Documentación proyecto Battleship SignalR

Contenido

Introducción

En el presente proyecto se presentará una prueba de concepto para demostrar el uso, ventajas y desventajas de la utilización de la tecnología WebSockets usando la librería de .NET 6 llamada SignalR. Nuestro objetivo principal es ver cómo varios clientes se comunican en tiempo real a través de un servidor que implemente esta tecnología.
Para demostrar esta comunicación bidireccional implementaremos de manera sencilla y superficial el juego de batalla naval (Battleship) usando un cliente web, para ello como lo explicaremos más adelante por fines demostrativos tratamos de imitar la comunicación que se tendría en una partida real cara a cara del juego de mesa.

Objetivos

- Demostrar ventajas y desventajas de implementar Websockets con SignalR
- Ver posibles casos en los que sea conveniente usar esta tecnología

Diagrama de Gantt

Battleship websockets-signalR

Sophos Solutions
Santiago Palmet Salcedo

Inicio del proyecto: Fri, 5/12/2023
Semana para mostrar: 1

TAREA	ASIGNADO A	PROGRESO	INICIO	FIN
Diseño y Arquitectura				
Planificar la estructura y arquitectura generales	Santiago Palmet Salcedo	100%	12-5-23	16-5-23
Integración con SignalR				
Estudio de la documentación	Santiago Palmet Salcedo	100%	17-5-23	18-5-23
Implementación de la librería	Santiago Palmet Salcedo	100%	18-5-23	19-5-23
Implementación del websocket	Santiago Palmet Salcedo	100%	22-5-23	23-5-23
Implementacion de peticiones Servidor-Cliente	Santiago Palmet Salcedo	100%	23-5-23	24-5-23
Motor del juego				
Implementación de la UI basica	Santiago Palmet Salcedo	100%	25-5-23	26-5-23
Manejo de la lógica	Santiago Palmet Salcedo	100%	27-5-23	31-5-23
Implementación de peticiones cliente-servidor	Santiago Palmet Salcedo	100%	27-5-23	31-5-23
Documentación				
Redactar y revisar en su totalidad la documentación	Santiago Palmet Salcedo	100%	1-6-23	1-6-23

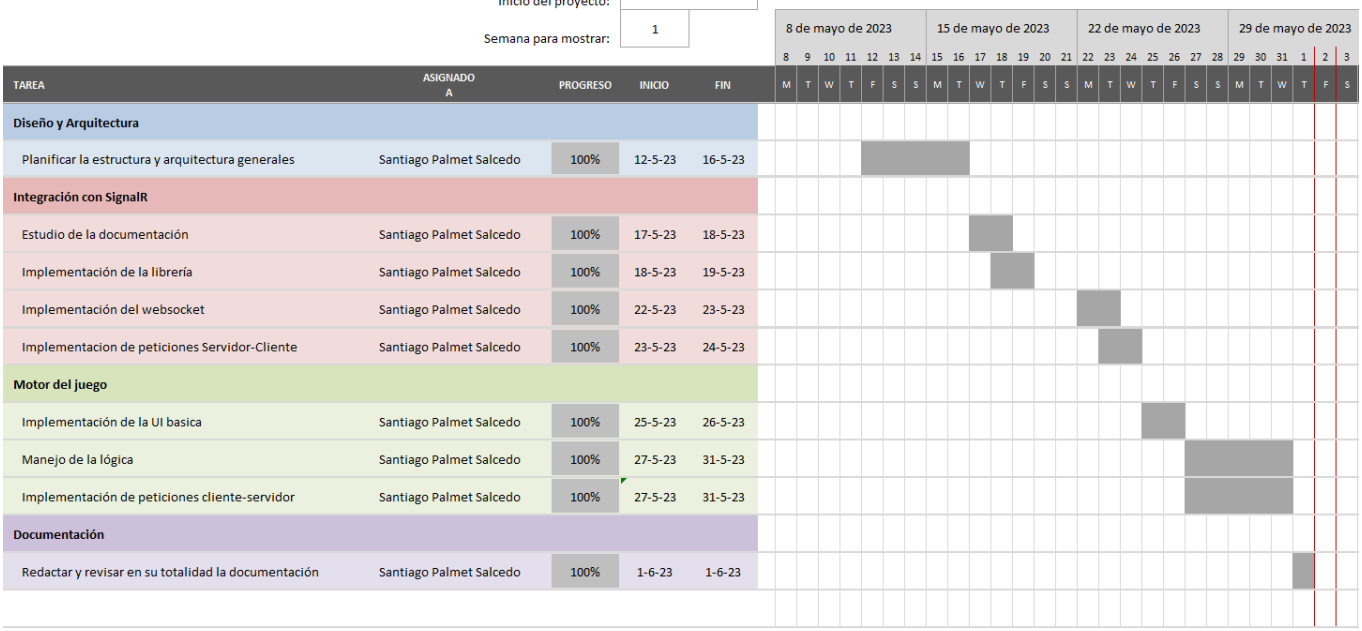
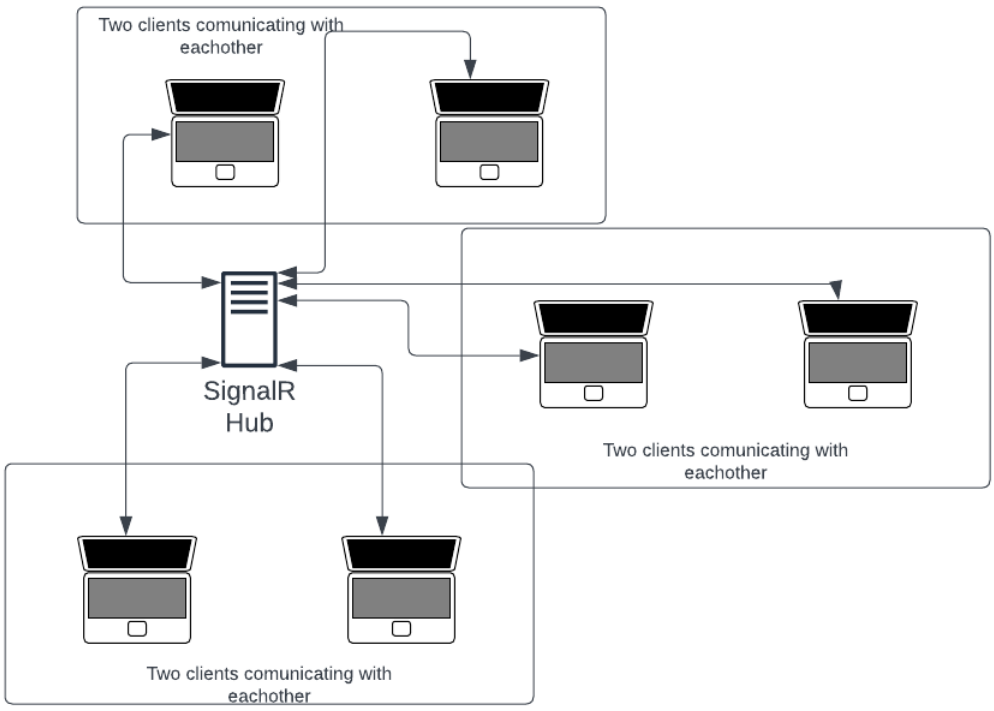


