
[75.07 / 95.02]

Algoritmos y programación III

Trabajo práctico 2: GPS Challenge

(trabajo grupal)

Alumno:

Nombre	Padrón	Mail
Ignacio Rodriguez Justo	108267	irodriguezj@fi.uba.ar
Ezequiel Tosto Valenzuela	108321	etosto@fi.uba.ar
Agustín de la Rosa	108275	adelarosa@fi.uba.ar
Juan Cruz Hernandez	105711	jchernandez@fi.uba.ar
Bruno Pascaretta	107840	bpascaretta@fi.uba.ar

Tutor: Maia Naftali

Nota Final:

Supuestos

Tomamos la suposición de que la penalización se suma al total de movimientos del jugador y también se le suma el costo de hacer un movimiento. Por ejemplo, si un auto se encuentra con un pozo, se le sumaría 3 de penalización al total y 1 por el movimiento ejecutado (4 total).

Otra suposición sería que al encontrarse un piquete, el cual te deja en el mismo lugar, suma 1 movimiento a movimientos totales.

Un supuesto es que una calle puede tener como máximo un obstáculo y/o una sorpresa.

Una calle que tiene una sorpresa y un obstáculo, primero activa la sorpresa luego choca con el obstáculo. Este caso incluye el cambio de vehículo.

La calle que tiene la Meta no tiene ningún obstáculo ni sorpresa.

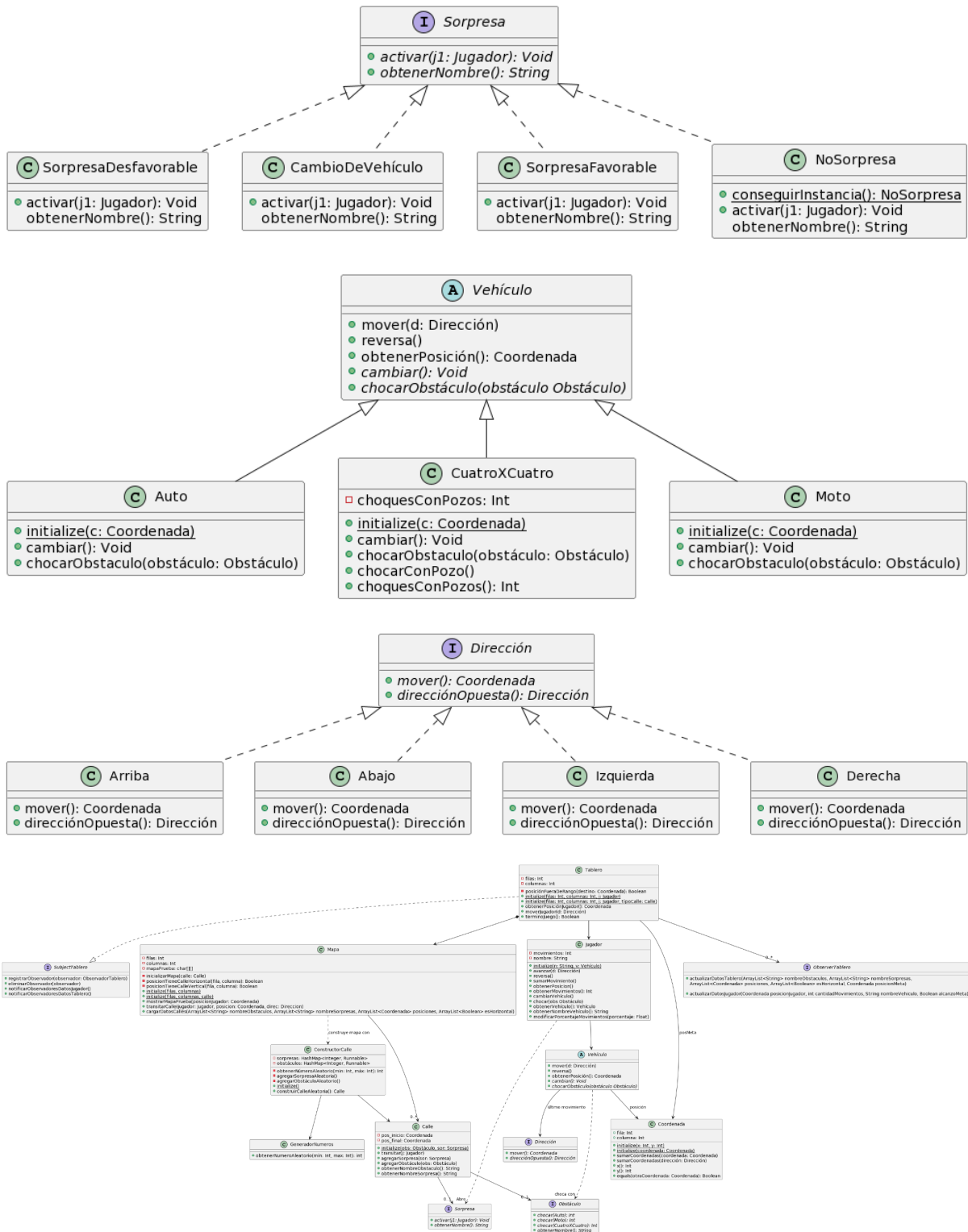
El tablero tiene que tener dimensiones cuadradas. Que la cantidad de filas sea igual a la cantidad de columnas.

Cada tamaño de tablero tiene una distinta lista de máximas puntuaciones.

