FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo práctico 2: GPS Challenge

[7507/9502] Algoritmos y Programación III
Primer cuatrimestre de 2022

Nombre	Padrón	Email
Olivera, Tomas Nahuel	106817	tolivera@fi.uba.ar
Polese, Martin Alejo	106808	mpolese@fi.uba.ar
Sedek, Jorge	91979	jsedek@fi.uba.ar
Lin, Cristian Martin	107825	clin@fi.uba.ar

TP2 - Algoritmos y Programación 3

Olivera, Tomas Nahuel Polese, Martin Alejo Lin, Cristian Martin

Sedek, Jorge

Mayo 2022

1. Introducción

El presente informe se muestra la documentación de la solución del segundo trabajo práctico de la materia Algoritmos y Programación III, consistiendo en la realización de un Juego utilizando el lenguaje Java, utilizando los conceptos aprendidos del paradigma POO (OOP en inglés).

2. Supuestos

Para el modelado del problema se hicieron los siguientes supuestos:

- El vehiculo comienza en una interseccion de calles(entrecalle).
- Las sorpresas y obstaculos solo se encuentran dentro de las calles.
- La meta solo puede aparecer en una calle.
- La meta y el vehiculo se generan al mismo tiempo cuando comienza la partida.
- El vehículo del jugador siempre comienza en la primera columna de casilleros(Entre Calle)
- La meta siempre está en la última columna de casilleros (Calle)

