FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico — Java
Algo-thief

[7507/9502] Algoritmos y Programación III
Curso 1
Segundo cuatrimestre del 2021

Alumno:	CODINA, CAMILA
Número de padrón:	106339
Email:	ccodina@fi.uba.ar
Alumno:	GRÜNER, TOMÁS
Número de padrón:	105798
Email:	tgruner@fi.uba.ar
Alumno:	LOURENGO CARIDADE, LUCÍA
Número de padrón:	104880
Email:	llourengo@fi.uba.ar
Alumno:	TOULOUSE, ALAN
Número de padrón:	105343
Email:	atoulouse@fi.uba.ar
Alumno:	VACCARELLI, SANTIAGO
Número de padrón:	106051
Email:	svaccarelli@fi.uba.ar

Índice

1. Introducción

El presente informe reune la documentación del segundo trabajo práctico de la materia Algoritmos y Programación III. Consiste en desarrollar una aplicación de manera grupal aplicando todos los conceptos vistos en el curso, utilizando un lenguaje de tipado estático (Java), con un diseño del paradigma de programación orientada a objetos y aplicando las técnicas de TDD e Integración Continua.

2. Supuestos

1. Las pistas dadas a los policías son: fáciles para los rangos Novato y Detective, medias para el rango Investigador, y difícil para el rango Sargento.
2. La rareza del objeto robado por el ladrón depende directamente de la cantidad de arrestos, y no solo del rango del detective. Más particularmente, 0-6 arrestos corresponde a valor común, 7-13 a valioso, y 14 en adelante a muy valioso.
3. El objeto robado (y la ciudad inicial), la ruta de escape del ladrón, y los edificios de las ciudades son elegidos aleatoriamente con igual probabilidad de aparecer cada uno.

3. Modelo de dominio

4. Diagramas de clase

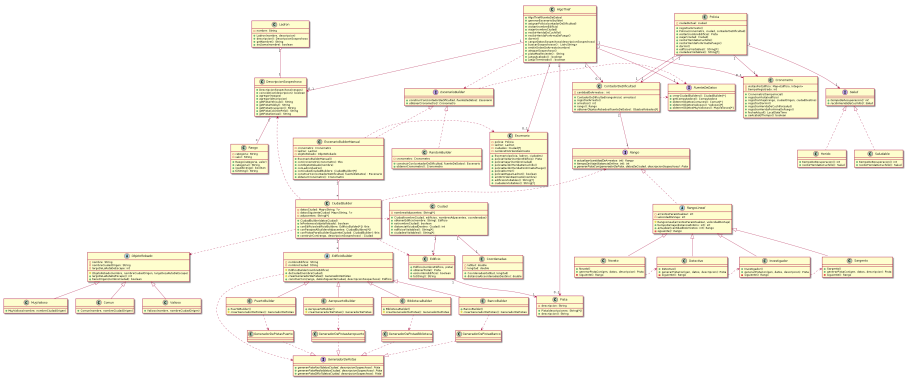

Figura 1: Diagrama de clase general.

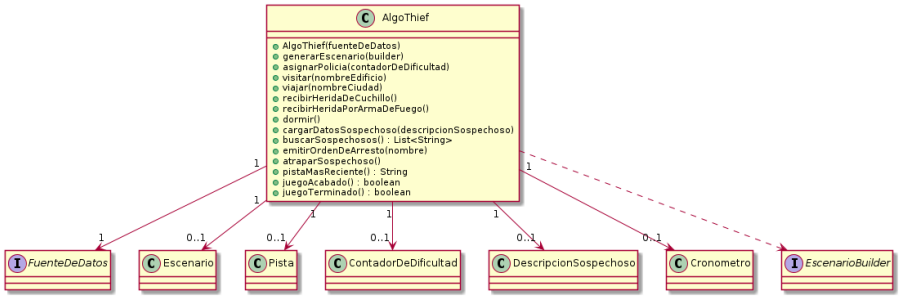

Figura 2: Diagrama de clase de algothief.

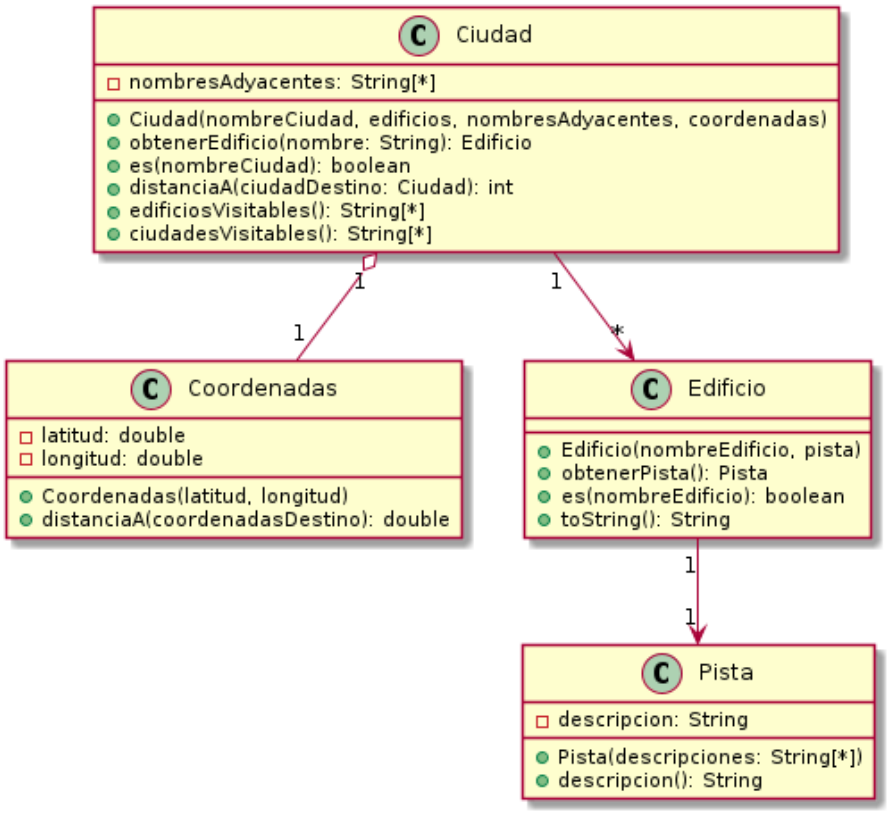

Figura 3: Diagrama de clase ciudad.

