Technical University of Cartagena

Telecommunications Engineering School

DIGITAL CONTENTS LABORATORY

Laboratory Content 1. Image Formats Converter

Professor:

Antonio Javier García Sánchez.

1. Objectives

The goal of this laboratory content consists on implementing in C# a Windows service to load image files and, later, to convert them to different formats. Therefore, the student must develop a program which converts any image format (e.g., .bmp) to another image format (e.g., .gif, .jpg, .tiff) and vice versa.

2. Laboratory Content Description

To achieve the purpose of this laboratory content, the student will use Visual Studio .NET where will open a new Project in C#, searching for the solution titled *Windows Application* assigning it a new name (the student must remember the path where she/he saves her/him solution). The initial format of her/his solution has to be similar to the following figure 1:

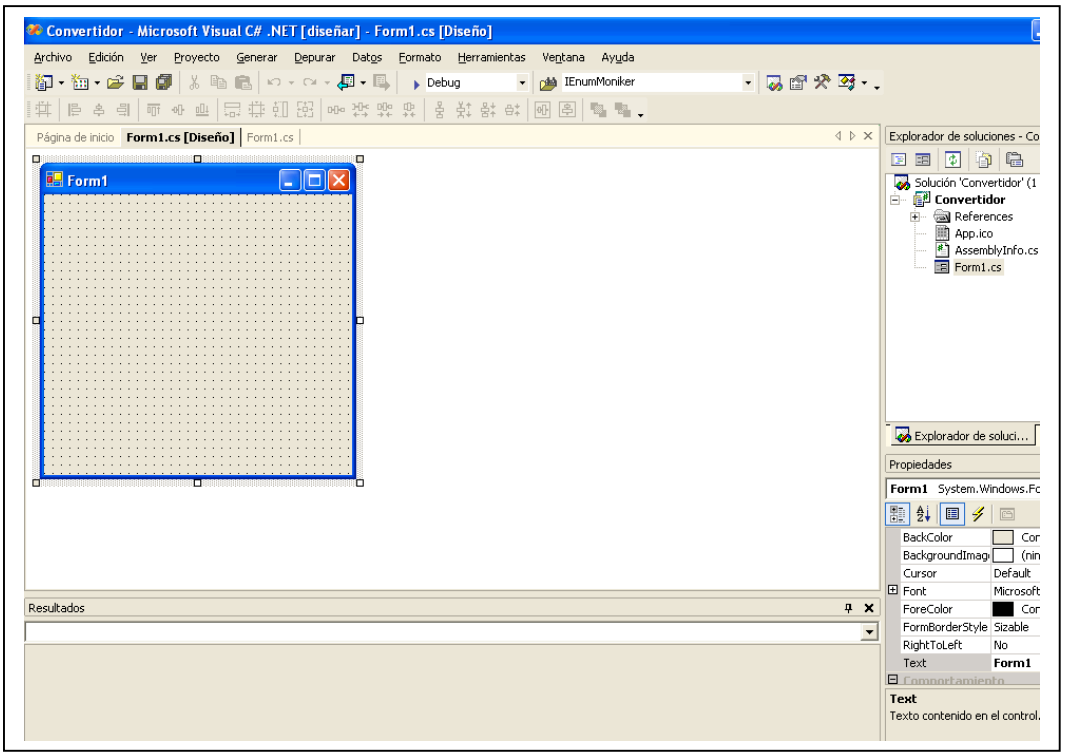

Figure 1. Initial aspect of a Windows Application Solution

And the initial code of the *form* must be the following one:

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace Convertidor
{

      public class Form1 : System.Windows.Forms.Form
```

```
	{
		private System.ComponentModel.Container components = null;

		public Form1()
		{
			InitializeComponent();


		//InitializeComponent

		}
		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
				if (components != null)
				{
					components.Dispose();
				}
			}
			base.Dispose( disposing );
		}

