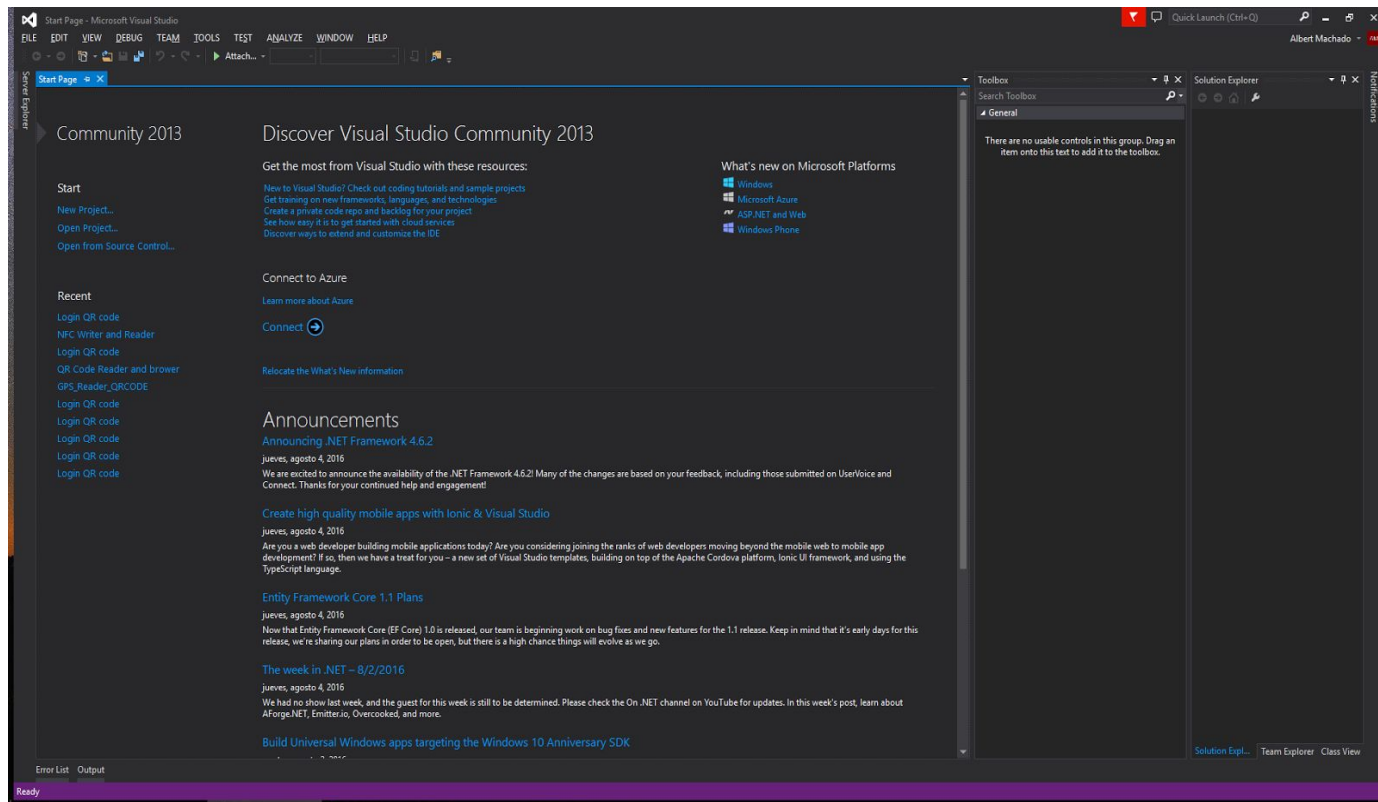
