Se pide realizar una aplicación cliente/servidor, utilizando las clases que se han visto en clase, que implemente un tablón de anuncios.

El funcionamiento general es el siguiente:

* Un servidor acepta conexiones y lleva el control de los clientes conectados y los mensajes publicados en una pequeña base de datos.
* Los clientes se conectan y mientras están conectados tienen la posibilidad de publicar mensajes y de recuperar los datos enviados por el resto de los clientes, e incluso filtrar según ciertos criterios.

# Servidor:

La base de datos en la que se basa el servidor es la siguiente:

**CLIENTES**

Ip (String)

UltimaConexion (date time)

**MENSAJES**

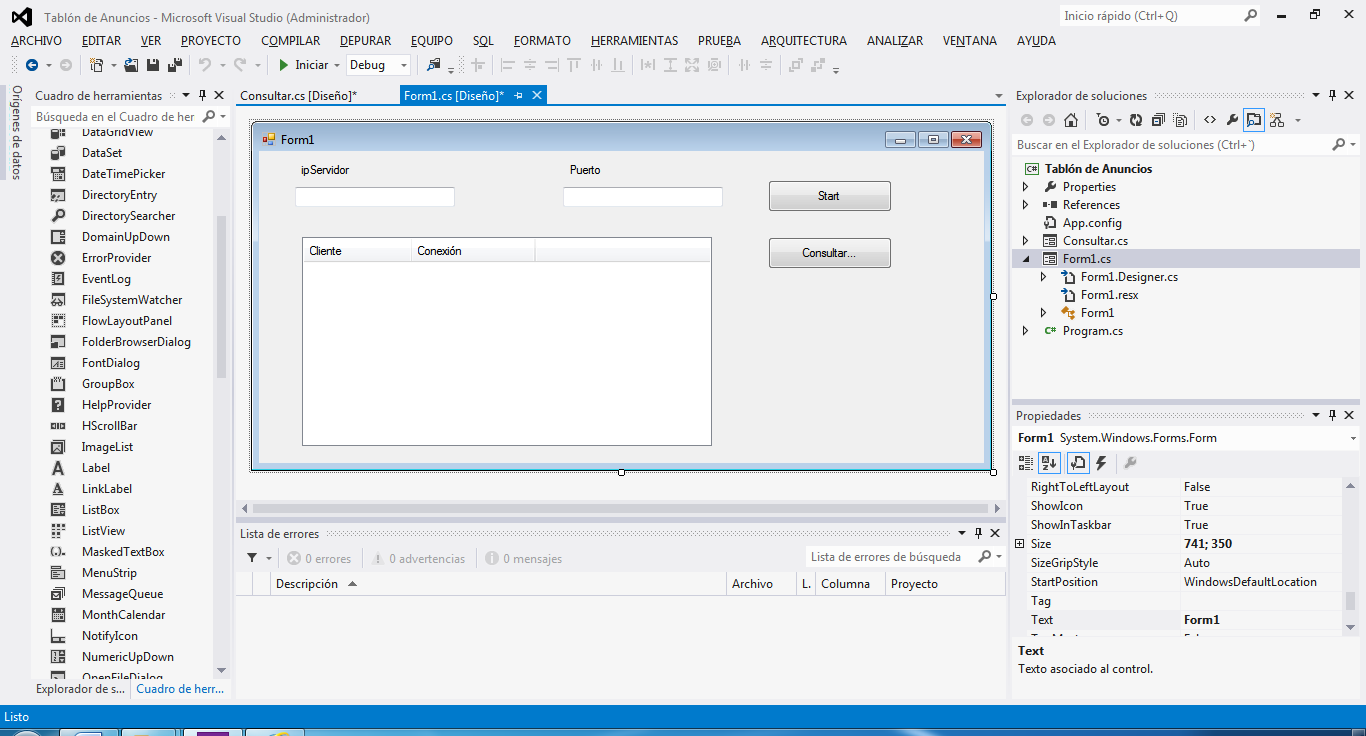
Id (int)

Texto mensaje (String)

Ip (string)

Publicacion (DateTime)

El formulario servidor es el siguiente:



