

Apéndice A

Código de las Interfaces de Usuario

Desarrolladas en Microsoft Visual Studio 2010

Interfaz de Configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;

namespace InterfazSATEDUTCIB
{
    public partial class SATEDUTCIB : Form
    {
        int opcion = 0; //variable de seleccion
        char c = "\r"; //variable retorno de carro
        public SATEDUTCIB()
        {
            InitializeComponent();
        }

        private void SATEDUTCIB_Load(object sender, EventArgs e)
        {
            puerto.SelectedIndex = 0; //seleccion de la opcion cero del combobox1
            baudrate.SelectedIndex = 0; //seleccion de la opcion cero del combobox2
            paridad.SelectedIndex = 0; //seleccion de la opcion cero del combobox3
            foreach (String s in SerialPort.GetPortNames()) //obtencion de los puertos COM disponibles
            {
                puerto.Items.Add(s);
            }
            botonmc.Enabled = false; //deshabilita el boton modo comando
        }
    }
}
```

```

botonenviar.Enabled = false;//deshabilita el boton enviar
comandotex.Enabled = false;//deshabilita el cuadro de texto de comandos
botoneco.Enabled = false;//deshabilita el boton activar eco
botonayuda.Enabled = false;//deshabilita el boton ayuda
botonec.Enabled = false;//deshabilita el boton configuracion extendida
botoncn.Enabled = false;//deshabilita el boton cambiar nombre
botoncne.Enabled = false;//deshabilita el boton cambiar nombre +mac
tbnombre.Enabled = false;//deshabilita el cuadro de texto del nombre
botoncp.Enabled = false;//deshabilita el boton de cambiar codigo pin
tbcp.Enabled = false;//deshabilita el cuadro de texto cambiar codigo pin
botonmb.Enabled = false;//deshabilita el boton modo 7 bits
botonvf.Enabled = false;//deshabilita el boton restablecer valores de fabrica
botonvt.Enabled = false;//deshabilita el boton establecer velocidad de transmision
botonp.Enabled = false;//deshabilita el boton establecer paridad
paridad.Enabled = false;//deshabilita el combobox4
veltrans.Enabled = false;//deshabilita el cuadro de texto de baud rate
botonhe.Enabled = false;//deshabilita el boton habilita encriptacion
botonha.Enabled = false;//deshabilita el boton habilita autentificacion
botontc.Enabled = false;//deshabilita el boton establecer tiempo
tbtc.Enabled = false;//deshabilita el cuadro de texto de tiempo
botoncm.Enabled = false;//deshabilita el boton de configuracion temporal
botoncb.Enabled = false;//deshabilita el boton de configuracion basica
botonlimpiar.Enabled = false;//deshabilita el boton de limpiar
}

private void botonAC_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == false)//si el puerto esta cerrado
    {
        serialPort1.PortName = puerto.Text;//nombre del puerto seleccionado
        serialPort1.BaudRate = int.Parse(baudrate.Text);//velocidad de transmision seleccionada
        serialPort1.Open();//abre el puerto
        botonAC.Text = ("Cerrar Puerto");//cambia el nombre del boton
        botonmc.Enabled = true;//habilita el boton modo comando
        botonenviar.Enabled = true;//habilita el boton enviar
        comandotex.Enabled = true;//habilita el textbox para enviar otros comandos o datos
    }
    else
    {
        serialPort1.Close();//cierra el puerto
        botonAC.Text = ("Abrir Puerto");//cambia el nombre del boton
        botonmc.Enabled = false;//deshabilita el boton modo comando
        botonenviar.Enabled = false;//deshabilita el boton enviar
        comandotex.Enabled = false;//deshabilita el cuadro de texto de comandos
        botoneco.Enabled = false;//deshabilita el boton activar eco
        botonayuda.Enabled = false;//deshabilita el boton ayuda
        botonec.Enabled = false;//deshabilita el boton configuracion extendida
        botoncn.Enabled = false;//deshabilita el boton cambiar nombre
        botoncne.Enabled = false;//deshabilita el boton cambiar nombre +mac
        tbnombre.Enabled = false;//deshabilita el cuadro de texto del nombre
        botoncp.Enabled = false;//deshabilita el boton de cambiar codigo pin
        tbcp.Enabled = false;//deshabilita el cuadro de texto cambiar codigo pin
        botonmb.Enabled = false;//deshabilita el boton modo 7 bits
        botonvf.Enabled = false;//deshabilita el boton restablecer valores de fabrica
        botonvt.Enabled = false;//deshabilita el boton establecer velocidad de transmision
        botonp.Enabled = false;//deshabilita el boton establecer paridad
    }
}

```

