

Dossier del Proyecto

# Bingo

Por David Ferrandez Molla

Actualización 19/02/2023

GITHUB: <https://github.com/hopesend/BingoProject>

## **1. Descripción General del Proyecto**

Se pretende desarrollar una aplicación que simule un juego de Bingo, tanto en su versión de Servidor y su Versión de Cliente. El servidor generará el juego y distribuirá los números al azar a los Clientes. Estos a su vez podrán cantar una Línea o un Bingo según se distribuyan los números en su cartón.

## **2. Objetivos**

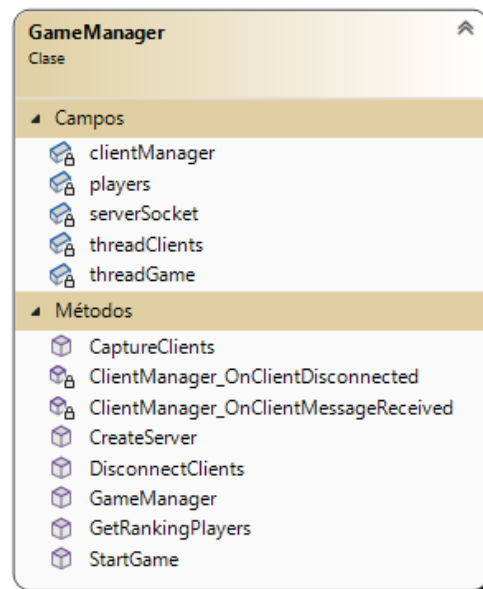
- Crear el Servidor del Bingo.
  - Crear el socket del servidor
  - Capturar los clientes que conecten al socket
  - Crear el algoritmo del juego
  - Crear el sistema de mensajes de servidor a clientes
  - Crear el sistema de mensajes de clientes a servidor.
- Crear el Cliente del Bingo
  - Crear el socket de conexión al servidor
  - Crear el sistema de recepción de mensajes del Servidor
  - Crear el sistema de envío de mensajes al servidor
  - Creación de visual para el marcado del cartón del Bingo

### 3. Módulos/Unidades funcionales del Proyecto

- **Bingo Server:**

- **GameManager.cs**

Clase encargada de la lógica del juego, creara el servidor, maneja los eventos de los clientes, tanto de recepción de mensajes como de desconexión de los mismos, capturara los clientes y su maneja su desconexión.



- **PARAMETROS**

- clientManager: Gestor de Clientes.
    - players: Cantidad de jugadores en el juego actual.
    - sockectSever: Gestor del Servidor
    - threadClients: Hilo de Captura de Clientes.
    - threadGame: Hilo del juego.

- **METODOS**

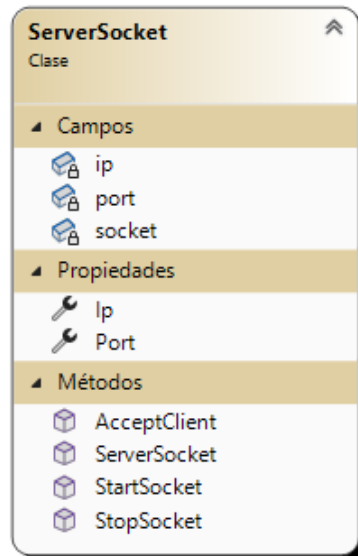
- CaptureClients: Método para el Hilo de captura de Clientes
    - Eventos de Cliente Desconectado y Cliente a enviado mensajes
    - CreateServer: Crea el servidor.
    - DisconnectClients: Fuerza la desconexión de los clientes.
    - GetRankingPlayers: Obtiene el ranking de jugadores en la sesión del servidor
    - StartGame; Metodo para el Hilo de Juego

- **TIEMPO ESTIMADO**

- 1 horas.

- **ServerSocket.cs**

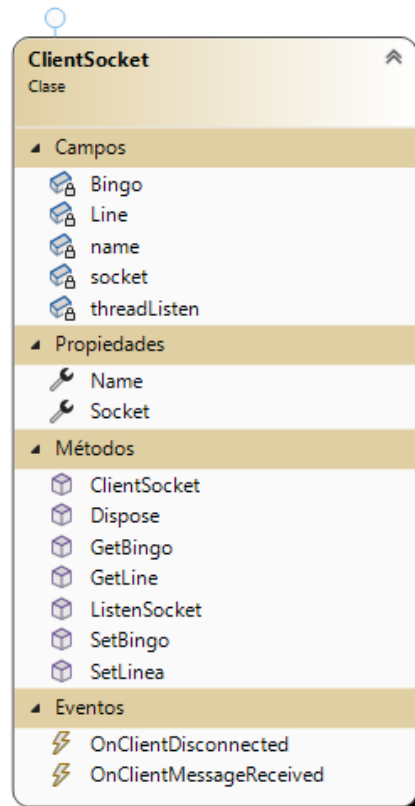
Encapsula los parámetros de información del servidor y su socket.



