Alumnos—
165647 - Iván Alejandro Ochoa Vega
247284 - Cuauhtémoc Eliseo Vásquez Salcido
117262 - Paulina Rodríguez Rayos
241400 - Paul Alejandro Vázquez Cervantes

Entrega Final —
Diseño de la Solución

Fecha—
20 de mayo de 2025

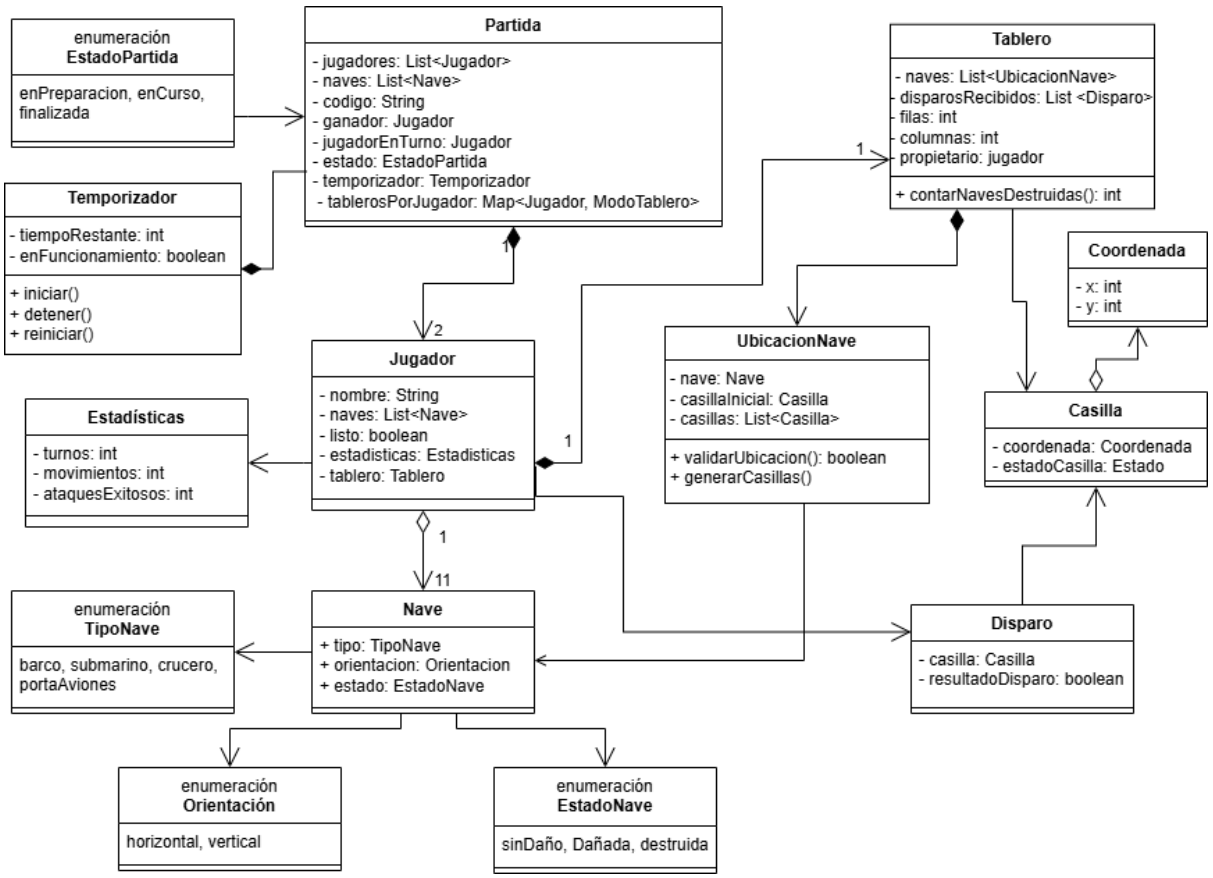
Materia—
Arquitectura de Software

Profesor—
José Gamaliel Rivera Ibarra

Batalla Naval

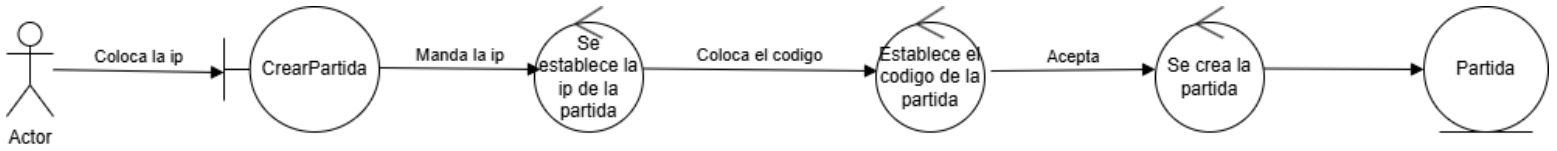
Avance 2. Diseño de la Solución

- **Modelo del dominio (diagrama de clases)**

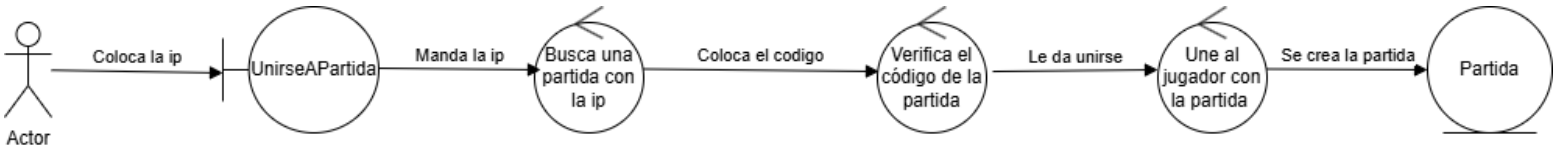


- **Un diagrama de robustez o secuencia del flujo principal de cada caso de uso.**