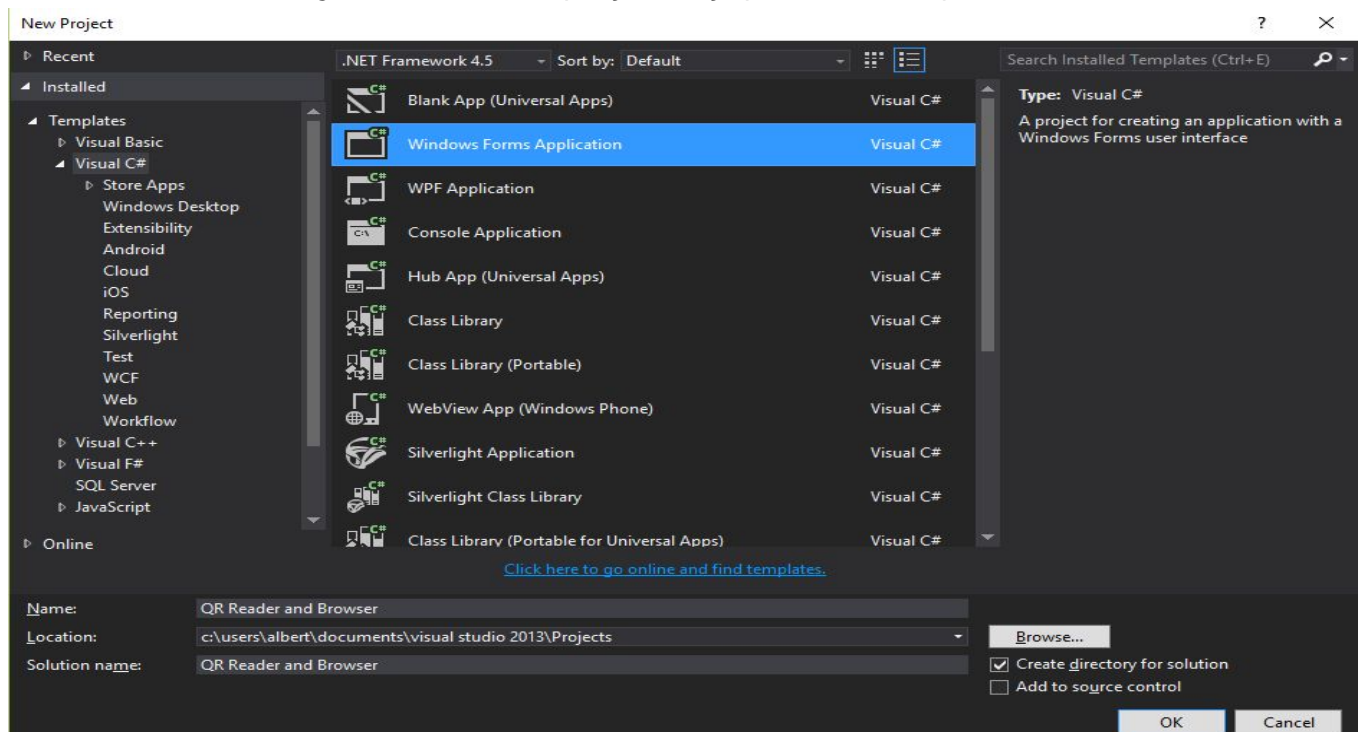