- **PARAMETROS**
  - `Ip`: Información de la ip del servidor
  - `Port`: Información del puerto del servidor
  - `Socket`: Socket servidor
- **METODOS**
  - `AcceptClients`: Encapsula el método del socket para aceptar clientes
  - `StartSocket`: Encapsula el método del socket para inicializar el servidor
  - `StopSocket`: Encapsula el método del socket para parar el servidor
- **TIEMPO ESTIMADO**
  - 1 hora.

- **ClientSocket.cs**

Encapsula es socket del cliente y genera la escucha de mensajes entrantes.



- **PARAMETROS**

- Bingo: Registra el conteo de los bingos generados por el usuario en la sesión, se usará para el ranking.
- Line: Registra el conteo de las líneas generadas por el usuario en la sesión, se usará para el ranking.
- Name: Nombre del jugador en la sesión.
- Socket: Socket Cliente.
- threadListen: Hilo gestor de la escucha del cliente.

- **METODOS**

- Dispose: Cierra el socket y libera el objeto.
- ListenSocket: Metodo para el hilo de escucha del cliente.

- **EVENTOS:**

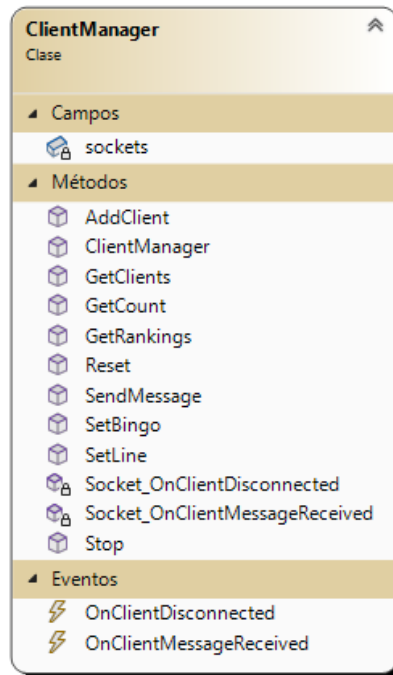
- OnClientDisconnected: Cuando el cliente se desconecta de la sesión
- OnClientMessageRecieved: Cuando el cliente ha lanzado un mensaje al servidor.

- **TIEMPO ESTIMADO:**

- 1 hora.

- **ClienteManager.cs**

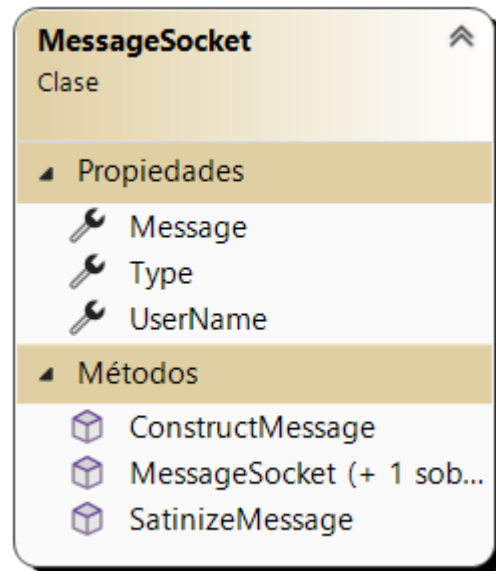
Es el gestor de clientes que se encargara de enviar información de los clientes al juego.



- **PARAMETROS:**
  - sockets: colección de objetos ClientSocket, encapsulando su acceso
- **METODOS:**
  - AddClient: Añade un cliente a la colección.
  - GetClients: Obtiene los sockets de los clientes.
  - GetCount: Cuenta los clientes.
  - GetRankings: Obtiene los rankings de la sesión.
  - Reset: Libera los clientes y crea una nueva colección.
  - SendMessage: Envía un mensaje a todos los clientes de la colección
  - SetBingo: Incrementa el parámetro de Bingo en el Cliente seleccionado
  - SetLine: Incrementa el parámetro de Line en el Cliente Seleccionado.
  - Stop: Aborta todos los sockets de cliente
- **TIEMPO ESTIMADO**
  - 2 horas.