Diagrama de la implementación de la solución

Para lograr el objetivo planteado queremos dividir a los clientes en grupos de 2 para así poder ejecutar partidas de manera independiente sin hacer esperar a los demás a que acabe la que se está jugando para ello la comunicación se vería de esta manera:



Esto se logra haciendo que el servidor genere nuevas partidas cada vez que hay 2 jugadores en espera, para que así la cantidad de gente que pueda usar el servicio dependa de la capacidad del servidor en el que esté corriendo el Hub.

Clases principales

La clase Protagonista en este caso es la de **GameHub.cs**, la cual contiene los métodos que manejan las comunicaciones entre clientes, y como se introdujo anteriormente, lo que se quiere lograr es que sea lo más parecido posible a como sería la comunicación jugando el juego de mesa real para ello tenemos pues las tareas asíncronas básicas necesarias como:

OnConnectedAsync(): La cual se encarga de manejar el tema de las conexiones de los clientes y de separarlos en “lobbys” diferentes.

OnDisconnectedAsync(): La cual se encarga manejar las desconexiones y

avisarle a los clientes cuando una desconexión ocurre.
Luego están las que manejan la comunicación como tal entre clientes de maneras más especificas como **Movements()**, **Attack()**, **RemoveTurn()**, las cuales se encargan de avisarle a los clientes donde se hizo el ataque y avisarle que su turno acabó.
Notifications(): Se encarga de avisarle al cliente lo que esté pasando en la partida como si su ataque acertó, si terminó su turno etc.
Y por último **Endgame()** la cual se ejecuta cuando la condición de victoria se cumple para alguno de los dos jugadores y envía a ambos clientes los resultados de la partida.

Luego el cliente es manejado por un archivo Javascript el cual se encarga de manejar la lógica del juego como calcular si el ataque recibido acertó o no y enviarle ese mensaje al otro jugador a través del hub.

Conclusión

Al terminar esta implementación podemos decir que signalR es una librería muy poderosa que nos facilita la implementación de websockets para la comunicación en tiempo real, está muy bien documentada lo cual facilita bastante el aprendizaje y entendimiento de cómo funciona realmente este tipo de comunicación, pero esta no es perfecta para todos los casos ya que dependiendo del proyecto puede ser más conveniente usar otras alternativas debido a que si es un proyecto muy pequeño el uso de esta librería puede terminar aumentando la complejidad del proyecto innecesariamente.