El tamaño del tablero tiene que ser de la siguiente forma, $2 \leq \text{tamaño tablero} \leq 14$. El usuario tiene que ingresar por lo menos un caracter en el input de nombre.

La meta se encuentra siempre en la misma posición, en la última columna, en la fila del medio.

Diagrama de clases



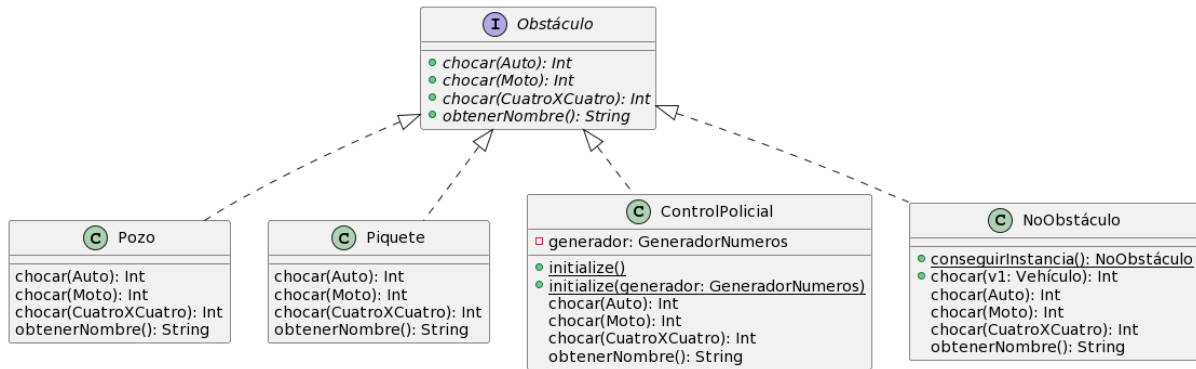


Diagrama de clases del ejercicio de **Sorpresata**:

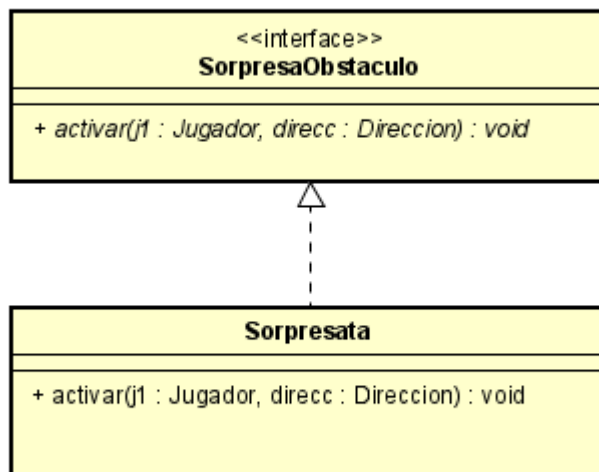
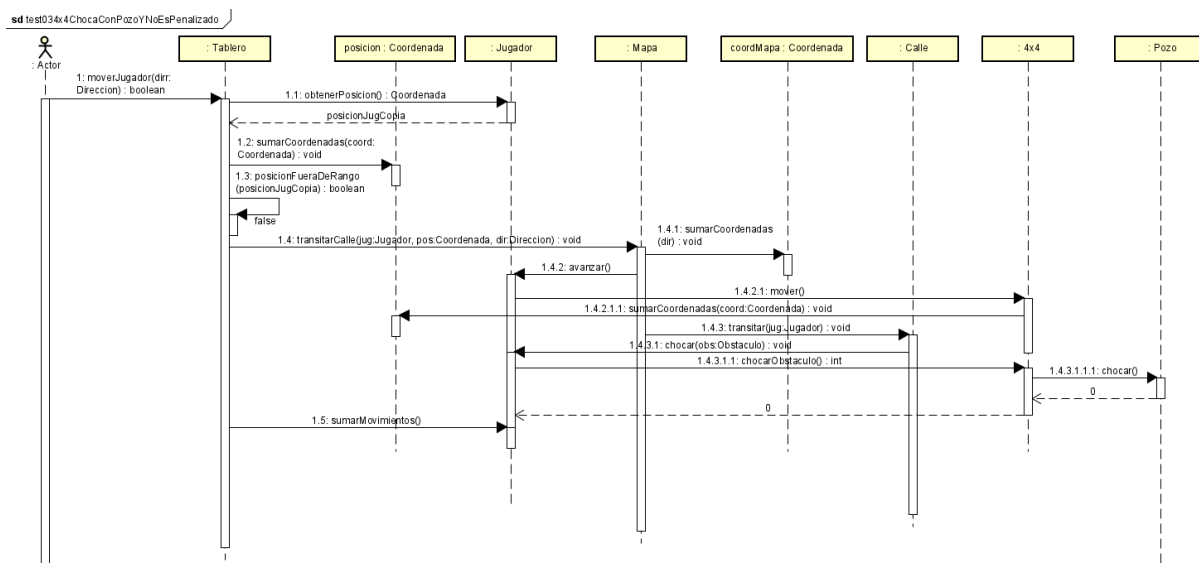
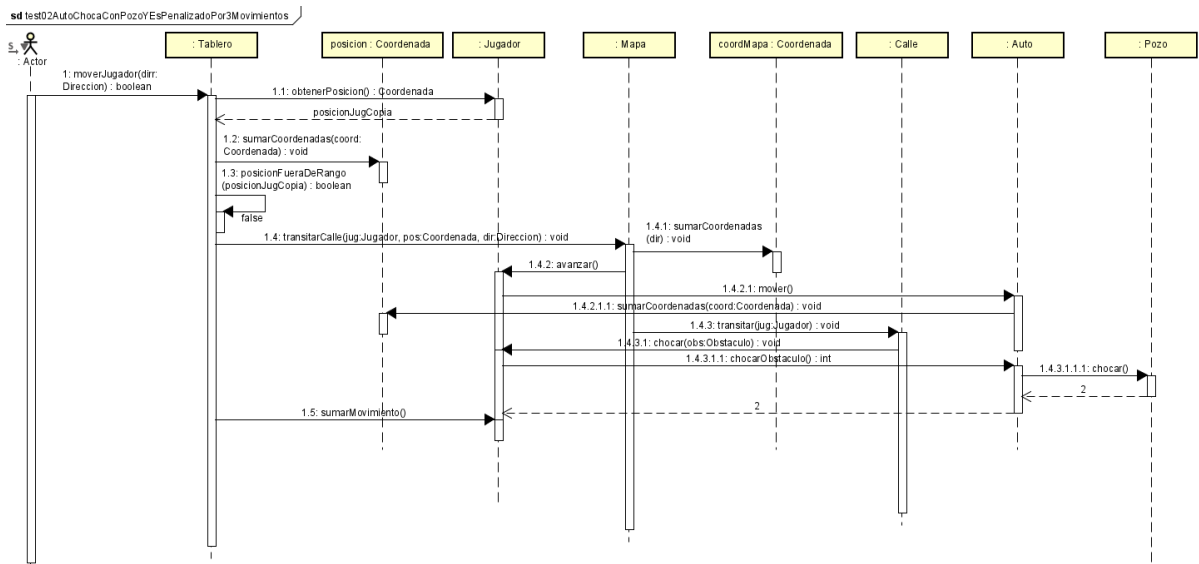
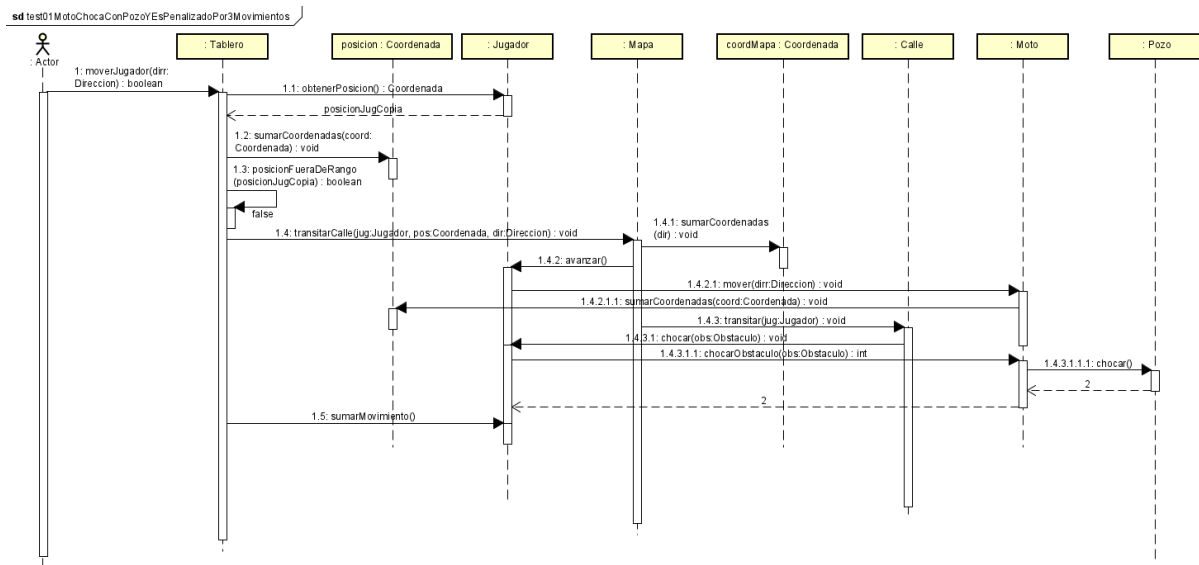