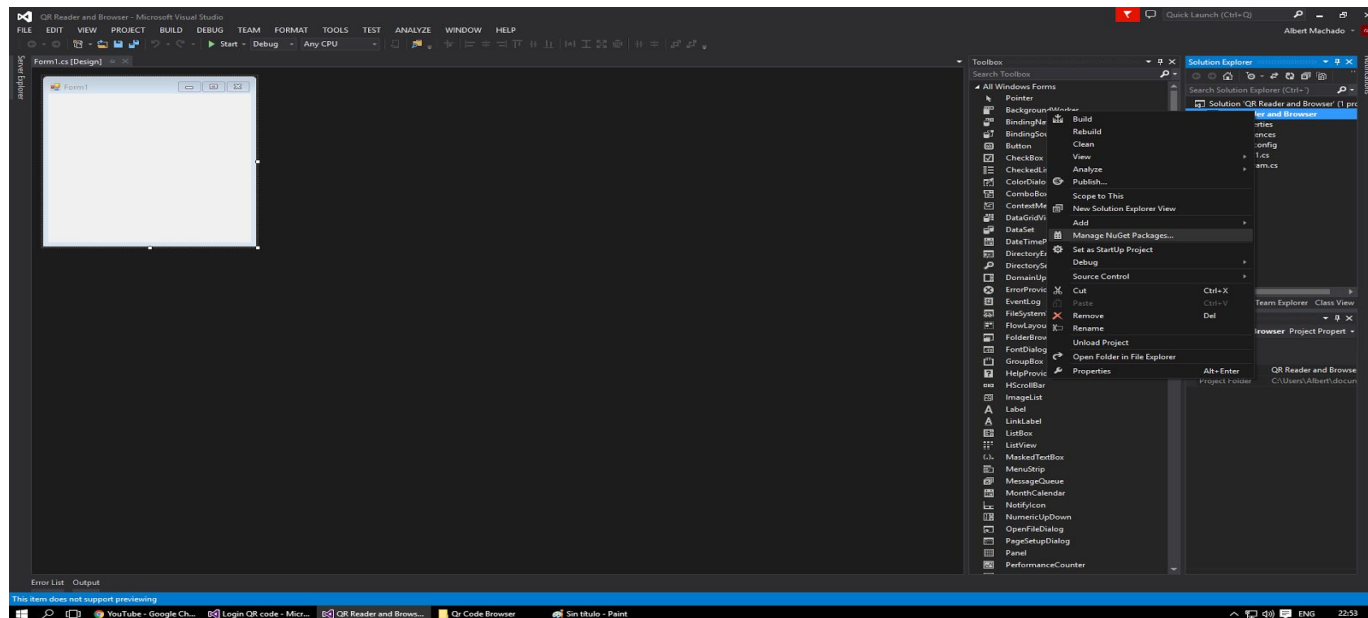
1. Iniciamos Visual studio



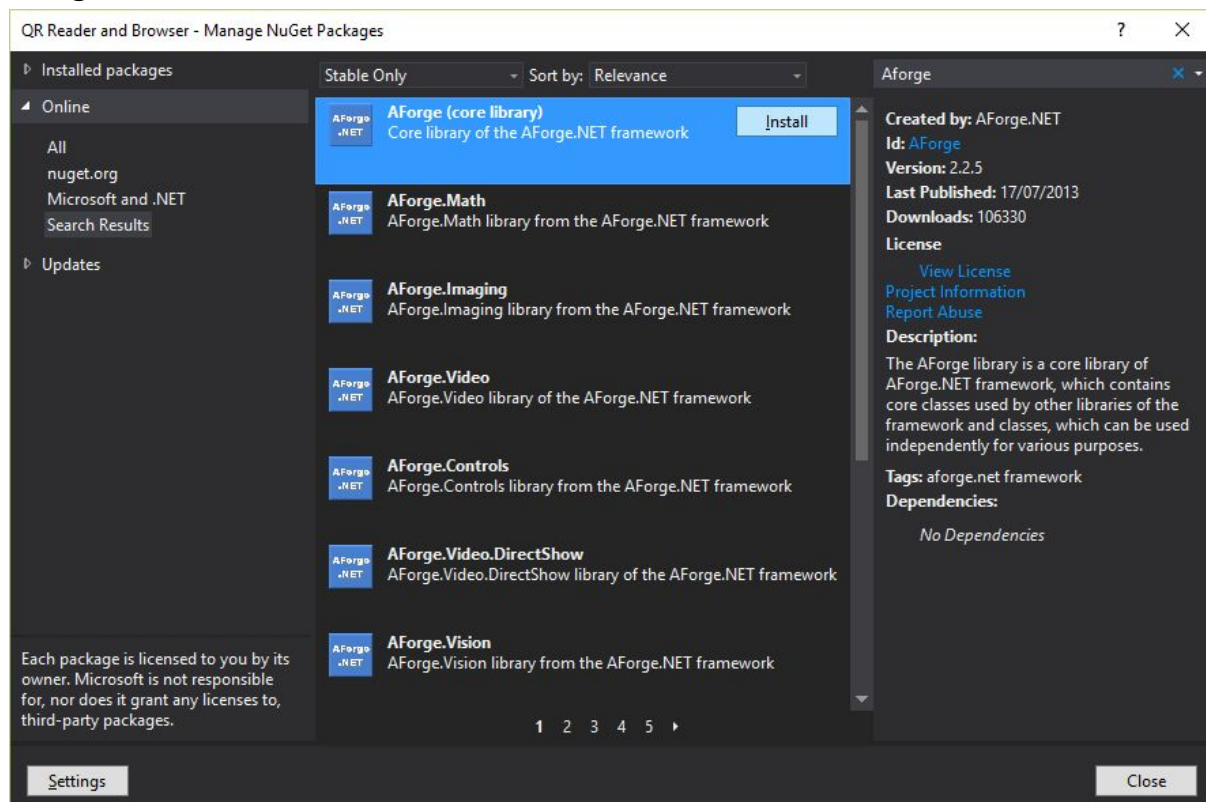
2. Usando la combinación de teclas **Ctrl + Shift + N** creamos un nuevo proyecto, escogemos la opción *Windows Forms Application*, nuestro directorio de trabajo, nombre del proyecto y por último apretamos OK.