- **MessageSocket.cs**

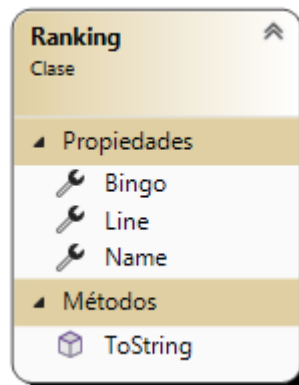
Es la encargada de castear los mensajes tanto de clientes como de servidor



- **METODOS:**
  - ConstructMessage: Maqueta un mensaje para su envío
  - MessageSocket: Dos constructores tanto para envío como para recepción.
- **TIEMPO ESTIMADO**
  - 15 minutos.

- **Ranking.cs**

Información del ranking de usuario en la sesión

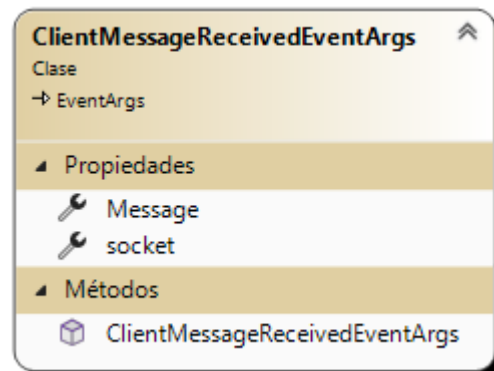


- PROPIEDADES
  - Bingo: Cuenteo de bingos en la sesión
  - Line: Cuenteo de líneas en la sesión.
  - Name: Nombre de Usuario.
- METODOS:
  - ToString: Override del método para la muestra en mensaje.
- TIEMPO ESTIMADO
  - 15 minutos.



- **ClientMessageRecievedEventArgs.cs**

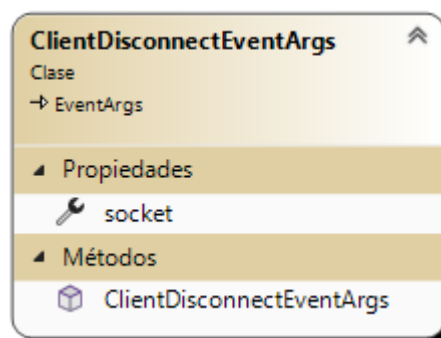
Evento de envío de mensaje de cliente.