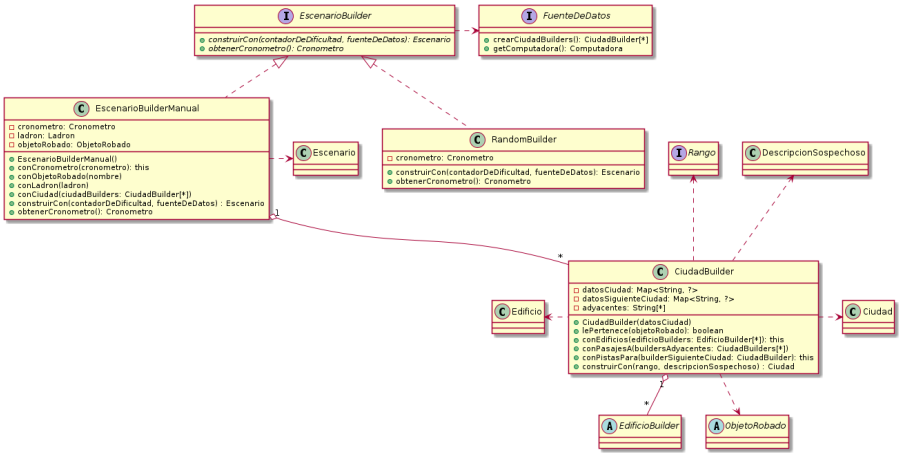

Figura 4: Diagrama de clase de Builder.

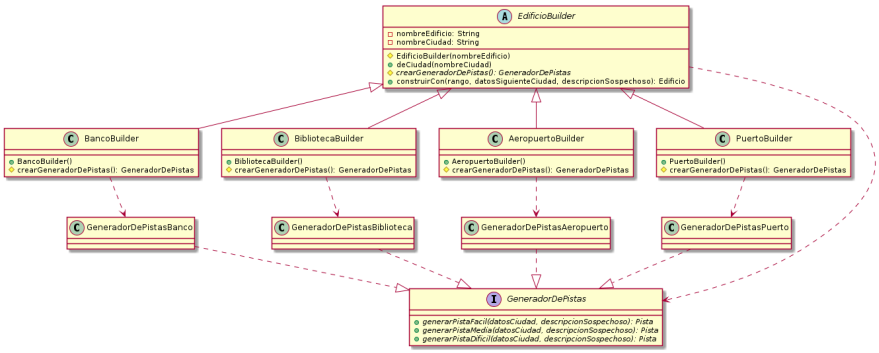

Figura 5: Diagrama de clase de Edificio Builder.

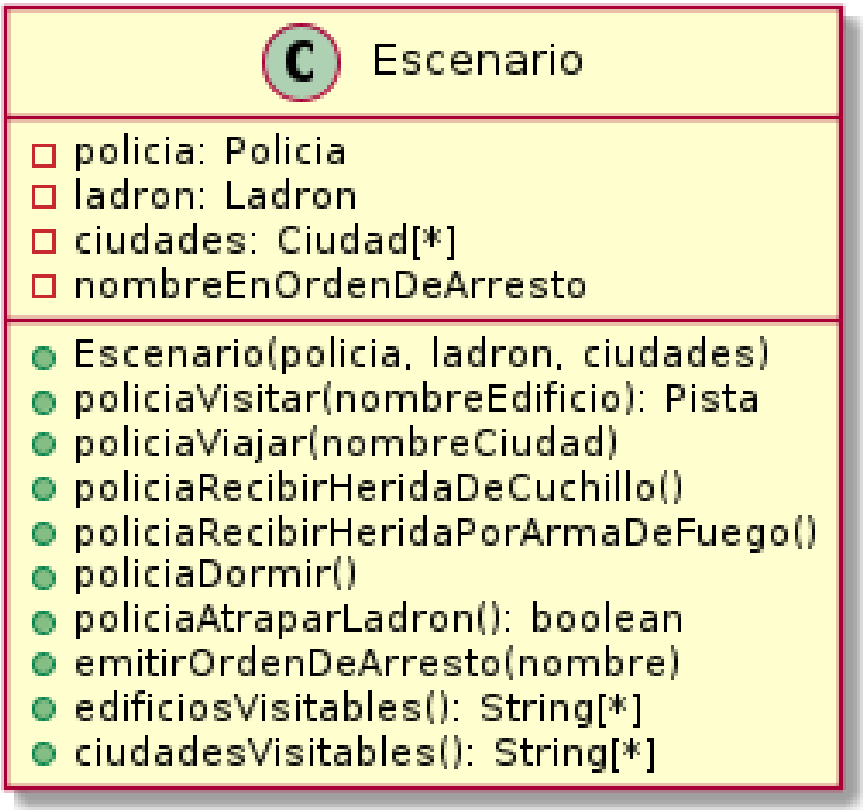

Figura 6: Diagrama de clase de Escenario.

