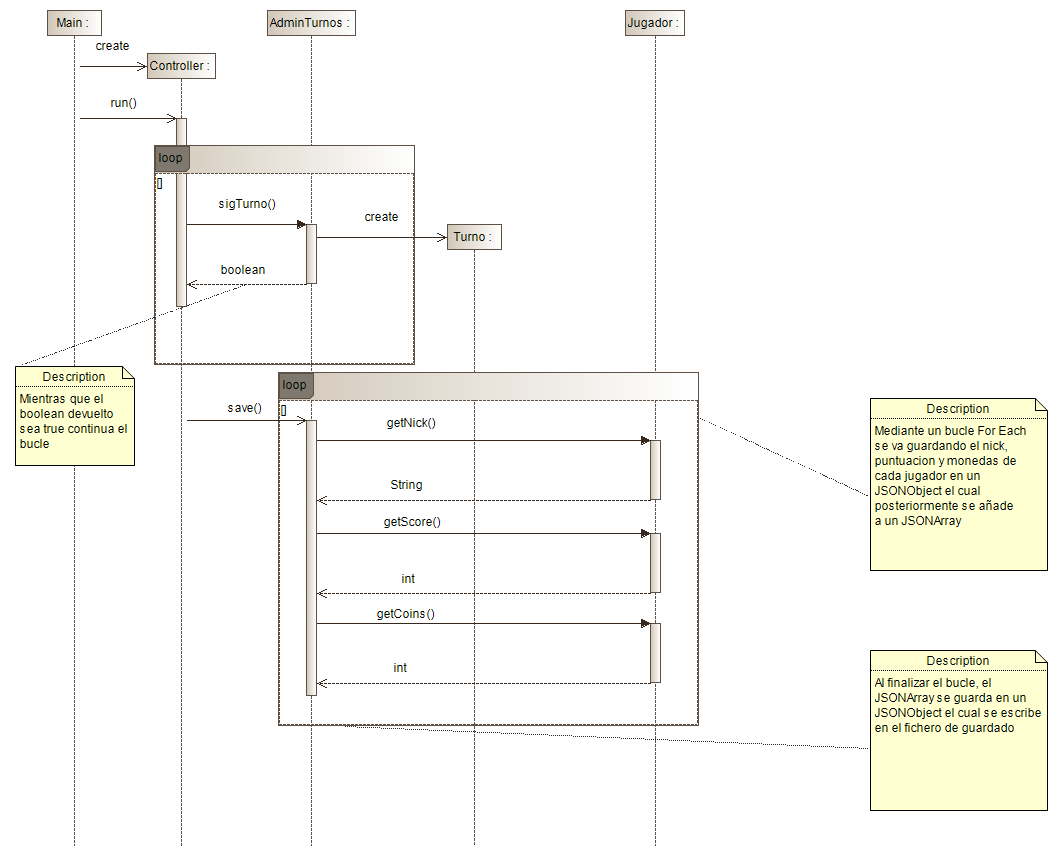
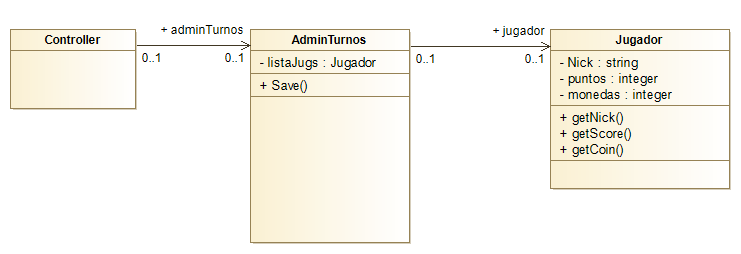
**SPRINT 3 DIAGRAMAS**

1. **Poner la primera ficha de la partida en el centro**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
2. **Guardar partida a medias**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**



* 1. **Explicación**

Una vez que se sale del bucle de partida del controller se llama al método save de AdminTurnos, una vez dentro de este método se le da la posibilidad al usuario de elegir el nombre del fichero de guardado y se procede a guardar la información de los jugadores, para ello, se recorre la lista de jugadores extrayendo nick, puntuación y monedas de cada jugador y guardándolo primero en un JSONObject y posteriormente en un JSONArray, una vez que se sale del bucle el JSONArray se guarda en otro JSONObject y este último se procede a guardar en el fichero que ha seleccionado el usuario.

