

Trabajo Práctico AlgoThief

[7507/9502] Algoritmos y Programación III

Curso 2

Segundo cuatrimestre de 2021

104229	PONT, Maria Fernanda
106171	DAVICO, Mauricio
-	POMI, Franco
Github:	https://github.com/mdavic0/algo3tp2

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	3
5. Diagramas de secuencia	5
6. Diagramas de secuencia	7
7. Detalles de implementación	8
7.1. Policia	8
7.2. Ladron	8
7.3. Robo	8
8. Excepciones	8

1. Introducción

El presente informe reúne la documentación acerca del trabajo práctico AlgoThief del 2do cuatrimestre de Algoritmos III de 2021.

2. Supuestos

1. A menos que se indique lo contrario se obedecen las reglas definidas en el documento AlgoThief definiendo la consigna. Faltando especificación en el documento, se da preferencia a la funcionalidad del juego de Carmen Sandiego para MSDOS.

2. El jugador es un policía que recorre distintos países y entra a sus edificios con el objetivo de obtener pistas y seguir el rastro de un ladrón que ha cometido el robo de un artefacto o tesoro valioso.

3. Los países están conectados entre sí de tal manera que solo se puede viajar a ciertos países a la vez. En cada país el jugador es presentado con dos países incorrectos donde no estuvo el ladrón y uno correcto donde sí estuvo. La excepción son aquellos países donde no estuvo el ladrón, en tal caso el ladrón sólo puede regresar al país en donde ya estuvo. También se permite tener menos de tres opciones en el país donde se encuentra el ladrón.

4. El recorrido por los países del juego consiste en una secuencia de países 'correctos' por los que pasó el ladrón. Sólo se puede llegar a un país en dicha secuencia desde el país que lo precede en la secuencia.

5. En cada país hay una serie de edificios dentro de los cuales se provee un texto que provee 'pistas' sobre la identidad del ladrón y el país al que fue posteriormente al actual. En caso de estar en un país donde nunca estuvo el ladrón, se le notificará al usuario con un mensaje genérico.

6. El policía necesita una orden de arresto para poder arrestar al ladrón al encontrarlo. Puede conseguir esta orden de arresto durante el juego al consultar con una computadora, donde puede ingresar las características conocidas del ladrón. Si las características reducen la lista de sospechosos a una persona, se emitirá una orden de arresto para ella. No necesariamente una orden de arresto emitida pertenece al verdadero ladrón - se debe deducir éste a partir de las pistas obtenidas en los edificios.

7. El tiempo in-game se consume únicamente con actividades discretas como entrar a un edificio, viajar, y exitosamente emitir una orden de arresto.

8. El jugador pierde el juego si alcanza al ladrón pero no tiene una orden de arresto para arrestarlo. También perderá si se le acaba el tiempo. Sólo ganará si al momento de entrar al edificio donde se encuentra el ladrón, ya tiene una orden de arresto para él.

9. Por propósitos de simplicidad y tiempo, cualquier artefacto puede ser robado en cualquier país.

3. Modelo de dominio

El modelo consiste de múltiples clases, de las cuales las más importantes son Policía, Robo y EstadoDeJuego. El Policía es controlado por el jugador y realiza todas las actividades que podría ordenar el jugador. El robo indica toda la información correspondiente al robo que ocurrió en la partida actual, incluyendo el Ladrón y la vía que tomó, y es responsable de conectar todos los Países del modelo entre sí, definiendo cómo se puede navegar el mundo en el juego. El EstadoDeJuego es suscrito a todas las clases que podrían desencadenar un fin de juego, sea por pérdida o victoria, y los usuarios del modelo pueden a su vez ser notificados de dichos eventos.

El modelo toma como parámetro al ser inicializado entre otras cosas todos los países posibles, ladrones, y artefactos a ser robados. (Es responsabilidad de las clases Controlador de crear estas clases a partir de los archivos JSON en la carpeta de recursos.) De esta manera el modelo es independiente de cualquier proceso de lectura de archivos.

Otra clase importante es Pista, cuyas subclases sirven para parsear la sintaxis de distintos tipos de pistas (PistaDePresidente, PistaDeVehiculo etc). Además permite combinar muchas pistas mediante la clase PistaCompuesta.

Finalmente las clases Computadora y Orden de arresto colaboran con el policía para emitir órdenes de arresto y que el policía pueda validar el arresto de un ladrón con ellas.

El modelo hace uso del patrón de estado en clases como `LugarActual`, usado para representar las distintas funcionalidades disponibles al policía dependiendo de si está dentro o fuera de un edificio. También tiene un `Rango`, que es actualizado dependiendo de la cantidad de arrestos del policía.