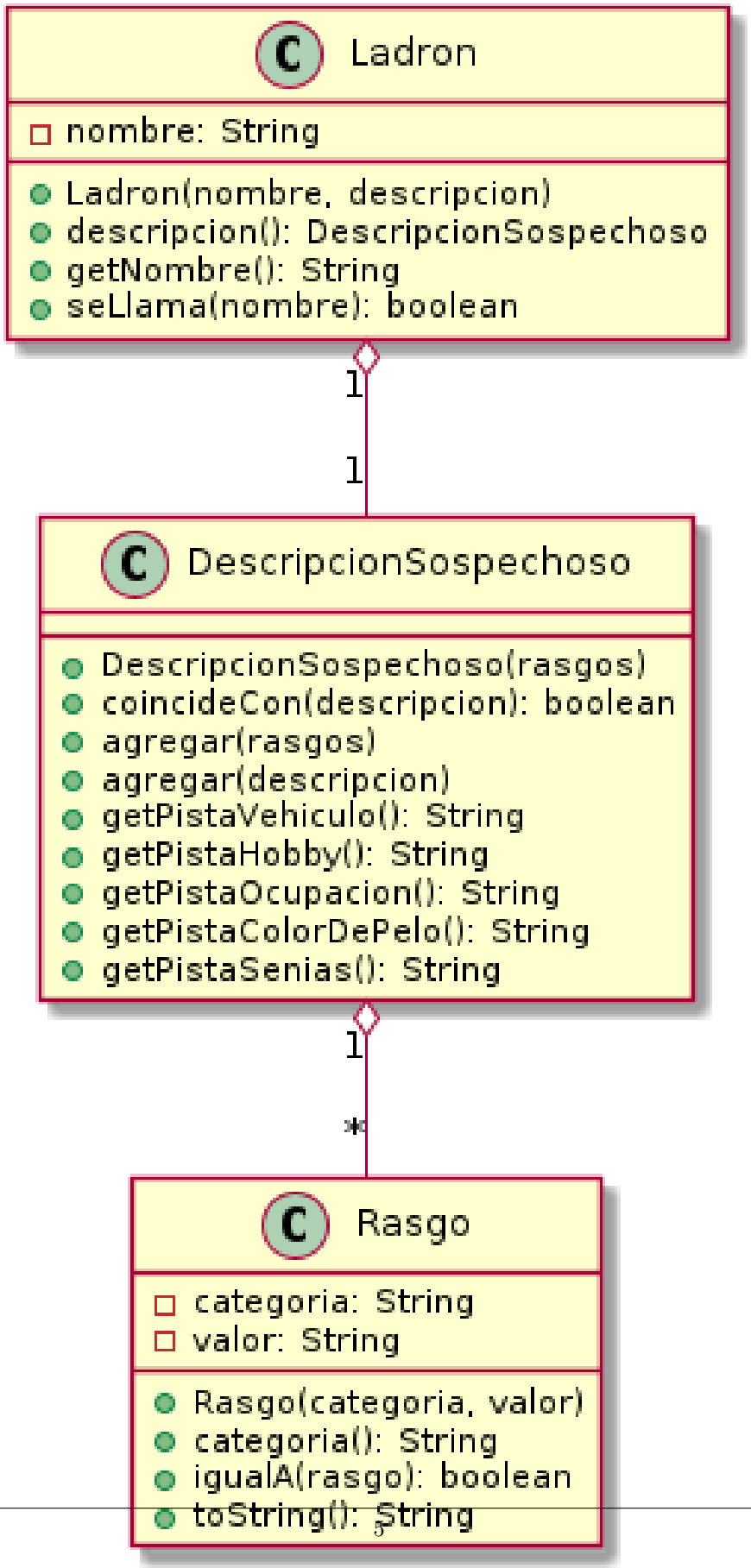

Figura 7: Diagrama de clase de Ladron.

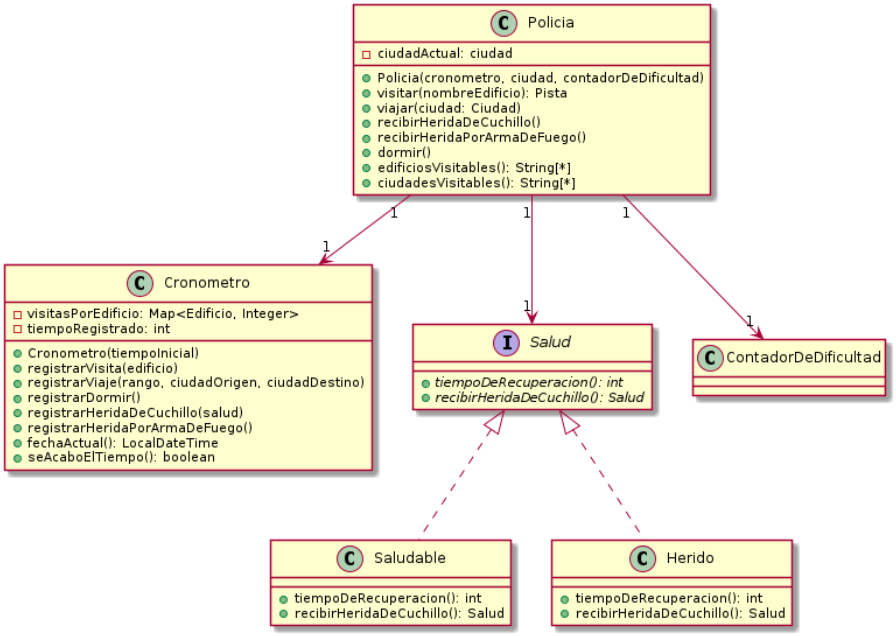

Figura 8: Diagrama de clase de Policia.

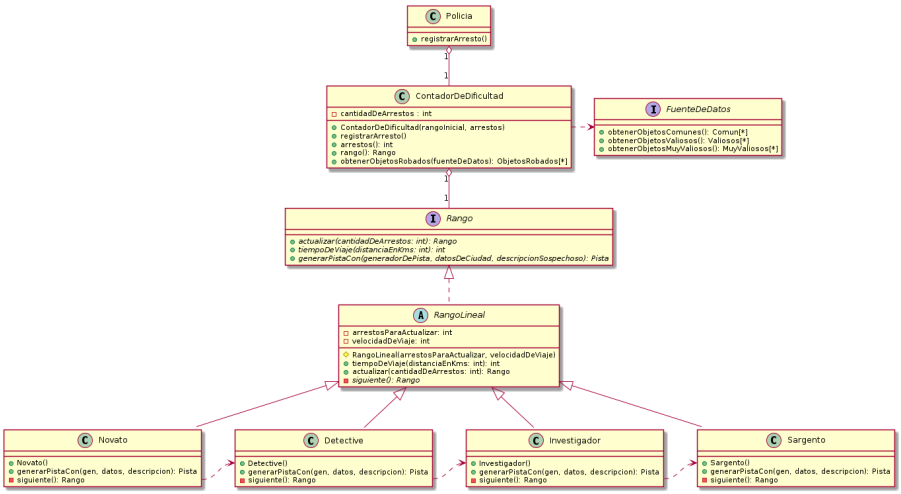

Figura 9: Diagrama de clase de Rango.