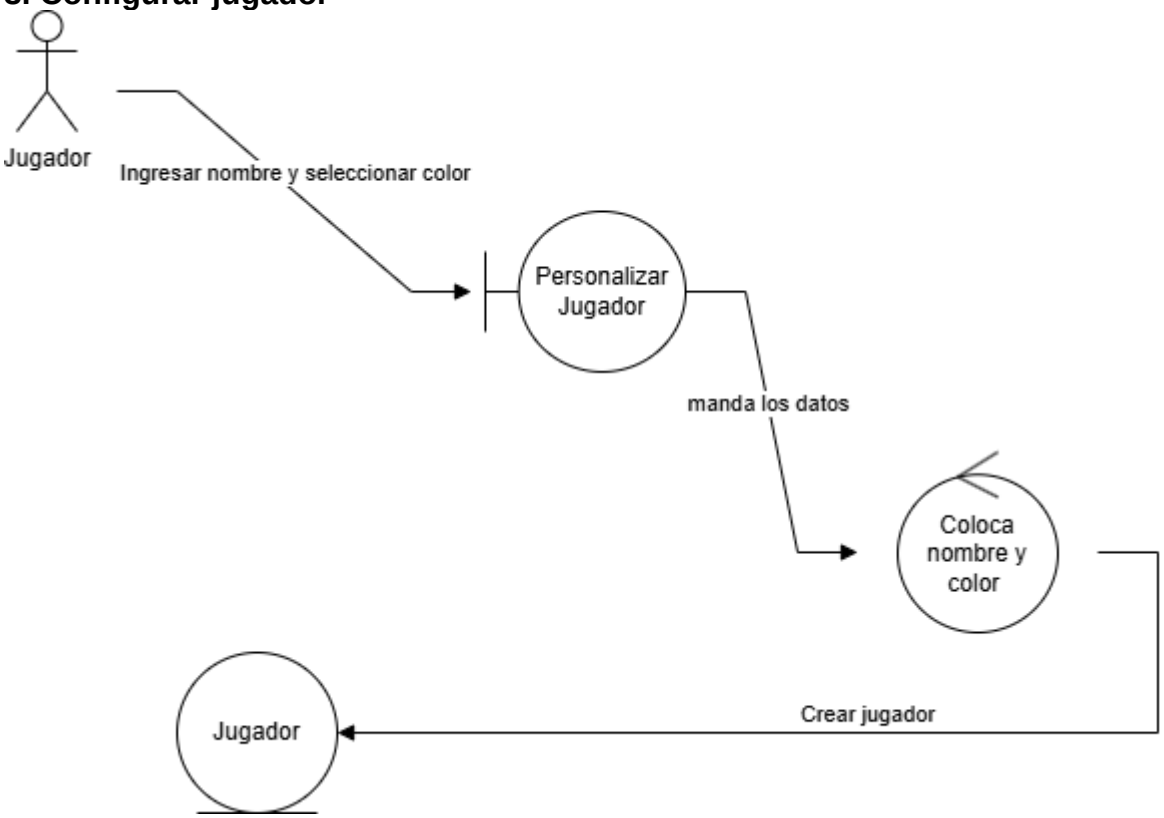
1. Crear partida



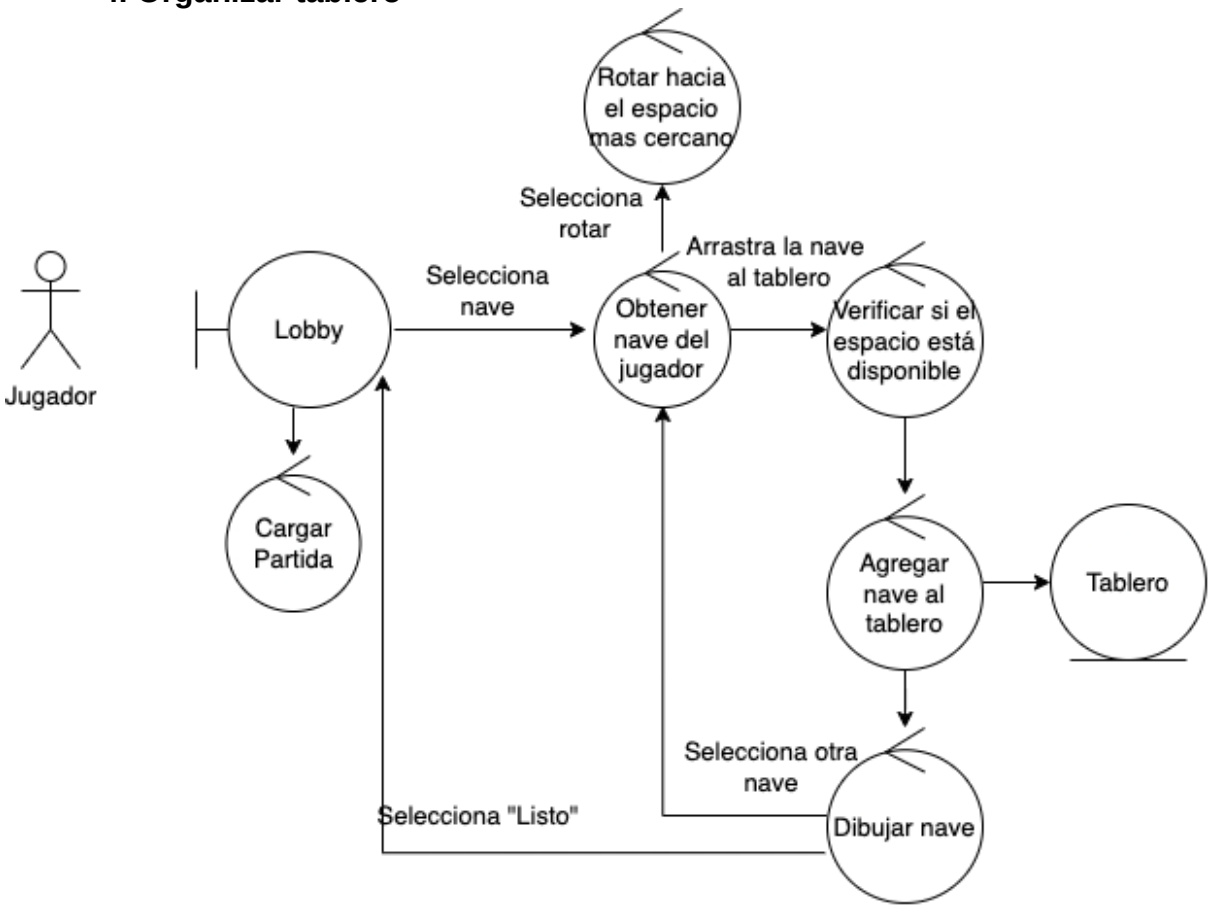
2. Unirse a partida



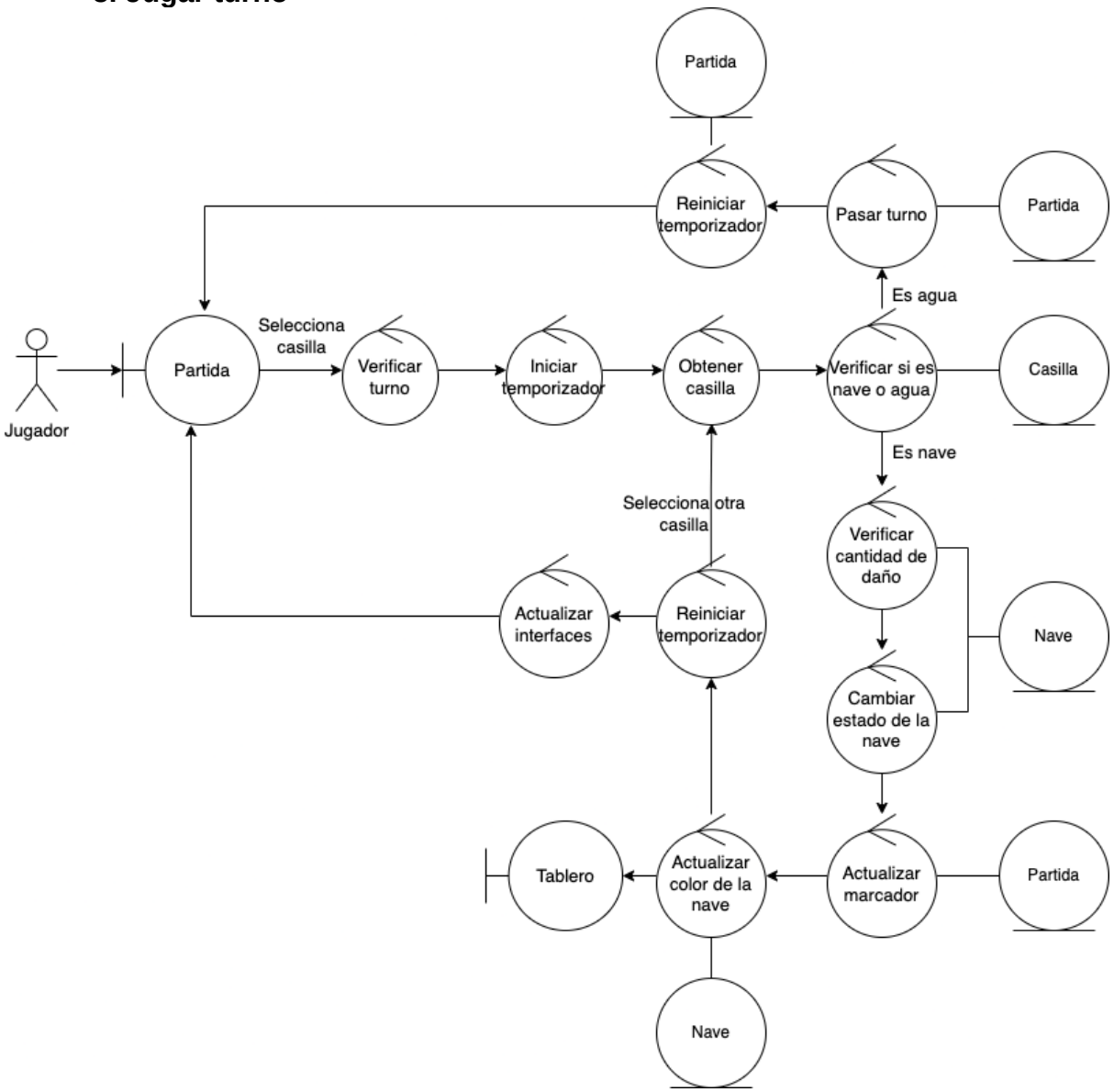
3. Configurar jugador



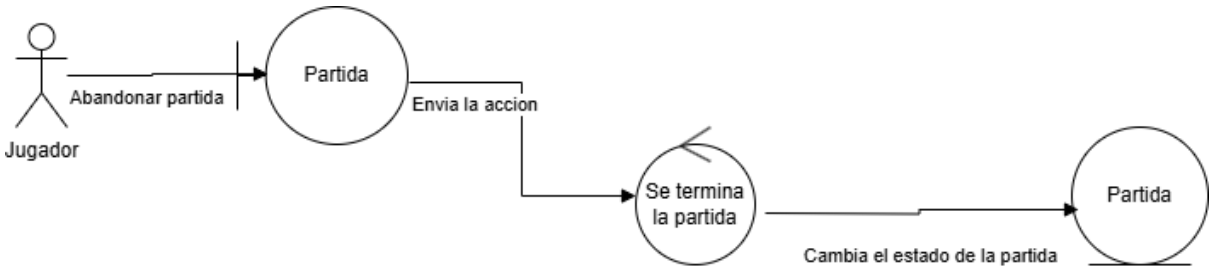
4. Organizar tablero