3. Diagramas de Clases

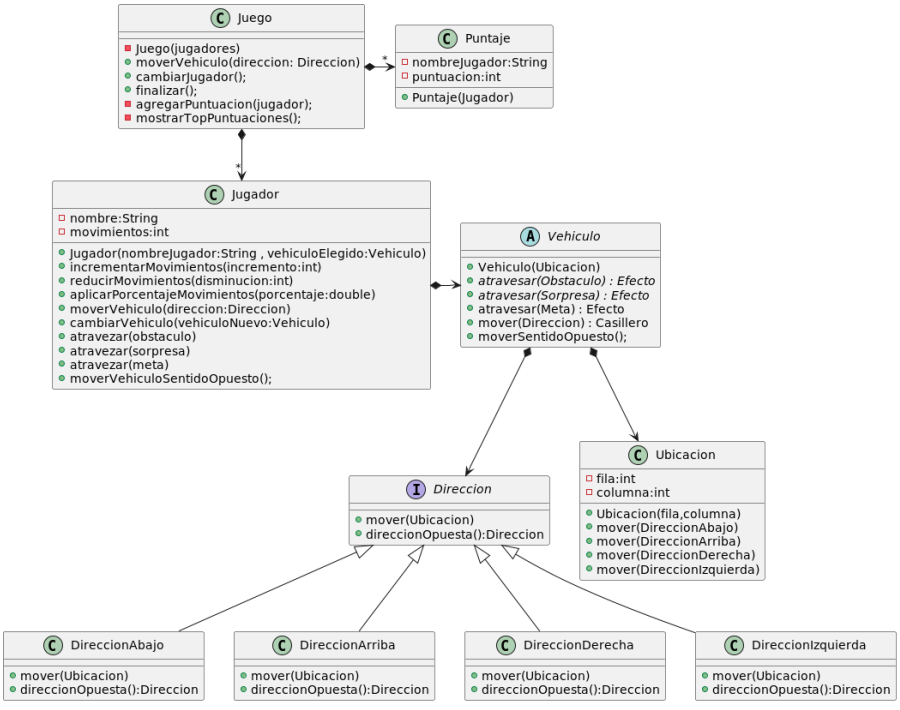

Figura 1: .Diagrama de clases general

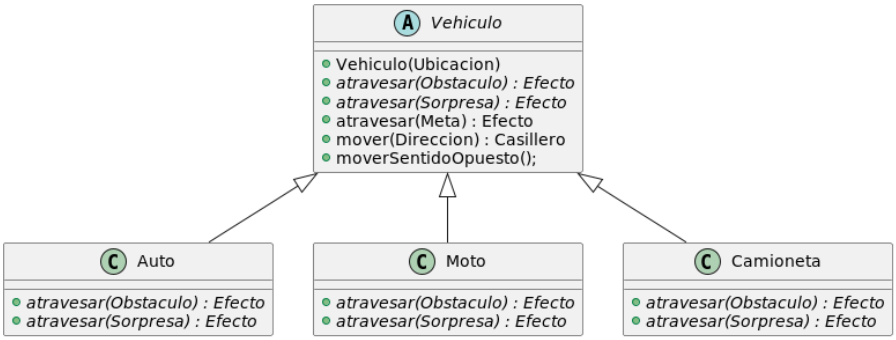

Figura 2: .Diagrama de clases: Vehiculos y la relacion con jugador

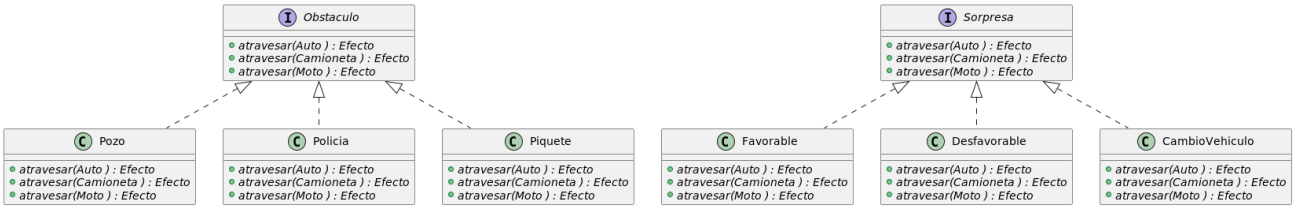

Figura 3: .Diagrama de clases: Intefaz de obstaculos y sorpresas con objeto urbano

