

**Esempio Pilota: DadiInRete1Byte1ConnApriChiudi**

Realizzare un'**Applicazione di Rete** che consenta ad un **Client** di connettersi al **Server** per ricevere un numero (un singolo byte) ottenuto simulando il **Lancio di un Dado**. Il Server ammette solo un Client per volta. La connessione viene chiusa appena inviato il numero al Client.

**Applicazione CLIENT****Definizione della Form del Client**
**Codice della Form del Client (Linguaggio C#)**

```
using System;
using System.Windows.Forms;
```

```
// necessaria per TcpListener, TcpClient, NetworkStream ...
using System.Net.Sockets;
```

```
namespace DadiCLN
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        TcpClient CN;
```

```
private void plsInvia_Click (object sender, EventArgs e)
{
    // crea l'oggetto connessione ...
    CN = new TcpClient ( );

    // invia al server la RICHIESTA DI CONNESSIONE ...
    CN.Connect ( txtIP.Text, 57 );

    // avvia il timer per l'attesa dei dati ...
    tmrAD.Start ( );
}
```

```
private void tmrAD_Tick (object sender, EventArgs e)
{
    // se sono arrivati dati dal server ...
    if ( CN.Available > 0 )
    {
        // ottieni il Flusso Dati della connessione...
        NetworkStream NS = CN.GetStream ( );

        // ricevi (leggi) dal Flusso Dati, il byte arrivato ...
        byte LancioDado = (byte) NS.ReadByte ( );

        lbltotaleDadi.Text = LancioDado.ToString ( );

        // ferma il timer per l'attesa dei dati ...
        tmrAD.Stop ( );

        // Chiudi la connessione ...
        CN.Close ( );
    }
}
```

## Applicazione SERVER

### Definizione della Form del Server



### Codice della Form del Server (Linguaggio C#)

```
using System;
using System.Windows.Forms;

using System.Net.Sockets;

namespace DadiSRV
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // crea l'oggetto R per generare numeri casuali ...
            Random R = new Random ();

            TcpListener LST; // gestisce l'attesa per richieste di conness.
            TcpClient CN;    // gestisce la singola connessione

            private void Form1_Load (object sender, EventArgs e)
            {
                // imposta l'attesa sulla porta di rete N. 57 ...
                LST = new TcpListener (57);

                // attiva l'attesa per le richieste di connessione ...
                LST.Start ();

                // avvia il Timer per verificare l'arrivo di richieste ...
                tmrAC.Start ();
            }

            private void tmrAC_Tick (object sender, EventArgs e)
            {
                // se sono arrivate richieste di connessione ...
                if ( LST.Pending () )
                {
                    // ottieni l'oggetto connessione dal TcpListener ...
                    CN = LST.AcceptTcpClient ();

                    // ferma l'attesa di richieste e il relativo timer ...
                    LST.Stop ();
                    tmrAC.Stop ();
                }
            }
        }
    }
}
```

```
// genera il lancio del dado (numero da 1 a 6) ...
byte LancioDado = (byte) R.Next (1, 7);

// ottieni l'oggetto flusso dati dalla connessione ...
NetworkStream NS = CN.GetStream ();

// invia il numero sul flusso dati della connessione ...
NS.WriteByte ( LancioDado );

lblMessaggio.Text = "Inviato il numero: " + LancioDado;

// Chiudi la connessione e riattiva l'attesa ...
CN.Close ();
LST.Start ();
tmrAC.Start ();
}
}
```