4. Diagramas de clase

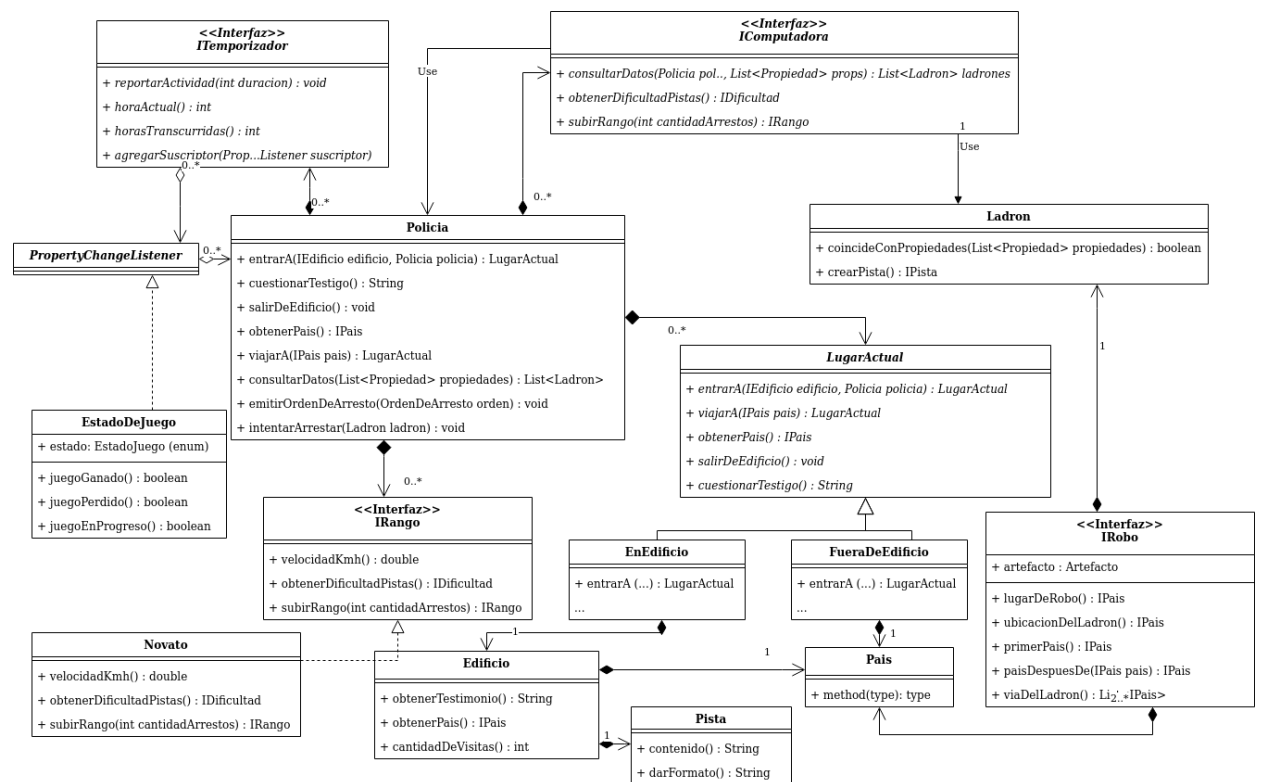


Figura 1: Diagrama global del modelo.

