# Sprint Review 6

Durante este sprint no hemos conseguido cumplir todos los **objetivos** que habíamos planteado. Esto se debe a varias razones. En lo que a iniciar una nueva partida desde la GUI se refiere, no se ha podido finalizar dado que se ha presentado un error que no se ha sabido resolver antes de la finalización del sprint. Por otra parte, durante este sprint una parte del trabajo se ha centrado en la refactorización del diseño y algunos de estos elementos no se han podido incluir en la rama principal del proyecto antes de la finalización del sprint, dado que es necesario tanto terminar algunas como someter a otras a pruebas antes de su inclusión.

Otra parte del trabajo durante este sprint, ha sido la inclusión de **multijugador en red**. Con estas modificaciones, cada uno de los jugadores que participan en la partida puede tener su propia vista. Para incorporar a nuestro proyecto esta parte, hemos seguido la arquitectura Cliente-Servidor. En nuestro caso, hemos considerado el sistema de la siguiente forma:

El servidor es un proveedor de servicios y atiende a las demandas de los clientes, por tanto en él reside el modelo de nuestro proyecto (la lógica del juego).El cliente es un demandante de servicios, en él reside la vista (la interfaz del juego).La comunicación entre el cliente y el servidor la hemos conseguido gracias al uso de sockets.

Además, se han añadido una serie de clases a la vista que nos permiten poder hacer login con nuestro nick y otra para poder ver los jugadores conectados al juego en tiempo real. (para más información sobre diseño ver documento “Multijugador en red”).

Por otro lado, durante este sprint también se ha **refactorizado el Modelo**, puesto que al incluir la GUI era necesario cambiar el comportamiento del Modelo para que dejara de estar guiado por bucles, y pasará a estar dirigido por los eventos que se producían en la GUI. Como consecuencia, se han eliminado los bucles que pedían comandos por la consola, y los que hacían avanzar el juego asignando turnos a los jugadores uno por uno. Toda la información referida a esta refactorización se encuentra en el documento ‘RefactorModelo’.

La **refactorización de la IA** ha permitido que el comportamiento de la máquina ahora sea más similar al de un jugador humano. Para facilitar este comportamiento, se ha decidido crear una nueva clase Diccionario con una estructura de datos trie (para más información leer el documento “refactorizaciónIA” almacenado en la carpeta del Sprint 6). Como consecuencia, se ha modificado el sistema por el que se diferencian los niveles de dificultad y, para mejorar el diseño original, se han modificado también tanto las estrategias como las clases Integrante, Jugador y Maquina (de nuevo, para más información leer el documento “refactoriaciónIA”). Queda pendiente la incorporación de este nuevo sistema a la rama principal del proyecto.

En cuanto a la **parte visual de la aplicación**, se ha mejorado la claridad del código y se ha dado un gran salto en cuanto a la calidad de la GUI.

Respecto a la ventana principal (el tablero de juego y los diversos paneles que lo componen) se han cambiado tanto iconos, colores y formatos para mejorar la visibilidad de la información y se ha mejorado el reescalado de dichos paneles permitiendo así una mayor adaptabilidad.

Dicho tratamiento se ha extendido también a los menús con los que cuenta nuestra aplicación (Menú principal, Lobby.. etc) donde se ha mejorado notablemente la presentación, se ha ajustado el reescalado y se ha adecuado la paleta de colores respecto a la que usa la ventana principal (el tablero de juego)

En cuanto a **JUnit,**  aunque estamos a unas alturas ya bastante avanzadas del proyecto, las pruebas unitarias nos han sido de gran utilidad. La razón por la que no empezamos a generar pruebas desde un principio ha sido la falta de conocimiento sobre JUnit. Estas pruebas abarcan las clases Tablero, ComandoPasarTurno, ComandoColocarFicha, ComandoInvertirSentido y GeneradorMazo.

La idea de cara al próximo sprint es seguir generando pruebas unitarias hasta tener probadas por lo menos las partes más críticas de la aplicación y la funcionalidad básica del juego. También intentaremos crear nuevos casos de uso para las pruebas que sean superadas sin errores con el objetivo de encontrar nuevos fallos en el juego.

En cuanto a la **creación de jugadores y máquinas**, se pueden añadir y eliminar a los mismos hasta llegar a un máximo de cuatro componentes por partida, obligando a la partida a que al menos haya un jugador, por otra parte, al crear una máquina podremos seleccionar su dificultad mediante un diálogo el cual nos mostrará las tres dificultades disponibles.También se ha añadido la posibilidad de ver los comandos que se pueden utilizar así como las instrucciones del juego mediante un JMenuBar al que se puede acceder en todo momento.