5. Jugar turno

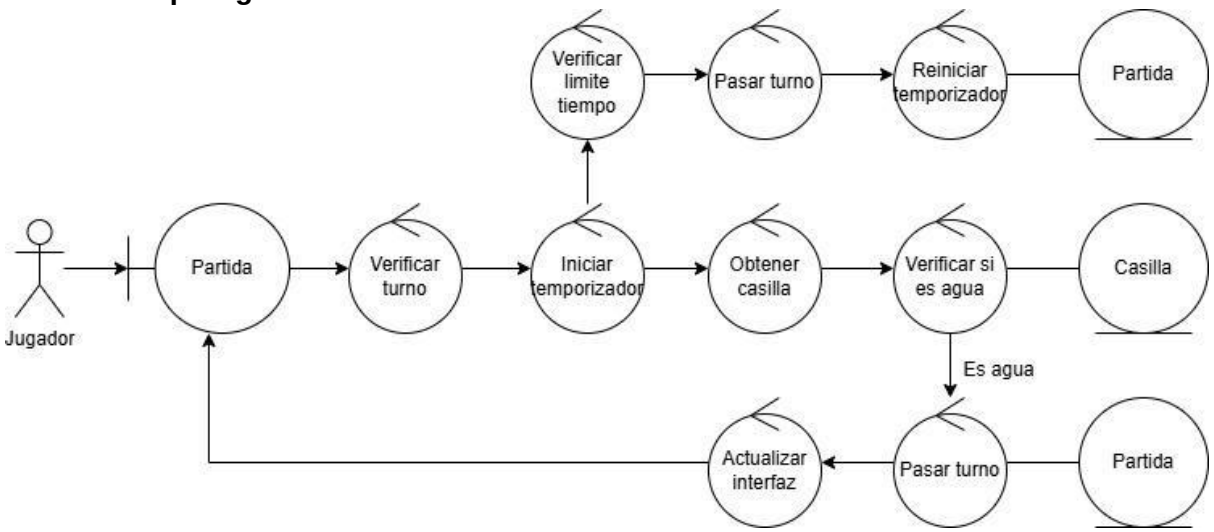


6. Abandonar partida



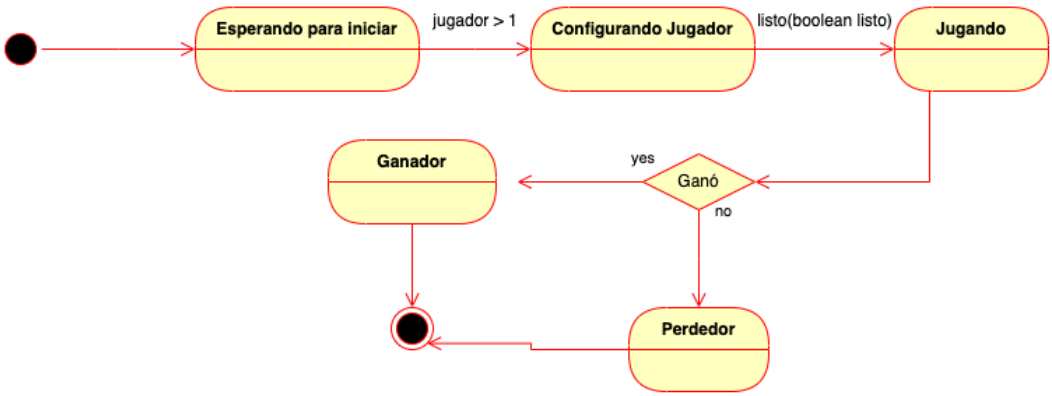
- **Un diagrama de robustez de los flujos alternativos del caso de uso más grande (jugar partida).**

Tiempo Agotado sin Acción:

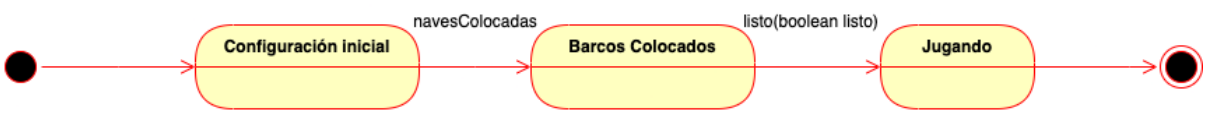


- **Un diagrama de estados para representar los estados del juego.**

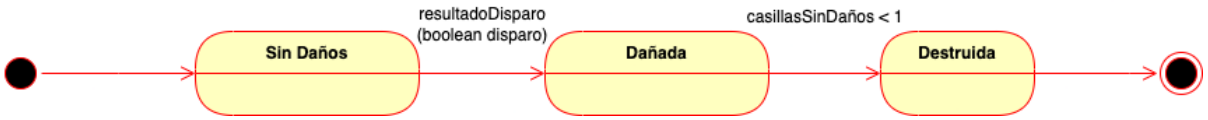
Estados Jugador



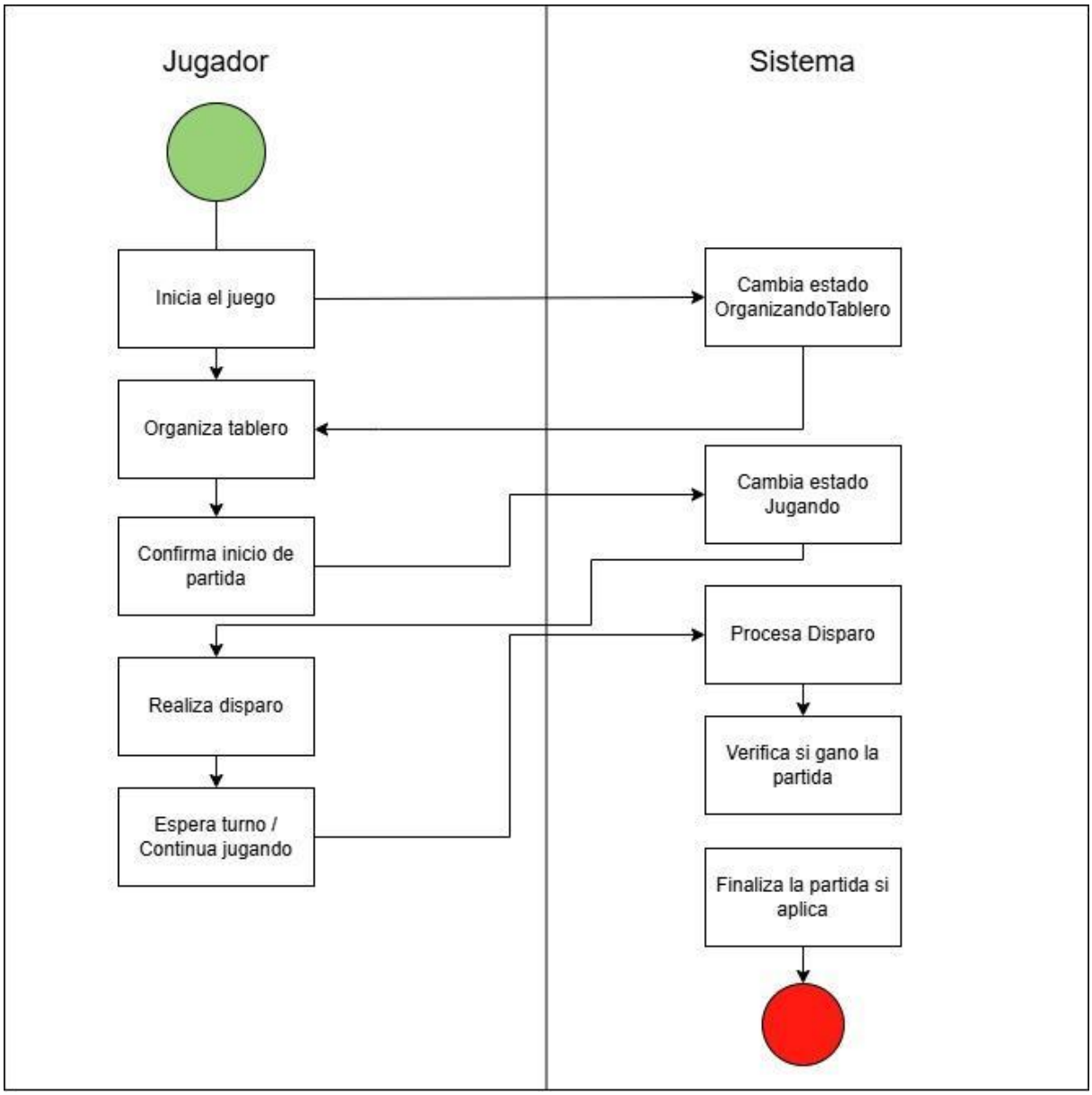
Estados Tablero



Estados Barco