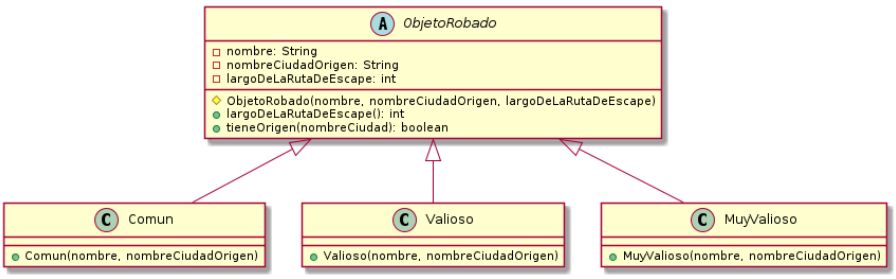

Figura 10: Diagrama de clase de ObjetoRobado.

5. Detalles de implementación

Con el fin de beneficiar el uso de las mejores prácticas durante la creación de nuestro proyecto, tuvimos en cuenta la posibilidad de implementar patrones de diseño para determinados casos que logramos identificar.

1. Utilizamos el patrón 'Builder', un patrón de diseño creacional para construir el escenario inicial de nuestro juego. Éste nos permite separar la construcción del Escenario, Ciudades y Edificios en distintas llamadas. Esto nos permite separar los parámetros del constructor en varios mensajes, y evitar enviar parámetros que no son necesarios. De esta forma cumplimos con los principios de 'Single Responsibility', al mover la lógica de creación de los objetos a una clase aparte.

2. También recurrimos al patrón 'Chain of Responsibility', un patrón de comportamiento que nos permite hacer una cadena de solicitudes: al recibir una solicitud, un manejador decide si debe procesarla o redirigirla al siguiente manejador de la cadena. El comportamiento de Rango≫actualizar lo utiliza.

3. A su vez, utilizamos el patrón 'State' para el manejo del rango actual del detective en la clase ContadorDeDificultad, donde se actualiza el objeto Rango (según lo explicado en el ítem anterior), y se le delega el comportamiento como estado actual. Junto con el ítem anterior, nos permite cumplir los principios de 'Single Responsibility' y 'Open-Closed'.

6. Excepciones

7. Diagramas de secuencia

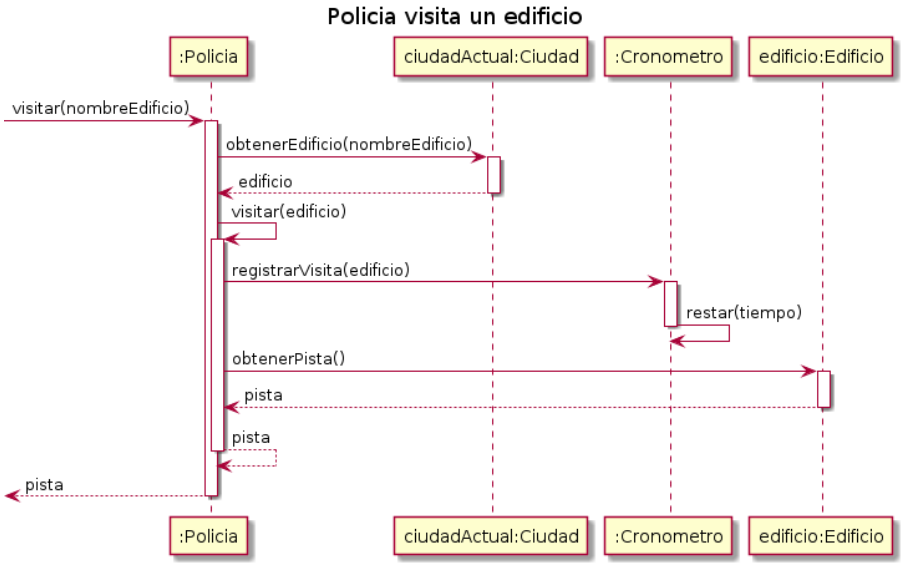

Figura 11: Diagrama de secuencia Entrega Cero.

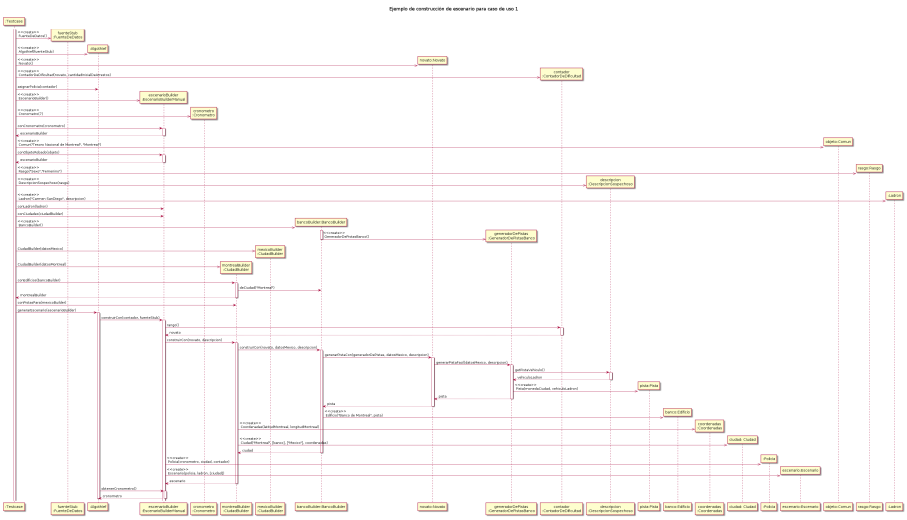

Figura 12: Diagrama de secuencia de la creación del Escenario.