```

    paridad.Enabled = false;//deshabilita el combobox4
    veltrans.Enabled = false;//deshabilita el cuadro de texto de baud rate
    botonhe.Enabled = false;//deshabilita el boton habilita encriptacion
    botonha.Enabled = false;//deshabilita el boton habilita autenticacion
    botontc.Enabled = false;//deshabilita el boton establecer tiempo
    tbtct.Enabled = false;//deshabilita el cuadro de texto de tiempo
    botoncm.Enabled = false;//deshabilita el boton de configuracion temporal
    botoncb.Enabled = false;//deshabilita el boton de configuracion basica
    botonlimpiar.Enabled = false;//deshabilita el boton de limpiar
}
}

private void botonsalir_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true)//cierra el puerto si esta abierto
    {
        serialPort1.Close();
    }
    Close();//cierra la aplicacion
}

private void botonmc_Click(object sender, EventArgs e)
{
    opcion = 0;//imprime en respuesta
    if (botonmc.Text == "Modo Comando")
    {
        serialPort1.Write("$$$");//manda tres signos de peso por el puerto serie
        botonmc.Text = "Salir Modo Comando";//cambia el nombre del boton
        botoneco.Enabled = true;//habilita el boton activar eco
        botonayuda.Enabled = true;//habilita el boton ayuda
        botonec.Enabled = true;//habilita el boton configuracion extendida
        botoncn.Enabled = true;//habilita el boton cambiar nombre
        botoncne.Enabled = true;//habilita el boton cambiar nombre +mac
        tbnombre.Enabled = true;//habilita el cuadro de texto del nombre
        botoncp.Enabled = true;//habilita el boton de cambiar codigo pin
        tbcpc.Enabled = true;//habilita el cuadro de texto cambiar codigo pin
        botonmb.Enabled = true;//habilita el boton modo 7 bits
        botonvf.Enabled = true;//habilita el boton restablecer valores de fabrica
        botonvt.Enabled = true;//habilita el boton establecer velocidad de transmision
        botonp.Enabled = true;//habilita el boton establecer paridad
        paridad.Enabled = true;//habilita el combobox4
        veltrans.Enabled = true;//habilita el cuadro de texto de baud rate
        botonhe.Enabled = true;//habilita el boton habilita encriptacion
        botonha.Enabled = true;//habilita el boton habilita autenticacion
        botontc.Enabled = true;//habilita el boton establecer tiempo
        tbtct.Enabled = true;//habilita el cuadro de texto de tiempo
        botoncm.Enabled = true;//habilita el boton de configuracion temporal
        botoncb.Enabled = true;//habilita el boton configuracion basica
        botonlimpiar.Enabled = true;//habilita el boton limpiar
    }
    else
    {
        serialPort1.Write("---" + c);//manda tres menos para salir del modo comando
        botonmc.Text = "Modo Comando";//cambia el nombre del boton
        botoneco.Enabled = false;//deshabilita el boton activar eco
        botoneco.Enabled = false;//deshabilita el boton activar eco
    }
}

```

```

        botonayuda.Enabled = false;//deshabilita el boton ayuda
        botonec.Enabled = false;//deshabilita el boton configuracion extendida
        botoncn.Enabled = false;//deshabilita el boton cambiar nombre
        botoncne.Enabled = false;//deshabilita el boton cambiar nombre +mac
        tbnombre.Enabled = false;//deshabilita el cuadro de texto del nombre
        botoncp.Enabled = false;//deshabilita el boton de cambiar codigo pin
        tcbcp.Enabled = false;//deshabilita el cuadro de texto cambiar codigo pin
        botonmb.Enabled = false;//deshabilita el boton modo 7 bits
        botonvf.Enabled = false;//deshabilita el boton restablecer valores de fabrica
        botonvt.Enabled = false;//deshabilita el boton establecer velocidad de transmision
        botonp.Enabled = false;//deshabilita el boton establecer paridad
        paridad.Enabled = false;//deshabilita el combobox4
        veltrans.Enabled = false;//deshabilita el cuadro de texto de baud rate
        botonhe.Enabled = false;//deshabilita el boton habilita encriptacion
        botonha.Enabled = false;//deshabilita el boton habilita autentificacion
        botontc.Enabled = false;//deshabilita el boton establecer tiempo
        tbtc.Enabled = false;//deshabilita el cuadro de texto de tiempo
        botoncm.Enabled = false;//deshabilita el boton de configuracion temporal
        botoncb.Enabled = false;//deshabilita el boton configuracion basica
        botonlimpiar.Enabled = false;//deshabilita el boton limmpiar
    }
}