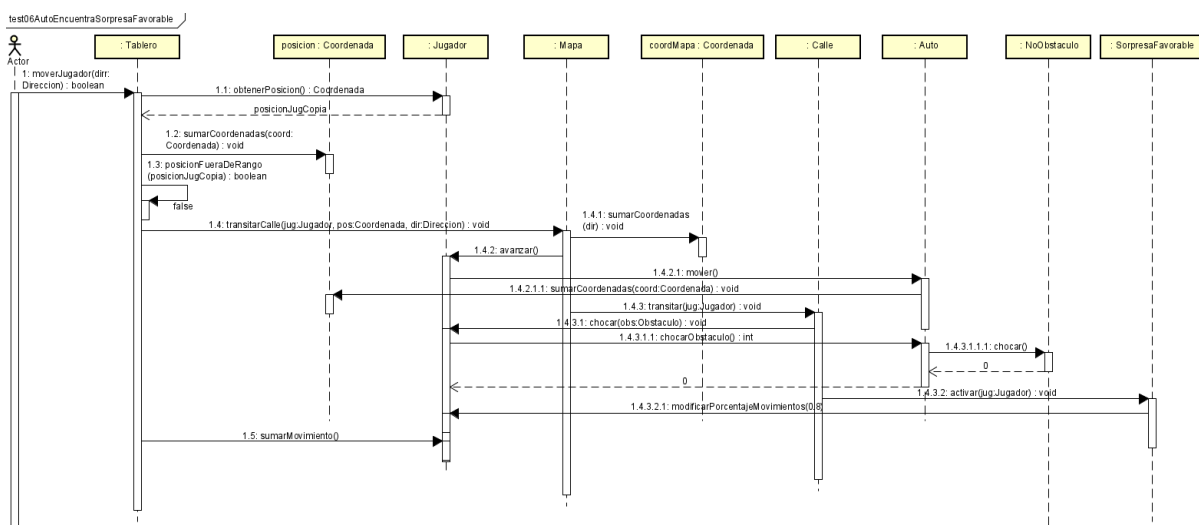
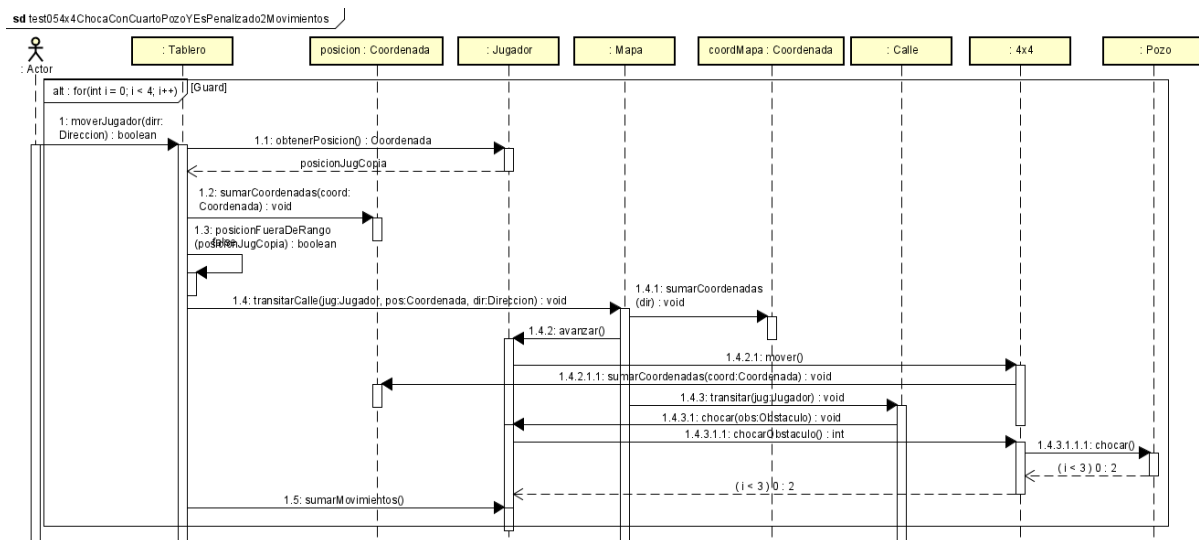
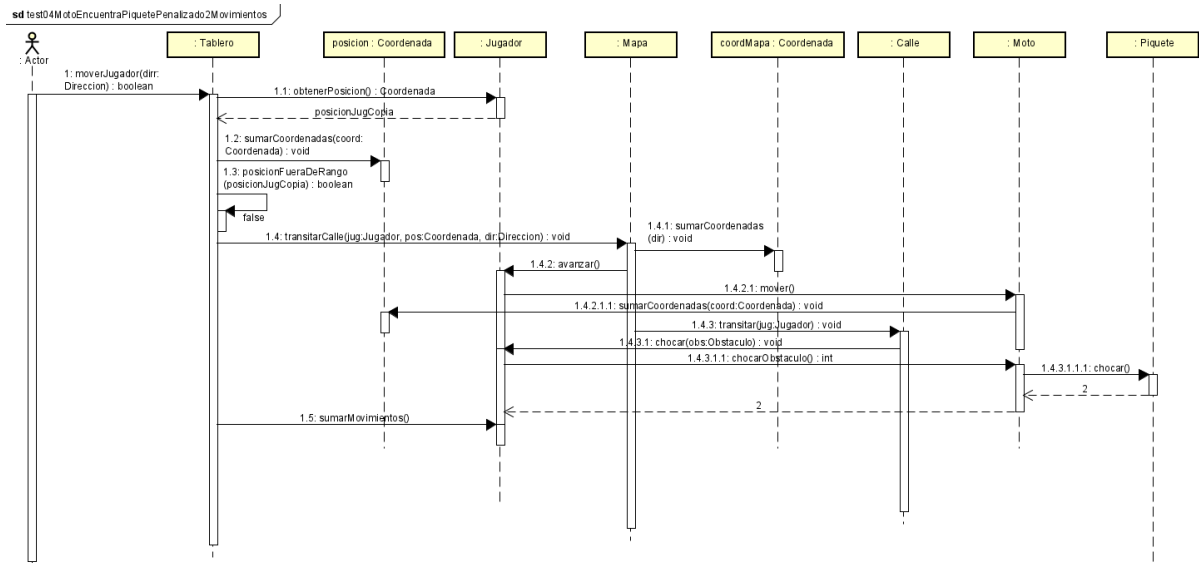