1. **Cargar partida a medias**
   1. **Imagen que contiene mapa

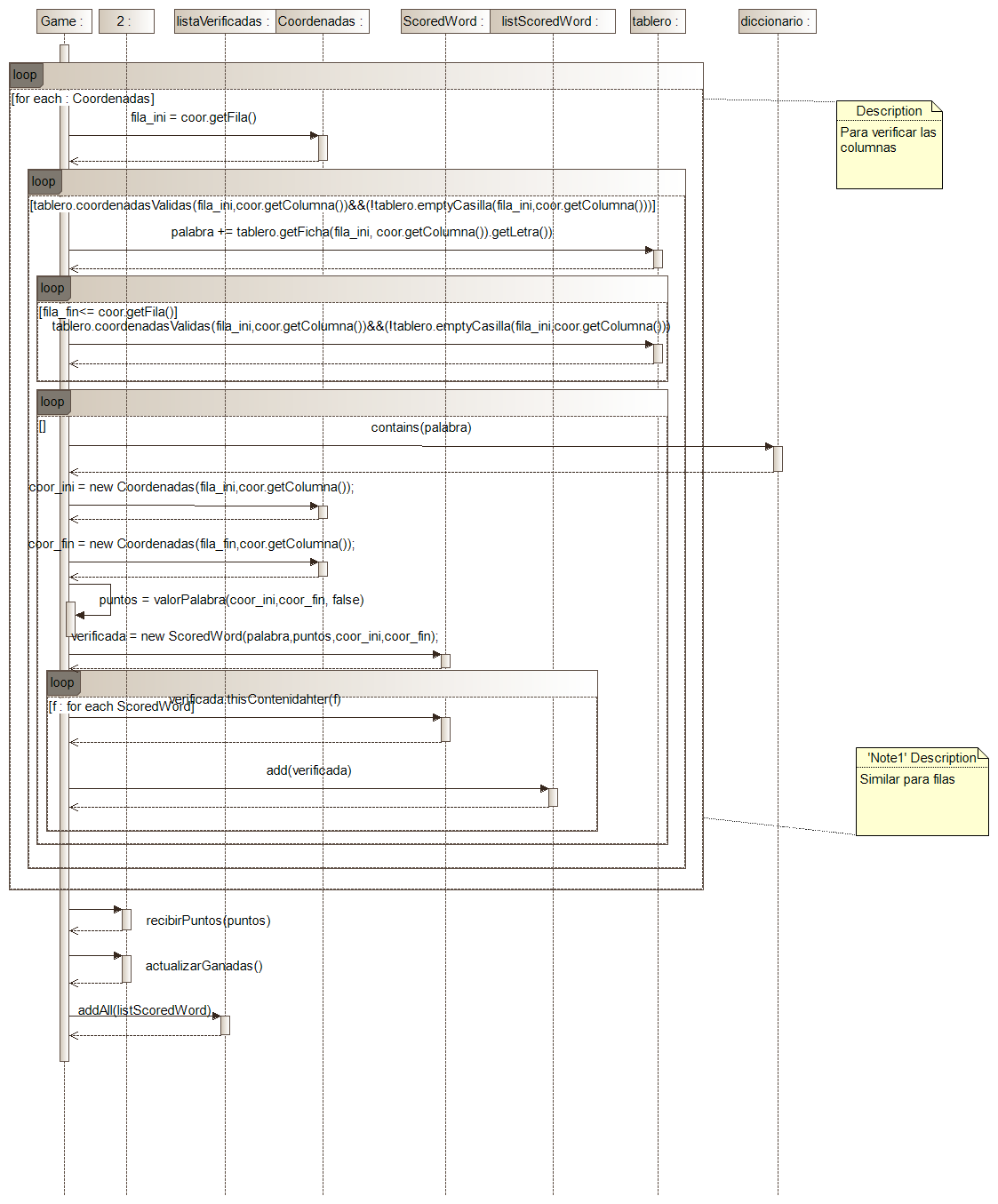
      Descripción generada automáticamenteDiagrama de secuencia**
   2. **Imagen que contiene captura de pantalla

