

# Comunicacións industriais

## Práctica 04: Chat por porto serie. Comunicación entre dous dispositivos serie

---

### Descrición da práctica:

Comunicación duplex entre dous dispositivos serie. Elaborarase un programa que constará de dous formularios, un principal no que se efectuará a comunicación e outro secundario que nos permitirá configurar os parámetros do porto. Para probar o funcionamento da práctica poderase empregar un cable null-modem conectado a dúas computadoradoras ou ben simular mediante portos serie virtuais a comunicación.

### Coñecementos previos:

- Programación en CSharp.
- Programación de parámetros característicos dunha comunicación serie (baudrate, databits, paridade, stopbits).
- Traballo con ficheiros **.xml** para gardar datos.

### Material necesario:

- PC con Visual Studio 2017 instalado (CSharp).
- Cable null-modem ou software de simulación de portos serie (Eltima Virtual Serial Port Driver ou similar).

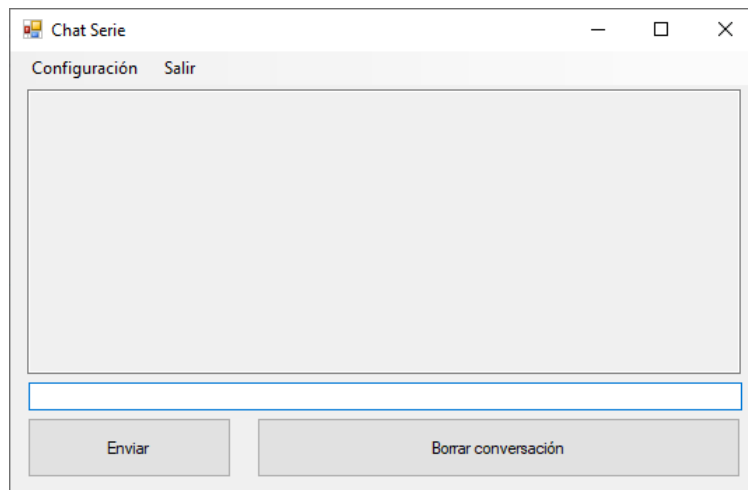
### Especificacións:

- Elaborarase un programa en CSharp con dous formularios. Un principal e un secundario de configuración da comunicación.
- No programa principal existirá un cadro de texto multiliña para ir seguindo a conversa e un cadro de texto dunha soa liña para introducir o texto a enviar ao interlocutor.
- Haberá que programar un menú principal (Main menu) para acceder ao formulario secundario de configuración e para saír.
- Disporase de dous botóns, un para enviar o texto e outro para borrar a conversa si se desexa.
- No formulario secundario teremos cinco ComboBoxes para seleccionar os parámetros típicos dunha comunicación serie (porto, baudrate, databits, paridade e stopbits). Haberá tamén un botón para establecer a comunicación, unha vez configurado o porto.
- Empregaremos un ficheiro **.xml** para gardar a última configuración e, dese xeito, non ter que estar a introducila cada vez que se arranque a aplicación.

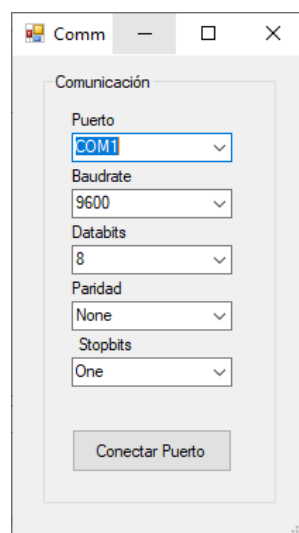
### Solución:

Creamos a interface gráfica cos dous formularios.

O formulario principal terá un aspecto similar ao seguinte.



O formulario para establecer os parámetros de comunicación será similar ao seguinte.



No primeiro formulario temos dúas variables globais.

```

public static Frm_Principal frm_principal = null; // Crear variable de tipo
Frm_Principal

string mensaje = "";

```

Creamos o evento para recibir caracteres polo porto.

```

public Frm_Principal()
{
    InitializeComponent();
    Puerto.DataReceived += new
System.IO.Ports.SerialDataReceivedEventHandler(Recepcion); // Línea imprescindible para
transmitir
}

```

No momento de arrancar a aplicación, configuramos o porto para que os caracteres en idiomas como o castelán ou galego se visualicen correctamente.

```

private void Frm_Principal_Load(object sender, EventArgs e)
{
    Txt_Envio.Focus();
    // Para traballar cos caracteres no noso idioma
    this.Puerto.Encoding = Encoding.GetEncoding(28591);
}

```

O sistema para recoller os caracteres que entran polo porto é o seguinte.

```
private void Recepcion(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    //Recibir Datos
    mensaje = Puerto.ReadLine();
    this.Invoke(new EventHandler(Actualizar));
}

private void Actualizar(object s, EventArgs e)
{
    Txt_Chat.Text = Txt_Chat.Text + "El: " + mensaje + "\r\n";
}
```

Fixémonos que empregamos “**ReadLine**” para recoller os caracteres. Isto é unha opción entre as varias que dispón o porto serie. Se escollemos dita opción non se parará de recibir caracteres ata que se atope o de nova liña.