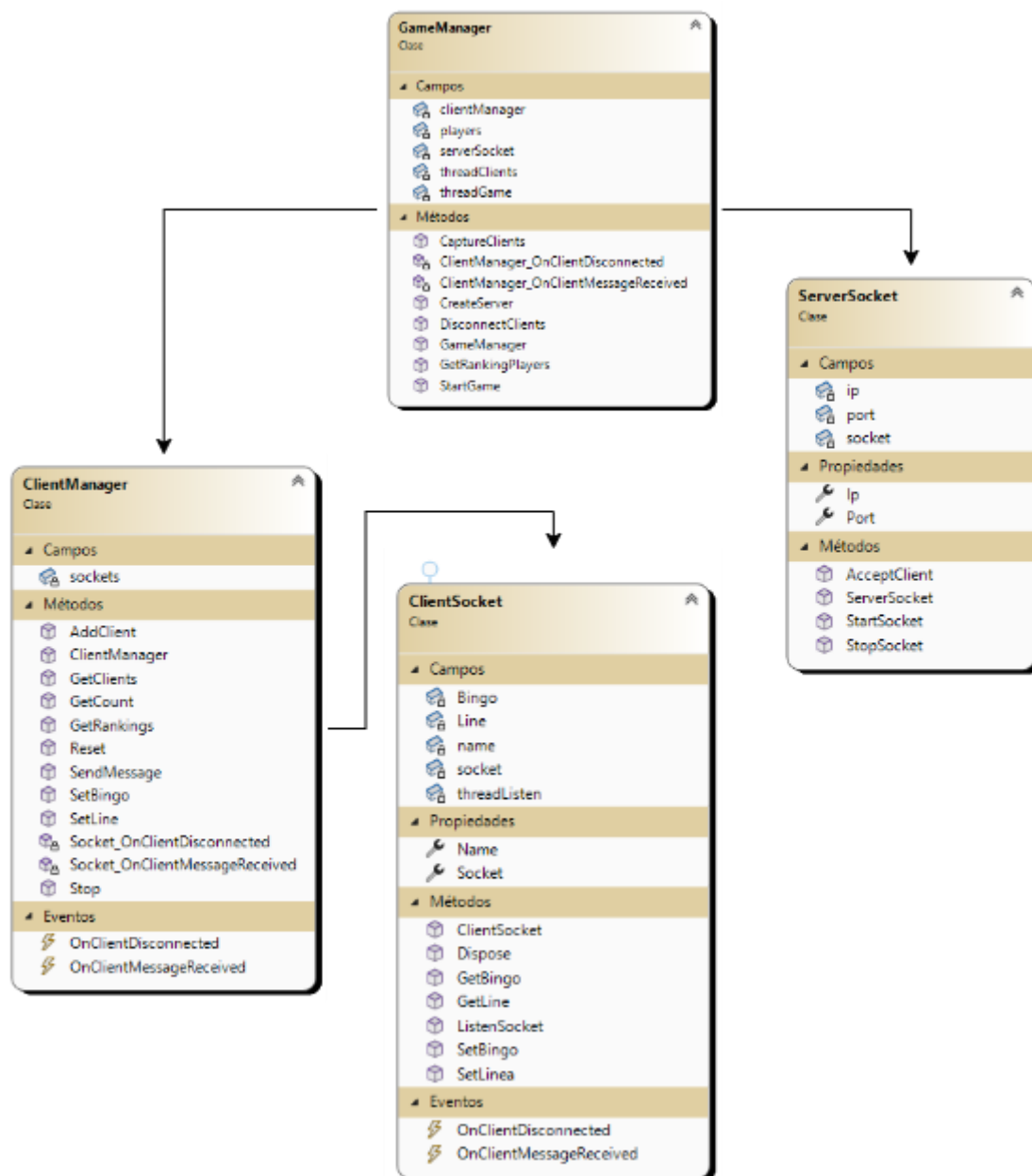
- PROPIEDADES
  - Mensaje: Mensaje formateado para servidor
  - socket: ClientSocket que lanza el mensaje.
- TIEMPO ESTIMADO
  - 15 minutos.

- **ClientDisconnectEventArgs.cs**

Evento de cliente desconectado.

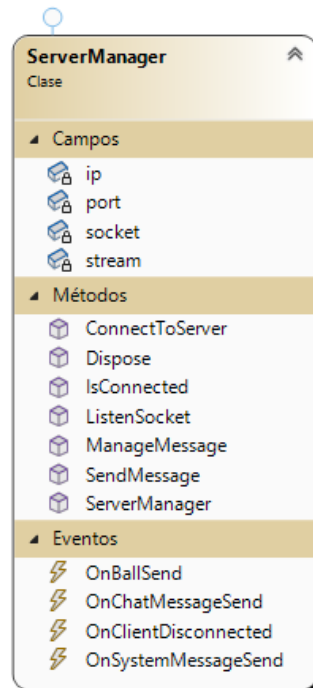


- PROPIEDADES
  - socket: ClientSocket que se desconecta
- TIEMPO ESTIMADO
  - 15 minutos.



- **Bingo Client:**
  - **ServerManager.cs**

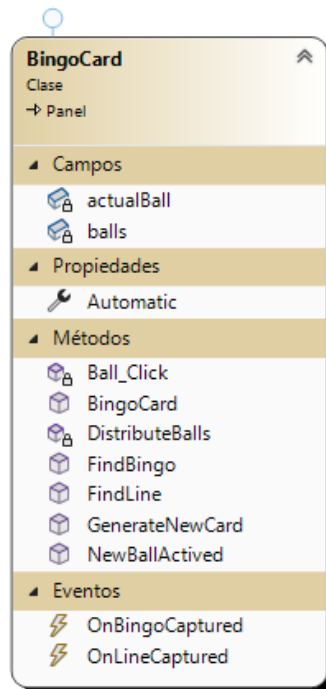
Gestor del servidor del bingo, maneja eventos de servidor y envía mensajes.



- PROPIEDADES
  - ip: información de la ip del servidor
  - port: información del puerto del servidor
  - socket: socket del servidor
  - stream: stream del socket servidor
- METODOS:
  - ConnectToServer: Conecta con el servidor.
  - Dispose: Desconecta el servidor.
  - IsConnected: Checkea que el servidor este operativo.
  - ListenSocket: Escucha a espera de mensajes
  - ManageMessage: Formatea el mensaje recibido
  - SendMessage: Envía mensaje al servidor
- TIEMPO ESTIMADO
  - 2 horas.

