

FIUBA - 7507

Algoritmos y programación 3

Trabajo práctico 2: GPS Challenge

2do cuatrimestre, 2013

(trabajo grupal)

Nombre	Padrón	Mail
Alic, Mirko	92774	turco807@gmail.com
Calcabrini, Eduardo	92662	educ_907@hotmail.com
Rozenzwaig, Ezequiel	92754	ezequielrozen@gmail.com

Fecha de entrega final: 13/12/13

Tutor: Massuh, Pablo

Nota Final:

Tabla de contenidos

Introducción

-Objetivo del trabajo

Consigna general

Descripción de la aplicación a desarrollar

-Contexto

Objetivo del juego

Puntos

Dinamica del juego

Disparos

Tablero

Naves

Entregables

Forma de entrega

Informe

-Supuestos

Modelo de dominio

Diagramas de clases

Detalles de implementación

Excepciones

Diagramas de secuencia

Checklist de corrección

Código

Introducción

Objetivo del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando en forma grupal y utilizando un lenguaje de tipado estático (Java)

Consigna general

Desarrollar la aplicación completa, incluyendo el modelo de clases e interface gráfica. La aplicación deberá ser acompañada por prueba unitarias e integrales y documentación de diseño. En la siguiente sección se describe la aplicación a desarrollar.

Descripción de la aplicación a desarrollar

Contexto

La empresa Algo Ritmos SA dedicada al desarrollo de video juegos a decidido contratar a un grupo de programadores para implementar el juego GPS Challenge

Objetivo del juego

GPS es un juego de estrategia por turnos. El escenario es una ciudad y el objetivo, guiar un vehículo a la meta en la menor cantidad de movimientos posibles.

Dinámica del juego

El juego se jugará por turnos, y en cada turno el usuario decide hacia cual de las 4 esquinas posibles avanzará.

Vehículos

El jugador podrá optar por tres diferentes tipos de vehículos.

- moto
- auto
- 4x4

Obstáculos

Al atravesar una cuadra el jugador se podrá encontrar con alguno de los siguientes obstáculos:

- Pozos: Le suma 3 movimientos de penalización a autos y motos, pero no afecta a las 4x4.
- Piquete: Autos y 4x4 deben pegar la vuelta, no pueden pasar. Las motos puede pasar con una penalización de 2 movimientos.
- Control Policial: Para todos los vehículos la penalización es de 3 movimientos, sin embargo la probabilidad de que el vehículo quede demorado por el control y sea penalizado es de 0,3 para las 4x4, 0,5 para los autos y 0,8 para las motos ya que nunca llevan el casco puesto.

Sorpresas

También se podrán encontrar diferentes tipos de sorpresas:

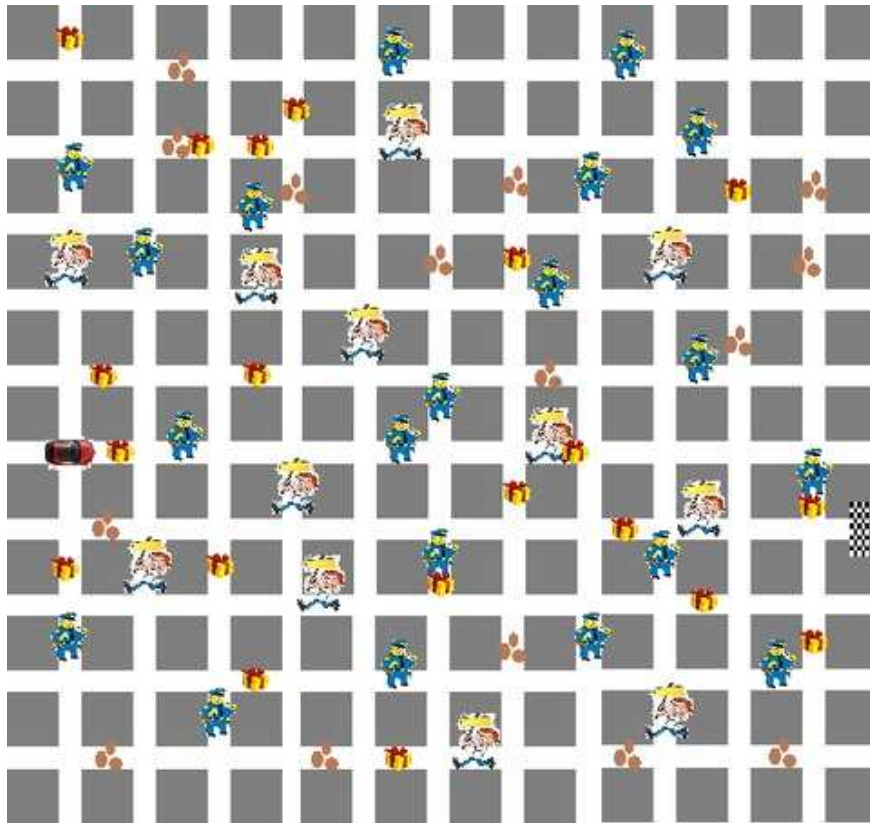
- Sorpresa Favorable: Resta el 20% de los movimientos hechos.
- Sorpresa Desfavorable: Suma el 25% de los movimientos hechos.
- Sorpresa Cambio de Vehículo: Cambia el vehículo del jugador. Si es una moto, la convierte en auto. Si es un auto lo convierte en 4x4. Si es una 4x4 la convierte en moto.

Las sorpresas figuraran en el mapa como un regalo y no se sabrá que es hasta que el vehículo la accione.

