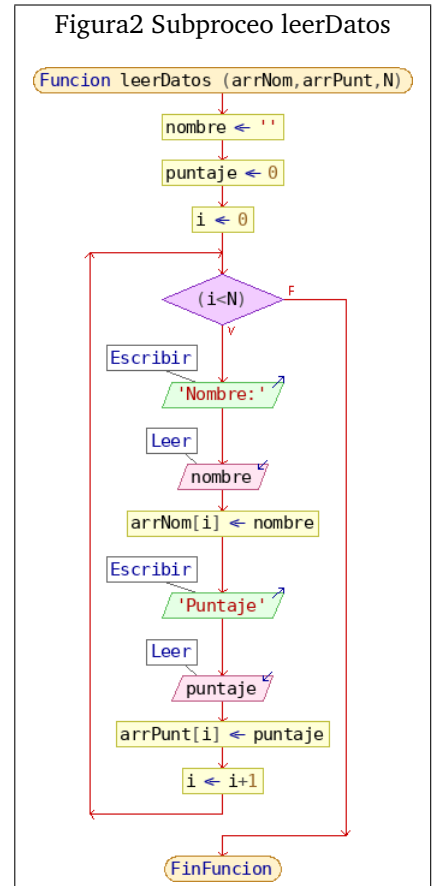
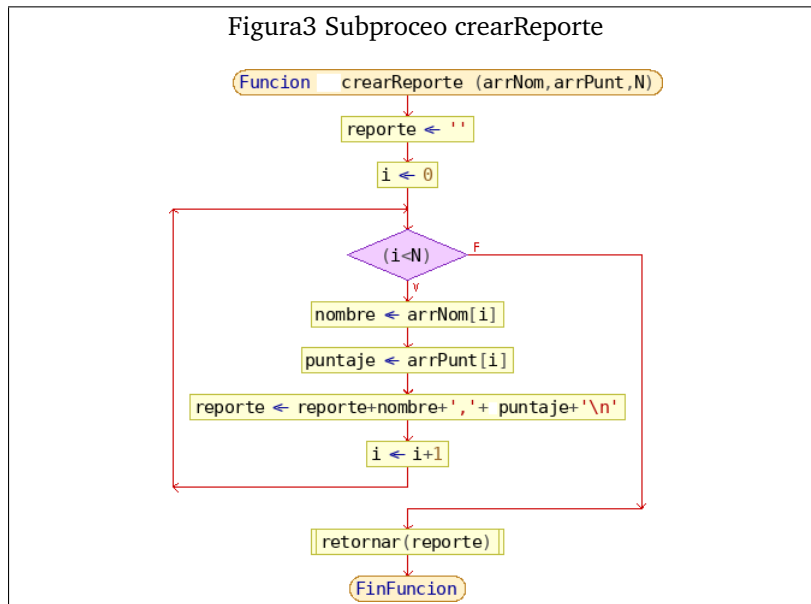
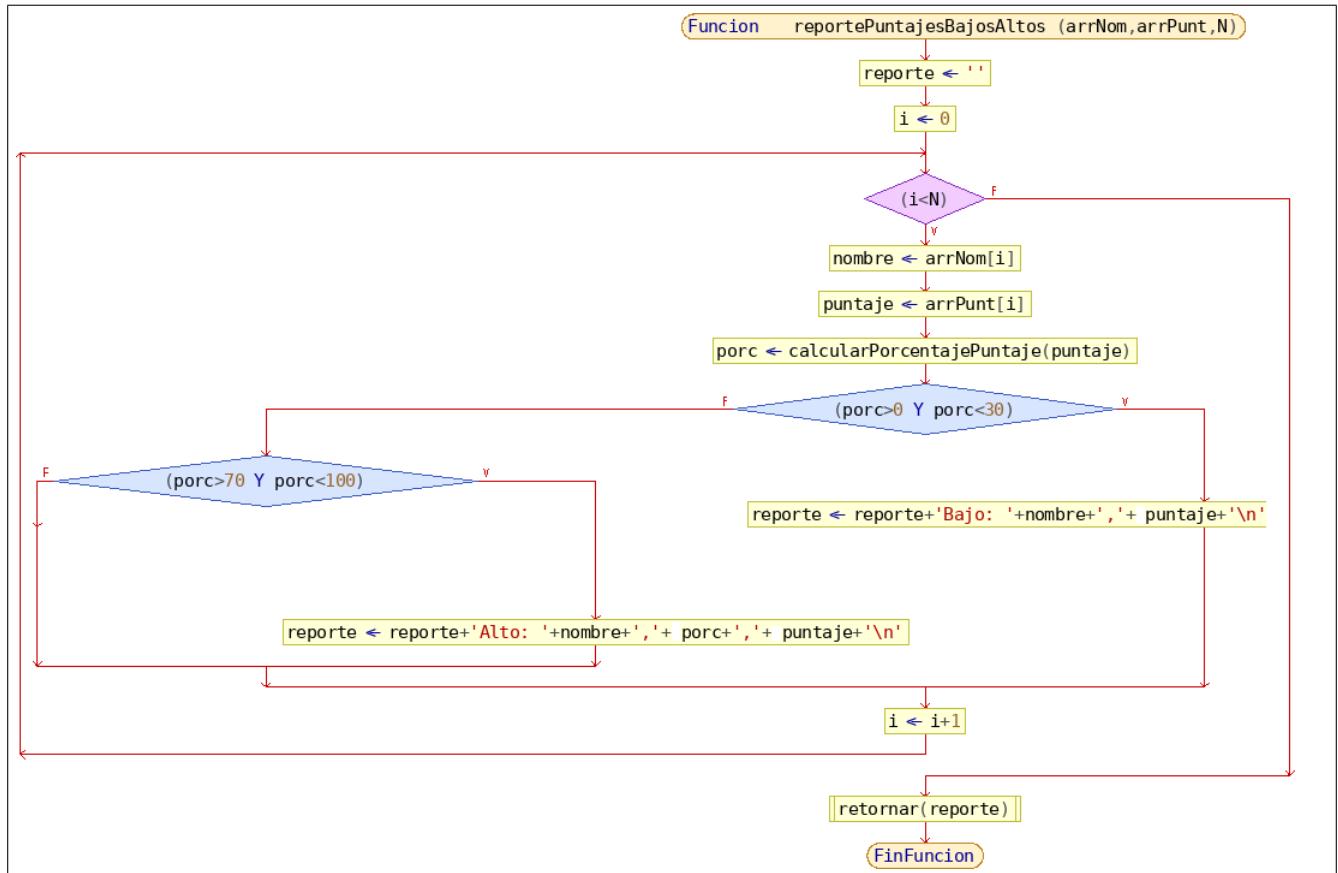