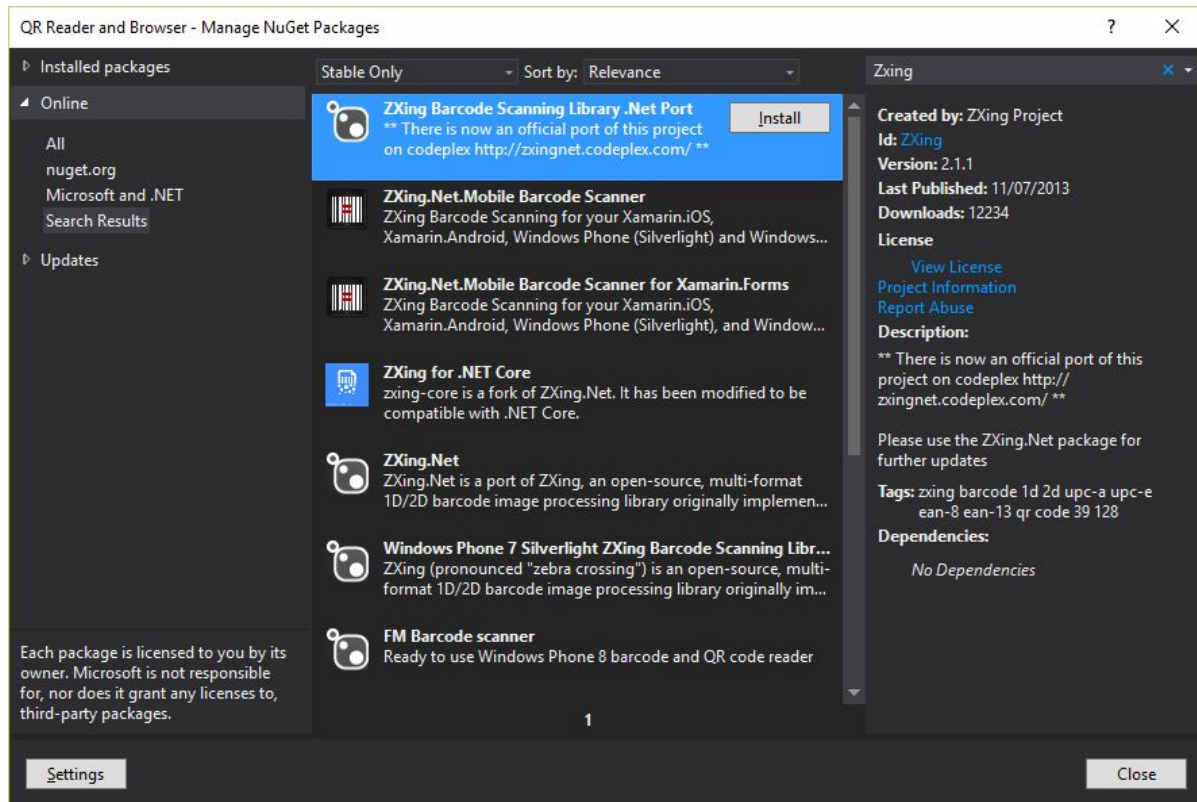
3. Agregamos las librerías necesarias para el proyecto utilizando NuGet Package, para eso seguimos los siguientes pasos:



4. Buscamos en la barra de la esquina superior derecha la librería AForge e instalamos la librería AForge(core library), AForge.Video y AForge.Video.DirectShow

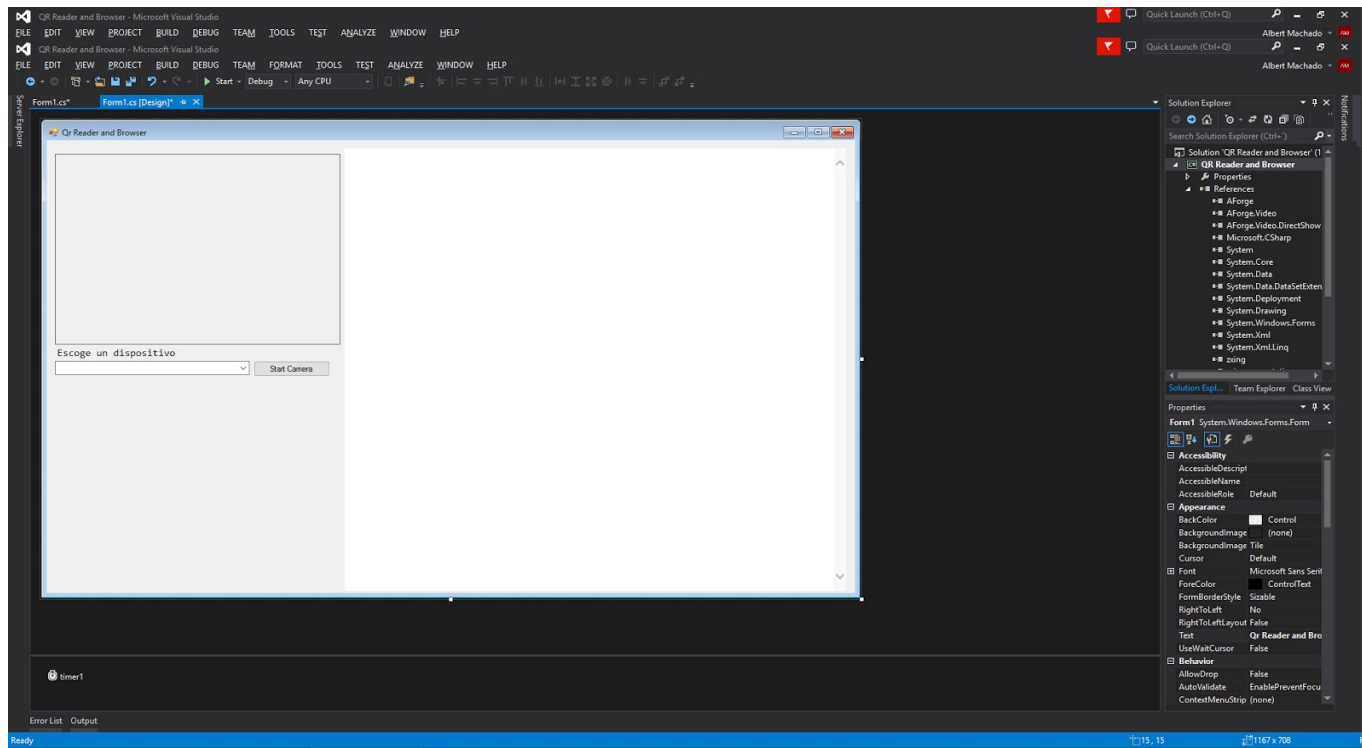


5. Ahora buscamos las librerías para Zxing instalamos **Zxing Barcode Scanning Library .Net Port** y **Zxing for .NET Core**