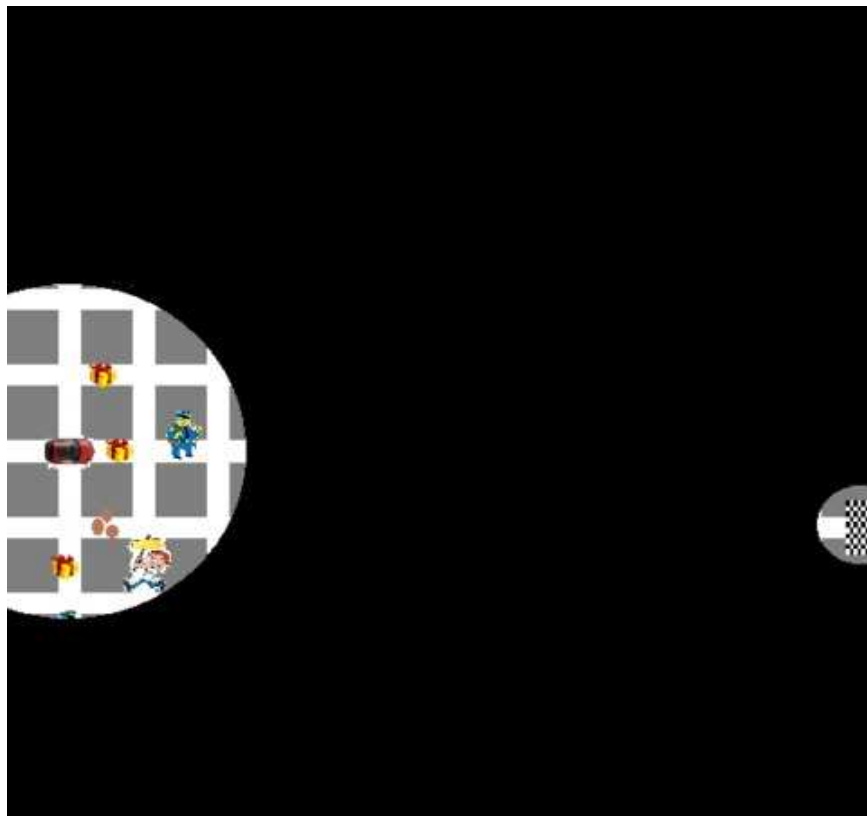
Escenario

Para hacerlo más interesante y jugable, el jugador no podrá ver más que dos manzanas a la redonda de la posición de su vehículo y la bandera a cuadros que marca la meta. El resto del mapa permanecerá en sombras.

El tamaño del escenario no será fijo, y tendrá un punto de partida y una meta.



Ejemplo de escenario



Ejemplo de escenario como lo visualiza el jugador

Puntajes altos

Se debe almacenar un ranking donde figuren los mejores puntajes asociados a un nickname que indique el usuario.

Entregables

- Código fuente de la aplicación completa, incluyendo también: código de la pruebas, archivos de recursos
- Script para compilación y ejecución (ant)
- Informe, acorde a lo especificado en este documento

Forma de entrega

A coordinar con el docente asignado.

Fechas de entrega

Se deberá validar semanalmente con el docente asignado el avance del trabajo. El docente podrá solicitar ítems específicos a entregar en cada revisión semanal.

La entrega final deberá ser en la semana del 13 de diciembre, en la fecha del curso en que se está inscripto.

Informe

Supuestos

- Decidimos que el juego se base en tiempo y no en movimientos por lo tanto hubo modificaciones en los diferentes objetos que interactúan con el vehículo.
- Siguiendo con el punto anterior, los obstáculos que incrementaban cantidades de movimientos (Pozo, ControlPolicial, Piquete) se modificaron con el fin de que el auto quede inmóvil durante esa cantidad de movimientos penalizados. De esta forma se logra que transcurra tiempo.
- Con las sorpresas que suman o restan movimientos en un porcentaje, se decidió tener en cuenta el porcentaje transcurrido en el momento que interactúa con el vehículo y sumarlo o restarlo al tiempo total.
- Nos referimos a mapa básico o mapa simple al tablero de juego que solo tiene cuadras y calles distribuidas de la forma más sencilla, como se muestra en las imágenes en la parte de Escenario de este informe.

Modelo de dominio

- Mapa: Es la encargada de contener las posiciones habilitadas y almacenar diferentes elementos para el momento de jugar, como puede ser un vehículo o obstáculos/sorpresas teniendo en cuenta las posiciones habilitadas.
- EditorMapa: Tiene la responsabilidad de dejar al mapa en condiciones optimas para jugar. Para tener una referencia visual es la encargada de establecer que posiciones son calles para que los obstáculos y vehículos puedan posicionarse.
- Vehículo: Esta clase se relaciona de forma directa con auto, moto y 4x4 a través de un estado y se verá afectada de forma diferente cuando actúe con los distintos obstáculos.
- Piloto: Es el que se encarga de conducir al Vehículo y a su vez un intermediario entre los obstáculos/sorpresas y el vehículo.
- Obstáculo y Sorpresa: Hay diferentes clases que depende de estas dos clases. Cada una interactuara con el vehículo de diferentes formas teniendo en cuenta penalizaciones, beneficios o cambios de vehículos.

Diagramas de clases

Diagrama de clase del juego:

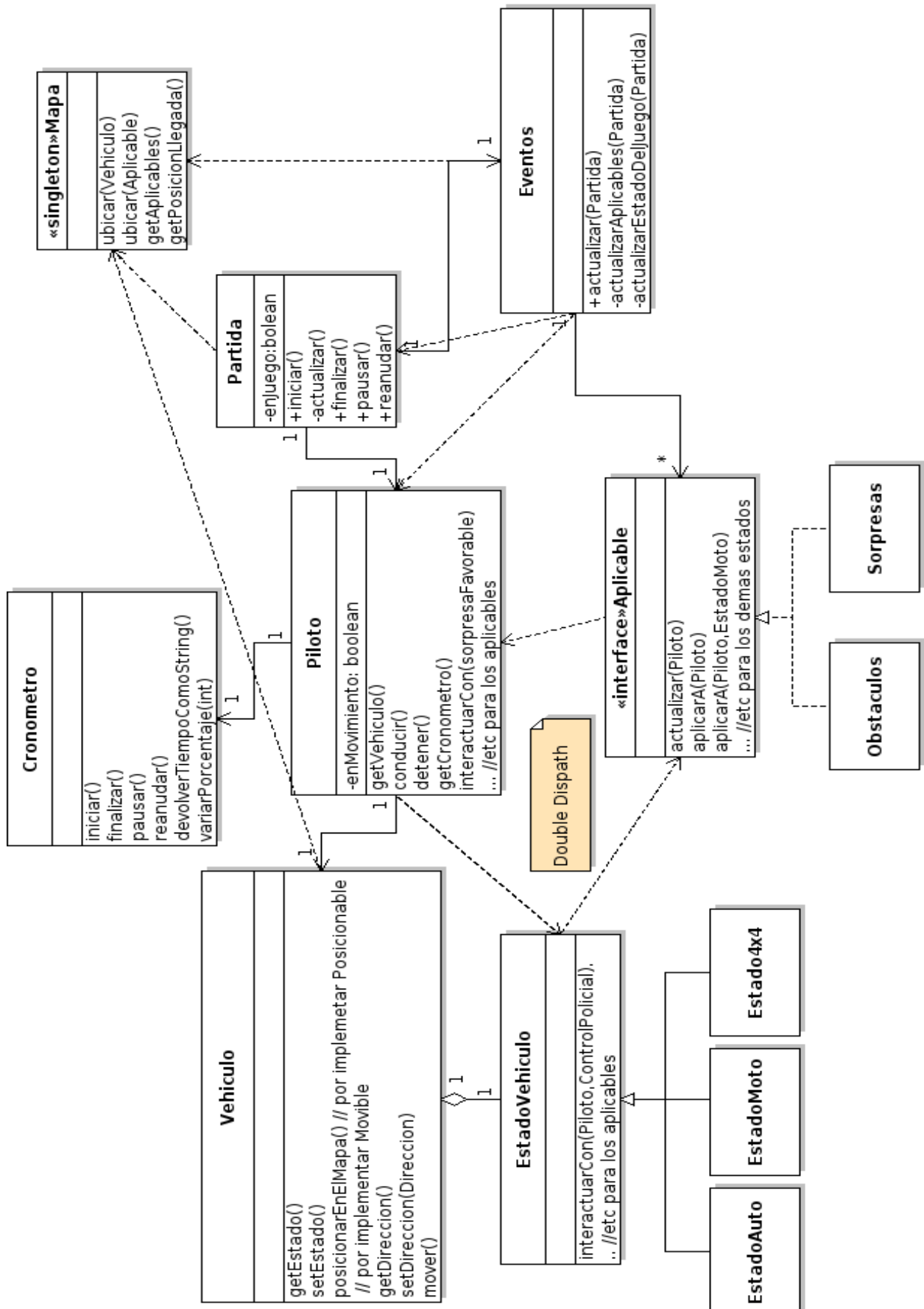


Diagrama de clase de aplicables:

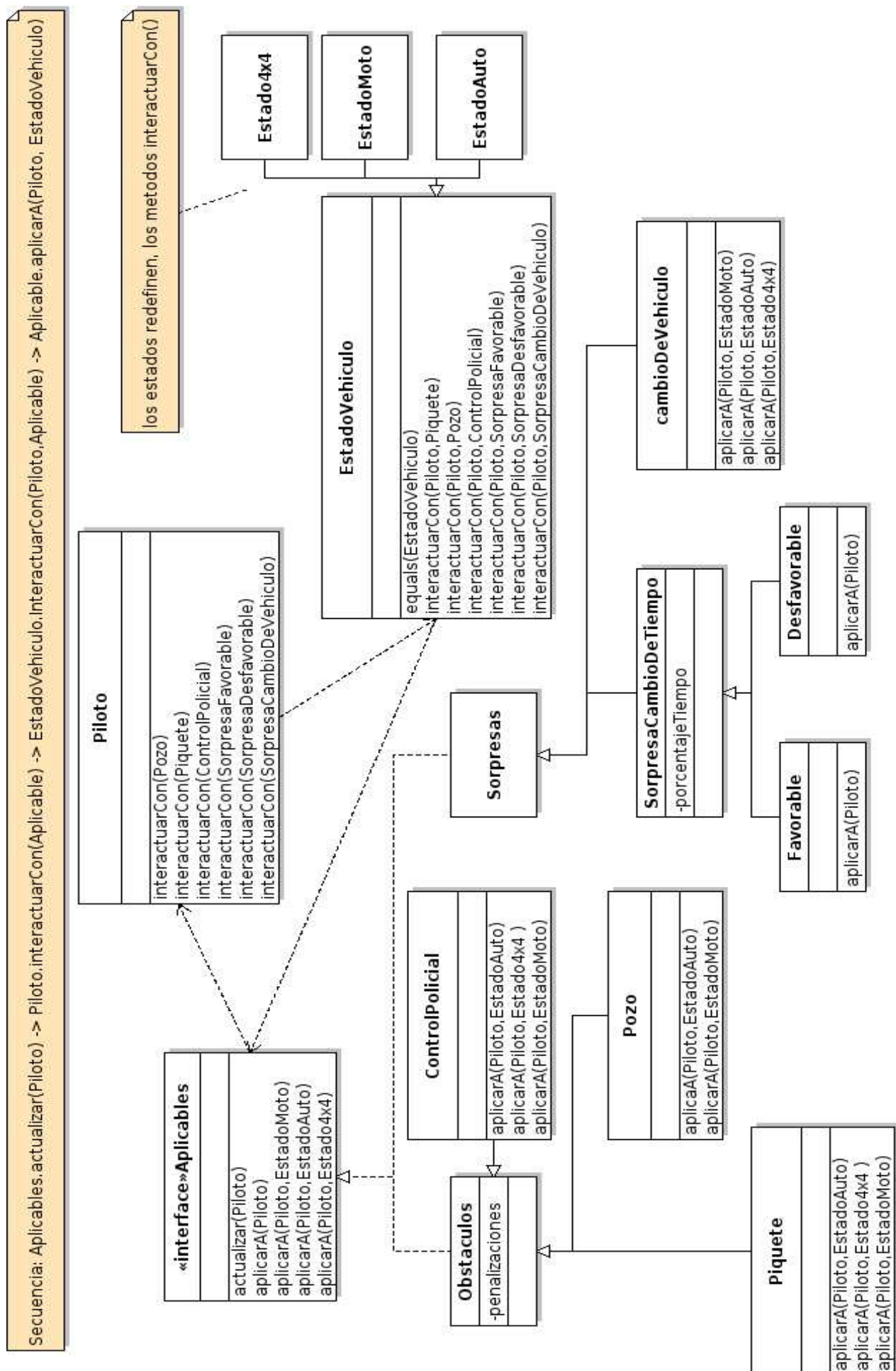


Diagrama de clase de un vehículo:

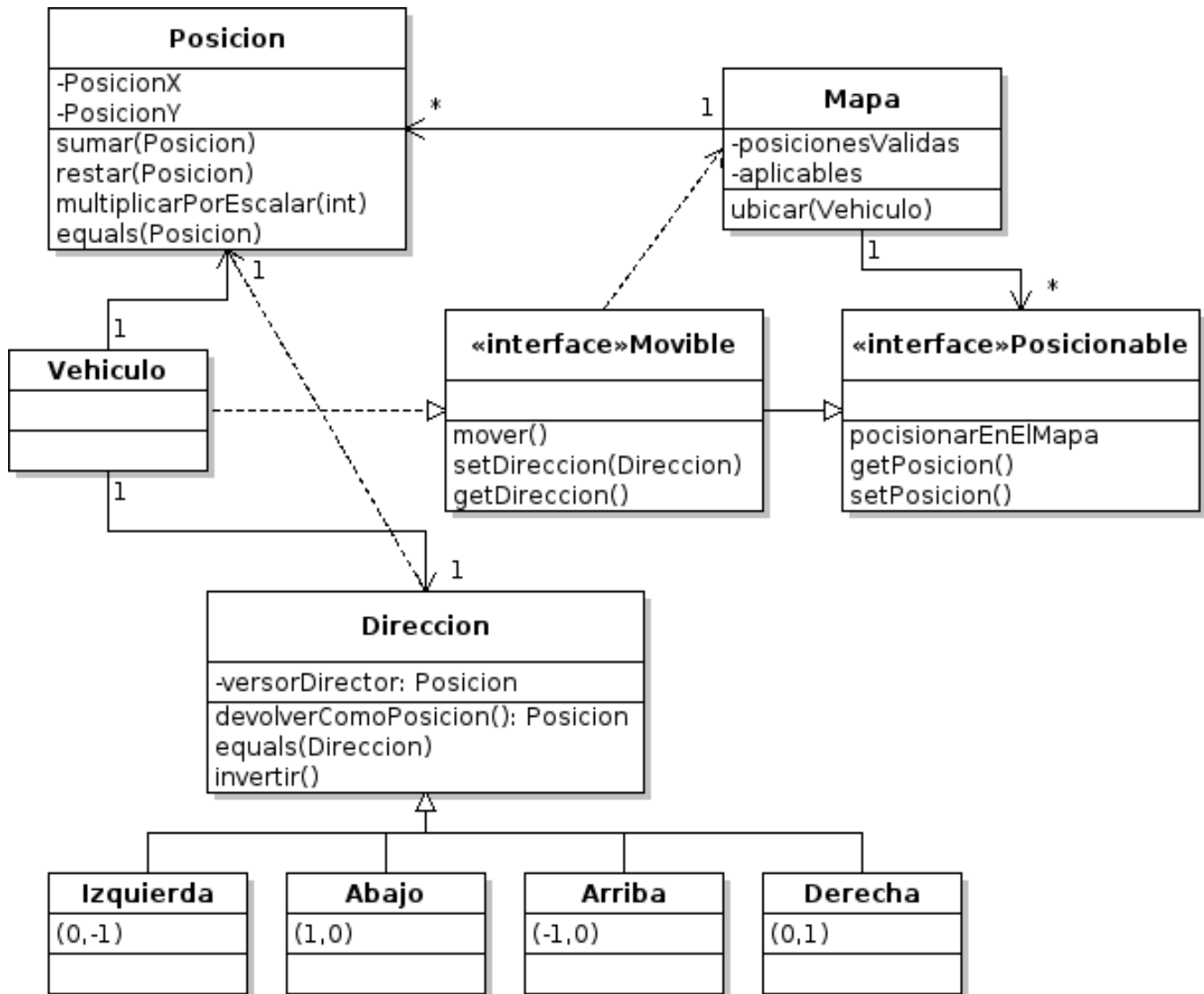
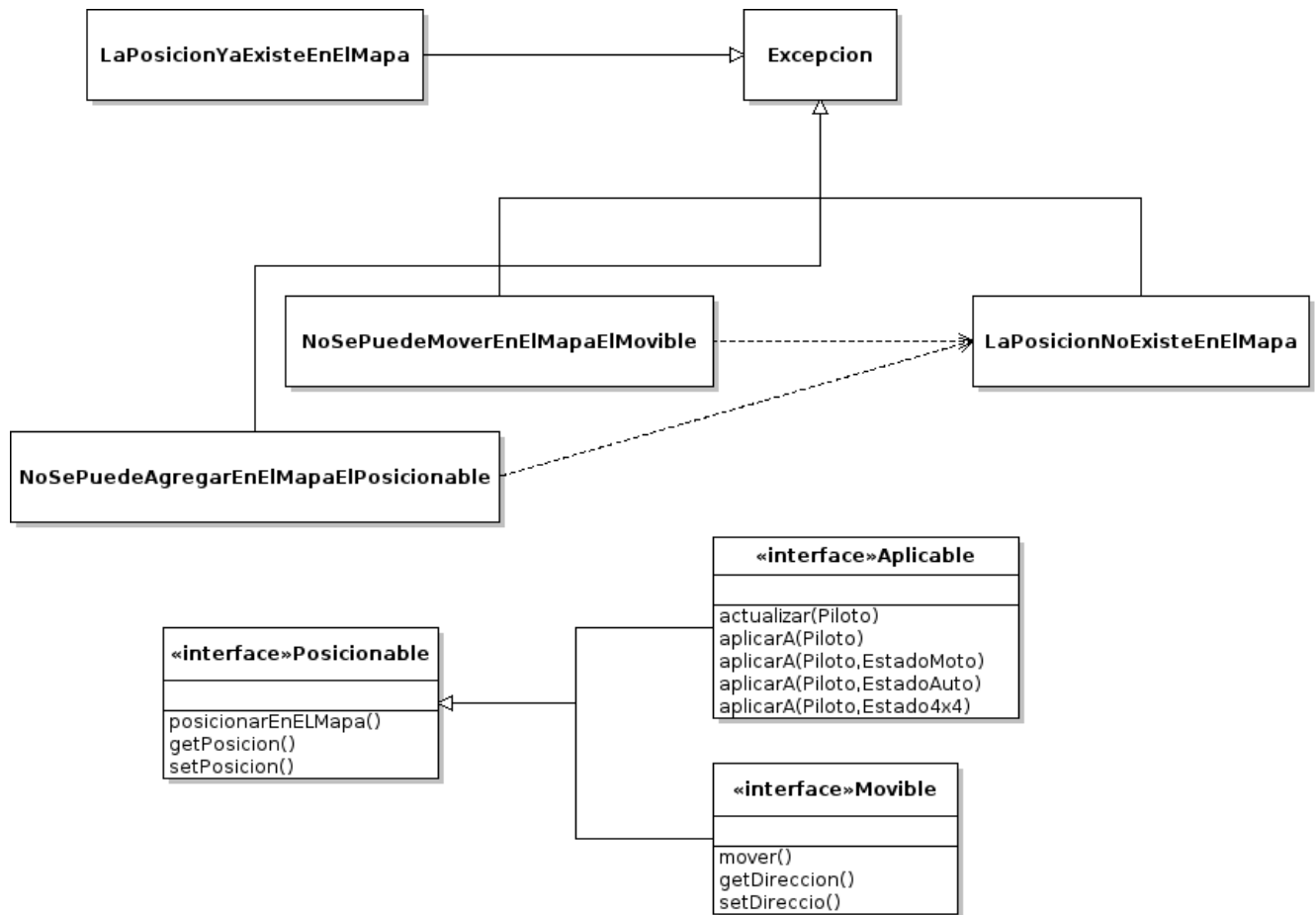


Diagrama de clase de interfaces y excepciones



Detalles de implementación

La clase Eventos es una clase que nos resulta interesante explicar, se encarga de actualizar los eventos de la partida en juego.

Por una parte tiene el comportamiento necesario para acoplar la interacción entre los aplicables que posea el mapa y el piloto que se encuentre conduciendo, es decir, les enviara los mensajes correspondientes a dichas clases para que puedan comunicarse, logrando así, un mayor nivel de encapsulamiento y una mejor agregación.

Por otra parte, estará al tanto del estado en el que se encuentra la partida, y en caso de que verifique su finalización se lo comunicara a la misma.

De esta forma la clase Eventos logra encapsular todo el comportamiento necesario para actualizar y actuar respecto a los eventos que van sucediendo.

