# Informe Guía grupo 2

* Para comprender mejor el funcionamiento del código, recomendamos ejecutar el mismo en Progam.

Para analizar y confrontar la segundo entrega, decidimos observar el código en 3 subsecciónes:

1 – Creación y registro de usuarios:

2 - Procesos previos al juego

3 – Juego

## 1 – Creación y registro de usuarios:

Existen las clases User y User Register. Se crean los usuarios y se registran en User Register, que contiene una lista de usuarios creados.

En este caso User Register funciona como information holder.

User tiene un atributo muy importante, llamado estado (status). Con el se decide que comandos están o no disponible en los Handles.

## 2 - Procesos previos al juego

Primero, Cuando un usuario decide jugar una partida, tiene dos opciones de juego: normal y con predicción. Cuando elige una búsqueda, existen dos opciones. Si hay alguien en la clase “lobby” (sala de espera), y también quiere jugar al mismo modo de juego, automáticamente se crea un juego con ambos y se elimina al otro jugador del lobby. Si no hay nadie en la sala de espera, este pasa a esperar en la misma. Una vez hecho el match, se crea un juego y se guarda en GamesRegister, el cual también funciona como information holder. El Game en si, se encarga de contener a los usuarios pero no es el encargado de ofrecer los servicios. Quién ofrece servicios es Logic, que se encuentra en la sección de juego.

## 3 – Juego

User, posee un atributo llamado Player, el cual es una clase que conoce los datos y tiene las operaciones necesarias para poder jugar. Cada vez que se crea un juego se vuelve a crear un Player nuevo en user, eliminando el anterior, su ciclo de vida empieza cuando iniciar el juego, y termina cuando se inicia un juego nuevo.

Player contiene el tablero (Board), y a su vez, el tablero contiene a los barcos (Ships). Después existe la clase Logic, la cual contiene varios métodos, siendo quien ofrece distintos servicios.

### Interfaces.

Poseemos dos interfaces, una para recibir datos por parte del usuario (input), y otra para mostrarle información al mismo. Ambas interfaces son implementadas por subclases, en está ocasión son implementadas por clases que funcionan y trabajan a Trávez de la consola. La creación de las interfaces nos permite, en un futuro, crear clases con nuevos medios de obtención y entrega de información con el usuario.

## Handles

* Luego está la sección de Handles. El mismo funciona como una estructura, en el cual se entrega un mensaje, y dependiendo de esté, un handler determinado lo maneja o no. El código base fue entregado por los profesores, y nosotros realizamos distintas extensiones del mismo, ya que esta escrito con la utilización de herencia e interfaz, permitiendo su fácil extensión, sin necesidad de modificar el código base.