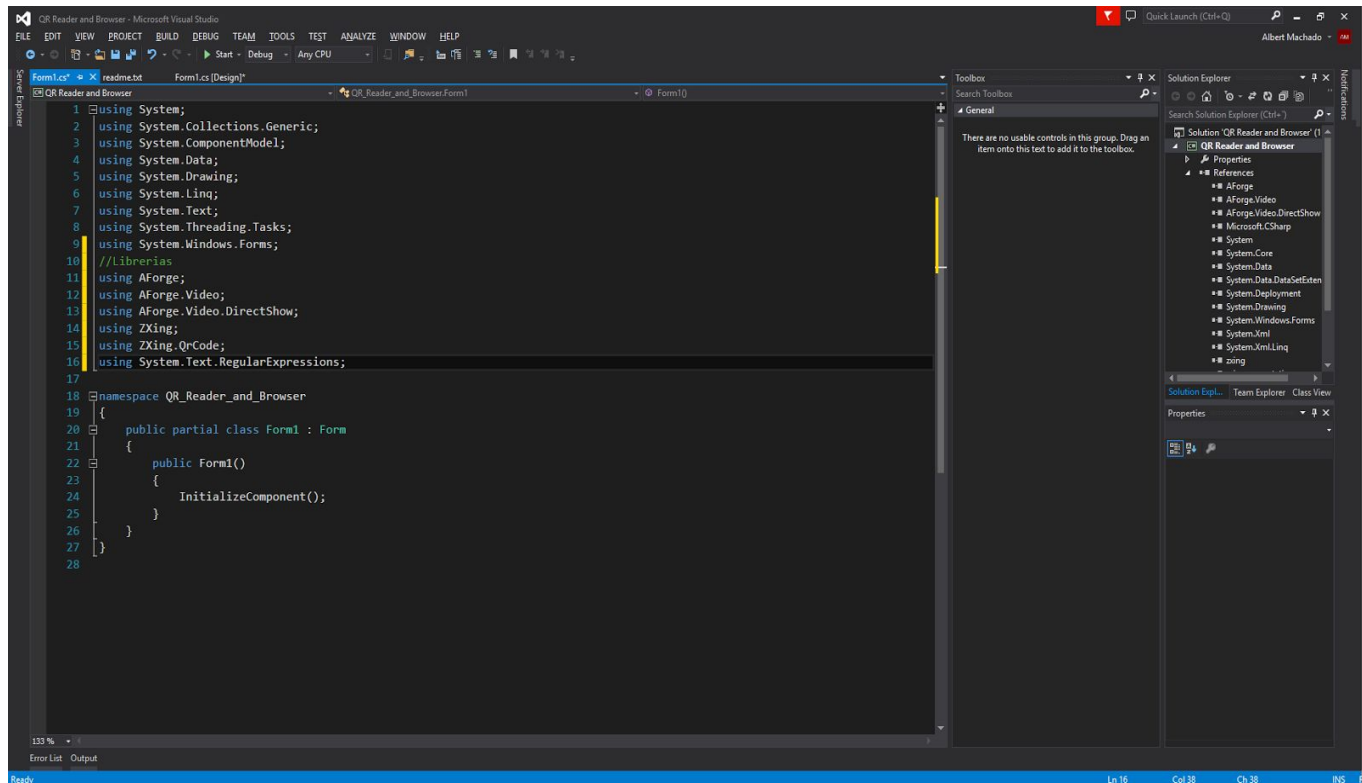


6. Hacemos el diseño de nuestra aplicación agregamos los siguientes elementos un título, el tamaño de la ventana principal y los elementos necesarios:

- 1 picturebox donde se mostrará lo capturado por la cámara
- 1 label que mostrará el mensaje (“Escoge un dispositivo”)
- 1 Combobox donde estara la lista de los dispositivos de captura
- 1 boton
- 1 webBrowser
- 1 timer



7. Nos vamos al código de *Form1* y hacemos uso de las librerías como se muestra a continuación, la librería AForge nos permite usar la cámara de nuestro equipo, ZXing es la encargada de leer el código Qr y con *System.Text.RegularExpressions* Podemos hacer uso de las expresiones regulares que las usaremos para validar la información contenida en el código qr.



