

## 1. INTRODUCCIÓN Y OBJETIVOS

En esta sesión se pretende que el alumno complete el uso de las clases del .NET Compact Framework dedicadas a las comunicaciones mediante sockets TCP. Finalmente, se sugiere la creación de una aplicación para un teléfono móvil basado en Windows CE

## 2. TAREAS DE LA PRÁCTICA

### 2.1. CREACIÓN DE UN SERVIDOR DE FECHA

Una vez practicada la creación de aplicaciones clientes, procederemos a contruir un un servicio de Fecha en el PC. En este caso, se usará la plantilla de aplicación *Proyectos de Visual C#* → *Aplicación para Windows*.

En primer lugar, se creará un servidor que devuelva una cadena de caracteres (“Hola Mundo”). El puerto al que se tendrán que conectar los clientes es el 1300. Comprueba en las transparencias de teoría como crear un socket TCP servidor.

Una vez que los clientes del servicio de fecha que habían sido creados tras completar los apartados 2.2 ó 2.3 de la sesión anterior, se conectan al servidor creado y reciben la cadena “Hola Mundo” (recuerda hacer que se conecten al puerto 1300 y no al 13), habrá que modificarla para que devuelva una cadena con la fecha y hora actuales. Para ello, utiliza la clase *DateTime* para obtener la fecha actual (`DateTime.Now.ToLongDateString()`)

Recuerda que el servidor será una aplicación de Windows y el cliente será el programa creado en 2.2 ó 2.3 de la sesión anterior, modificando el puerto al que se accede. **El emulador de Pocket PC sólo puede emular sockets cliente. Si se crea un socket servidor, el emulador no recibirá las peticiones de conexión.**

Crea una nueva versión del servidor utilizando la clase *TcpListener* en lugar de la clase *Socket*.

### 2.2. CLIENTE DEL SERVICIO DE HORA

Este ejercicio consiste en crear un cliente en .NET CF para el servicio de hora de Unix. Al conectarse al servicio y leer del socket, envía un entero de 32 bits con los segundos pasados desde el 1 de Enero de 1900 hasta ese instante. El servicio escucha en el puerto 37.

La milonga consiste en pasar ese valor a una fecha dentro de la PDA. Para eso usaremos el método *Ticks* de la clase *DateTime* que nos ofrece el valor en ticks de reloj de una determinada fecha desde el 1 de enero del año 1.

Será necesario convertir a ticks (centenas de nanosegundo) el valor enviado por el servidor de hora (en segundos). Para ellos habrá que realizar los siguientes pasos:

1. Obtener el número y convertirlo a un valor usable dentro de la PDA. Utilizaremos el método *IPAddress.NetworkToHostOrder()*. En el otro extremo, el servicio de hora habrá enviado el entero utilizando una función en C similar al método *IPAddress.HostToNetworkOrder()* de .NET. Todo esto se hace para asegurar que máquinas con distinta ordenación de los bytes (little endian vs big endian) que

componen un dato numérico con tamaño superior a un byte puedan comunicarse con éxito.

2. El uint obtenido debe ser convertido a long (los ticks son long).
3. Se multiplica ese valor por  $10^9$  (para convertir en nanosegundos) y se divide el resultado por 100 (para pasar a ticks). En resumen, se multiplicará por  $10^7$ .
4. Se crea una instancia de *DateTime* y se inicializa a 1-1-1900.
5. A esta instancia, se le sumarán los ticks calculados antes, obteniendo la fecha actual.

### **2.3. CREACIÓN DE APLICACIONES PARA TELÉFONOS MÓVILES BASADOS EN WINDOWS CE**

Este último ejercicio opcional pretende simplemente que el alumno sea consciente acerca de la dificultad que presenta la limitación de entrada de usuario en un teléfono móvil que pueda ejecutar aplicaciones .NET CF. Se pretende que el alumno cree una aplicación calculadora para uno de estos dispositivos. El tipo de plantilla de aplicación a utilizar cuando se crea el nuevo proyecto es, de nuevo, un *Proyecto para SmartDevice*, pero cuya plataforma de destino es *Smartphone*. Si Visual Studio .NET no ofrece esa plataforma de destino, será necesario instalar el SDK para Smartphone, disponible en la página de Microsoft.

Se recomienda observar cómo funciona la calculadora incluida con el emulador de Smartphone, para obtener ideas acerca de cómo implementar nuestra calculadora.