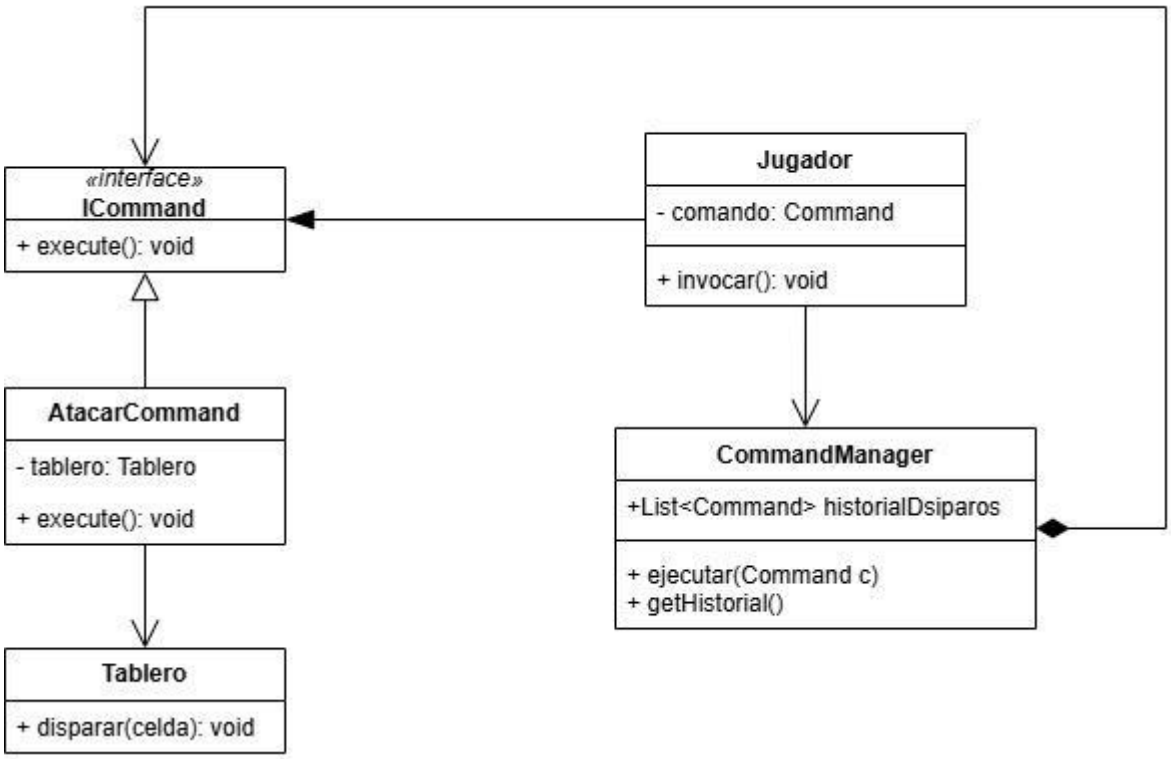


- **Diagrama de actividades**

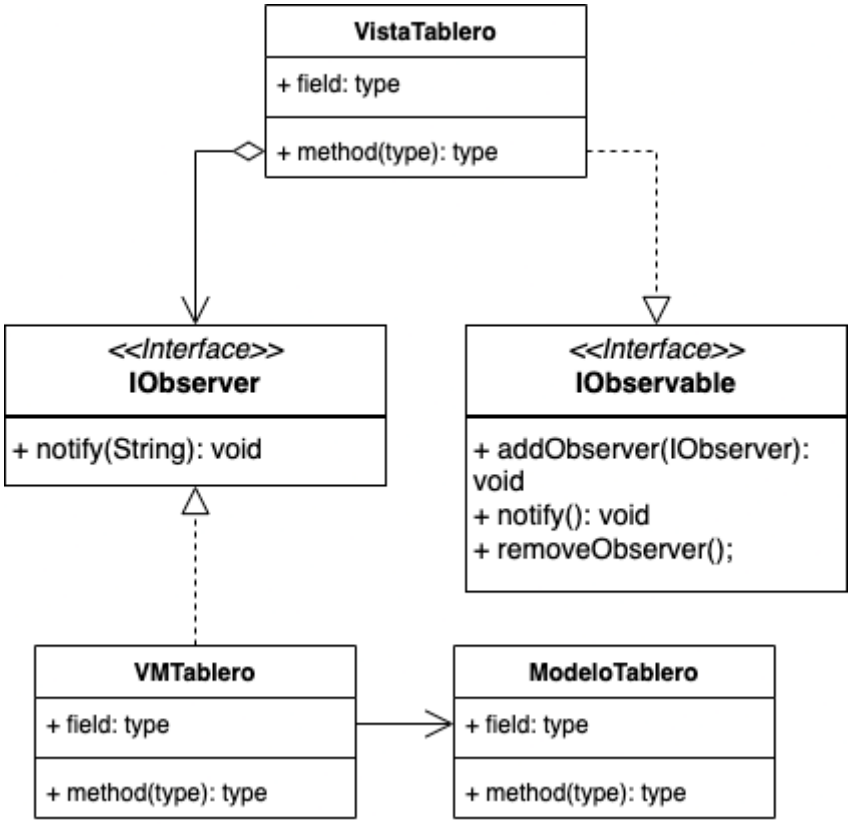


- **Un diagrama de clases y/o secuencia que represente cada uno de los patrones de diseño a utilizar en su solución.**