- **BingoCard.cs**

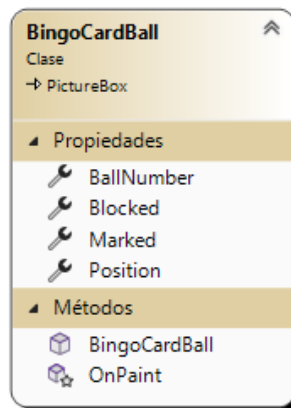
Control del cartón del bingo, crea los visuales para el formulario



- PROPIEDADES
  - actualBall: la bola generada en el servidor
  - balls: colección con las bolas del cartón
  - Automatic: Activa/Desactiva el juego automático.
- METODOS:
  - Ball\_Click: Evento de click en la bola.
  - DistributeBalls: Distribuye las bolas en el cartón
  - FindBingo: Busca un Bingo en el cartón.
  - FinLine: Busca una Línea en el cartón.
  - GenerateNewCard: Genera un nuevo cartón.
  - NewBallActivated: Captura de la bola generada en el servidor.
- EVENTOS
  - OnBingoCaptured: Evento para la captura de un nuevo Bingo.
  - OnLineCaptured: Evento para la captura de una nueva Línea.
- TIEMPO ESTIMADO
  - 1 horas.

- **BingoCardBall.cs**

Control de la bola del cartón.



- PROPIEDADES

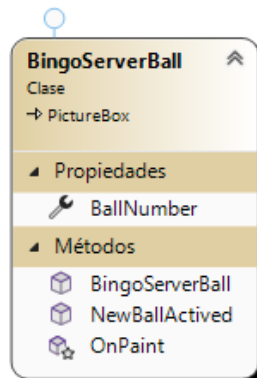
- BallNumber: Numero de la bola.
    - Blocked: En estado bloqueada por Bingo o Línea.
    - Marked: En estado marcada por click.
    - Position: Posición en el cartón de la bola.

- TIEMPO ESTIMADO

- 1/2 hora.

- **BingoServerBall.cs**

Control de la bola del Formulario.



- PROPIEDADES

- BallNumber: Numero de la bola.
    - Blocked: En estado bloqueada por Bingo o Línea.
    - Marked: En estado marcada por click.
    - Position: Posición en el cartón de la bola.

- METODOS

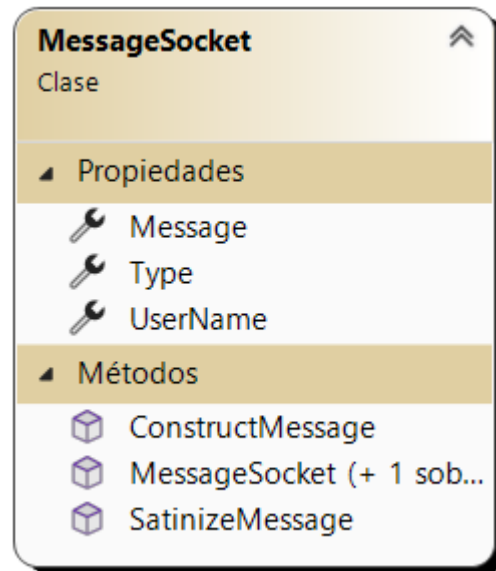
- NewBallActivated: Captura la nueva bola del servidor.

- TIEMPO ESTIMADO

- 1/2 hora.