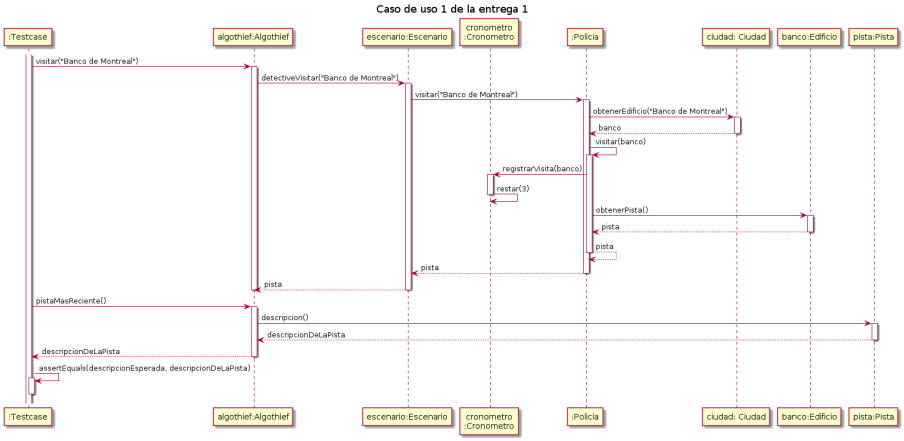

Figura 13: Diagrama de secuencia Entrega 1 TestCase01.

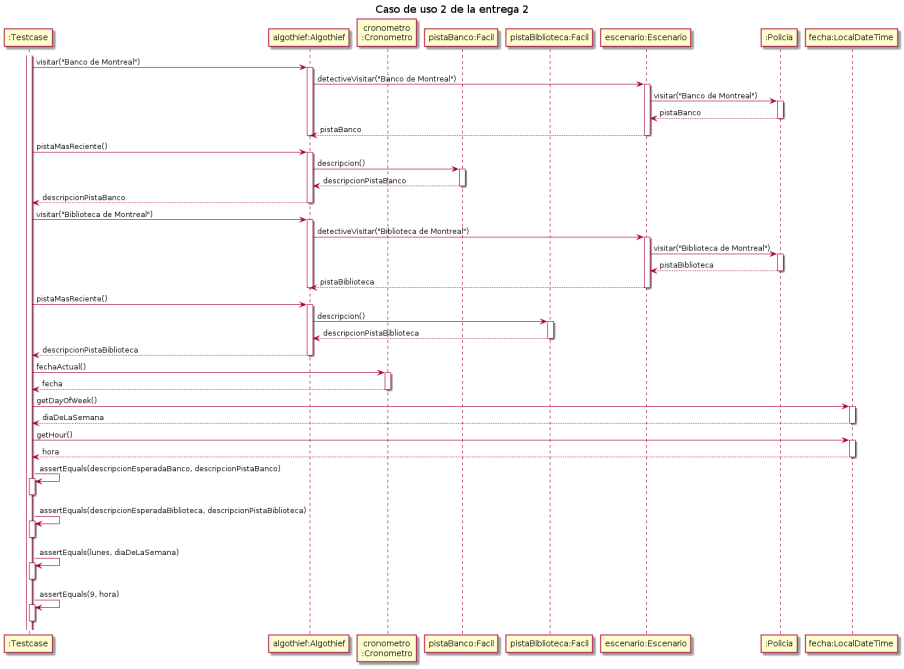

Figura 14: Diagrama de secuencia Entrega 1 TestCase02.

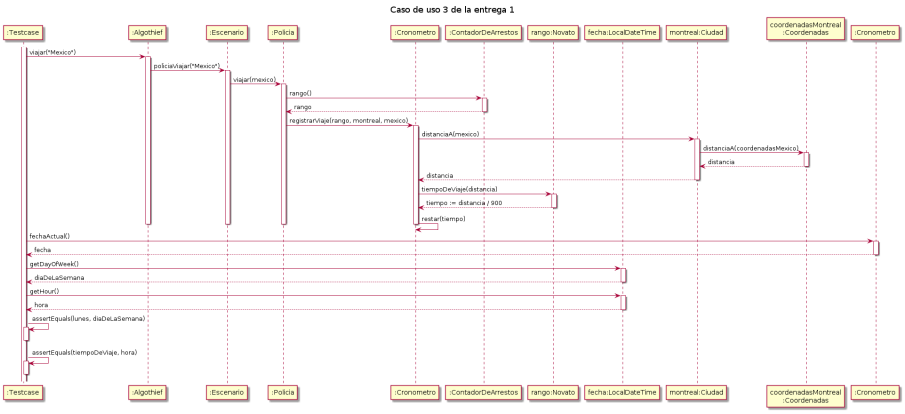

Figura 15: Diagrama de secuencia Entrega 1 TestCase03.