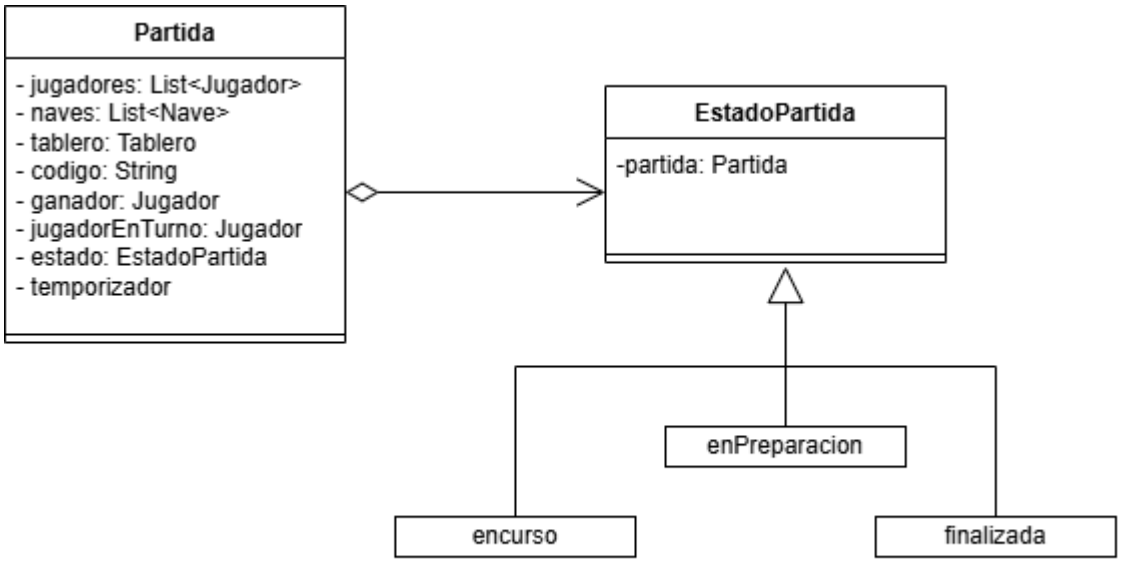
1. Comando (Command) – Para manejar los disparos

