

Crear extensiones en C# (Extension Methods)

Los Extension Methods, son funciones que crea el programador para realizar operaciones específicas sobre un tipo de dato.

Supongamos que tenemos una lista de cadenas (List<string>) y queremos que una extensión nos de la cantidad de elementos en una cadena (por ejemplo La lista tiene 2 elementos.)

Primero, para crear una extensión necesitamos crear una clase auxiliar la cual tiene que ser estática(static)

```
public static class Extensiones
```

Dentro de ella, creamos la función, que sea pública y estática, en los parámetros pondremos this, seguido del tipo de dato al cual le vamos a crear la extensión, la variable que recibiremos en el parámetro será el valor de la variable fuente.

```
public static string CantidadElementos(this List<string> fuente)
{
    int Cantidad = fuente.Count();
    return ("La lista tiene " + Cantidad.ToString() + " elementos.");
}
```

para utilizar esta extensión, tendremos que definir una variable que sea List<string> y después al poner “punto” aparecerá en la lista la nueva extensión que hemos creado

```
var lista = new List<string>();
    lista.Add("Azul");
    lista.Add("Amarillo");
    lista.Add("Rojo");
    lista.Add("Verde");
    lista.Add("Cafe");
    lista.Add("Negro");
    lista.Add("Blanco");

lista.CantidadElementos();
```

Resultado:

La lista tiene 7 elementos.

MÉTODO PARA AMPLIAR LA FUNCIONALIDAD DE LOS DATAGRIDVIEW

La siguiente función convierte un DataGridView en un DataSet

```
/// <summary>
/// Esta función sirve para ampliar la funcionalidad de un DataGridView
/// </summary>
/// <param name="dgv">Instancia del DataGridView</param>
/// <returns> Devuelve un DataSet a partir de un DataGridView</returns>

public static DataSet GetDataSet(this DataGridView dgv) {
    var ds = new DataSet();
    var dt = new DataTable();

    foreach (var column in dgv.Columns.Cast<DataGridViewColumn>()) {
        if (column.Visible) {
            dt.Columns.Add(column.Name);
        }
    }

    var cellValues = new object[dgv.Columns.Count];
    foreach (var row in dgv.Rows.Cast<DataGridViewRow>()) {
        for (int i=0; i< row.Cells.Count; i++) {
            cellValues[i] = row.Cells[i].Value;
        }
        dt.Rows.Add(cellValues);
    }

    ds.Tables.Add(dt);
    return ds;
}
```

Por tanto, en nuestro proyecto se debe crear una extensión de los DataGridView en una clase nueva, de manera que una vez que el dataGridView ya tuviera sus datos, se pueda utilizar de la siguiente forma:

```
//AHORA VOY A METER EL CODIGO PARA PASAR EL CONTENIDO DEL DATAGRIDVIEW A UN
//FICHERO XML
var dataset = dgvConsulta.GetDataSet();
dataset.WriteXml(File.OpenWrite("consulta.xml"));
```

El código de arriba convierte el DataGridView en un DataSet, paso necesario para poder guardarlo como fichero XML, a través del método propio "WriteXml" del DataSet que ya viene implementado en C#.