

## 1. INTRODUCCIÓN Y OBJETIVOS

En esta sesión se pretende que el alumno practique el uso de las clases del .NET Compact Framework dedicadas a las comunicaciones mediante sockets TCP.

## 2. TAREAS DE LA PRÁCTICA

### 2.1. ACCESO A SERVICIO DE FECHA MEDIANTE UN SOCKET TCP

El alumno debe confeccionar una aplicación basada en formulario que conecte con el servicio de fecha de una máquina unix:

- El programa debe mostrar un Textbox con la dirección de la máquina (en la forma: xx.xx.xx.xx; por ejemplo, 156.35.151.2) y un botón de conexión.
- El resultado se debe mostrar en una etiqueta.
- Tiene que haber otro botón para terminar la aplicación.

El servicio Fecha escucha en el puerto 13 (tanto TCP como UDP) y se limita a devolver una cadena de caracteres con la fecha y hora actuales. Para recibir estos datos, el programa sólo deberá conectarse al puerto del servicio Fecha mediante un socket TCP (utiliza los ejemplos de uso de la clase Socket vistos en clase) y leer de éste la cadena de caracteres. Para leer la cadena de caracteres con la fecha y hora del *byte[]* devuelto por el socket, utiliza el método *System.Text.Encoding.ASCII.GetString()*.

### 2.2. USO DE NOMBRES DNS

A continuación, se creará una nueva versión del programa que admita direcciones basadas en nombre DNS. Por ejemplo, en el caso de la IP mostrada anteriormente como ejemplo (156.35.151.2), el nombre DNS asociado es sirio.edv.uniovi.es.

Para conseguir esto, se utilizarán las clases *IPEndPoint*, *IPHostEntry* y la clase estática *Dns*, de la siguiente manera:

```
byte []buf=new byte[80];
string maq;
IPEndPoint sirio;
if (this.textBox1.Text.Length != 0)
{
    maq=this.textBox1.Text;
    try
    {
        sirio= new EndPoint(IPAddress.Parse(maq),13);
    }
    catch (FormatException)
    {
        IPHostEntry hos= new IPHostEntry();
        hos=Dns.Resolve(maq);
        sirio= new IPEndPoint(hos.AddressList[0],13);
    }

    // Crear el socket
    ...
}
```

```
        // Conectarse al IPEndPoint sirio
        ...

        // Leer mensaje en buf y extraer la cadena, copiándola en un
        // Label
        ...
    }
```

También se podría usar el método *Dns.GetHostByName()* en vez de *Dns.Resolve()*.

### 2.3. ACCESO A SERVICIO DE FECHA MEDIANTE UN CLIENTE TCP

Crema una nueva versión de la aplicación anterior, utilizando ahora la clase *TcpClient* en lugar de la clase *Socket*.

### 2.4. USO DEL SERVICIO DE ECO

Construir un cliente del servicio de eco (echo) mediante la clase *Socket*. El servicio echo escucha en el puerto 7, tanto TCP como UDP, y devuelve los datos que le son enviados.

El programa debe ser similar al anterior, con posibilidad de indicar la dirección IP o el nombre DNS a la que conectarse. Existirán un *TextBox* en el que escribir el mensaje a enviar, un botón para ordenar el envío del texto del *TextBox* y otro botón para salir de la aplicación. Además, el formulario contendrá un *Label* para mostrar el mensaje retornado por el servidor echo.

Las acciones a ejecutar al pulsar el botón de envío serán:

- Crear un socket TCP y conectarse con el puerto 7 de la máquina especificada en el primer *TextBox*.
- Enviar a través del socket el mensaje previamente introducido en el segundo *TextBox*.
- Leer del socket y mostrar la cadena en el *Label*.

Utiliza dos variables *byte[]* distintas para enviar el mensaje al servidor de eco y para recoger su respuesta.

### 2.5. ACCESO AL SERVICIO ECHO MEDIANTE TCPCLIENT

Crema una nueva versión de la aplicación anterior, utilizando ahora la clase *TcpClient* en lugar de la clase *Socket*.