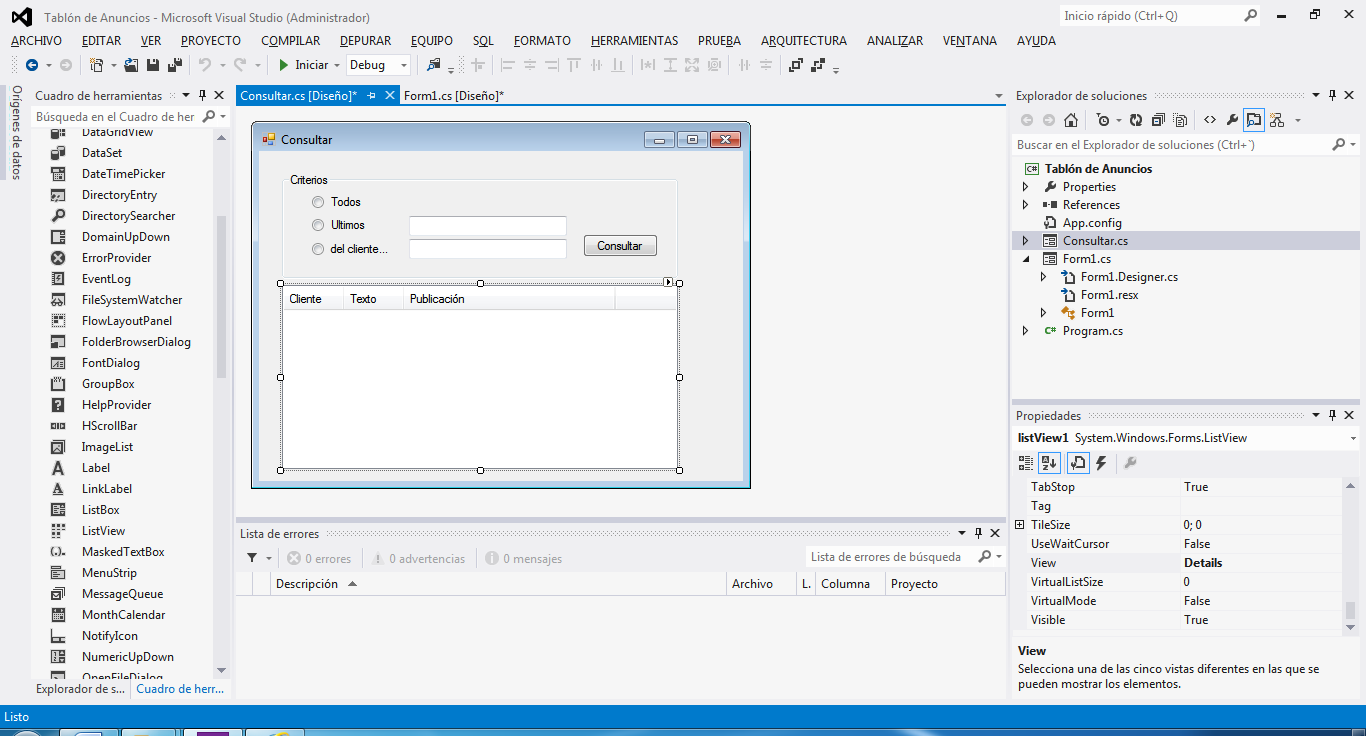
* Al pulsar “Start” Crearía un objeto de la clase servidor vista en clase e iniciaría el servidor.
* Al recibir una conexión añadiría los datos a la listview y a la base de datos y, si es la primera vez que conecta ese cliente, añadirá la ip a la base de datos. Si no es la primera vez, simplemente modificará la fecha de la última conexión.
* Al desconectar un cliente, borraría el cliente de la listview.
* Al recibir datos, los datos podrán recibirse de la siguiente forma:
  + **Comando All:** si el mensaje solo contiene la palabra “All” formará una cadena como la siguiente “texto1;texto2;texto3…” con todos los mensajes de la base de datos y la transmitirá al cliente que lo ha solicitado.
  + **Comando Last:** se recibirá algo como lo siguiente “Last;número”, formará una cadena similar a la del apartado anterior con los *número* últimos mensajes insertados con el formato indicado anteriormente.
  + **Comando Msg:** se recibirá algo como lo siguiente “Msg;texto” agregará un nuevo registro a la tabla de mensajes con la siguiente información:
    - **Id:** el id del último mensaje almacenado +1. Si no hay mensajes 1.
    - **Texto:** el texto recibido.
    - **Ip:** la del cliente que envía el texto.
    - **Fecha:** la de hoy.
* En cualquier momento se podrá desconectar a un cliente pulsando doble sobre el en la listview y previa confirmación.
  + **Comando Client:** se recibirá algo como lo siguiente “Client;*ipcliente”* formará una cadena similar a la de apartados anteriores, con los textos de los mensajes de *ipcliente.*

Para descomponer la cadena recibida del cliente utilizar el método Split

String []cadena=datos.Split(new char[]{';'});

La función Split toma una cadena y la divide en palabras, tomando como separador el carácter que se le indique. Es por eso que lo que nos envía el cliente, son los ítems separados por “;” en una misma cadena.