5. Diagramas de secuencia

Diagrama de estados de Rango

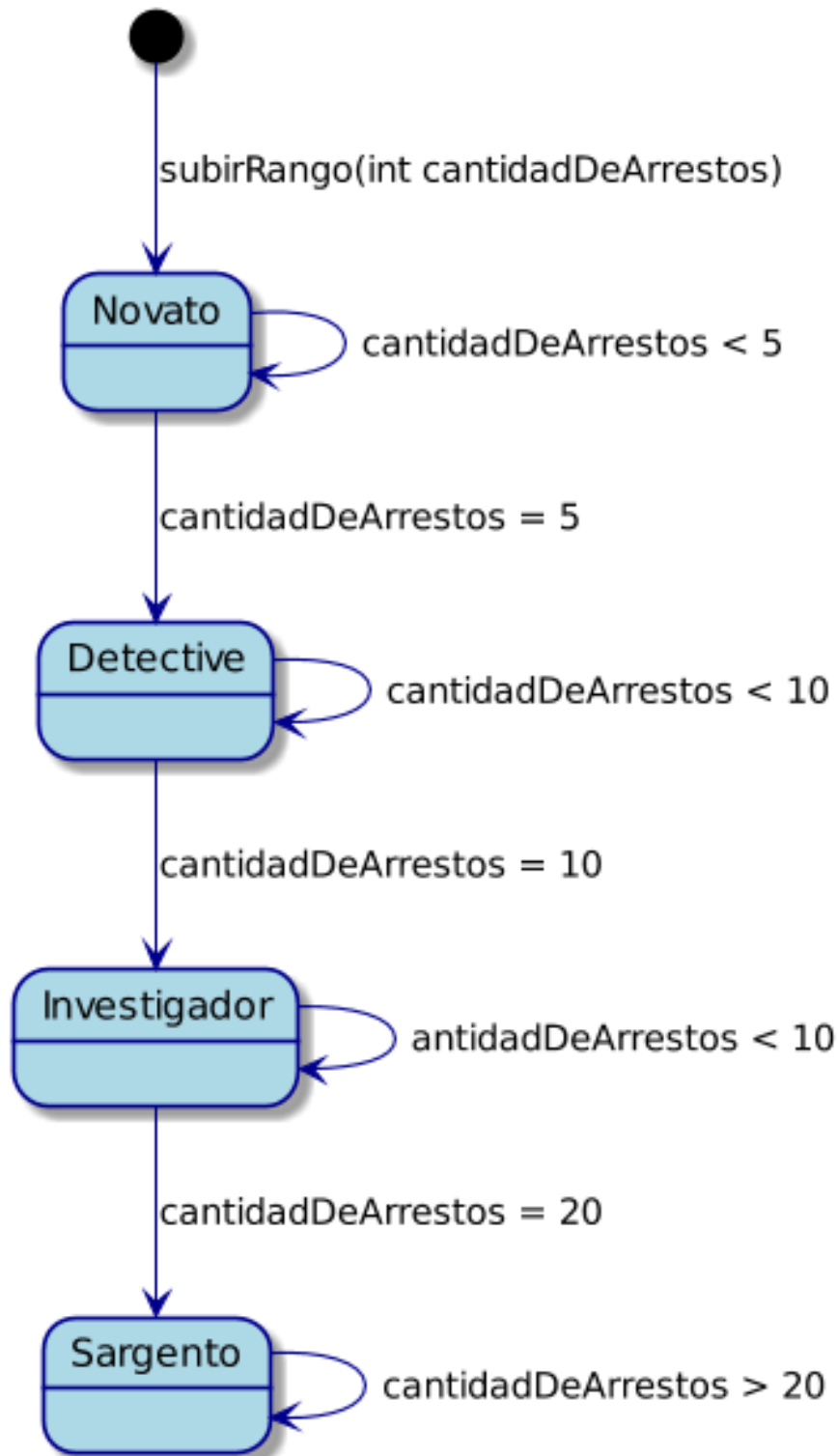


Figura 2: Diagrama de estado del rango del policía.

Diagrama de estado de Lugar Actual del Policia



Figura 3: Diagrama de estado del rango del policía.

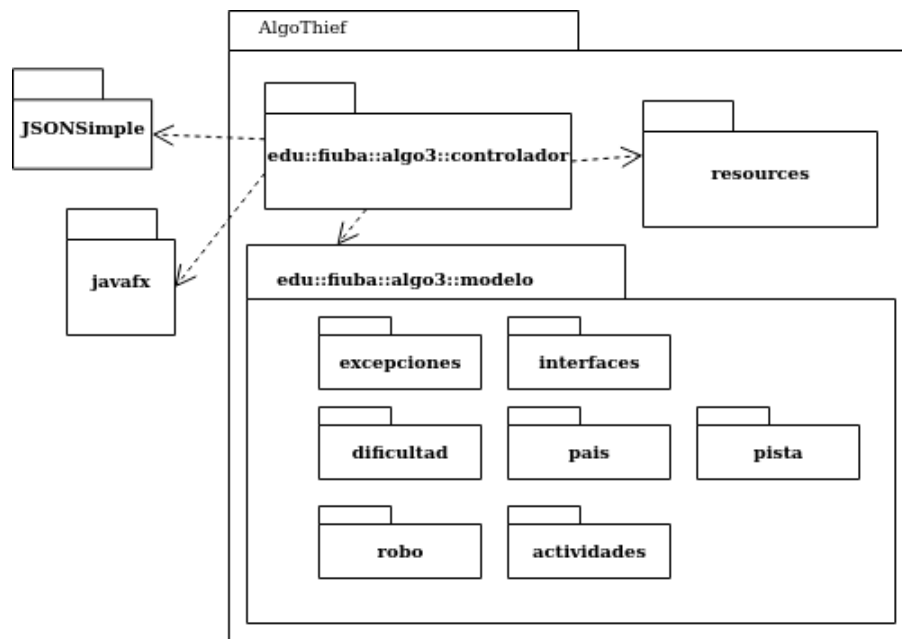


Figura 4: Diagrama de paquetes.