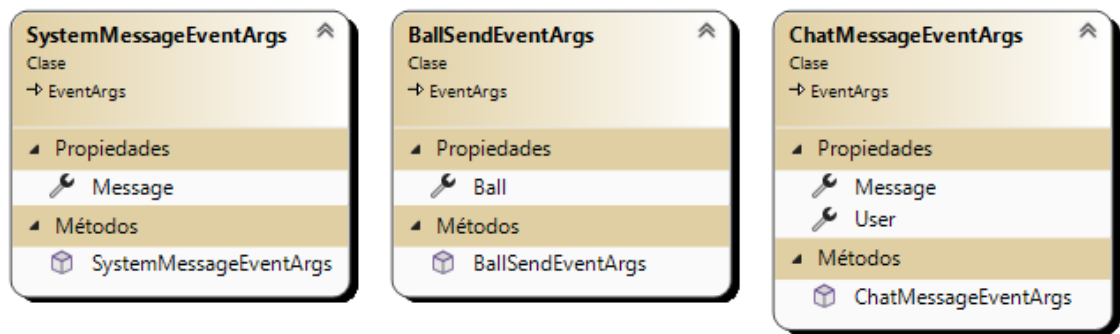
- **MessageSocket.cs**

Es la encargada de castear los mensajes tanto de clientes como de servidor



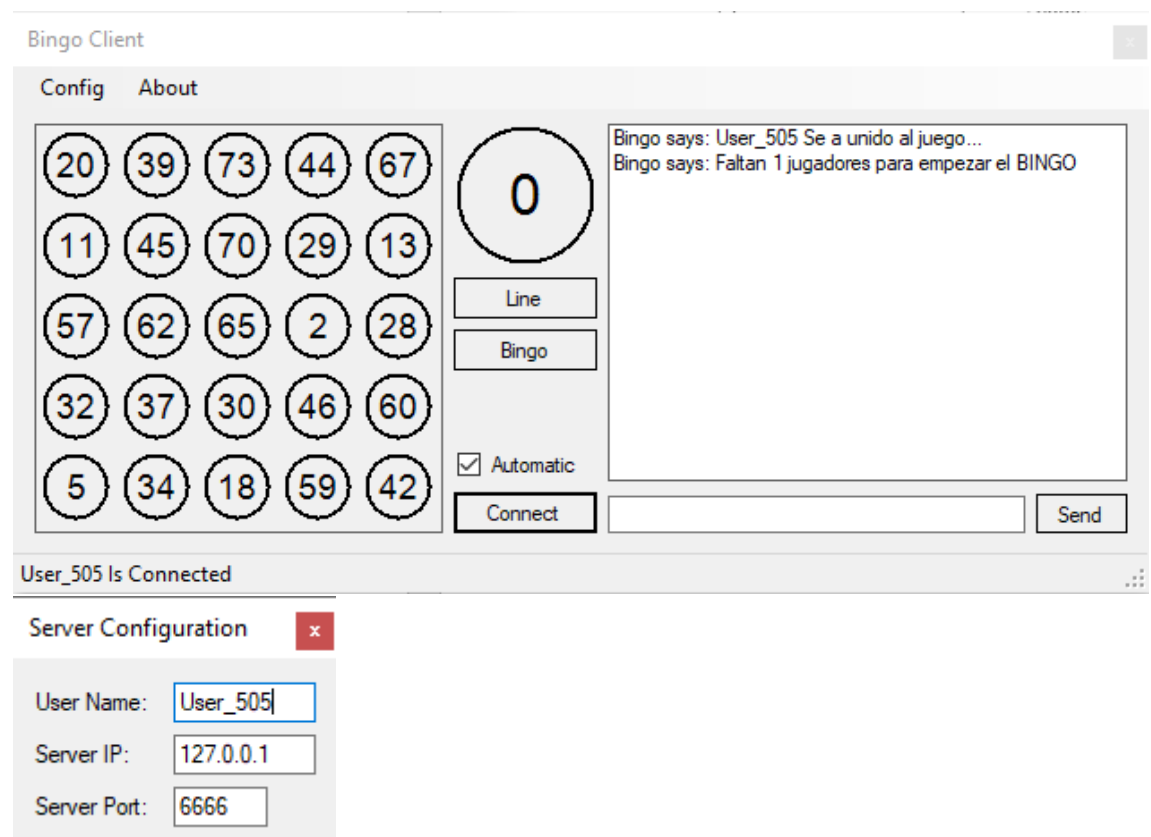
- **METODOS:**
  - ConstructMessage: Maqueta un mensaje para su envío
  - MessageSocket: Dos constructores tanto para envío como para recepción.
- **TIEMPO ESTIMADO**
  - 15 minutos.