Al pulsar el botón consultar aparecerá el siguiente formulario:

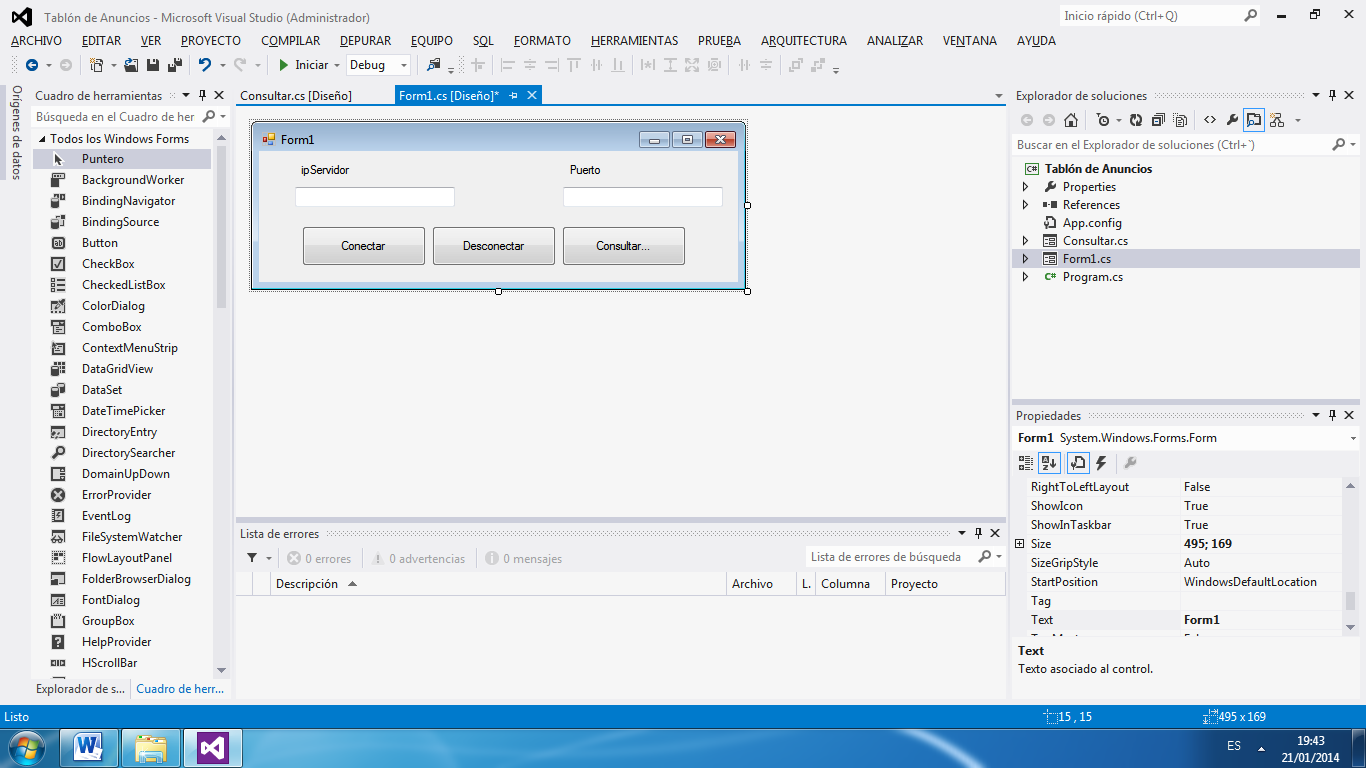


De manera que al pulsar consultar:

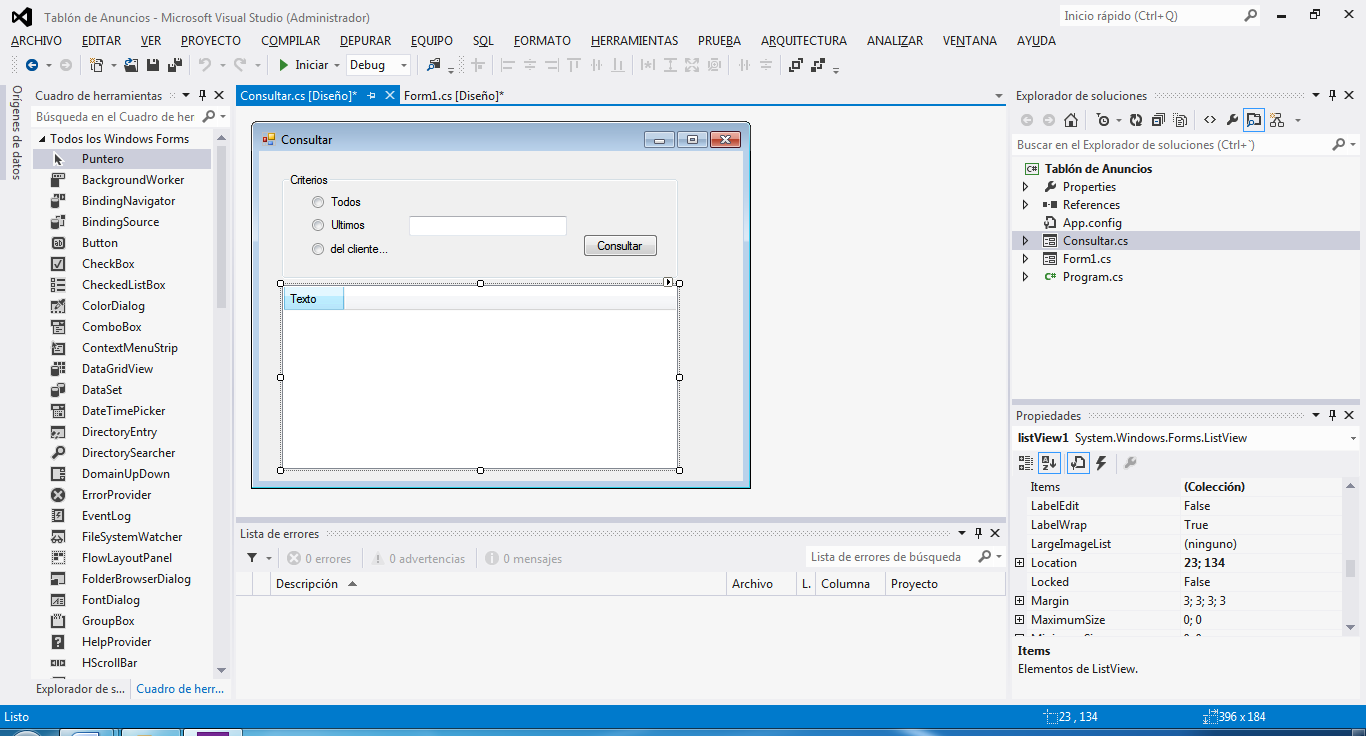
* Si está seleccionado todos mostrará en la listview todos los mensajes, con los datos que se indican.
* Si está seleccionado últimos y hay un número (***n)*** en la caja de texto correspondiente mostrará los n mensajes más recientes. Si no hay número mostrará un msgbox para que se introduzca.
* Si está seleccionado “del cliente…” y está introducida la ip del cliente, todos los mensajes subidos desde esa dirección. Si no hay ip mostrará un msgbox para que se introduzca.

# Cliente:

El cliente tendrá el siguiente formulario principal.



* Al pulsar conectar se conectará al servidor.
* Al pulsar desconectar se desconectará del servidor.
* Al pulsar consultar aparecerá un formulario similar al que aparece en el servidor:



Y hará lo siguiente:

* Si está seleccionada la opción todos, enviará el comando **All** al servidor.
* Si está seleccionado ***Últimos*** e introducido un número se enviará al servidor una cadena como la siguiente.

“All;numero” donde número es un entero introducido en la caja de texto correspondiente. Si no hay número, se mostrará un msgbox para indicar que se debe de introducir.

* Si está seleccionado ***Del Cliente*** se enviará al servidor una cadena “Client;*ipcliente”* donde *IpCliente* es la ip del socket del lado de cliente.

Al recibir información habrá que descomponer la cadena recibida mediante la orden Split explicada en 1 y visualizar la información correspondiente en la listView.