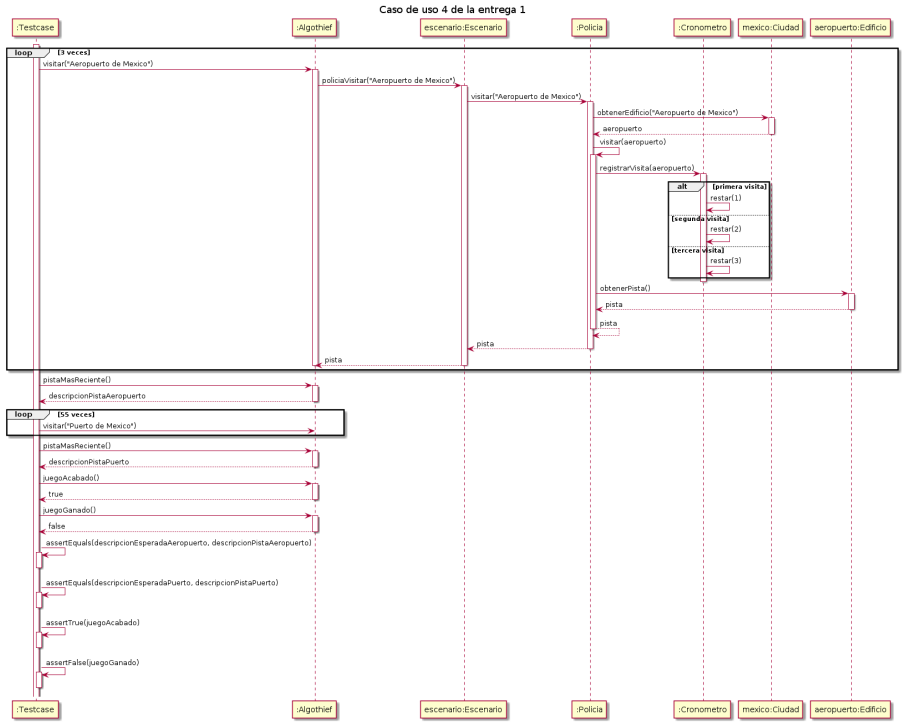

Figura 16: Diagrama de secuencia Entrega 1 TestCase04.

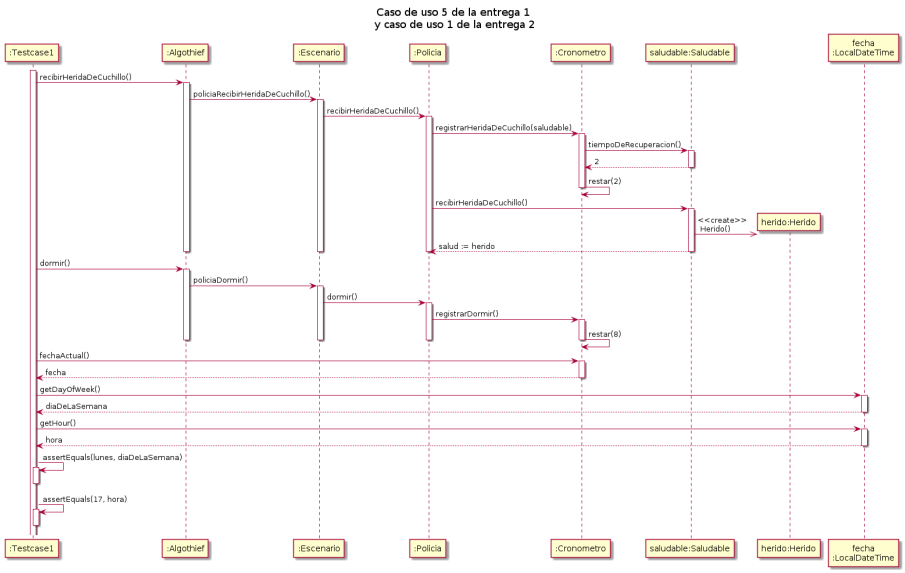

Figura 17: Diagrama de secuencia Entrega 1 TestCase05 y Entrega 2 TestCase01

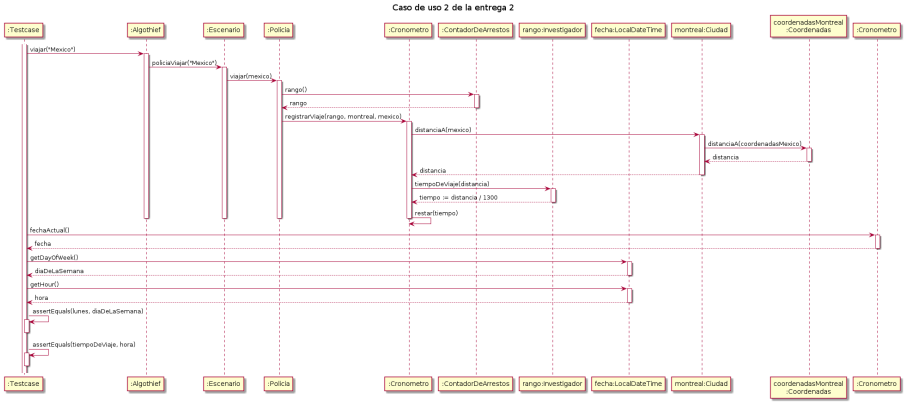

Figura 18: Diagrama de secuencia Entrega 2 TestCase02.

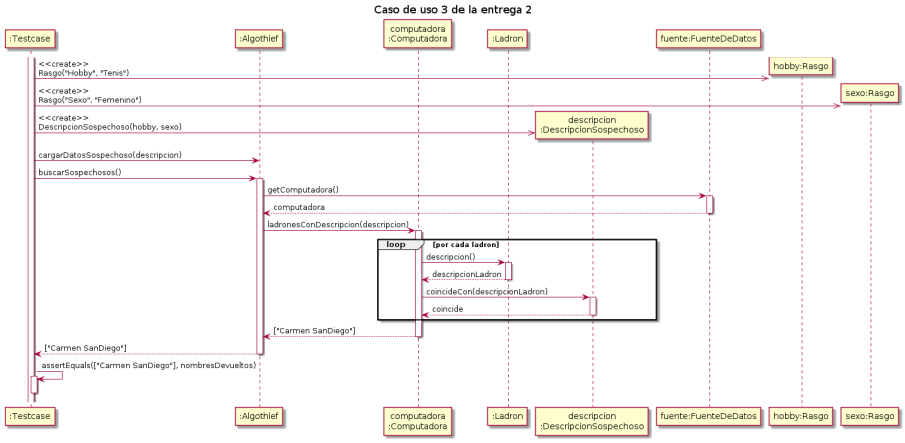

Figura 19: Diagrama de secuencia Entrega 2 TestCase03.