4. Diagramas de Secuencias

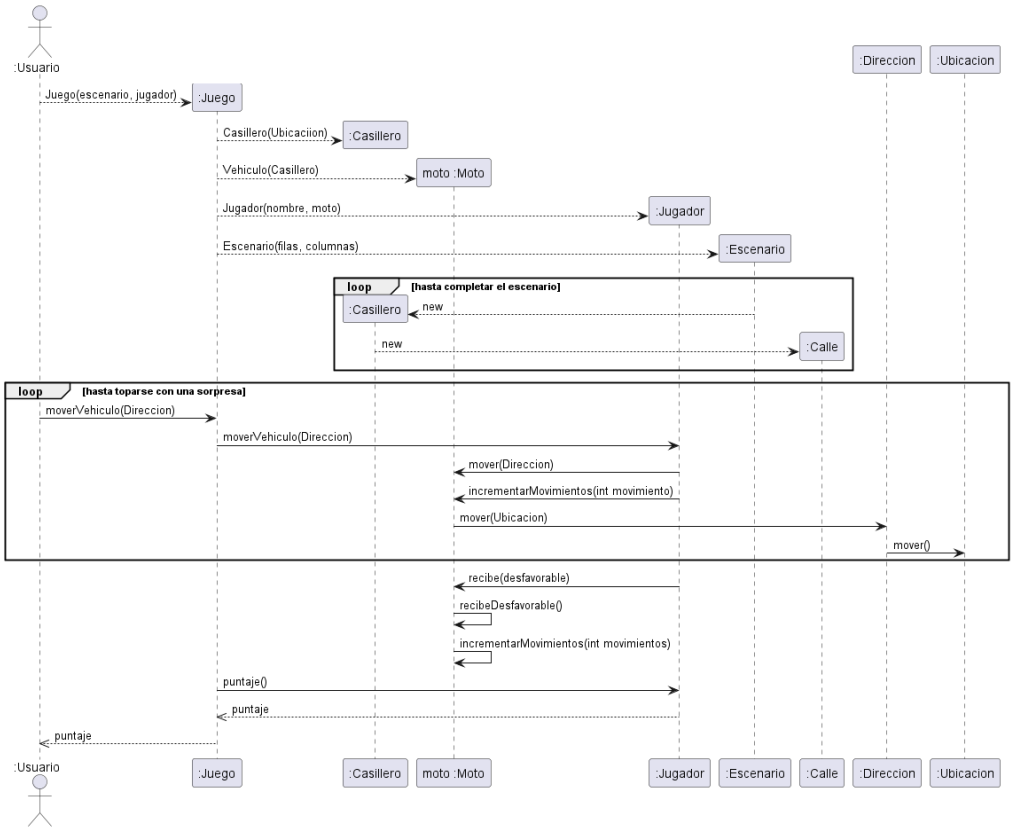

Figura 4: .UnaMotoAtraviesaLaCiudadYSeEncuentraConUnaSorpresaDesfavorable

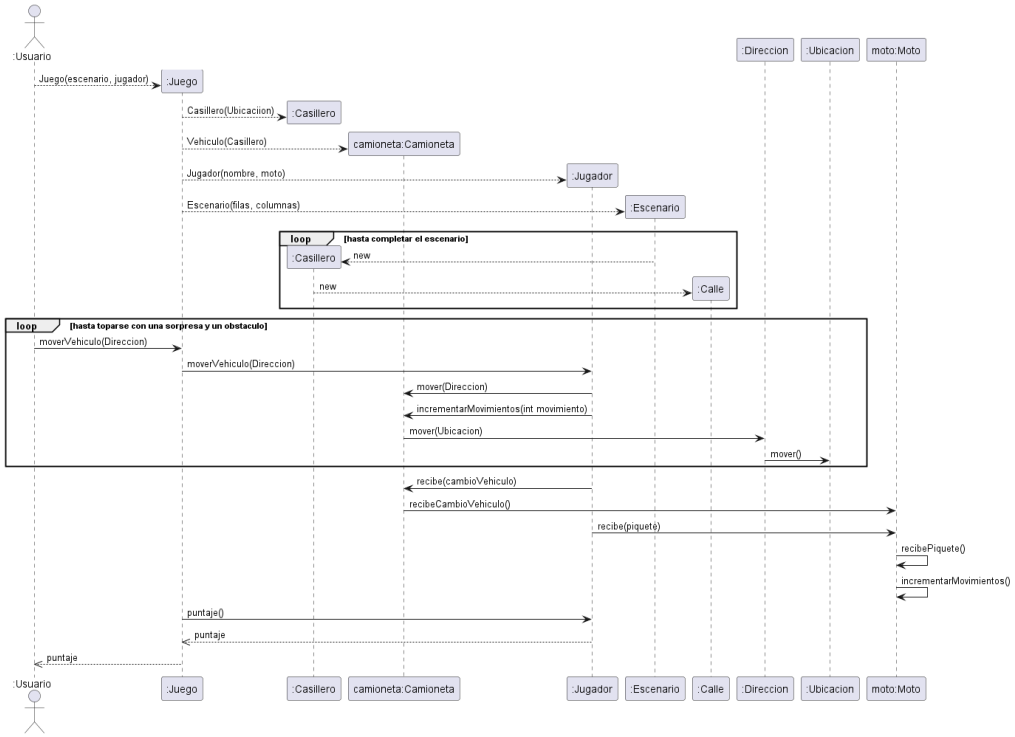

Figura 5: .UnaCamionetaSeEncuentraConUnCambioVehiculoYPasaPorUnPiquete

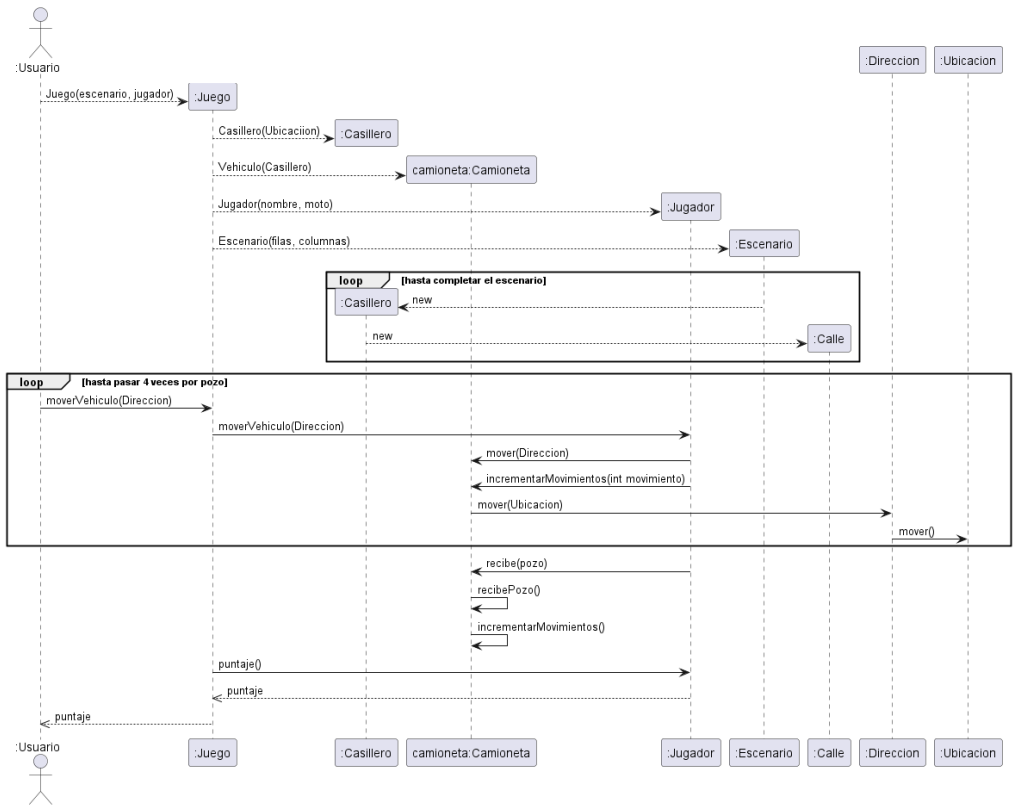

Figura 6: .UnaCamionetaAtraviesaLaCiudadYSeEncuentraConUnPozo4veces

5. Diagrama de Paquetes

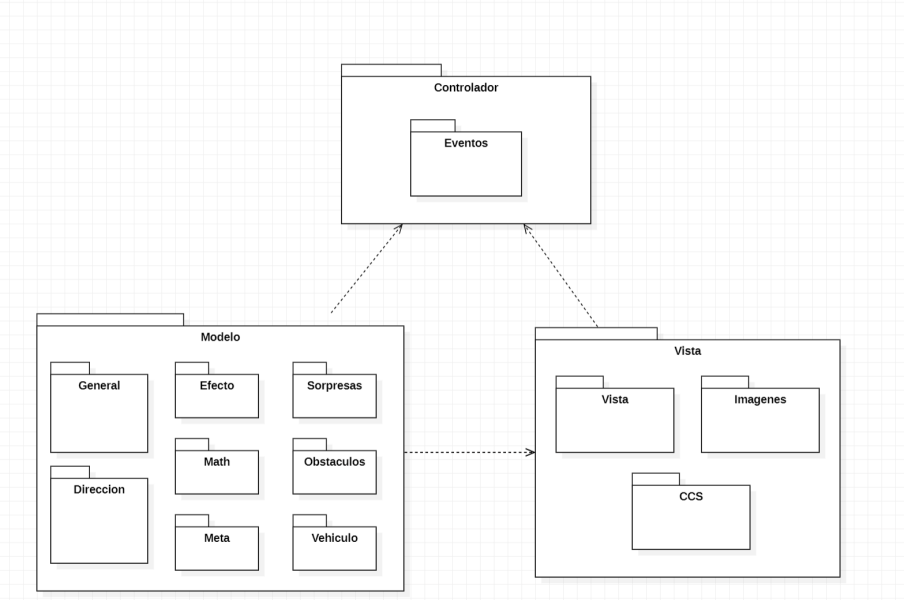

Figura 7: .Diagrama de paquete

6. Diagramas de Estado

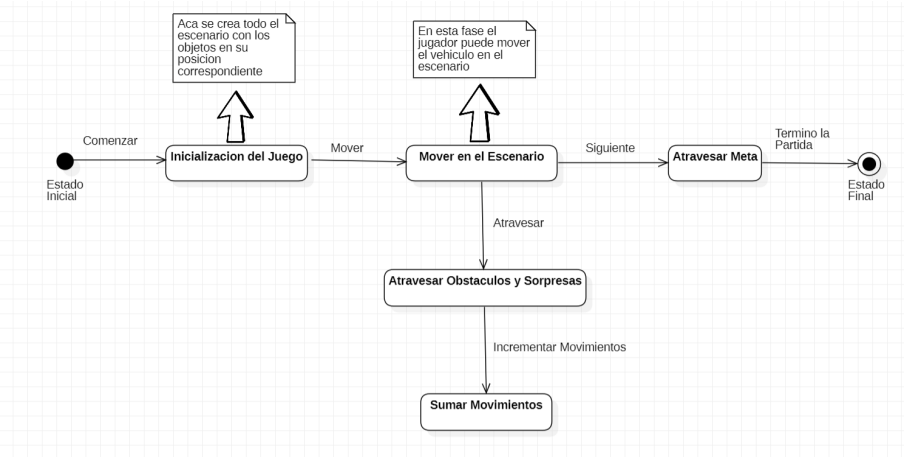

Figura 8: .Diagrama de Estado para el juego

7. Detalles de implementación

7.1. Juego

El juego es la clase "base"del modelo, esta compuesto por los jugadores, el jugador activo (el jugador al que le toca moverse), y utiliza un escenario con el que los jugadores interactuan.

7.2. Jugador

La clase Jugador representa a los jugadores que estan interactuando con nuestro juego. Un jugador tendra un nombre, un vehículo y sus movimientos.

7.3. Puntaje

El puntaje esta dado por el nombre del jugador y sus movimientos.

