//Uso del diccionario

Declaracion:

Dictionary<string, int> ejemploMiDiccionario = new Dictionary<string, int>();

Si hay una lista que almacena un dictionario se declara:

List<KeyValuePair<string, int>> listaDePalabras = new List<KeyValuePair<string, int>>();

Si se desea saber si contiene la palabra

foreach (string palabra in strRichBox.Split(';'))//Ojo con poner comillas dobles!

{

if (ejemploMiDiccionario.ContainsKey(palabra))

{

//El dictionario puede ser usado como array para acceder

//a los elementos, se logra poniendo el KEY o el KEYVALUE

//entre los []..

//Ojo: NO se usan los index, se usan los KEY

ejemploMiDiccionario[palabra]++;

}

else

{

ejemploMiDiccionario.Add(palabra, 1);

}

}

No olvidar de pasar a LISTA antes de ordenarlo

listaDePalabras = ejemploMiDiccionario.ToList();

listaDePalabras.Sort(ordenarDiccionario);

//Llamo a la funcion sorting

public static int ordenarDiccionario(KeyValuePair<string, int> x, KeyValuePair<string, int> y)

{

if (x.Value > y.Value) { return 1; }

else if (x.Value == y.Value) { return 0; }

else { return -1; }

}

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Declarar un enumerado

public enum Franja { Franja\_1, Franja\_2, Franja\_3 };

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Si se tiene que hacer un override del Equals

public override bool Equals(object obj)

{

return (obj is Provincial);

}

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sobrecarga de operadores

public static bool operator ==(Llamada l1, Llamada l2)

{

}

public static bool operator !=(Llamada l1, Llamada l2)

{

return !(l1 == l2);

}

public static Centralita operator +(Centralita c, Llamada nuevaLlamada)

{

bool esta = false;

foreach (Llamada llamadaAux in c.listaDeLlamadas)

{

if(llamadaAux == nuevaLlamada)

{

esta = true;

break;

}

}

if (esta == false)

{

c.AgregarLlamada(nuevaLlamada);

}

return c;

}

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ganancias Centralita

public float GananciasPorLocal { get { return CalcularGanancia(Llamada.TipoLlamada.Local); } }

public float GananciasPorProvincial { get { return CalcularGanancia(Llamada.TipoLlamada.Provincial); }}

public float GananciasPorTotal { get { return CalcularGanancia(Llamada.TipoLlamada.Todas); } }

public List<Llamada> Llamadas { get { return listaDeLlamadas; } }

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DATASOURCE COMBOBOX

cmbTipoEnumerado.DataSource = Enum.GetValues(typeof(Tipo));

/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Agregar al combobox un string

public void AgregarAlCmbNombreListas(CLASEPADRE nombrelista)

{

foreach (Astro astro in planetas)

{

if(astro is Planeta && planetas.Contains(planeta))

{

cmbPlaneta.Items.Add((string)astro);

}

}

}

Indexadores

La declaración de un indexador luce cómo una propiedad, sólo que:

Este recibe parámetros.

La palabra clave this se utiliza para su definición

class EjemploIndexadores {

// Declaro un array

private string[] palabras = new string[100];

// Defino el indexador

public string this[int i] {

get { return palabras[i]; }

set { palabras[i] = value; }

}

}

EjemploIndexadores ejemplo = new EjemploIndexadores();

ejemplo[0] = "Hola";

ejemplo[1] = "Chau";