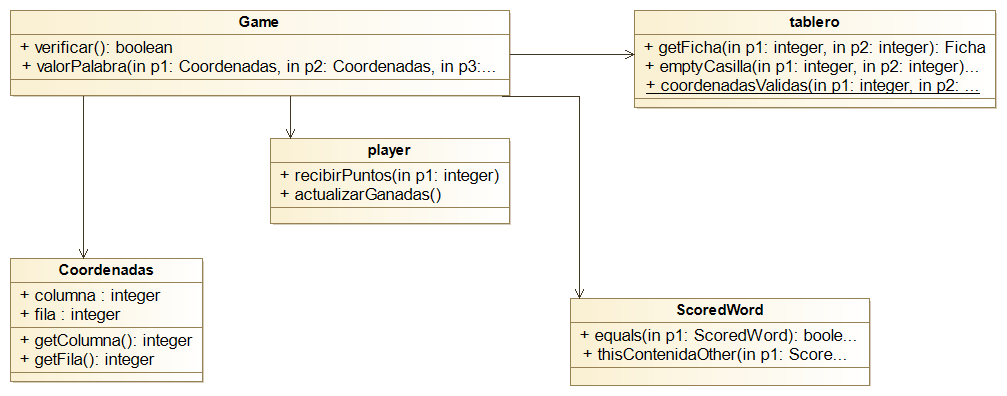
      Descripción generada automáticamenteDiagrama de clases**
   3. **Explicación**

Para cargar una partida hemos usado el patrón Memento. Tenemos una clase Memento que implementa la interfaz IMemento y permite crear Mementos guardando la información en formato JSONObject. El resto de clases implementan la interfaz Originator, que contiene dos métodos: setMemento(Memento) para cargar datos de una partida y createMemento(), que devuelve un Memento con los datos para guardar una partida.

Inicialmente el Controller crea tres mementos principales con la información relativa a las clases Game, AdminTurnos y Turnos. Tras guardar la información (JSONObject) en los mementos se llama a los métodos setMemento() de la clase Game para guardar el memento del Game, y al setMemento() de la clase AdminTurnos para guardar tanto el memento del AdminTurnos como el de Turnos. Esto es debido al formato en que decidimos guardar la información en el JSONObject.

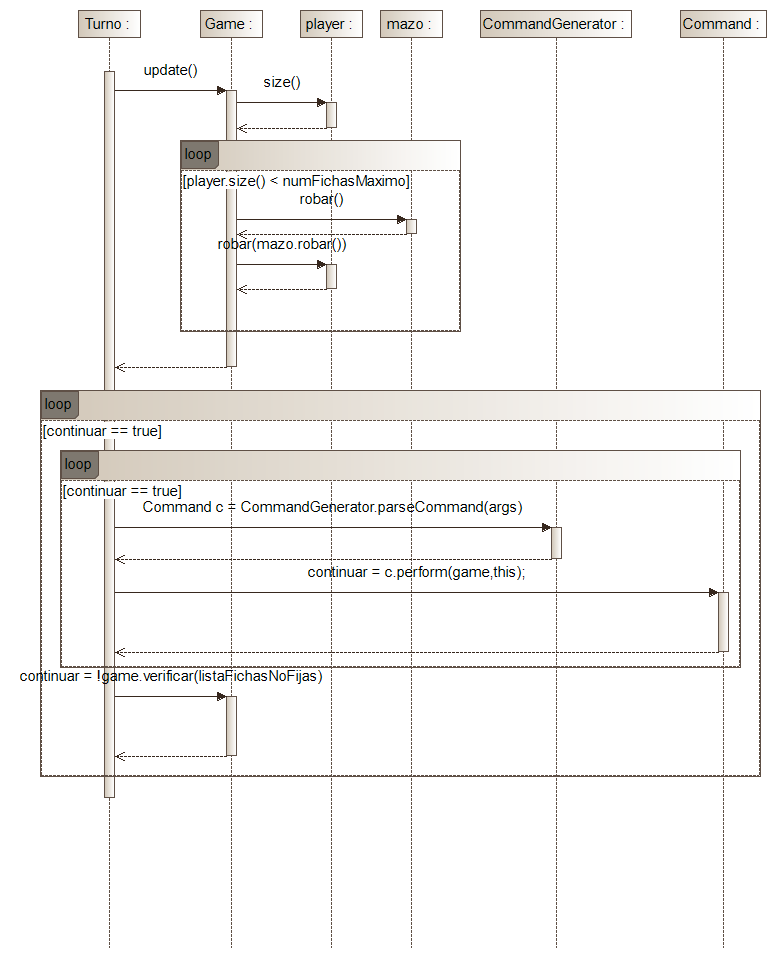
Tanto el diagrama de clases como el diagrama de secuencia pretenden ser generales para visualizar el funcionamiento del patrón memento. Por tanto, faltaría por incluir en los diagramas todas las clases que implementan la interfaz Originator, es decir, todas aquellas que tienen que cargar datos de una partida.