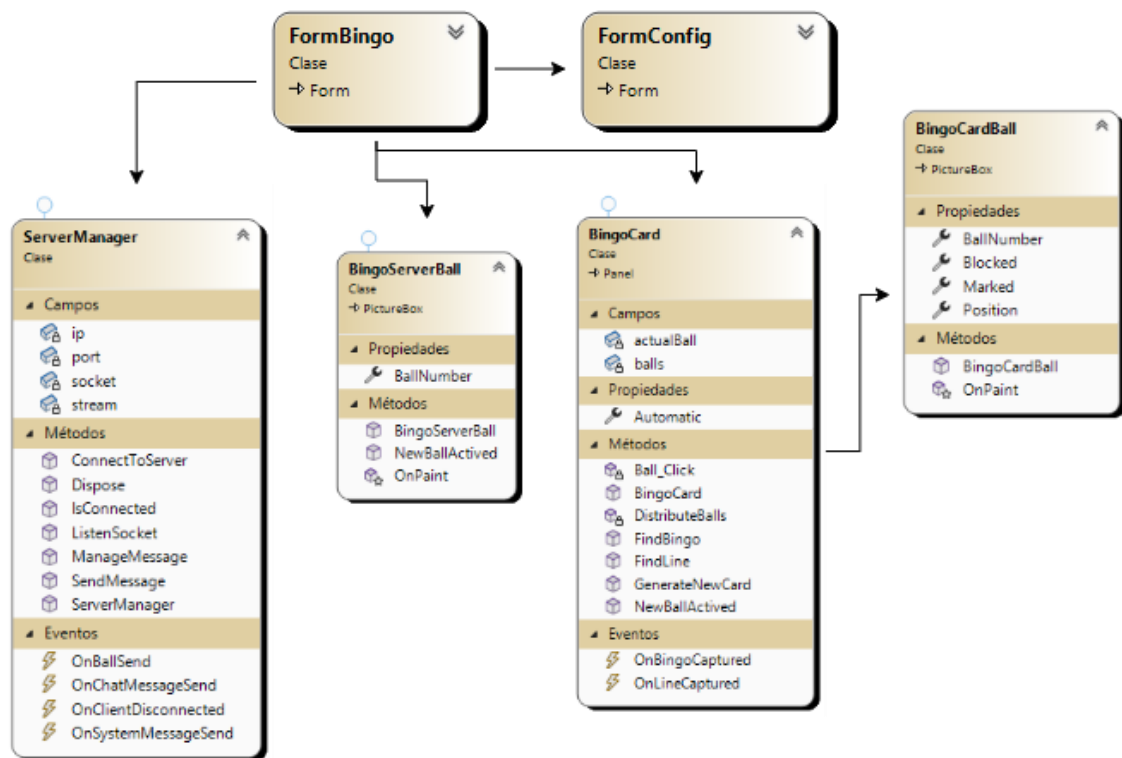
- **Eventos para la captura de los mensajes**