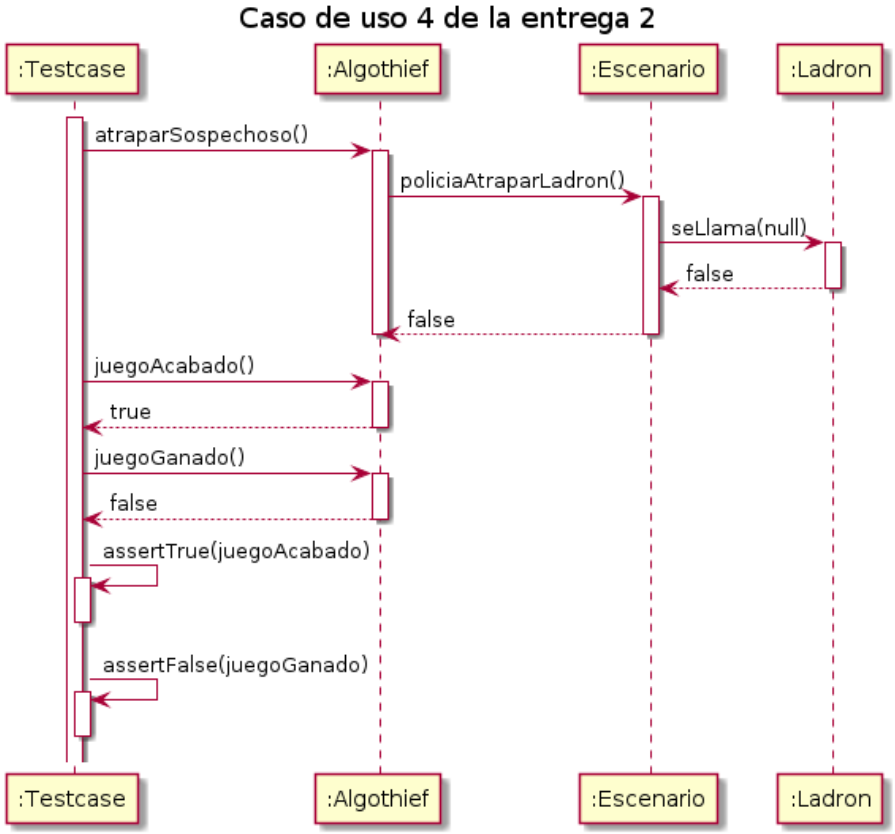

Figura 20: Diagrama de secuencia Entrega 2 TestCase04.

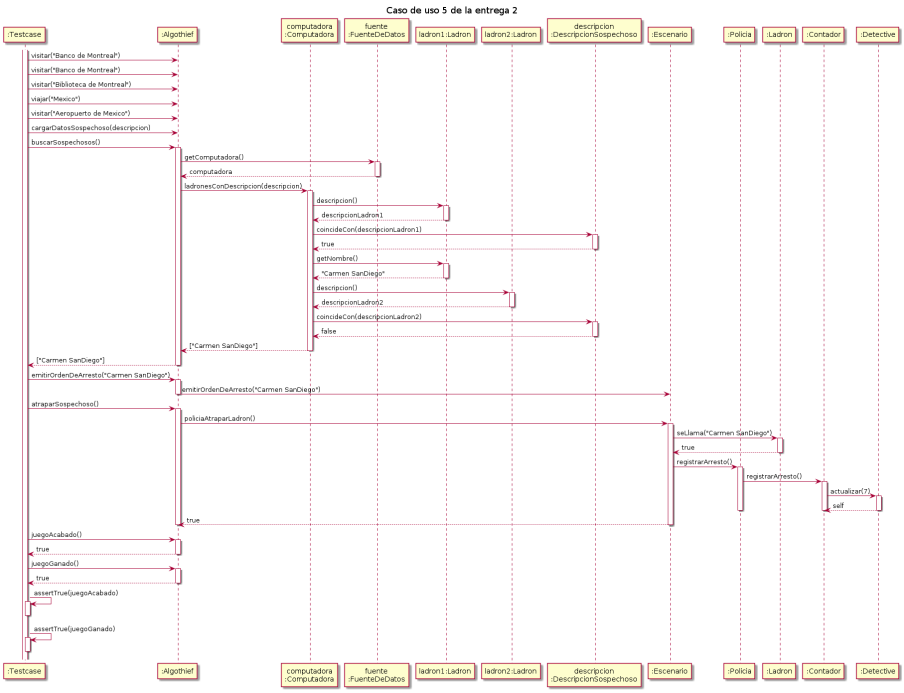

Figura 21: Diagrama de secuencia Entrega 2 TestCase05.