1. **Pedir si quieres New Game/Load Game**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
2. **Verificar palabras correctas automáticamente al finalizar turno**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**

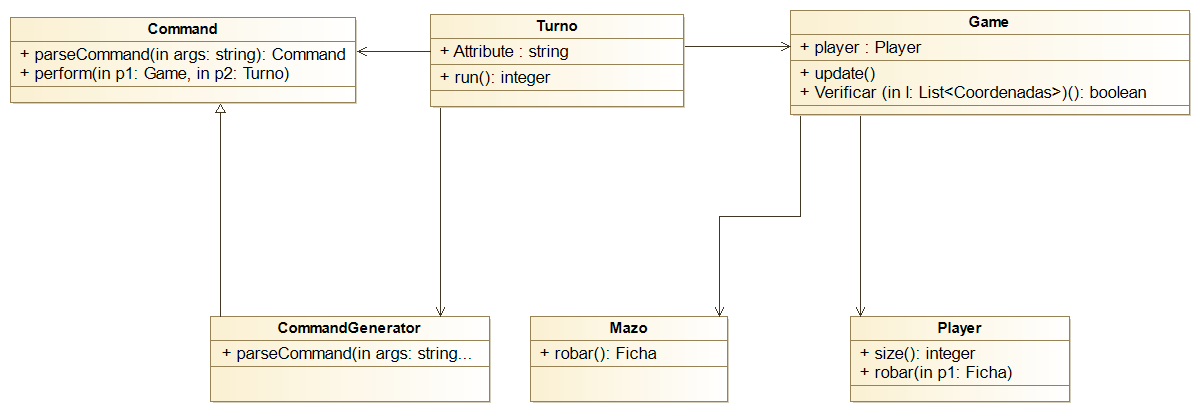


* 1. **Explicación**

1. **No pasar de turno si he colocado alguna ficha que no forme palabra**
   1. **Diagrama de secuencia**