Excepciones

Las diferentes excepciones creadas son:

- **LaPosicionYaExisteEnElMapa:** Debido a que las posiciones son objetos únicos, el propósito de esta excepción es advertir al programador que esa posición ya fue agregada al mapa.
- **LaPosicionNoExisteEnElMapa:** En algún momento del desarrollo es necesario pedirle al mapa posiciones determinadas por lo tanto si alguna de las pedidas nunca fue agregada al mapa saldrá esta excepción. Por otro lado sirve para cuando se intenta recurrir a posiciones que superan las dimensiones del mapa.
- **NoSePuedeMoverEnElMapaElMovable:** En casos como el vehículo al querer moverse a una determinada posición puede que esta no exista (si es una cuadra) por lo tanto no se le permite llevar a cabo la tarea.
- **NoSePuedeAgregarEnElMapaElPosicionable:** Esta excepción esta con el fin de avisarle al programador que se esta intentando colocar un posicionable en alguna posición invalida ya sea una cuadra o fuera del rango del mapa

Diagramas de secuencia

Diagrama de secuencia de un juego:

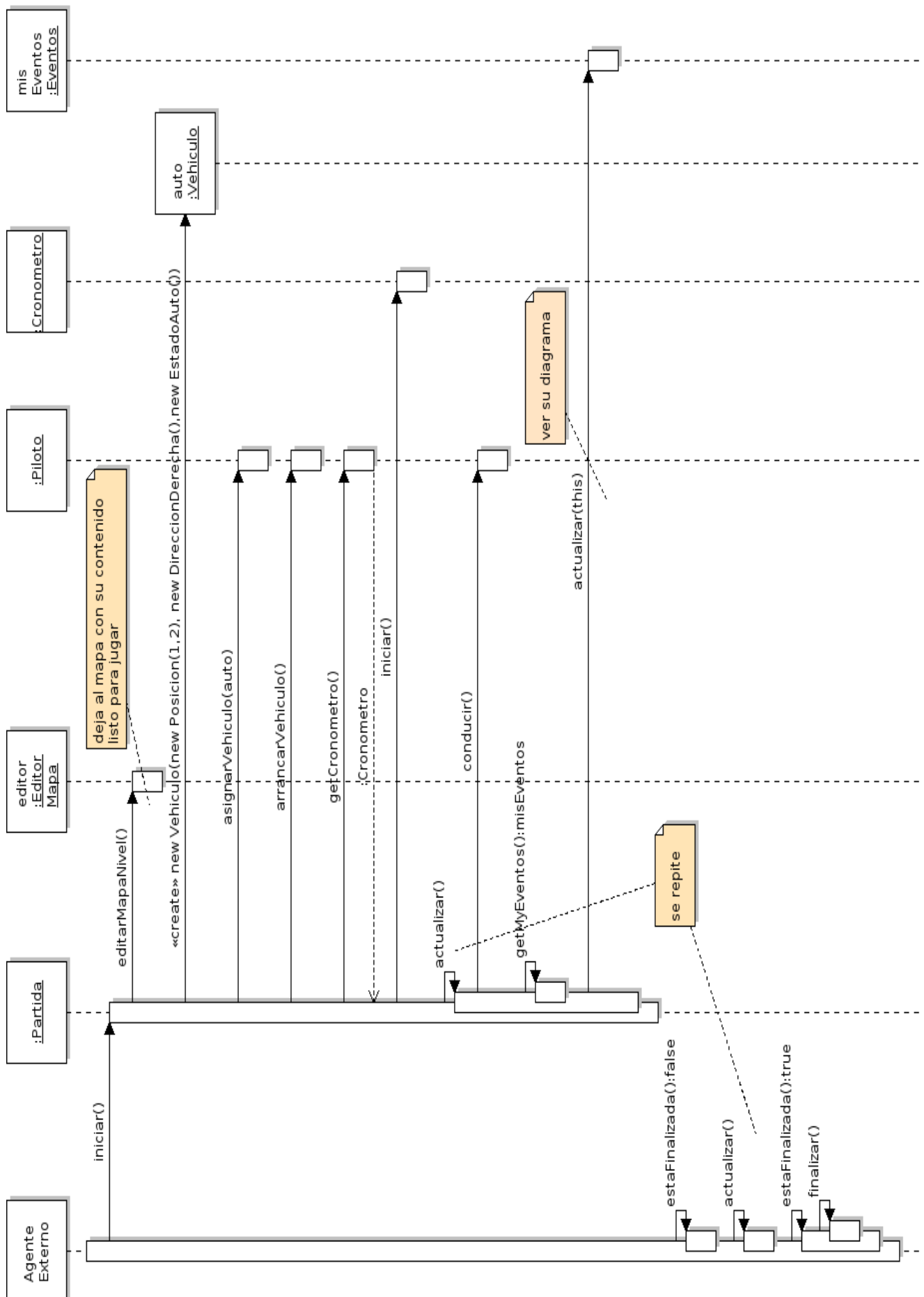


Diagrama de secuencia de la actualización de la clase Eventos:

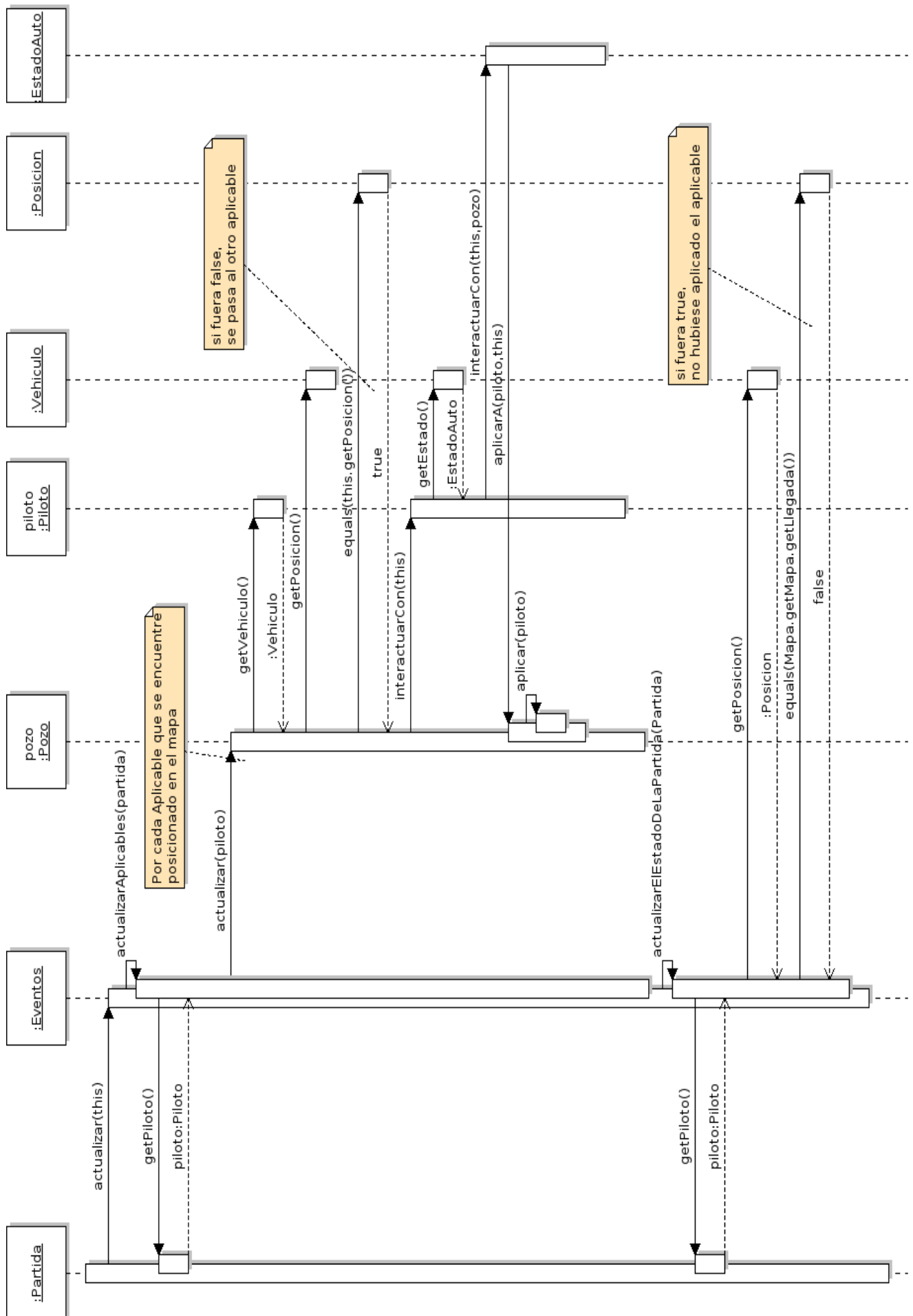


Diagrama de secuencia de un movable en una posición que no existe:

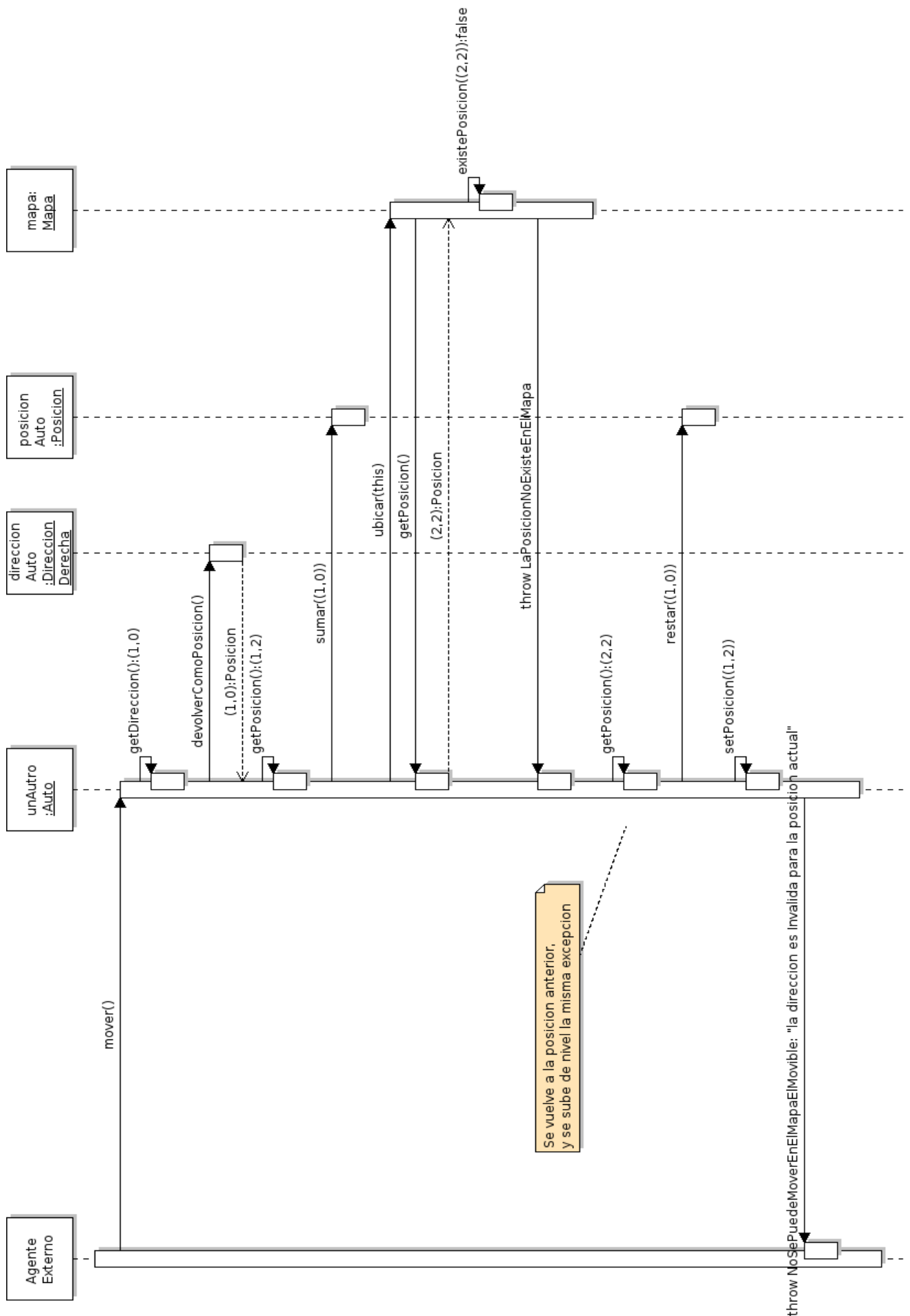


Diagrama de secuencia de un movable en un mapa:

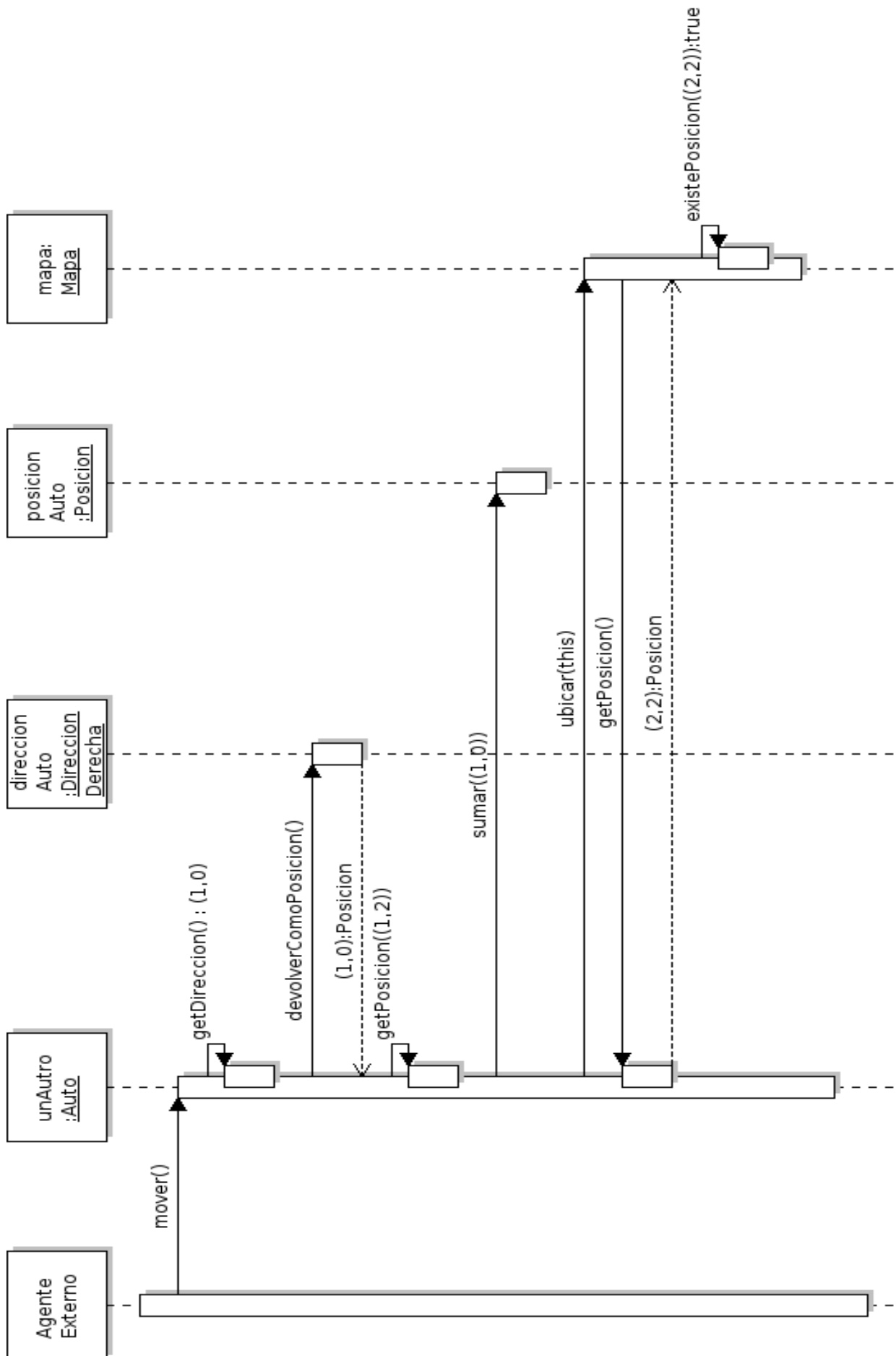


Diagrama de secuencia de Posicionable colocado en un Mapa:

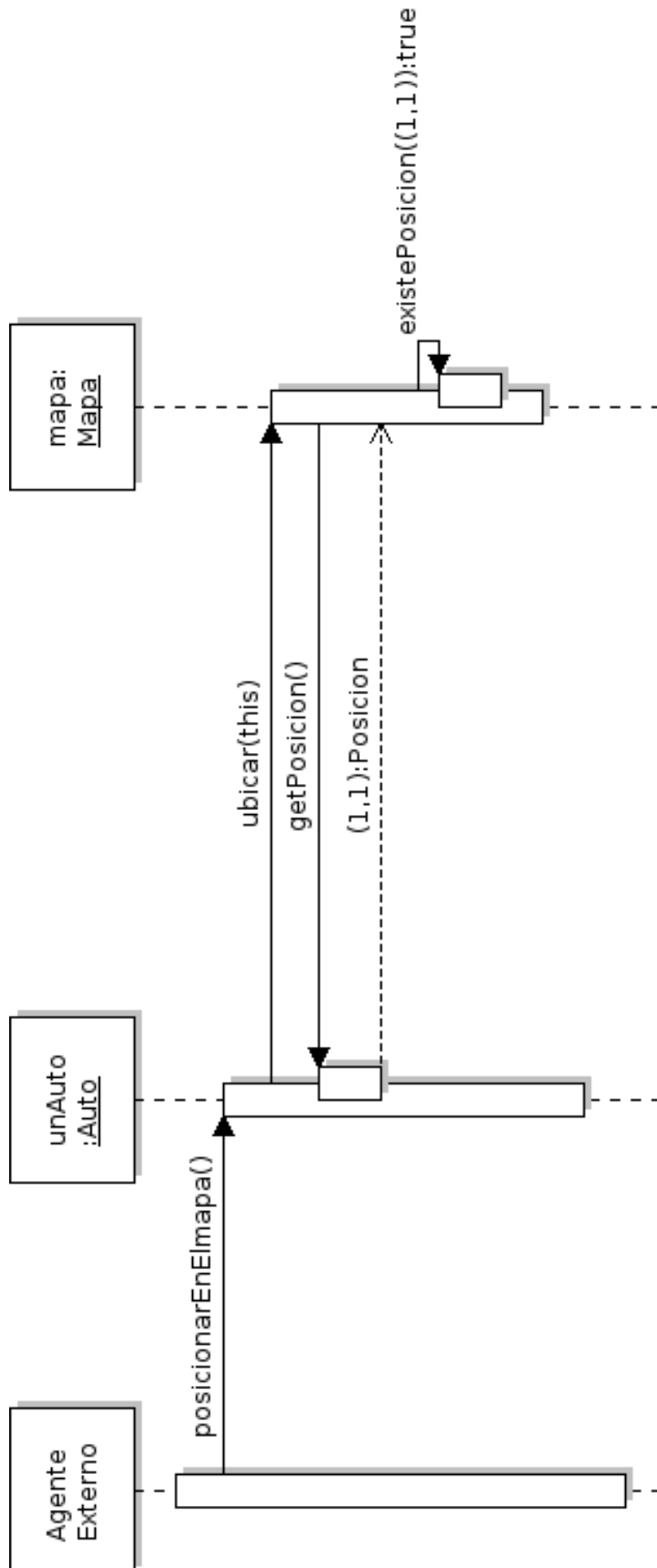


Diagrama de secuencia ubicar un posicionable que ya existe en un mapa:

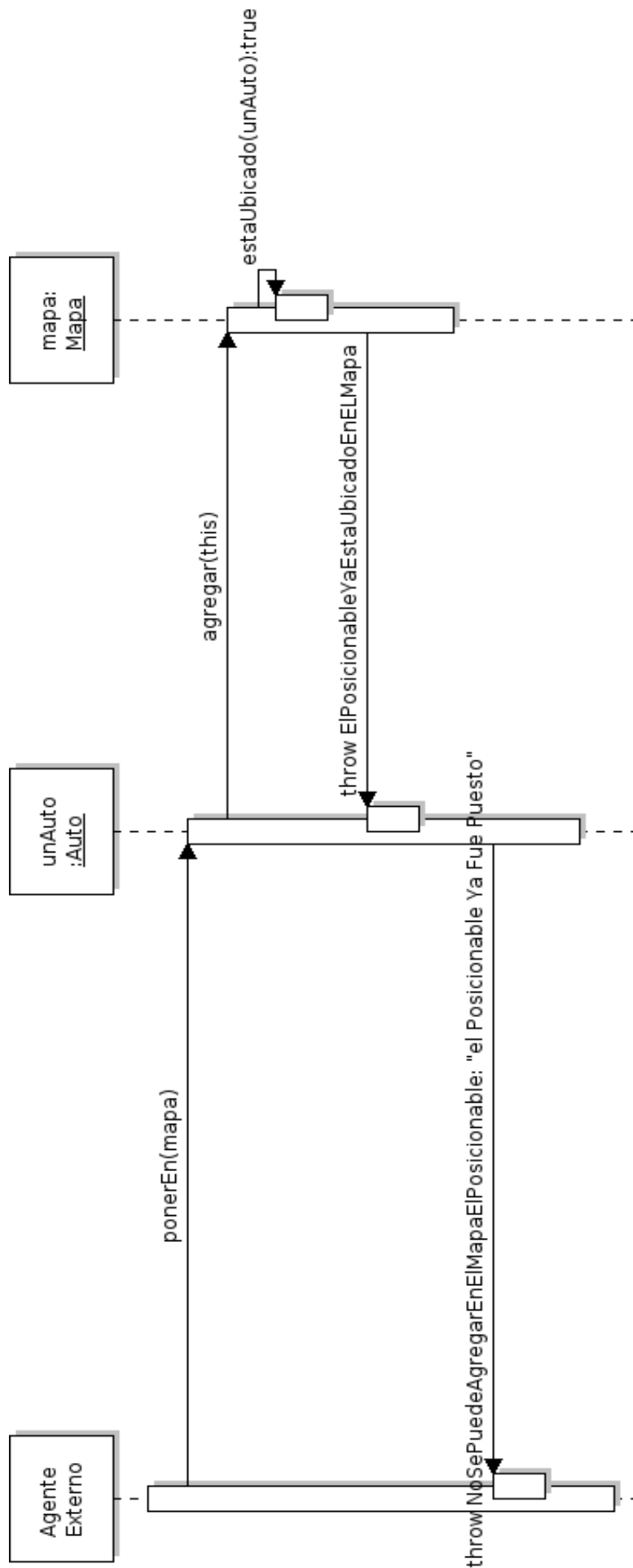
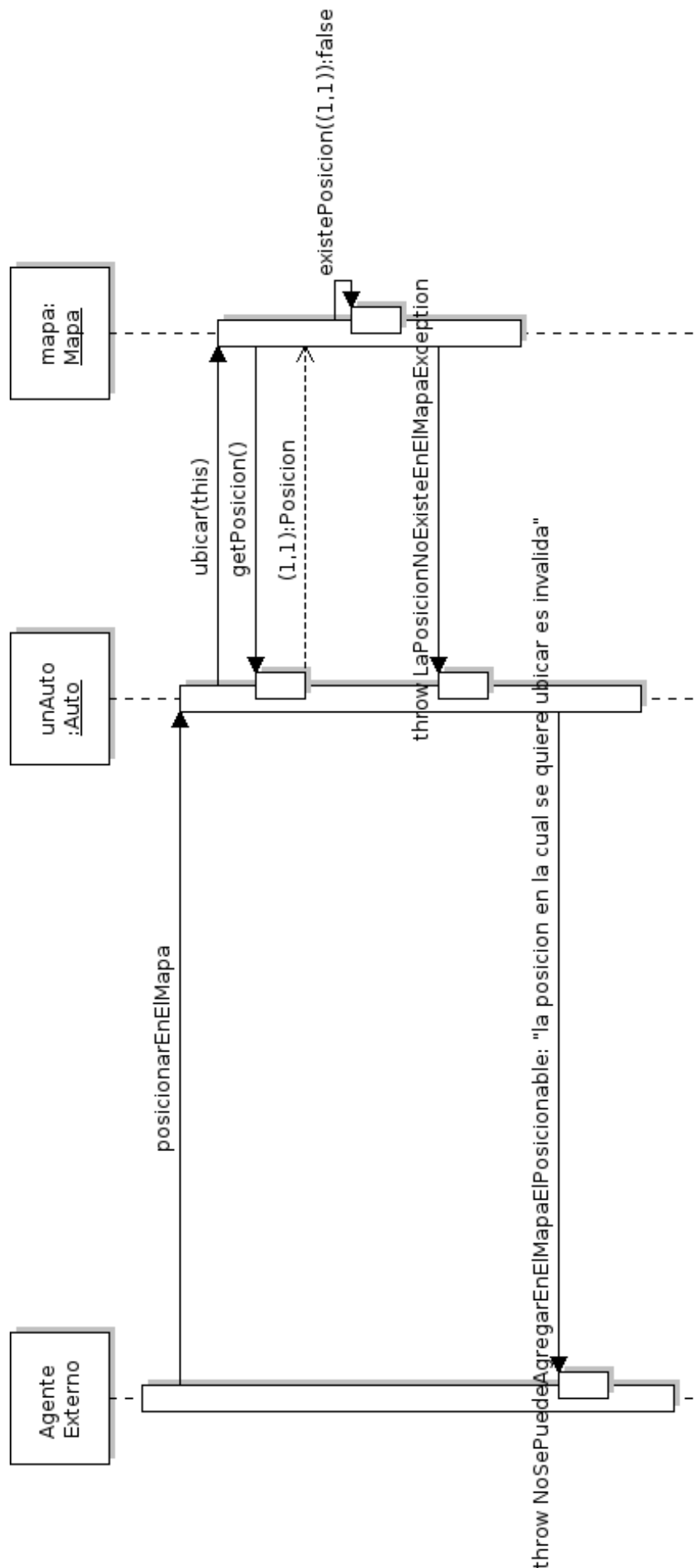


Diagrama de secuencia ubicar un posicionable en posición inexistente:



Checklist de corrección

Esta sección es para uso exclusivo de la cátedra, por favor no modificar.

Carpeta

Generalidades

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

Modelo

- ¿Está completo? ¿Contempla la totalidad del problema?
- ¿Respeta encapsulamiento?
- ¿Hace un buen uso de excepciones?
- ¿Utiliza polimorfismo en las situaciones esperadas?

Diagramas

Diagrama de clases

- ¿Está completo?
- ¿Está bien utilizada la notación?

Diagramas de secuencia

- ¿Está completo?
- ¿Es consistente con el diagrama de clases?
- ¿Está bien utilizada la notación?

Diagrama de estados

- ¿Está completo?
- ¿Está bien utilizada la notación?

Código

Generalidades

- ¿Respeta estándares de codificación?
- ¿Está correctamente documentado?