- **Interfaces visuales para el juego**

Es la encargada de castear los mensajes tanto de clientes como de servidor

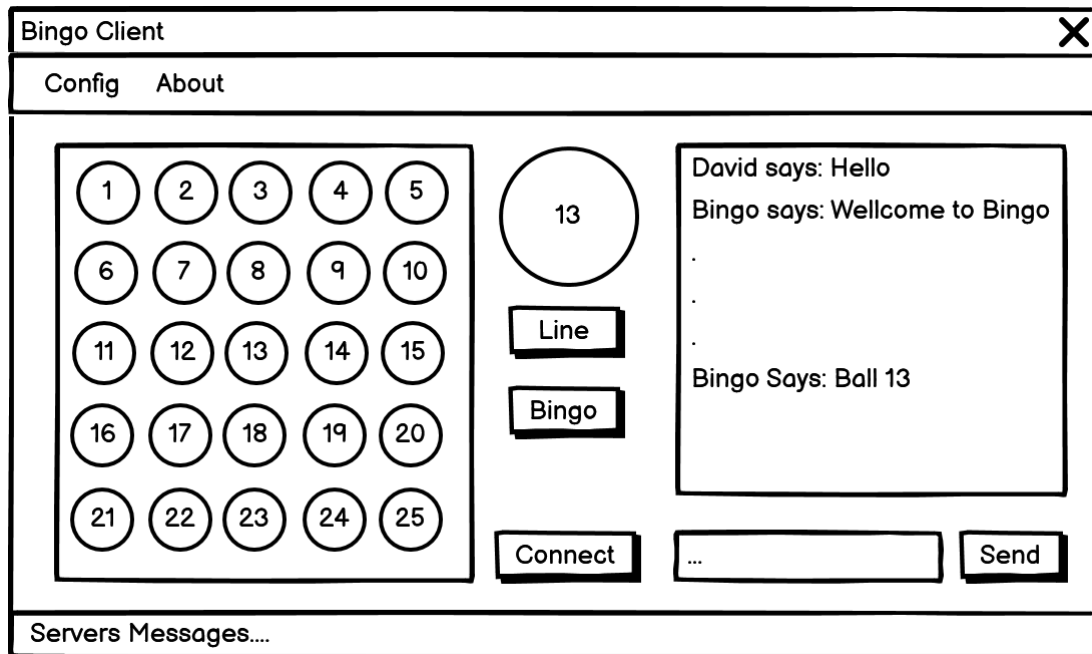




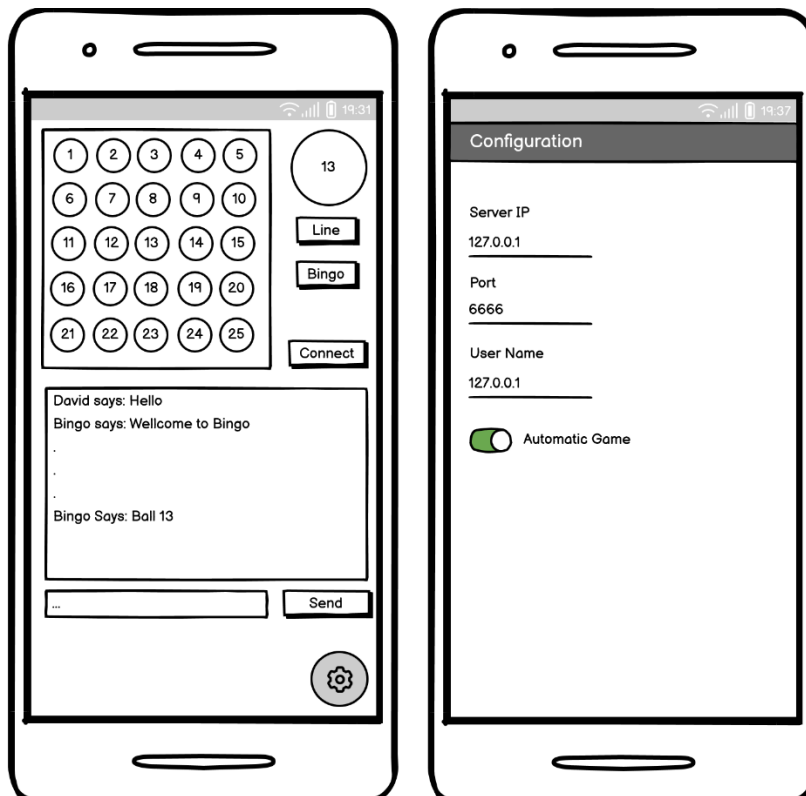


#### 4. Wireframes

- **Bingo Server**
  - Aplicación de consola sin Interface Visual
- **Bingo Client**
  - Versión Escritorio Winforms



- Versión Android



## 5. Información de la Implementación

Toda la implementación inicial se generará en C# con Visual Studio 2022, tanto el Servidor como el Cliente. (Completado)

- Posibles Implementaciones hasta la fecha de entrega
  - Servidor en Java (En Proceso) (Tiempo Estimado: 6h)
  - Cliente en Java
  - Cliente en Android (En Proceso) (Tiempo Estimado: 6h)
  - Servidor en PHP (En Proceso) (Tiempo Estimado: 6h)
- Posibles añadidos y mejoras al proyecto
  - Modo Manual/Automático (Completado)
  - Ranking de Jugadores (Completado)
  - Diferentes modalidades de Bingo (No Completado)
  - Implementación de Encriptación de Mensajes (Completado)

## 6. Conclusiones

- Implementación de Sockets:
  - La complejidad de la implementación de los Sockets tanto en Cliente como en el Servidor a sido baja, refactorizando al final del proyecto con un sistema de NetworkStream más en consonancia con los bytes recibidos y no forzados a 1024. Quijas para una mejora en un futuro en la aplicación se debería de implementar SslStream con certificado para la encriptación del canal de comunicación. Se a solventado con una encriptación en BASE64 del mensaje.
- Implementación de Interface Visual y Componentes Visuales en Cliente:
  - Se han creado componentes únicos tanto para el cartón como para las bolas, al implementar el sistema automático y la parametrización del tiempo de la bola en el servidor, la curva de complejidad en los controles ha sido baja.
- Aplicación en Conjunto:
  - No ha sido una aplicación muy compleja para su implementación, su fase de bosquejo y diseño a sido corto y funcional. Como he dicho arriba, quizás con una implementación de comunicación SSL habría terminado dejando bastante atada la aplicación final. He considerado que las aplicaciones finales no necesitaban de un motor de instalación, dejando los dos ejecutables en las carpetas correspondientes.