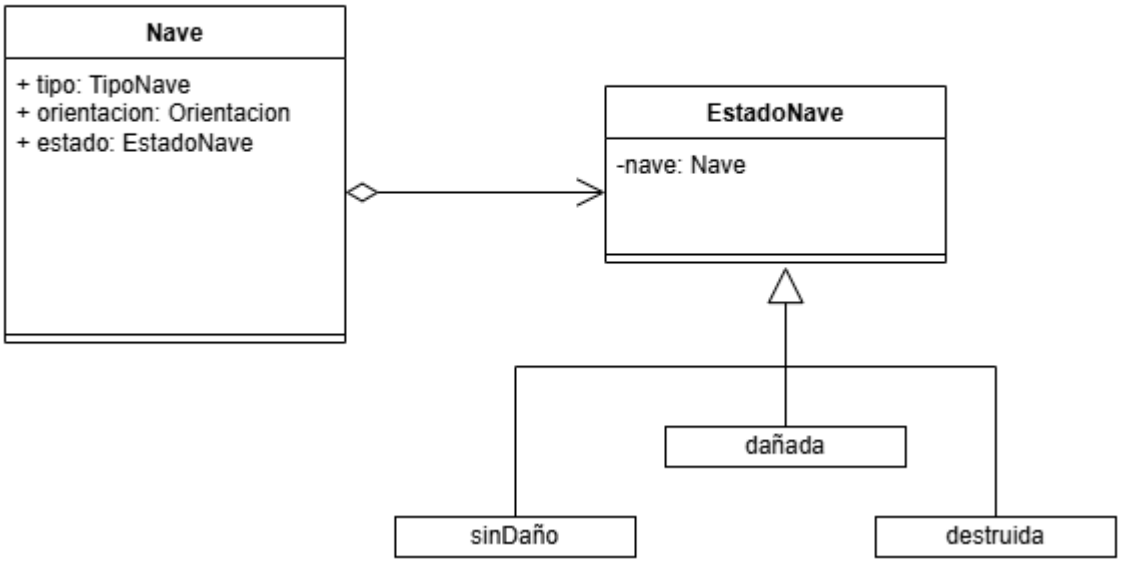


2. Observador (Observer) – Para actualizar la UI en tiempo real

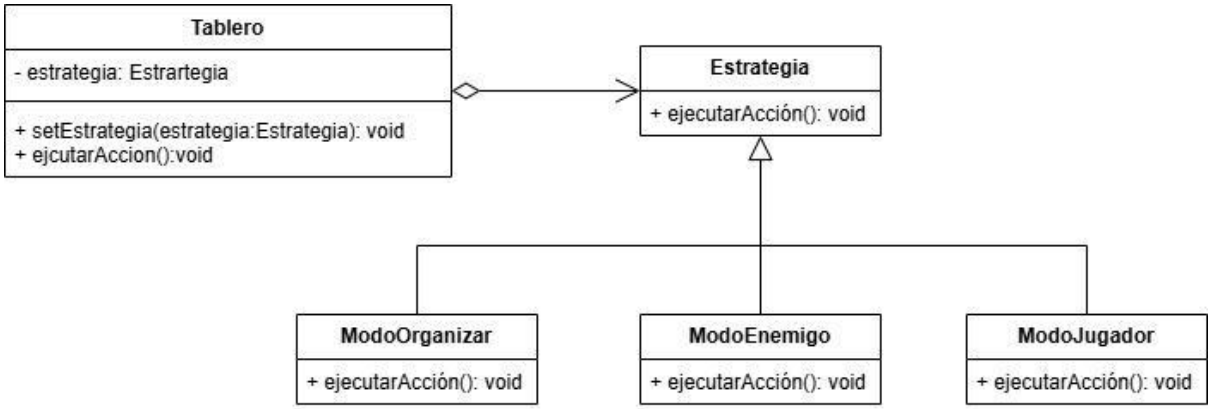


3. Estado (State) – Para manejar los diferentes estados del juego y las naves.



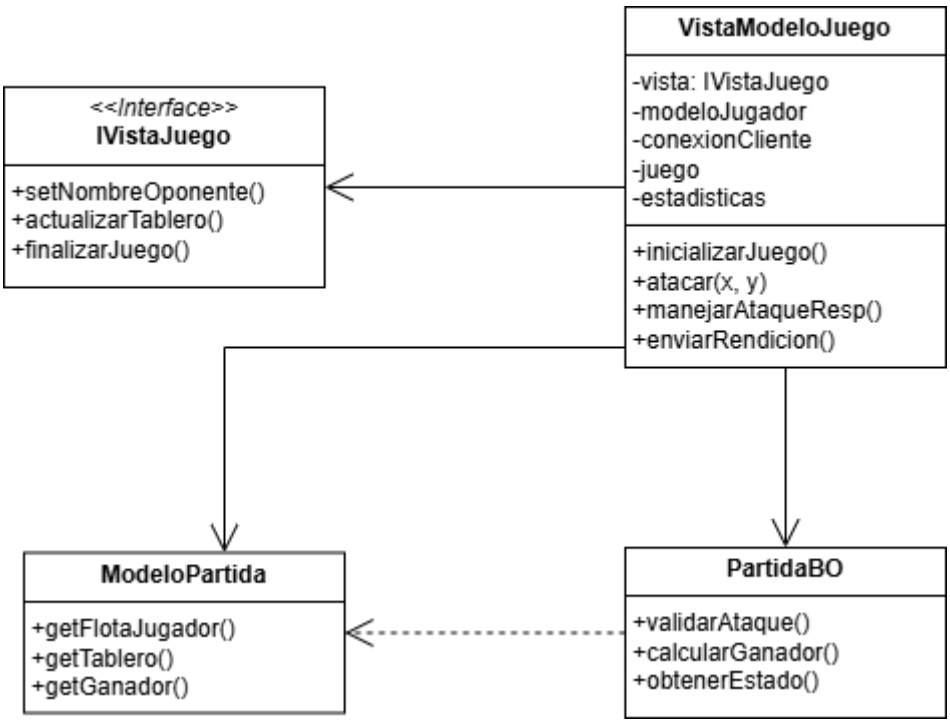


4. Strategy – Para manejar los diferentes modos de interacción con el tablero.



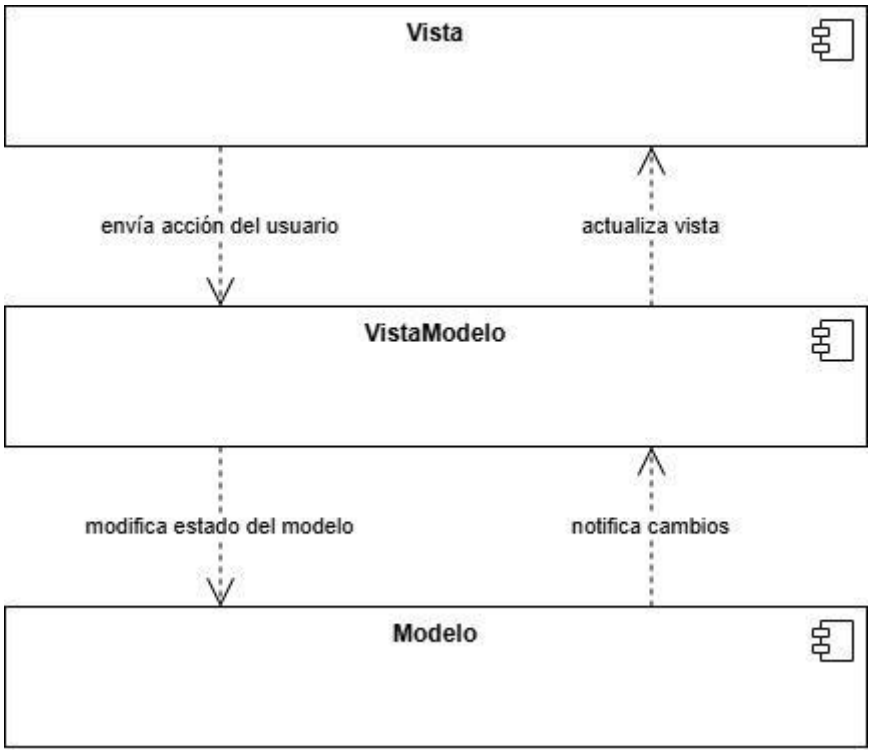
5. Business Object (BO) – Para encapsular la lógica de negocio del juego.

El patrón BO se usa para separar la lógica de negocio del acceso a los datos y de la VistaModelo. PartidaBO coordina las operaciones del juego como validar movimientos, gestionar turnos y aplicar reglas, utilizando modelos como ModeloPartida. De este modo, la VistaModelo se mantiene desacoplada de la lógica del juego, cumpliendo el principio de separación de responsabilidades de MVVM.

