**-Entidades**

**clsSala** → int codSala, String nombreSala,clsJugador JugadorArriba, clsJugador JugadorAbajo, Long Tiempo, int cantidadFichasArriba = 12, int cantidadFichasAbajo = 12

**clsJugador** → String nombre, String password, List<Partida> Partidas

**-DAL**

**Conexión →**

**UriBase →** GetUriBase() : Uri

**Listados →**

GetListadoCompletoUsuarios() : List<clsJugador> jugadores.

GetListadoPartidasDeJugador(String nombreJugador) : List<clsPartida> partidas

GetSalasConEspacio() : List<clsPartida> salas → Método que sirve para recoger todas las partidas que, en la base de datos, su atributo jugadorArriba o jugadorAbajo, sean null.

**Gestion →**

**GestionJugadores:**

InsertarJugador(clsJugador jugador) : clsJugador jugador

ModificarJugador(clsJugador jugador) : HttpResponseMessage

EliminarJugador(String nombreJugador) : HttpResponseMessage

**GestionPartidas:**

InsertarPartida (clsPartida partida) : clsPartida partida

ModificarPartida(clsPartida partida) : HttpResponseMessage

**-BL**

**Listados →**

GetListadoCompletoUsuariosBL() : List<clsJugador> jugadores.

GetListadoPartidasDeJugadorBL(String nombreJugador) : List<clsPartida> partidas

GetSalasConEspacioBL() : List<clsPartida> salas → Método que sirve para recoger todas las partidas que, en la base de datos, su atributo jugadorArriba o jugadorAbajo, sean null.

**Gestion →**

**GestionJugadores:**

InsertarJugadorBL(clsJugador jugador) : clsJugador jugador

ModificarJugadorBL(clsJugador jugador) : HttpResponseMessage

EliminarJugadorBL(String nombreJugador) : HttpResponseMessage

**GestionPartidas:**

InsertarPartidaBL(clsPartida partida) : clsPartida partida

ModificarPartidaBL(clsPartida partida) : HttpResponseMessage

EliminarPartidaBL(int idPartida) : HttpResponseMessage

**-UI**

**Resources:**

**Images →** fichaNegra, fichaBlanca, fichaDamaBlanca, fichaDamaNegra

**Models:**

**clsFicha** → int posX, int posY, String colorSource, bool isDama

**clsJugadorConPartidasGanadas : clsJugador →** int partidasGanadas

**clsPartidaConFichas : Partida** → List<Ficha> FichasArriba size = 12, List<Ficha> FichasAbajo size = 12.

**ViewModels:**

**BaseVM** → ViewModel de utilidad que implementa InotifyPropertyChanged y que añade los atributos isBusy (Hace referencia a que la página está ocupada cuando se intenta utilizar) y Title(añade el título a la página).

**LoginVM : BaseVM** → ViewModel que contendrá al jugador a instanciar a partir de los datos introducidos en la vista LoginPage. Esta página se encargará de instanciar al jugador, generar un diccionario y enviarlo a la Flyout del Shell.

**MainVM : BaseVM →** ViewModel que contendrá un listado de salas, una salaSeleccionada, un comando para crear una nueva sala, que tendrá un DisplayPromptAsync() en el que se pedirá al usuario un nombre de sala, un comando para buscar una sala específica y un comando para entrar en la sala seleccionada cuando se pulse el botón que acciona el comando dentro de la vista..

Si hay una sala elegida se mandará dicha sala en forma de IDictionary a la página JuegoPage.

**EstadisticasVM : BaseVM →** ViewModel que contiene un objeto clsJugador y varios métodos para sacar de su listado de partidas las estadísticas necesarias como son: tiempo jugado, partidas ganadas, partidas perdidas, partidas jugadas…

**RankingVM : BaseVM →** ViewModel que contendrá una lista de objetos clsJugadorConPartidasGanadas. Cuando JugadorSeleccionado != null podrá accederse a la página EstadisticasPage de ese objeto clsJugador

**GameVM : BaseVM →** ViewModel que contendrá dos objetos clsJugador, un objeto clsPartidaConFichas, donde el nombre de ambos jugadores debe ser igual a los atributos de la clase clsPartida jugadorArriba y jugadorAbajo.

Se necesitarán comandos que aporten el movimiento de las fichas, añadan uno a la puntuación cada vez que una ficha del lado contrario es eliminada, compruebe el resultado y sume tiempo al cronómetro. Cuando el contador de fichas de uno de los dos jugadores llegue a 0, se mostrará un DisplayAlert con los datos de la partida junto con el nombre del ganador. No importa donde se pulse, este nos enviará de nuevo a MainPage con los datos del usuario que en principio realizó el Log in.

**Views:**

**LoginPage** → Vista que pide al usuario introducir nombre y contraseña y, según si pulsa el botón “Sign up” (Nuevo usuario), abrirá un Alert que mostrará, si el nombre de usuario ya estaba registrado y se intenta registrar con el mismo nombre, un mensaje de error, volviendo de nuevo a reiniciar la página. En caso contrario se mostrará un mensaje afirmativo, mostrando que se creó un nuevo usuario.

Si se pulsa el botón “Sign in” (Usuario ya registrado) y el usuario y la contraseña no son correctos, también mostrará un mensaje de error, en caso contrario, se iniciará la vista Main.

**GamePage** → Vista que contendrá un Grid de (8 columnas x 8 filas) donde se posicionarán las imágenes de todas las fichas con su ordenación correspondiente. A cada lado del tablero, encontraremos el nombre de uno de los dos jugadores, junto con la cantidad de fichas restantes en el tablero. Cuando sea el turno de uno de los dos jugadores, este tendrá que elegir una de sus fichas, lo que accionará un DisplayActionSheet() con dos elecciones (izquierda alante, derecha alante) de las cuales, se podrá realizar solo una y cuando no haya ninguna ficha propia bloqueando el movimiento. Si la que bloquea el movimiento es una de las fichas contrarias, y el hueco que queda tras esta, está vacío, la ficha pasará al segundo hueco y la ficha contraría desaparecerá del tablero, junto con la reducción de la cantidad de fichas correspondientes al color.

**Flyout:**

**MainPage** → Vista de elección de salas. Se mostrará un listado de salas disponibles, junto con la cantidad de usuarios dentro de ellas. Si esta sala no está llena, el usuario podrá seleccionarla y pasar a la vista Juego, si no, se mostrará un Alert que informará al usuario sobre el estado completo de la sala.

Bajo esta lista, habrá 2 botones. El primero servirá para intentar conectarse a la sala seleccionada, el segundo permite crear una nueva sala y te introduce en ella a la espera de la llegada de otro jugador.

**EstadisticasPage** → Vista que mostrará las estadísticas completas del jugador logueado. Mostrará una lista con: el menor tiempo en el que ha conseguido ganar una partida, partidas ganadas, partidas jugadas, partidas perdidas y tiempo total de juego..

**RankingPage** → Vista que muestra un ListView con los nombres de todos los usuarios y su cantidad de partidas ganadas en orden descendente por número de partidas ganadas bajo un SearchBar que servirá para buscar a un usuario en concreto.