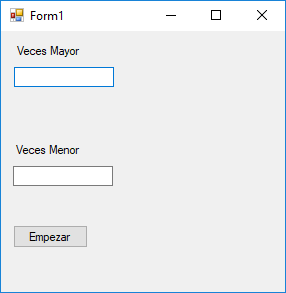
Tenemos una aplicación para jugar a adivinar el número. Para ello contamos con tres componentes: un formulario, un cliente y un servidor. (Tener en cuenta que las funciones para declarar y enviar mensajes, tienen que estar declaradas y accesibles tanto desde el formulario como desde el proceso que envía los mensajes y utilizadas correctamente, en caso contrario todas las operaciones con mensajes se puntuarán como incorrectas.).

1. El formulario:



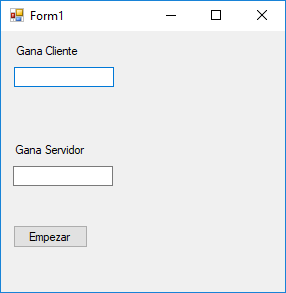
* + El botón empezar lanzará los procesos Cliente y Servidor.
  + Cuando se recibe el mensaje “WM\_MAYOR”, incrementará en 1 el contador “Veces mayor”.
  + Cuando se recibe el mensaje “WM\_MENOR” incrementará en 1 el contador “Veces menor”.
  + Cuando se recibe el mensaje “WM\_IGUAL” mostrará un MessageBox con el número de intentos que ha costado acertar el número (Veces Menor + Veces Mayor).

1. El Servidor. Será una aplicación de consola que:
   1. Crea un pipe de servidor para comunicarse con el cliente con el nombre que se quiera y la dirección necesaria.
   2. Se pone a la espera de conexión por parte del cliente.
   3. Crea el/los stream/s necesarios para enviar y/o recibir los datos.
   4. Saca un número al azar entre 1 y 100 (n1). (podemos usar la expresión ***new Random(DateTime.Now.Millisecond).Next(1, 100);*** para ello).
   5. Lee el número que le enviará el cliente (n2).
   6. Mientras n1 sea distinto de n2 repetir.
      1. Muestra n2 por pantalla.
      2. Si n2 es mayor que n1 enviará al cliente la palabra “Mayor”.
      3. Si n2 es menor que n2 enviará al cliente la palabra “Menor”.
      4. Lee nuevamente el número que le enviará el cliente (n2) y vuelve a f.
   7. Cuando ambos números sean iguales enviará al cliente la palabra “correcto” y terminará el programa.
2. El cliente. Será una aplicación de consola que:
   1. Crea un pipe de cliente para comunicarse con el pipe del servidor.
   2. Se conecta al pipe del servidor.
   3. Lee un número de consola.
   4. Envía el número al servidor.
   5. Lee la respuesta del servidor.
   6. Mientras la respuesta no sea “correcto”.
      1. Si la respuesta es “mayor” envía el mensaje “WM\_MAYOR” al formulario.
      2. Si la respuesta es “menor” envía el mensaje “WM\_MENOR” al formulario.
   7. Cuando la respuesta es “correcto” envía el mensaje “WM\_IGUAL” al formulario.
   8. Finaliza el programa.

Tanto en 2 como en 3 finalizar el programa implica el cierre ordenado de los pipes y los streams que permanezcan abiertos.

Tenemos una aplicación para jugar al número más alto. Para ello contamos con tres componentes: un formulario, un cliente y un servidor. (Tener en cuenta que las funciones para declarar y enviar mensajes, tienen que estar declaradas y accesibles tanto desde el formulario como desde el proceso que envía los mensajes y utilizadas correctamente, en caso contrario todas las operaciones con mensajes se puntuarán como incorrectas.).

1. El formulario.



* 1. El botón empezar lanzará los procesos Cliente y Servidor.
  2. Cuando se recibe el mensaje “WM\_CLIENTE”, incrementará en 1 el contador “Gana Cliente”.
  3. Cuando se recibe el mensaje “WM\_SERVIDOR” incrementará en 1 el contador “Veces Servidor”.
  4. Cuando se recibe el mensaje “WM\_FIN” sacará un msgbox mostrando el número de veces que ha ganado el cliente y el número de veces que ha ganado el servidor.

1. El Servidor. Será una aplicación de consola que:
   1. Crea un pipe de servidor para comunicarse con el cliente con el nombre que se quiera y la dirección necesaria.
   2. Se pone a la espera de conexión por parte del cliente.
   3. Crea el/los stream/s necesarios para enviar y recibir los datos.
   4. Saca un número al azar entre 1 y 100 (n1) y lo muestra por pantalla. (Podemos usar la expresión ***new Random(DateTime.Now.Millisecond).Next(1, 100);***para ello)
   5. Lee el número que le enviará el cliente (n2) y lo muestra por pantalla.
      1. Si n2 es mayor que n1 enviará al cliente la palabra “cliente”.
      2. Si n2 es menor que n1 enviará al cliente la palabra “servidor”.
      3. Si son iguales enviará el cliente la palabra iguales.
   6. Lee la respuesta del cliente y mientras sea “SI”, volverá al punto (d.
   7. Cuando la respuesta es “NO” finaliza el programa.
2. El cliente. Será una aplicación de consola que:
3. Crea un pipe de cliente para comunicarse con el pipe del servidor.
4. Se conecta al pipe del servidor.
5. Genera un número entre 1 y 100 al azar y lo muestra por pantalla.
6. Envía el número al servidor.
7. Lee la respuesta del servidor.
   * 1. Si la respuesta es “servidor” envía el mensaje “WM\_SERVIDOR” al formulario.
     2. Si la respuesta es “cliente” envía el mensaje “WM\_CLIENTE” al formulario.
8. Pregunta por consola al usuario si desea volver a jugar.
9. Envía la respuesta del usuario al servidor.
10. Mientras la respuesta sea “SI” vuele al paso c).
11. Lanza el mensaje “WM\_FIN” al formulario.
12. Finaliza el programa.

Tanto en 2 como en 3 finalizar el programa implica el cierre ordenado de los pipes y los streams que permanezcan abiertos.