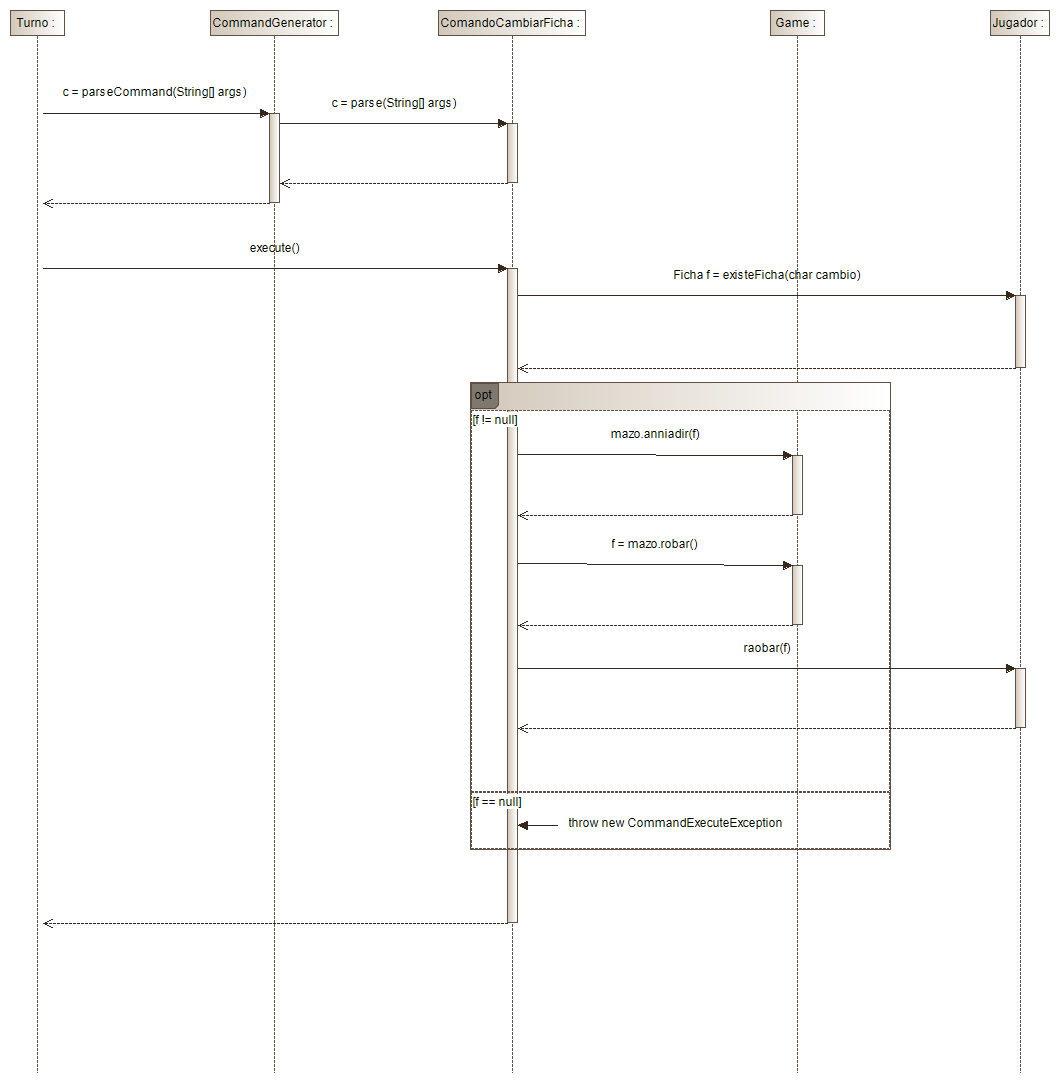
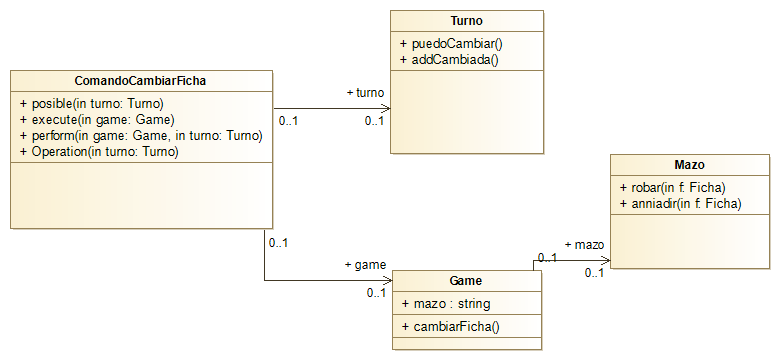


* 1. **Diagrama de clases**



* 1. **Explicación**

Cuando el jugador introduce el comando de pasar de turno, la variable booleana continuar se cambia a true si es posible pasar el turno. Entonces se llama al método verificar de game. Este método devuelve true si la listaFichasNoFija que le has pasado es vacia o si ha verificado alguna palabra (la lista ScoredWord no es vacia). Si devuelve true podemos pasar de turno saliendo del segundo bucle while. Si no continuar se pone a true y no podemos pasar de turno.

1. **Obtener monedas**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
2. **Solo fichas al lado de fichas**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
3. **Si finaliza la partida obtener puntos por las monedas sobrantes**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
4. **Cambiar ficha**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**

Se ha añadido un nuevo comando cambiar ficha. Su condición para poder ejecutarse es que se hayan cambiado menos de 7 fichas durante el turno, y su ejecución consiste en meter la ficha que se quiere cambiar en el mazo, sacar una ficha del mazo y ponerla en la mano del jugador.

1. **Cambiar ficha por comodín a cambio de monedas**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
2. **Saltar al jugador siguiente a cambio de monedas**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**
3. **Revertir orden de jugadores a cambio de monedas**
   1. **Diagrama de secuencia**
   2. **Diagrama de clases**
   3. **Explicación**