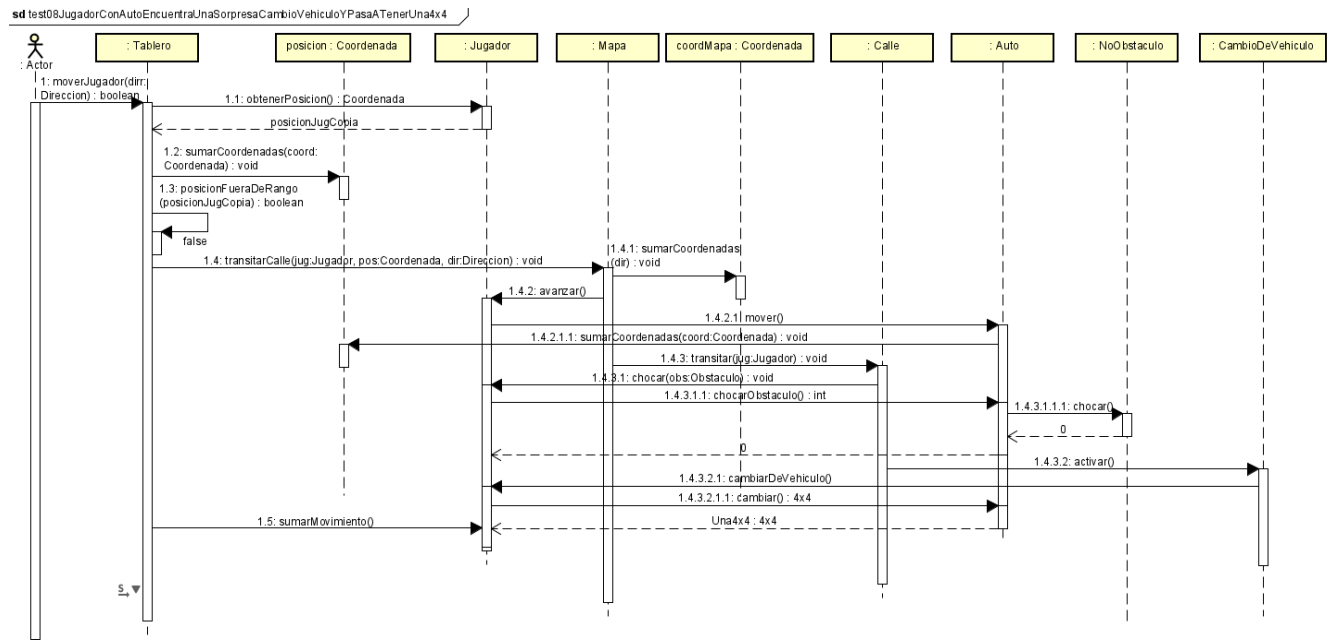
Diagrama de secuencia

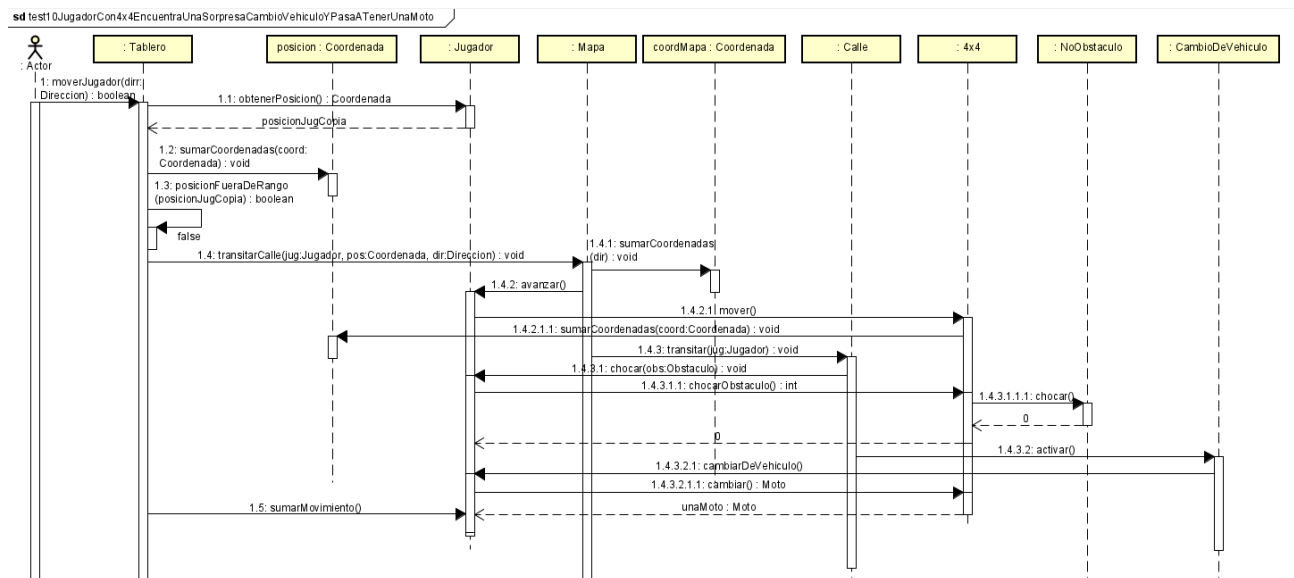
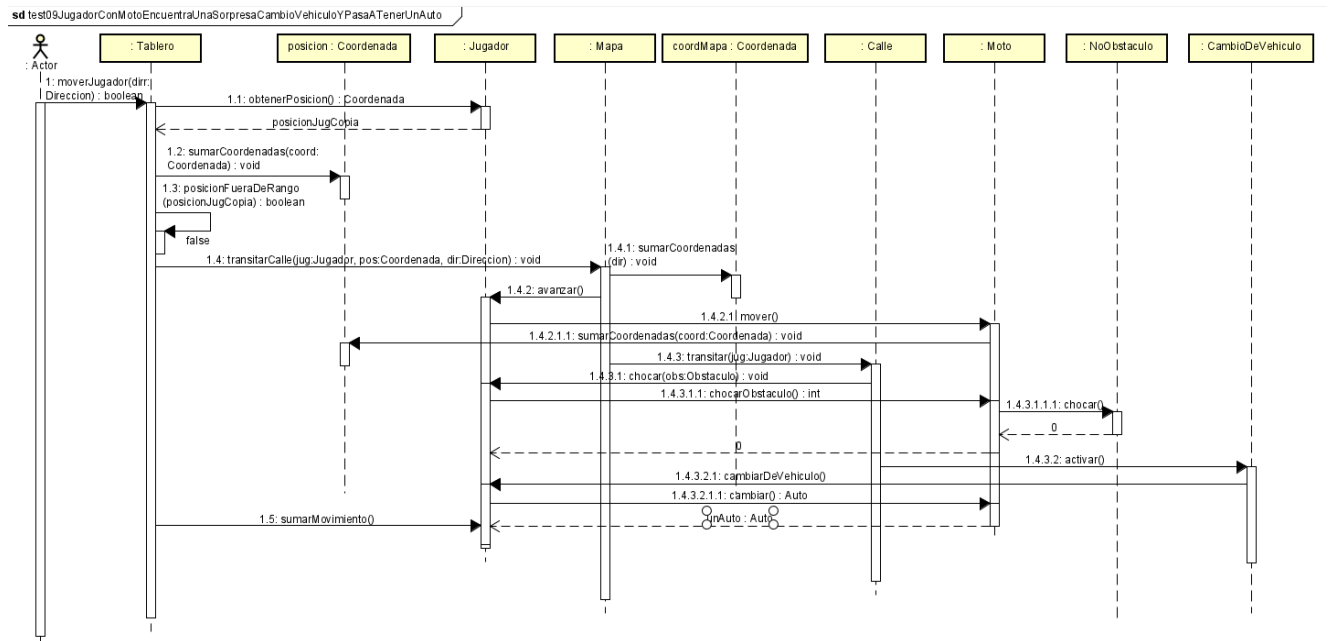
Recomendación, ver los diagramas desde el repositorio en GitHub
(*algo3_tp2 -> diagramas -> sequences -> ...*)