6. Diagramas de secuencia

El detective entra al edificio donde está el ladrón

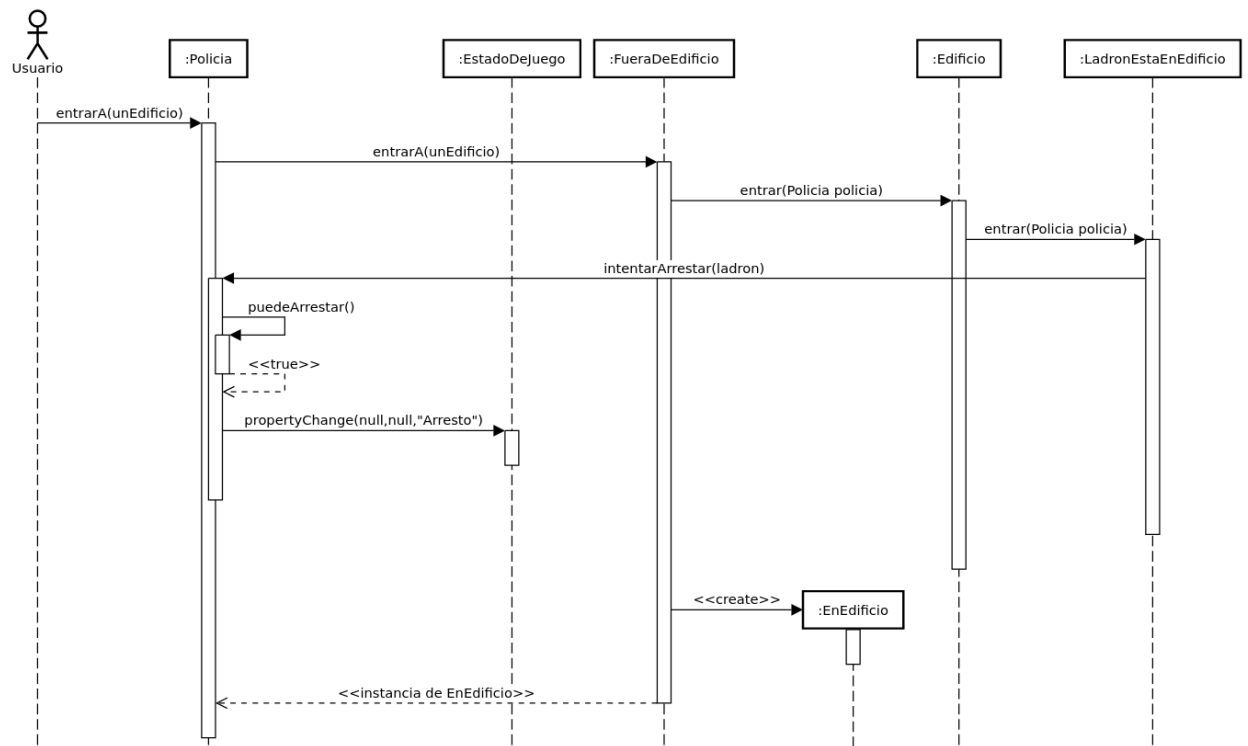


Figura 5: Diagrama de secuencia para entrar a un edificio donde está el ladrón.

7. Detalles de implementación

7.1. Policia

El policia delega funcionalidad como subir de rango, consultar a la computadora y realizar acciones dependientes de su ubicación, respectivamente a las clases Rango, Computadora y LugarActual.

LugarActual es una clase que representa el estado del policía, sea dentro o fuera de un edificio. Debido a que un policía puede persistir aunque no haya una partida en curso, el lugar actual del policía puede no ser Inactivo, indicando que no está en ningún lugar porque no hay una partida en curso.

7.2. Ladron

El ladrón tiene un campo para cada una de sus propiedades incluyendo su nombre, pero además guarda sus propiedades en un formato genérico como una lista de instancias de la clase Propiedad. Esto permite con más facilidad determinar si una lista de propiedades arbitraria descarta o no a un ladrón dado. Por ejemplo, si Carmen Sandiego tiene una moto, tiene cabello castaño, lleva un collar de rubíes, y es mujer, el conjunto de propiedades "Género: Femenino" "vehículo: Moto" no la excluyen, pero las propiedades "vehículo: Moto" "Género: Masculino" sí. Estos pares de 'tipo de propiedad' y 'propiedad' son representados por la clase propiedad, y una coincidencia de propiedad indica que el tipo y contenido de la propiedad son los mismos.

7.3. Robo

El robo es el ente alrededor del cual está diseñada una partida, incluyendo las conexiones entre países por las que un policía puede viajar. También debería ser responsable de inicializar todos los países con sus edificios dependiendo si estuvo, no estuvo, o está el ladrón en ellos, lo hace colaborando con el GeneradorDeEdificios.

8. Excepciones

AlgoThiefException es la excepción general de la que descende el resto.

PoliciaNoInicializadoException es la excepción que ocurre cuando el policía trata de hacer algo que requiere que el esté en un país o edificio, es decir, que tenga asignado un caso.

AccionInvalidaException ocurre cuando el policía trata de hacer algo que no puede hacer si está en un edificio, o viceversa.