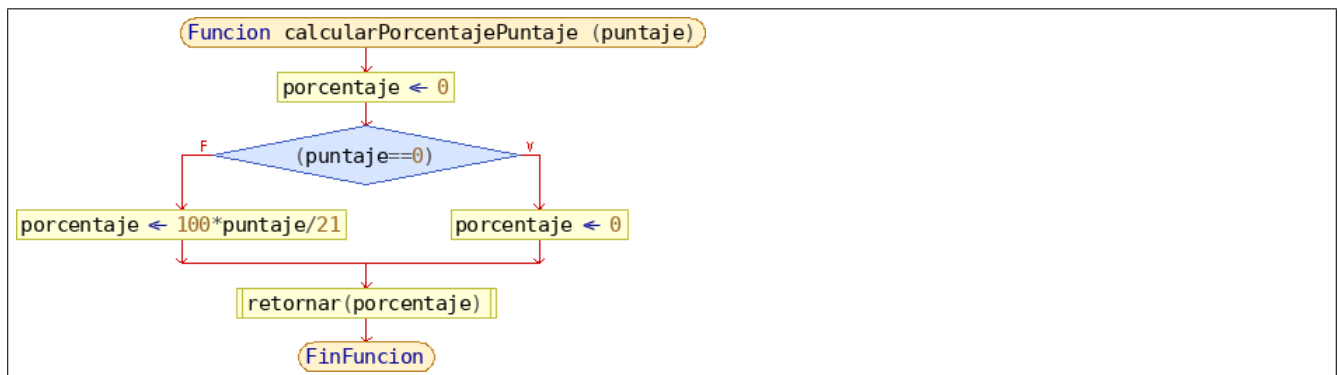
Figura3 Subproceso crearReporte



Funcion reportePuntajesBajosAltos



Funcion calcularPorcentajePuntaje



3. Código

No olviden de activar Microsoft.VisualBasic en el proyecto para poder utilizar inputBox

3.1. Forma Principal

```
public partial class Form1 : Form
{
    public Form1() {
        InitializeComponent();
    }

    private void btSalir_Click(object sender, EventArgs e) {
        Close();
    }
}
```

```

private void btIniciar_Click(object sender, EventArgs e) {
    principal();
}

public static void principal () {
    int N;
    string reporte;

    int[] arrPuntajes;
    string[] arrNombres;

    //N = Convert.ToInt32(Interaction.InputBox("Cuantos_Jugadores:_"));
    arrNombres = new string[N];
    arrPuntajes = new int[N];

    Funciones.leerDatos (arrNombres, arrPuntajes, N);
    reporte = Funciones.crearReporte(arrNombres, arrPuntajes, N);
    MessageBox.Show ("Reporte_sin_ordenar:_\n" + reporte);

    Funciones.ordenamientoBurbujaPorPuntajes(arrNombres, arrPuntajes, N); // solo llamarlo. Supone que ya existe
    reporte = Funciones.crearReporte(arrNombres, arrPuntajes, N);
    MessageBox.Show ("Reporte_ordenado_por_puntaje:_\n" + reporte);
}
}

```

3.2. Clase de Funciones

```

class Funciones {
    //public static void leerDatos (string[] arrNom, int[] arrPun, int N) {
    // Implementar de acuerdo al flujograms

    //public static string crearReporte(string[] arrNom, int[] arrPun, int N) {
    // Implementar de acuerdo al flujograms

    // Funcion que realiza el ordenamiento de acuerdo al puntaje
    public static void ordenamientoBurbujaPorPuntajes(string[] arrNom, int[] arrPun, int N) {
        // Variables temporales:
        String tempAlfabetico;
        int tempNumerico;
        for (int i = 0; i < N - 1; i++) {
            for (int j = 0; j < N - 1 - i; j++) {
                if (arrPun[j] > arrPun[j + 1]) {
                    // Intercambio: Ordena los estudiantes según la calificación —>

                    tempNumerico = arrPun[j];
                    arrPun[j] = arrPun[j + 1];
                    arrPun[j + 1] = tempNumerico;
                    // Intercambio: Ordena los arrNom según la calificación —>
                    tempAlfabetico = arrNom[j];
                    arrNom[j] = arrNom[j + 1];
                    arrNom[j + 1] = tempAlfabetico;
                }
            } // if
        } // for j
    } // for i
}
}

```