private void botonenviar_Click(object sender, EventArgs e)
{
    String comando;//variable para el comando extra
    opcion = 0;//imprime en respuesta
    comando = comandotex.Text;//comando extra
    serialPort1.Write(comando + c);//manda el comando extra
    comandotex.Clear();//limpia el textbox despues de enviar
}

private void botoneco_Click(object sender, EventArgs e)
{
    opcion = 0;//imprime en respuesta
    if (botoneco.Text == "Activar Eco")
    {
        serialPort1.Write("+" + c);//manda comando para activar eco
        botoneco.Text = "Desactivar Eco";//cambia nombre del boton
    }
    else
    {
        serialPort1.Write("+" + c);//manda comando para activar eco
        botoneco.Text = "Activar Eco";//cambia nombre del boton
    }
}

private void botonayuda_Click(object sender, EventArgs e)
{
    opcion = 1;//opcion 1
    serialPort1.Write("H" + c);//envia el comando para la ayuda
}

private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    String mensaje;//variable mensaje
    mensaje = serialPort1.ReadExisting();//lee el mensaje proveniente de la tarjeta
}

```

```

if (opcion == 0)
{
    textboxeco.AppendText(mensaje);//imprime mensaje en el cuadro de texto respuesta
}
if (opcion == 1)
{
    tbayuda.AppendText(mensaje);//imprime en el cuadro de texto de ayuda
}
if (opcion == 2)
{
    tbedc.AppendText(mensaje);//imprime en el cuadro de texto de configuracion
}
}

private void botonec_Click(object sender, EventArgs e)
{
    opcion = 2;//opcion 2
    serialPort1.Write("E" + c);//envia el comando para el estado de la configuracion
}

private void botoncn_Click(object sender, EventArgs e)
{
    String nombre;//variable nombre
    opcion = 0;//opcion 0
    nombre = tnombre.Text;//nuevo nombre
    serialPort1.Write("SN," + nombre + c);//cambio del nombre de la tarjeta
}

private void botoncne_Click(object sender, EventArgs e)
{
    String nombre;//variable nombre
    opcion = 0;//opcion 0
    nombre = tnombre.Text;//nuevo nombre
    serialPort1.Write("S-" + nombre + c);//cambio del nombre de la tarjeta mas ultimos 4 numeros de la direccion mac
}

private void botoncp_Click(object sender, EventArgs e)
{
    String codigo;//variable codigo
    opcion = 0;//opcion 0
    codigo = tcbp.Text;//nuevo codigo
    serialPort1.Write("SP," + codigo + c);//cambia el codigo PIN
}

private void botonmb_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (botonmb.Text == "Modo 7 bits")
    {
        serialPort1.Write("S7,1" + c);//manda comando modo 7 bits
        botonmb.Text = "Modo 8 bits";//cambia el nombre del boton
    }
    else
    {
        serialPort1.Write("S7,0" + c);//entra en modo de 8 bits
        botonmb.Text = "Modo 7 bits";//cambia el nombre del boton
    }
}

```

```

}
private void botonvf_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    serialPort1.Write("SF,1" + c);//restablece los valores de fabrica
}
private void botonp_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (paridad.Text == "PAR")
    {
        serialPort1.Write("SL,E" + c);//manda la paridad seleccionada
    }
    if (paridad.Text == "IMPAR")
    {
        serialPort1.Write("SL,O" + c);//manda la paridad seleccionada
    }
    if (paridad.Text == "NINGUNA")
    {
        serialPort1.Write("SL,N" + c);//manda la paridad seleccionada
    }
}
private void botonvt_Click(object sender, EventArgs e)
{
    int velocidad;//variable velocidad
    opcion = 0;//opcion 0
    velocidad = int.Parse(veltrans.Text);//guarda velocidad de transmision
    serialPort1.Write("SU," + velocidad + c);//envia velocidad de transmision
}

private void botonhe_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (botonhe.Text == "Habilitar Encriptación")
    {
        serialPort1.Write("SE,1" + c);//habilita encriptacion
        botonhe.Text = "Deshabilitar Encriptación";//cambia nombre del boton
    }
    else if (botonhe.Text == "Deshabilitar Encriptación")
    {
        serialPort1.Write("SE,0" + c);//deshabilita encriptacion
        botonhe.Text = "Habilitar Encriptación";//cambia nombre del boton
    }
}

private void botonha_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (botonha.Text == "Habilitar Autenticación")
    {
        serialPort1.Write("SA,1" + c);//habilita autenticacion
        botonha.Text = "Deshabilitar Autenticación";//cambia nombre del boton
    }
    else if (botonha.Text == "Deshabilitar Autenticación")
    {
        serialPort1.Write("SA,0" + c);//deshabilita autenticacion
    }
}

```