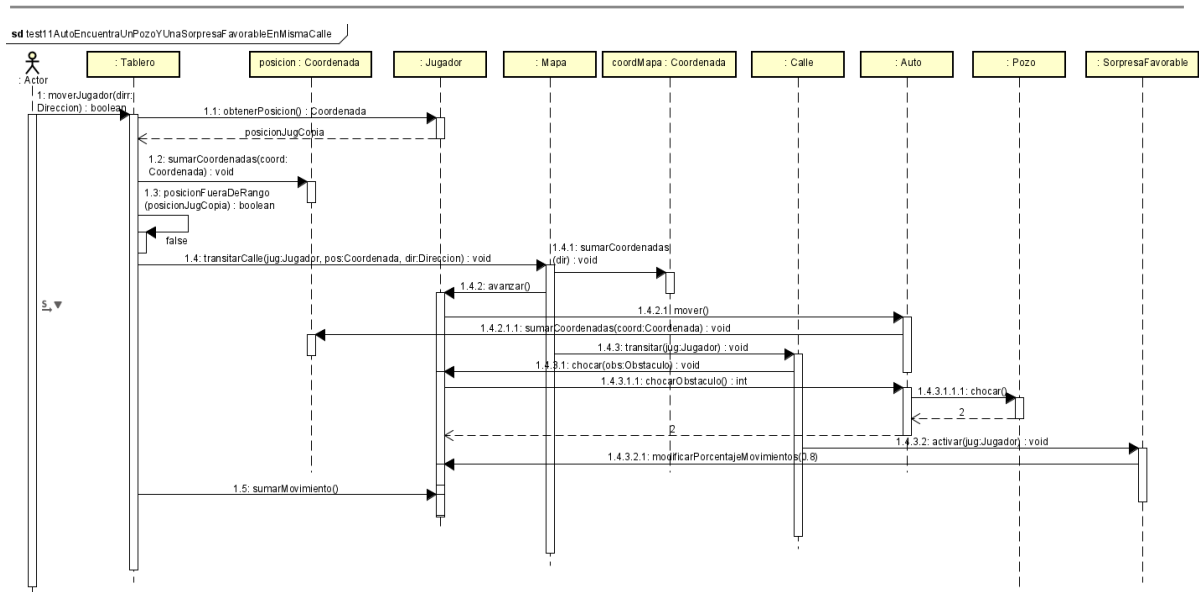
En los primeros tests, cabe aclarar que la calle además de llamar al obstáculo también llama a NoSorpesa que implementa Sorpesa.