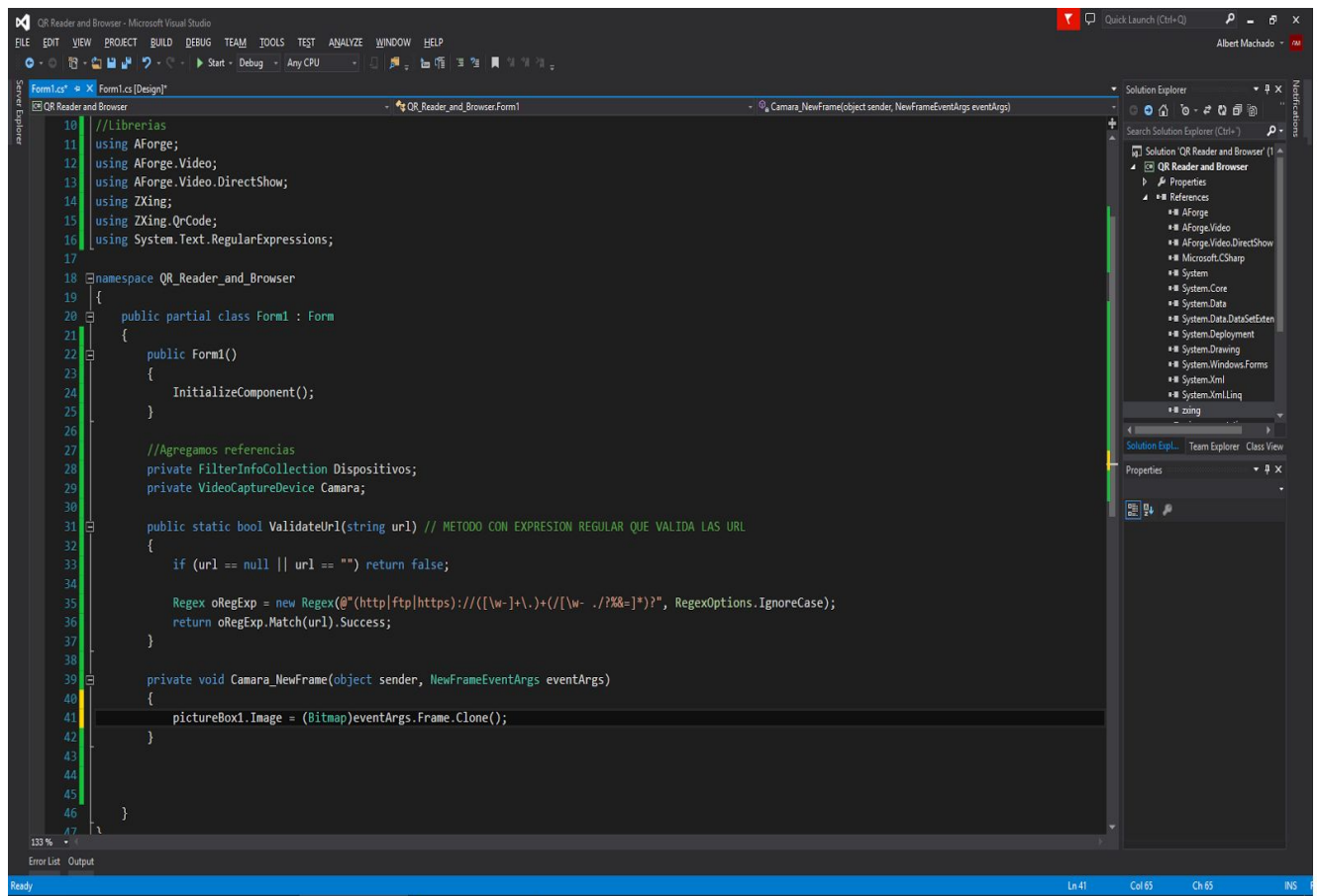
8. Agregamos el siguiente código así:

```
//Agregamos referencias
private FilterInfoCollection Dispositivos;
private VideoCaptureDevice Camara;

public static bool ValidateUrl(string url) // METODO CON
EXPRESION REGULAR QUE VALIDA LAS URL
{
    if (url == null || url == "") return false;

    Regex oRegExp = new
    Regex(@"(http|ftp|https):\/\/([\\w-]+\\.)+(/[\\w- ./?%&=]*)?",
    RegexOptions.IgnoreCase);
    return oRegExp.Match(url).Success;
}

private void Camara_NewFrame(object sender, NewFrameEventArgs
eventArgs)
{
    pictureBox1.Image = (Bitmap)eventArgs.Frame.Clone();
}
```