7.4. Ubicacion

Una ubicación representa una posición en el tablero. Esta dado por una fila y una columna, y se utiliza para el vehículo (para conocer su ubicación actual), y los casilleros.

7.5. Escenario

El escenario esta compuesto por casilleros, y es donde ocurre el Juego.

7.6. Casilleros

Un casillero puede representar una Entrecalle, Calle o Edificio según la ubicación del mismo. El vehículo siempre se movera entre Entrecalles y atravesara una Calle en su movimiento, donde esta ultima contendra los obstaculos y sorpresas pertinentes, como así tambien la meta (si es que contiene alguno de estos elementos).

7.7. Vehiculo

La clase Vehiculo es una clase abstracta que tendrá tres clases hijas: Moto, Auto, Camioneta.

7.8. Obstaculo

Obstaculo es una interfaz que implementan las clases: Piquete, Pozo, Policia y SinObstaculo. Estos se almacenan en un casillero, particularmente en los casilleros que representan una calle en el modelo.

7.9. Sorpresas

Sorpresa al igual que obstaculos es una intefaz que implementan las clases: CambioVehiculo, Desfavorable, Favorable, y SinSorpresa. Estas se almacenan en un casillero, particularmente en los casillleros que representan una calle en el modelo.

7.10. Direccion

Direccion es una interfaz que implementan las clases: DireccionAbajo, DireccionArriba, DireccionDerecha y DireccionIzquierda. Donde cada una de estas, representa un tipo de movimiento que puede realizar el vehiculo: vehiculo.mover(new DireccionDerecha), por ejemplo si se quisiera mover el vehiculo a la derecha.

8. Excepciones

De momento no tenemos excepciones en el modelo, pues con nuestra implementanción los únicos conflictos que encontramos son:

Que un jugador intente salirse del tablero. Que dos jugadores compartan una misma ubicación.

Pero ambos problemas pueden solucionarse internamente, sin la necesidad de levantar excepciones, pues si un jugador intenta salirse del tablero, simplemente vuelve para atras, y si dos jugadores comparten la misma ubicación, simplemente la comparten sin conflictos.