		#region Windows Form Designer generated code

		private void InitializeComponent()
		{
			this.components = new
			 System.ComponentModel.Container();
			this.Size = new System.Drawing.Size(300,300);
			this.Text = "Form1";
		}
		#endregion
		[STAThread]
		static void Main()
		{
			Application.Run(new Form1());
		}
	}
}
```

To carry out the image selection and the image storage in a new type of format (and, therefore, in a new file), the student will employ two *buttons*. Furthermore, both the image selection and the action of saving it, the student will use two *comboboxs*. Finally, to display the image, the student will employ the *PictureBox* tool. The aspect of the graphical interface will be the figure 2.

In the case of running the current code, *buttons, comboboxs and picturebox* are automatically generated as objects from their respective classes. Moreover, the code of the properties associated to those objects (e.g., size, position, color, etc.) is generated by default. However, other properties as *scrollbars* must be modified and adjusted by the student.

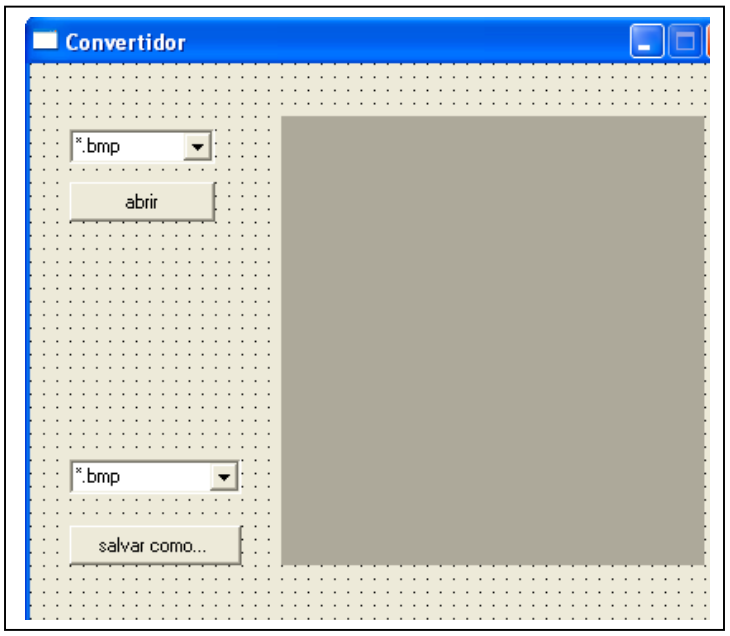

Figure 2. Graphical User Interface of the Image Formats Converter

Additionally to the automatic generation of these properties by the CLR belong to the framework of Visual Studio, the student has to include a reference to the set of data where the image is stored/saved:

```
private Bitmap m_bitmap;
```

`m_bitmap` will be an object of class *Bitmap*. To obtain further information, the student can look the Help of Visual Studio up.

2.1. Image Selection

The button labeled as *open* together with its *combobox* will carry out the functionalities of format selection and the image load. To this end, when we click the *open* button, a *dialog picture* will be opened with the files which have the same extension as the selected format. This *dialog picture* is loaded as an object from its class `OpenFileDialog()`.

```
private void b_abrir_Click(object sender, System.EventArgs e)
            {
                  OpenFileDialog ofd = new OpenFileDialog();
                  ofd.Filter = c_abrir.Text + "|" + c_abrir.Text;
                      .
                      .
                      .
            }
```

The selected image must be saved in the *m_bitmap* object, to be later displayed in the *PictureBox* object.

2.2. Save the selected image

Once the image is obtained and displayed in the *PictureBox*, we will save the image with another name, in another *path* into the *harddisk* and, above all, another format. So, the student must create an object of the `SaveFileDialog` class:

```
private void b_salvar_Click(object sender, System.EventArgs e)
            {
                  SaveFileDialog sfd = new SaveFileDialog();
                  sfd.Title = "Salvar Imagen Como...";

                  sfd.Filter = c_salvar.Text + "|" + c_salvar.Text;
                        .
                        .
                        .
              }
```

The formats to choose will be the following ones: .bmp, .gig, .tiff y .jpg. To achieve this goal, the student has to use the *ImageFormat* property, which is loaded from the *using*:

```
using System.Drawing.Imaging;
```

Finally, we will consider the case of those images that have a higher size than the designed *form*. Under these circumstances, the student has to implement the required code to adapt the graphical user interface to the image size.