9. Luego hacemos doble click en la ventana principal del programa en el diseñador para activar el evento *Form1_Load* Y agregamos el siguiente código:

```
private void Form1_Load(object sender, EventArgs e)
{
    int i = 0; // Contadora
    // Listamos dispositivos
    Dispositivos = new
FilterInfoCollection(FilterCategory.VideoInputDevice);

    //Cargamos dispositivos al combobox
    foreach (FilterInfo Device in Dispositivos)
    {
        //Contadora de camaras
        i = i + 1;
        comboBox1.Items.Add(Device.Name);
    }
    comboBox1.SelectedIndex = -1;
    if (i == 0) // Si no hay camara deshabilitar el boton
    {
        button1.Enabled = false;
    }
    else // Si hay camara el boton se habilita
    {
        button1.Enabled = true;
    }
    // Iniciar control de video
    Camara = new VideoCaptureDevice();
}
}
```

10. Activamos el evento del boton y agregamos el siguiente código:

```
private void button1_Click(object sender, EventArgs e)
{
    //Establecemos el dispositivo seleccionado como fuente de
video
    Camara = new
VideoCaptureDevice(Dispositivos[comboBox1.SelectedIndex].MonikerString)
;
    Camara.NewFrame += new
NewFrameEventHandler(Camara_NewFrame);
    //Iniciar recepcion de video
    Camara.Start();
    // Habilitamos y empezamos el timer
    timer1.Enabled = true;
    timer1.Start();
}
```

```
}
```

11. Activamos el evento del Tick o reloj dando doble click al elemento en el diseñador, este se encuentra en la parte de abajo de este y agregamos el siguiente código a su evento:

timer1

```
private void timer1_Tick(object sender, EventArgs e)
{
    // Si no es null de la caja de imagen
    if (pictureBox1.Image != null)
    {
        //Instanciamos lo necesario para leer el código
        BarcodeReader Reader = new BarcodeReader();
        //Variable que contiene el resultado.
        Result result =
Reader.Decode((Bitmap)pictureBox1.Image);
        try
        {
            //Convertimos el resultado en un string
            string decoded = result.ToString();
            //Si hay algo en el mensaje entonces
            if (decoded != "")
            {
                // Si no es una url
                if (!ValidateUrl(decoded))
                {
                    timer1.Stop(); // detenemos el scaneo
                    MessageBox.Show("El código QR no
corresponde a una url, pero corresponde al siguiente mensaje: " +
decoded);
                    Camara.SignalToStop(); // Detenemos la
camara
                    pictureBox1.Image = null; // Reiniciamos la
caja de imagen
                }
                else // Si es una url
                {
                    webBrowser1.Navigate(decoded); // Navegamos
con el WebBrowser
                    timer1.Stop(); // Detenemos el Scaneo
                    MessageBox.Show("Navegando a " + decoded);
                    // Mostramos un mensaje
                    Camara.SignalToStop(); // Detenemos la
camara
                    pictureBox1.Image = null; // Reiniciamos la
caja de imagen
                }
            }
        }
    }
}
```

```
        }  
    }  
    catch (Exception ex)  
    {  
    }  
}  
}
```

12. Por último agregamos el siguiente evento para cuando cerremos la aplicación la cámara deje de capturar.

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)  
{  
    if (Camara.IsRunning == true)  
    {  
        Camara.SignalToStop(); // Detenemos la camara  
    }  
}
```

13. Construimos, ejecutamos y testeamos.