```
        botonha.Text = "Habilitar Autenticación";//cambia nombre del boton
    }
}

private void botontc_Click(object sender, EventArgs e)
{
    int tiempo;//variable tiempo
    opcion = 0;//opcion cero
    tiempo = int.Parse(tbtc.Text);//guarda tiempo
    serialPort1.Write("ST," + tiempo + c);//envia instruccion para cambiar el tiempo para entrar a el modo comando
}

private void botoncm_Click(object sender, EventArgs e)
{
    int velocidad;//variable velocidad
    opcion = 0;//opcion 0
    velocidad = int.Parse(veltrans.Text);//guarda velocidad de transmision
    if (paridad.Text == "PAR")
    {
        serialPort1.Write("U," + velocidad + ",E" + c);//manda la configuracion temporal del puerto serie
    }
    if (paridad.Text == "IMPAR")
    {
        serialPort1.Write("U," + velocidad + ",O" + c);//manda la configuracion temporal del puerto serie
    }
    if (paridad.Text == "NINGUNA")
    {
        serialPort1.Write("U," + velocidad + ",N" + c);//manda la configuracion temporal del puerto serie
    }
}

private void botoncb_Click(object sender, EventArgs e)
{
    opcion = 2;//opcion 0
    serialPort1.Write("D" + c);//envia comando de configuracion basica
}

private void botonlimpiar_Click(object sender, EventArgs e)
{
    tbedc.Clear();
}
}
}
```

Monitor de Puertos

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;

namespace MonitorPuertos
{
    public partial class MonitorPuertos : Form
    {
        public MonitorPuertos()
        {
            InitializeComponent();
        }

        private void MonitorPuertos_Load(object sender, EventArgs e)
        {
            comboBox1.SelectedIndex = 0; //seleccion de la opcion cero del combobox1
            comboBox2.SelectedIndex = 0; //seleccion de la opcion cero del combobox2
            foreach (String s in SerialPort.GetPortNames()) //obtencion de los puertos COM disponibles
            {
                comboBox1.Items.Add(s);
            }
            button2.Enabled = false; //deshabilita boton enviar
            button3.Enabled = false; //deshabilita boton limpiar
            textBox2.Enabled = false; //deshabilita cuadro de texto recepcion
            textBox1.Enabled = false; //deshabilita cuadro de texto transmision
        }

        private void button4_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen == true) //cierra el puerto si esta abierto
            {
                serialPort1.Close();
            }
            Close(); //cierra la interfaz
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen == false) //si el puerto esta cerrado
            {
                serialPort1.PortName = comboBox1.Text; //nombre del puerto seleccionado
                serialPort1.BaudRate = int.Parse(comboBox2.Text); //velocidad de transmision seleccionada
                serialPort1.Open(); //abre el puerto
                button1.Text = ("Cerrar Puerto"); //cambia el nombre del boton
                button2.Enabled = true; //habilita boton enviar
                button3.Enabled = true; //habilita boton limpiar
            }
        }
    }
}
```



```
        textBox2.Enabled = true;//habilita cuadro de texto recepcion
        textBox1.Enabled = true;//habilita cuadro de texto transmision
    }
    else
    {
        serialPort1.Close();//cierra el puerto
        button1.Text = ("Abrir Puerto");//cambia el nombre del boton
        button2.Enabled = false;//deshabilita boton enviar
        button3.Enabled = false;//deshabilita boton limpiar
        textBox2.Enabled = false;//deshabilita cuadro de texto recepcion
        textBox1.Enabled = false;//deshabilita cuadro de texto transmision
    }
}
private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    String mensaje;//variable
    mensaje = serialPort1.ReadExisting();//guarda en mensaje lo que llega al rx del puerto serial
    textBox2.AppendText(mensaje);//agrega el texto que se recibe
}
private void button3_Click(object sender, EventArgs e)
{
    textBox2.Clear();//limpia el cuadro de texto recepcion
}
private void button2_Click(object sender, EventArgs e)
{
    String mensaje;//variable
    mensaje = textBox1.Text;//guarda el contenido del cuadro de texto transmision
    serialPort1.Write(mensaje);//envia el mensaje
    textBox1.Clear();//limpia el cuadro de texto transmision
}
}
}
```