8. Diagramas de estados

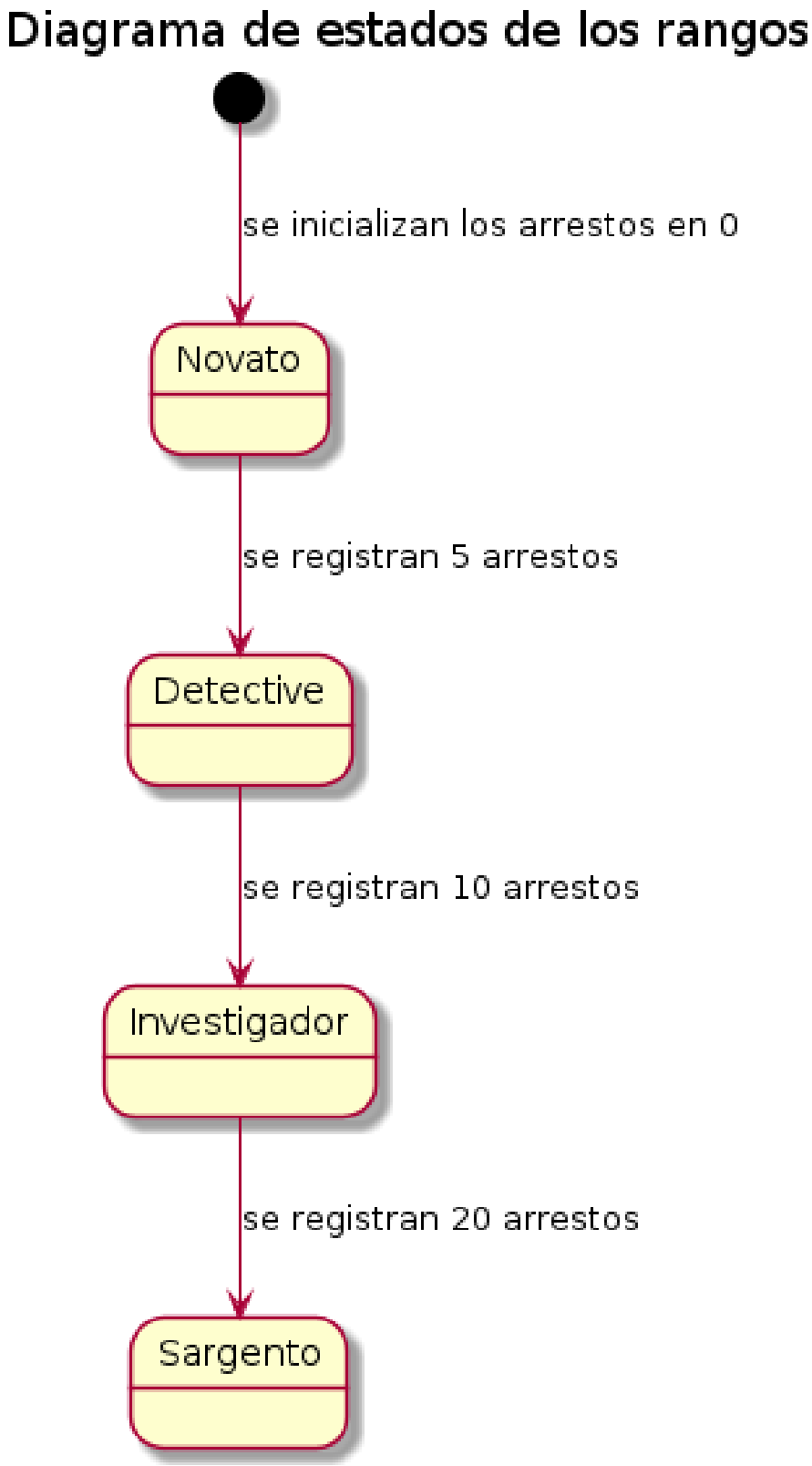

Figura 22: Diagrama de estado de rango.

9. Diagramas de Paquetes

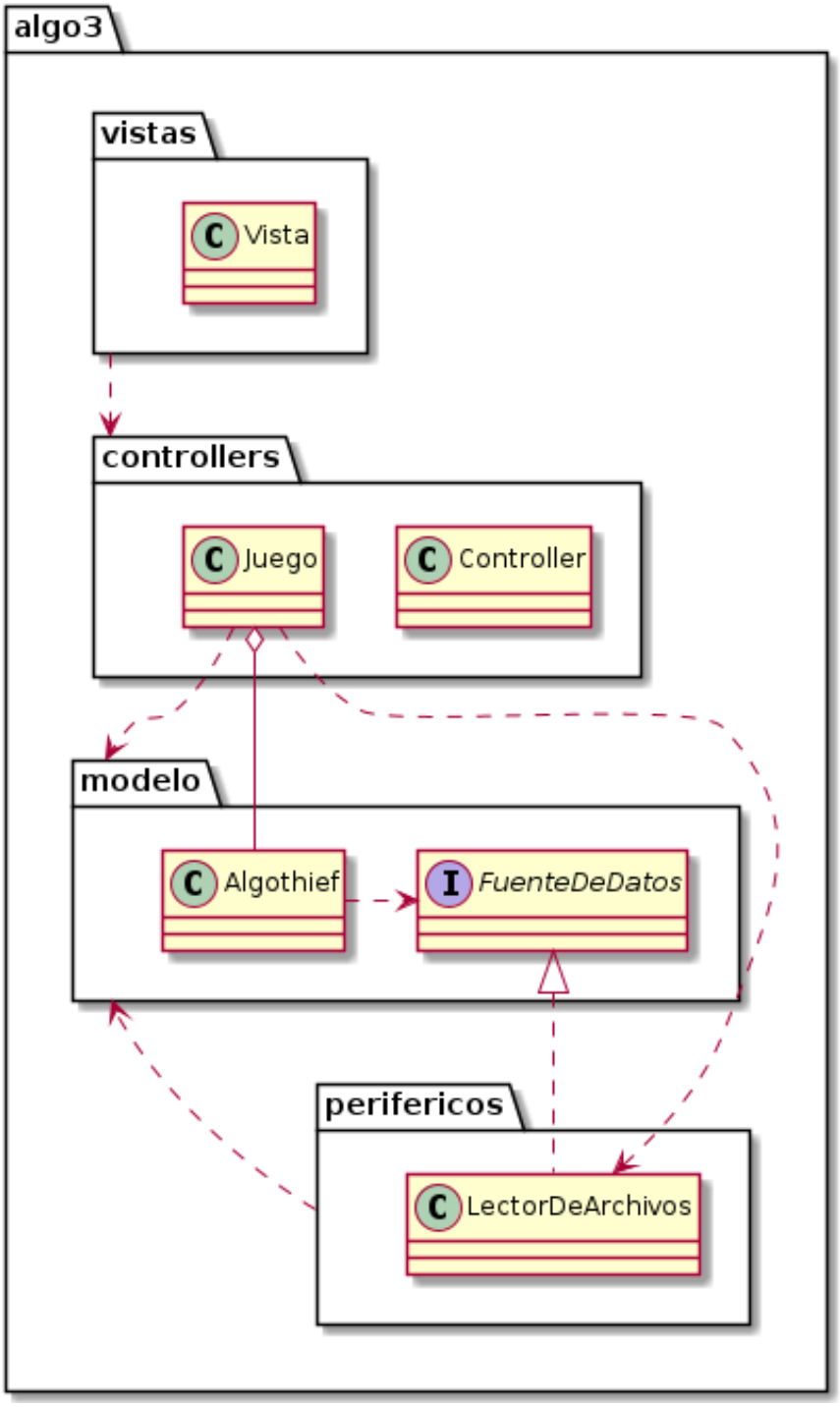

Figura 23: Diagrama de paquetes.