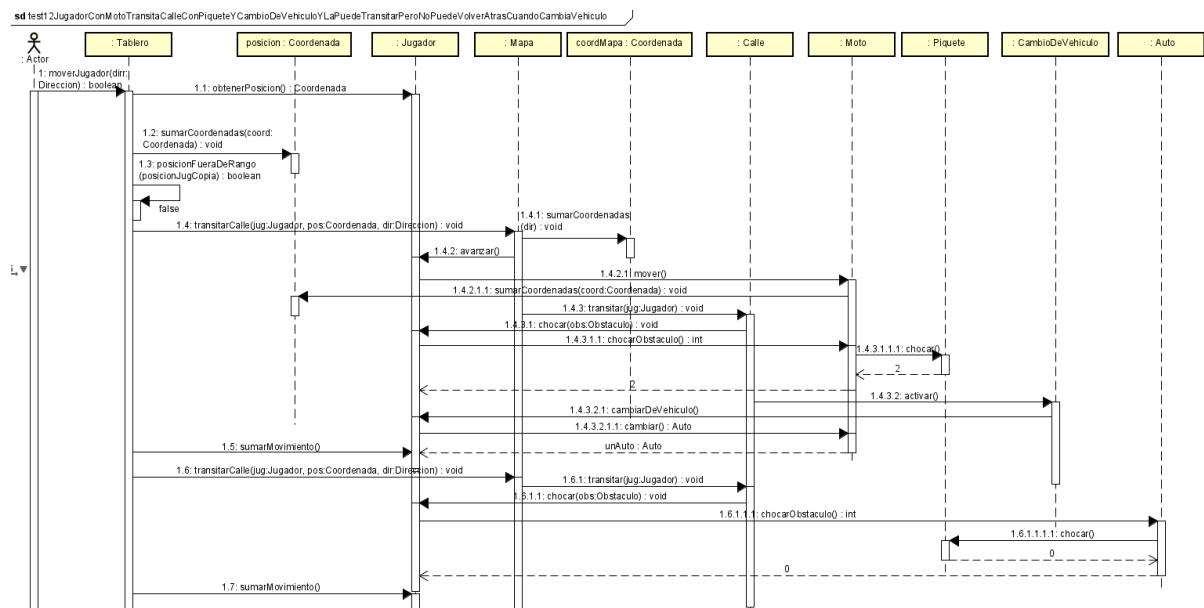


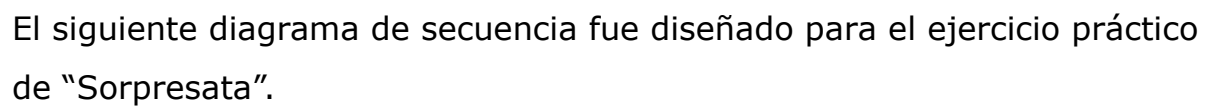


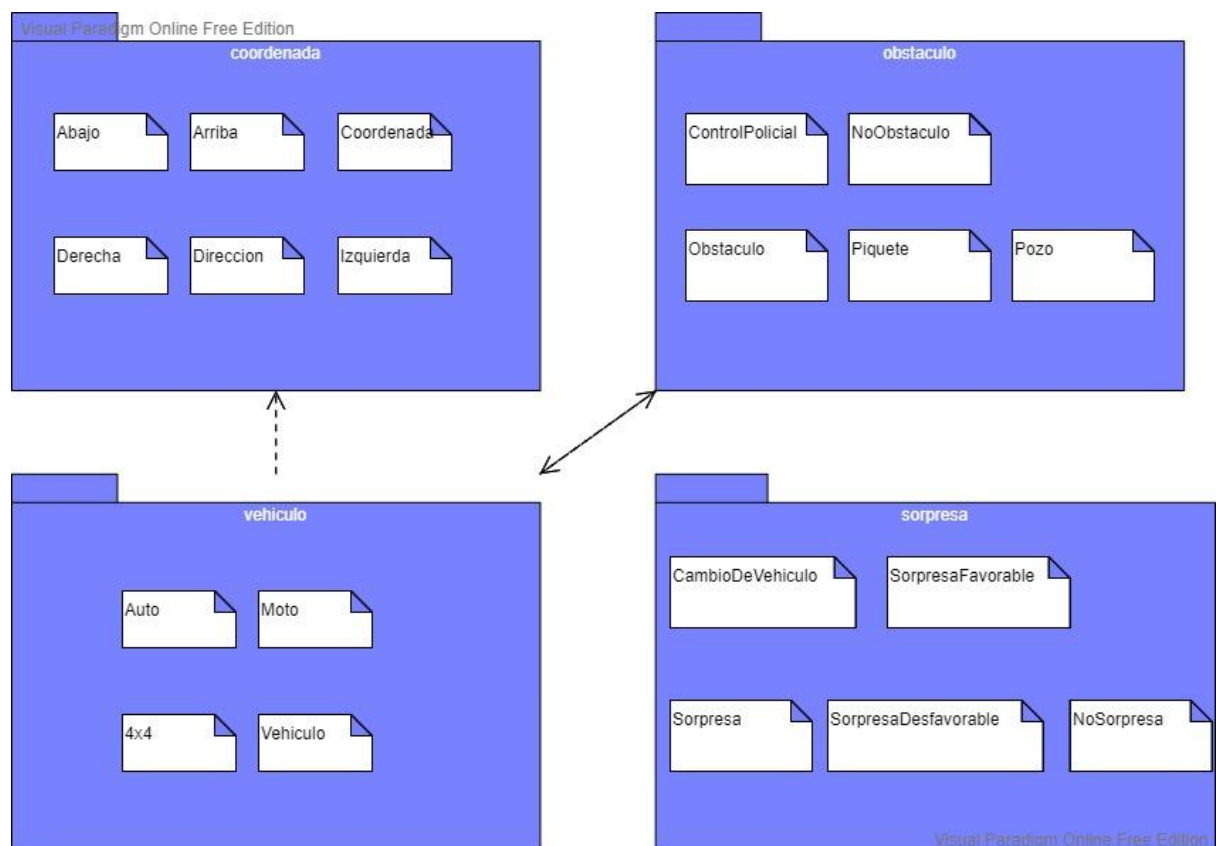
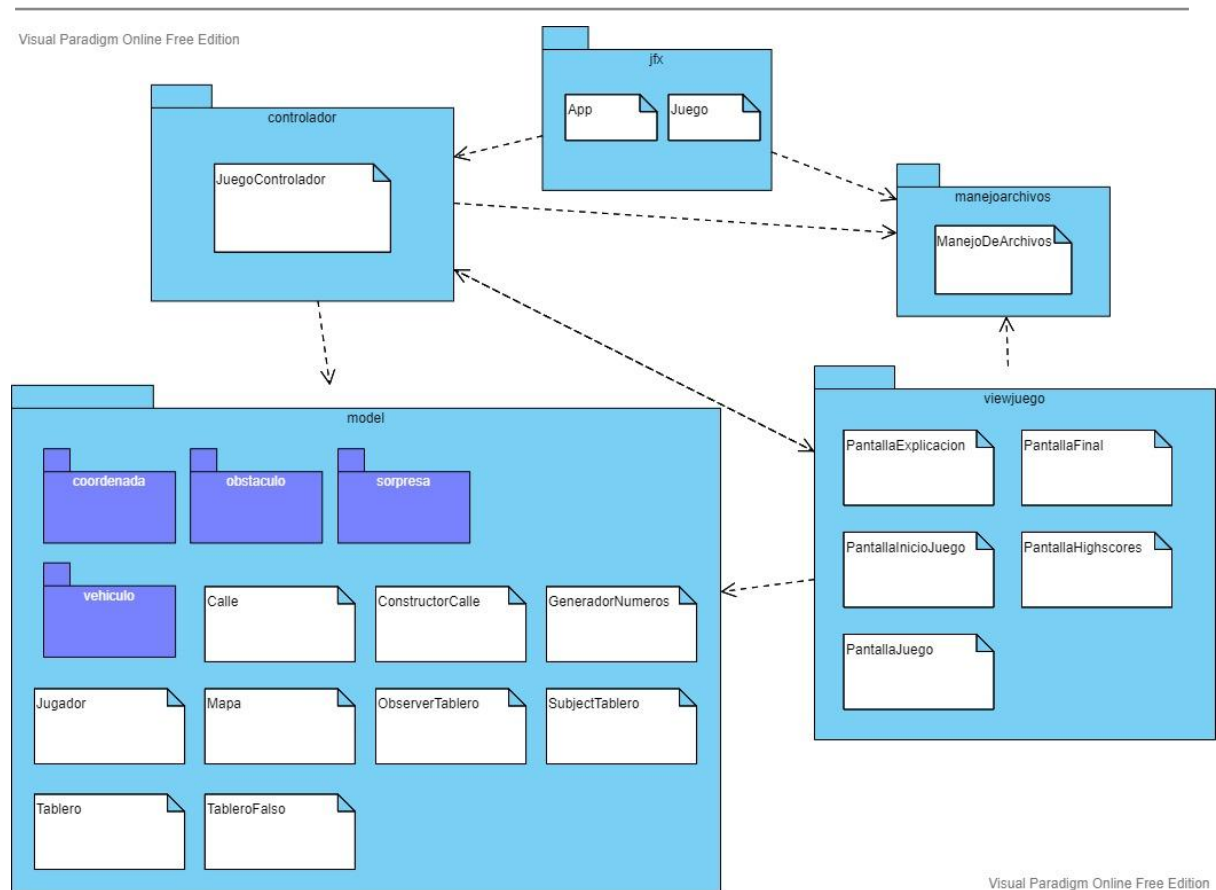




Otra aclaración, en los diagramas de secuencia de test12 y test13 obviamos algunos llamados los cuales se muestran previamente en el mismo diagrama, no queremos complicar el seguimiento.







Detalles de implementación

Utilizamos el diseño de patrón de Null Object y para contrarrestar una desventaja que sería el gran uso de memoria, también implementamos Singleton. Luego implementamos MVC para facilitar el manejo de la interfaz gráfica con respecto al modelo, en nuestro caso, el controller sería el `Main.java`. El diseño de patrón de Double Dispatch lo utilizamos cuando un vehículo choca contra un obstáculo. También utilizamos el patrón Observer para actualizar la vista del juego con los datos del tablero.

Excepciones

Para capturar las excepciones de manejo de archivo del tipo *IOException* y *FileNotFoundException* decidimos tener aislado en otro archivo todo lo relacionado a archivos para no perjudicar al usuario cortándole la ejecución del programa. No llegamos a implementar una forma de indicarle el error al usuario además de un `System.out.println(...)`.

Otra excepción que podría causar problema es la que ocurre cuando presionas una tecla que no es W, A, S o D. En nuestro caso, si el usuario se confunde, no pasa nada.

Para controlar que en los inputs de la pantalla inicial no hubiese ningún problema, controlamos que esté dentro de nuestros supuestos parámetros, $2 \leq \text{dimensión tablero} \leq 14$ y que se ingrese al menos un carácter en el nombre.