No menú Comunicación abrimos o segundo formulario.

```
private void Btn_Config_Click(object sender, EventArgs e)
{
    frm_principal = this; //frm_principal = 0 formulario actual
    Frm_Secundario frm_secundario = new Frm_Secundario(); //Creamos en memoria o
segundo formulario
    frm_secundario.ShowDialog(); //Amosa o segundo formulario poñendo o primeiro a
enabled false (no usable)
}
```

No menú Saír pechamos a aplicación.

```
private void Btn_Salir_Click(object sender, EventArgs e)
{
    DialogResult aviso = MessageBox.Show("Quere pechar o programa?", "Atención",
MessageBoxButtons.OKCancel, MessageBoxIcon.Warning); //Aviso de advertencia ao saír
    if(aviso == DialogResult.OK)
    {
        Puerto.Close();
        this.Close();
    }
}
```

No botón para borrar o chat temos o seguinte.

```
private void Btn_Borrar_Click(object sender, EventArgs e)
{
    DialogResult aviso = MessageBox.Show("Quere borrar o chat?", "Atención",
MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation); //Aviso de advertencia al borrar
    if (aviso == DialogResult.OK)
    {
        Txt_Chat.Text = "";
    }
}
```

No botón para enviar temos a programación seguinte.

```
private void Btn_Enviar_Click(object sender, EventArgs e)
{
    if(Txt_Envio.Text != "" && Puerto.IsOpen)
    {
        Puerto.WriteLine(Txt_Envio.Text);
        Txt_Chat.Text = Txt_Chat.Text + "Eu: " + Txt_Envio.Text + "\r\n";
        Txt_Envio.Text = ""; //Borrar o cuadro de texto ao enviar a mensaxe
    }
}
```

```

    }
    else
    {
        MessageBox.Show("Non se pode enviar unha mensaxe baleira ou co porto
pechado", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Podemos programar o evento KeyDown do cadro de texto para non ter que estar pulsando o botón de enviar e simplemente pulsando a tecla Intro executar o mesmo código.

No segundo formulario empezamos programando o evento Load para detectar os portos que existan na computadora na que se está a executar e recollemos a información do ficheiro **config.xml** para establecer as opcións por defecto do porto.

```

private void Frm_Secundario_Load(object sender, EventArgs e)
{
    //Declaramos unha variable matriz de cadeas para conter os nomes dos portos que
se atopen no ordenador
    string[] portos = SerialPort.GetPortNames();
    //Colocamos no combo a matriz de portos atopada
    foreach (string port in portos)
    {
        Cmb_Puerto.Items.Add(port);
    }
    Cmb_Puerto.SelectedIndex = 0;

    Cmb_Paridad.DataSource = Enum.GetValues(typeof(System.IO.Ports.Parity));
    Cmb_Stopbits.DataSource = Enum.GetValues(typeof(System.IO.Ports.StopBits));

    if(File.Exists(Application.StartupPath + "\\config.xml"))//Control de error por
se non existe config.xml
    {
        XDocument xdoc = XDocument.Load(Application.StartupPath + "\\config.xml");
        var elementos = from dato in xdoc.Descendants() select dato;
        foreach (var dato in xdoc.Descendants("Configuracion"))
        {
            Cmb_Puerto.SelectedIndex = Convert.ToByte(dato.Element("Puerto").Value);
            Cmb_Baudrate.SelectedIndex =
Convert.ToByte(dato.Element("Baudrate").Value);
            Cmb_Databits.SelectedIndex =
Convert.ToByte(dato.Element("Databits").Value);
            Cmb_Paridad.SelectedIndex =
Convert.ToByte(dato.Element("Paridad").Value);
            Cmb_Stopbits.SelectedIndex =
Convert.ToByte(dato.Element("Stopbits").Value);
        }
    }
}

```

E logo, cando pulsamos o botón Conectar, aplicamos a configuración ao porto, abrímolos e gardamos a configuración actual como valores por defecto.

```

private void Btn_Conectar_Click(object sender, EventArgs e)
{
    Frm_Principal.frm_principal.Puerto.Close();
    Frm_Principal.frm_principal.Puerto.PortName = Cmb_Puerto.Text;
    Frm_Principal.frm_principal.Puerto.BaudRate =
Convert.ToInt16(Cmb_Baudrate.Text);
}

```

```

        Frm_Principal.frm_principal.Puerto.DataBits =
Convert.ToInt16(Cmb_Databits.Text);
        Frm_Principal.frm_principal.Puerto.Parity =
(System.IO.Ports.Parity)Cmb_Paridad.SelectedItem;
        Frm_Principal.frm_principal.Puerto.StopBits =
(System.IO.Ports.StopBits)Cmb_Stopbits.SelectedItem;
        Frm_Principal.frm_principal.Puerto.Open();

        XmlDocument doc = new XmlDocument();
        XmlElement raiz = doc.CreateElement("Configuracion");
        doc.AppendChild(raiz);
        XmlElement puerto = doc.CreateElement("Puerto");

puerto.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Puerto.SelectedIndex)));
        raiz.AppendChild(puerto);
        XmlElement baudrate = doc.CreateElement("Baudrate");

baudrate.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Baudrate.SelectedIndex)));
        raiz.AppendChild(baudrate);
        XmlElement databits = doc.CreateElement("Databits");

databits.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Databits.SelectedIndex)));
        raiz.AppendChild(databits);
        XmlElement paridad = doc.CreateElement("Paridad");

paridad.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Paridad.SelectedIndex)));
        raiz.AppendChild(paridad);
        XmlElement stopbits = doc.CreateElement("Stopbits");

stopbits.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Stopbits.SelectedIndex)));
        raiz.AppendChild(stopbits);
        doc.Save(Application.StartupPath + "\\config.xml");

        this.Close();
    }

```