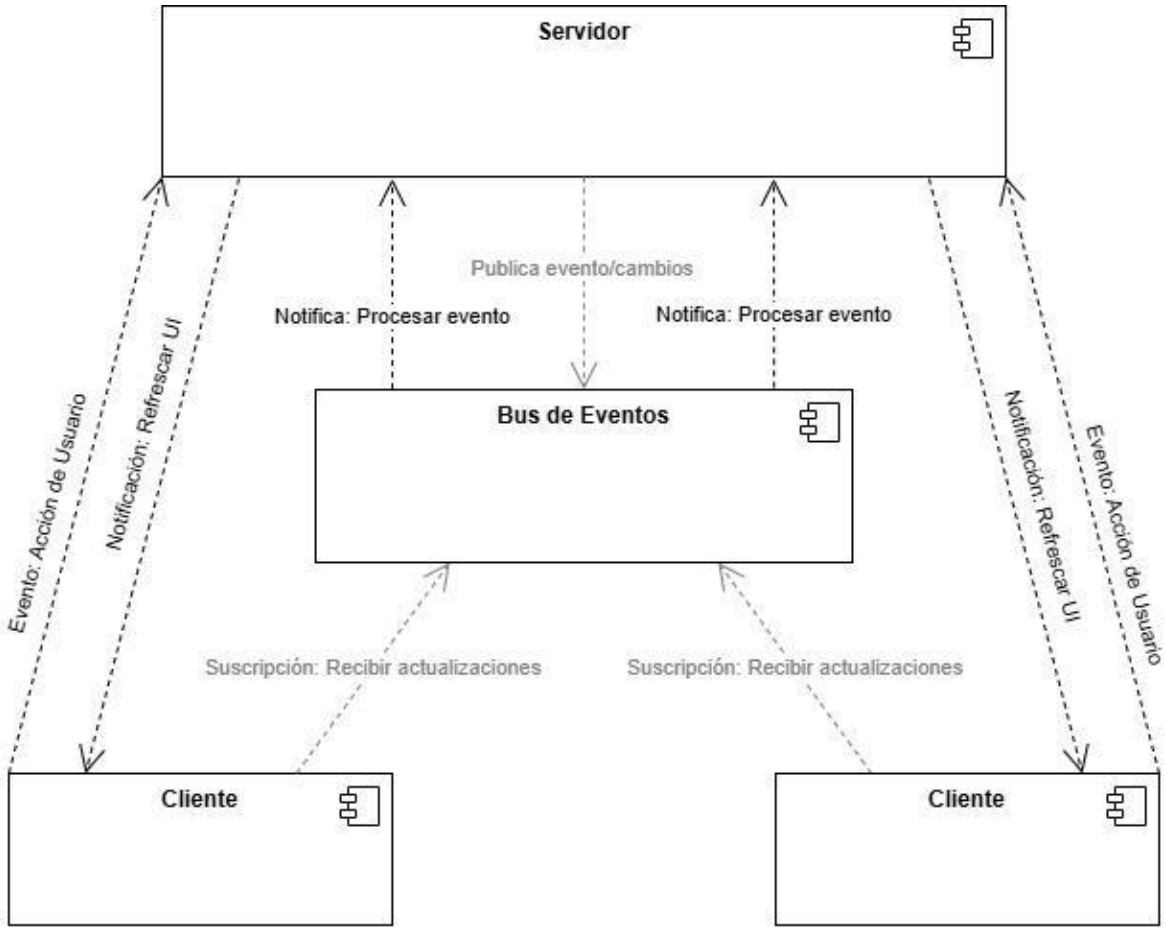


Patrones y Estilos Arquitectónicos

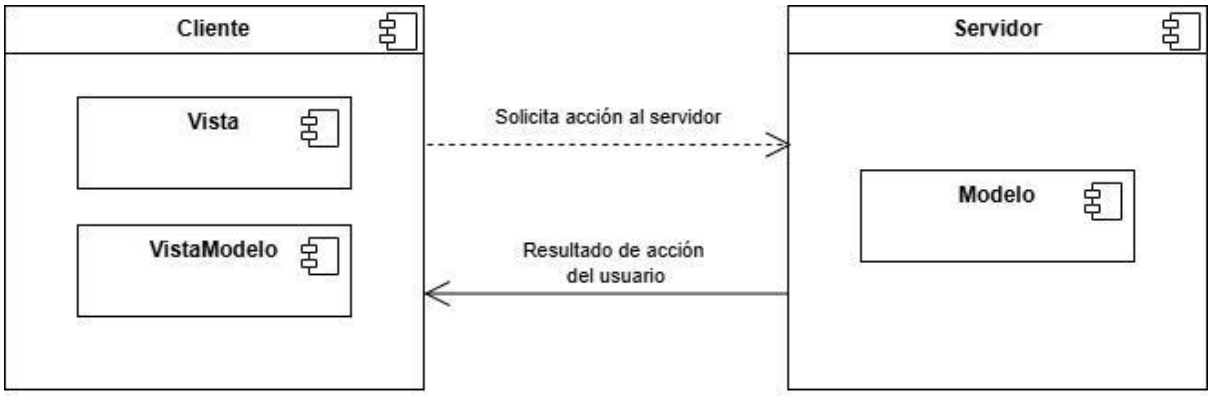
1. MVVM (Modelo-Vista-VistaModelo)



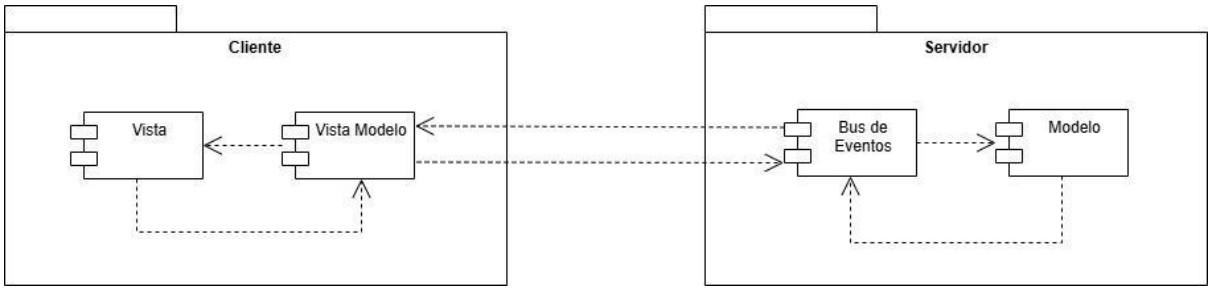
2. Evento-Reacción – Para la interacción del juego



3. Cliente-Servidor – Para el juego en línea



- **Diagrama de Paquetes**



- **Diagrama de Despliegue**