

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III
Curso 1
Segundo cuatrimestre de 2018

Alumnos:	Allo, Tomás
	Ponce, Antonella
	Cugliari, Pablo
	González, Andrés
Números de padrón:	101057
	100635
	100703
	95694
Emails:	tomasmanuelallo@gmail.com
	antonellaponce28@gmail.com
	pacugliari@hotmail.com
	andresgonzalez.fba@gmail.com

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clases	2
4. Diagramas de secuencia	4
5. Diagramas de paquetes	5
6. Diagramas de estado	6
7. Detalles de implementación	6
7.1. Mapa	6
7.2. Juego	6
7.3. Jugador	7
8. Excepciones	7

1. Introducción

El presente informe reúne la documentación de la solución del segundo trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar una aplicación de manera grupal utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso y las técnicas de TDD e Integración Continua en el lenguaje Java.

2. Supuestos

Para este trabajo práctico se tuvo en cuenta que las posiciones del mapa sólo pueden ser ocupadas por una sola entidad. Es decir, en cada posición puede haber una única unidad o una una parte de un edificio. Por lo tanto, ningún elemento del juego comparte posición.

Además, la incorporación de una unidad sólo depende de que el jugador involucrado tenga la cantidad de oro suficiente, no es necesario esperar un determinado tiempo ya que se crea de inmediato.

Teniendo en cuenta funcionalidades para la próxima entrega, se supone que al iniciar el juego los castillos de ambos jugadores se ubicarán en extremos opuestos del mapa.

3. Diagramas de clases

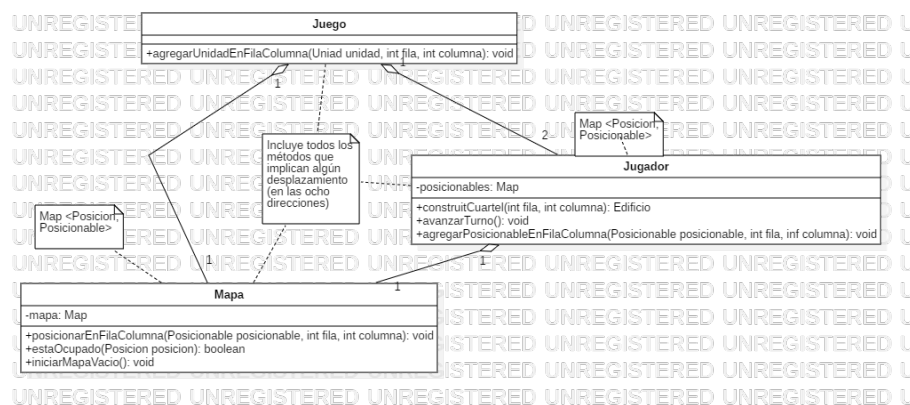


Figura 1: Diagrama que representa parte de la solución del trabajo práctico

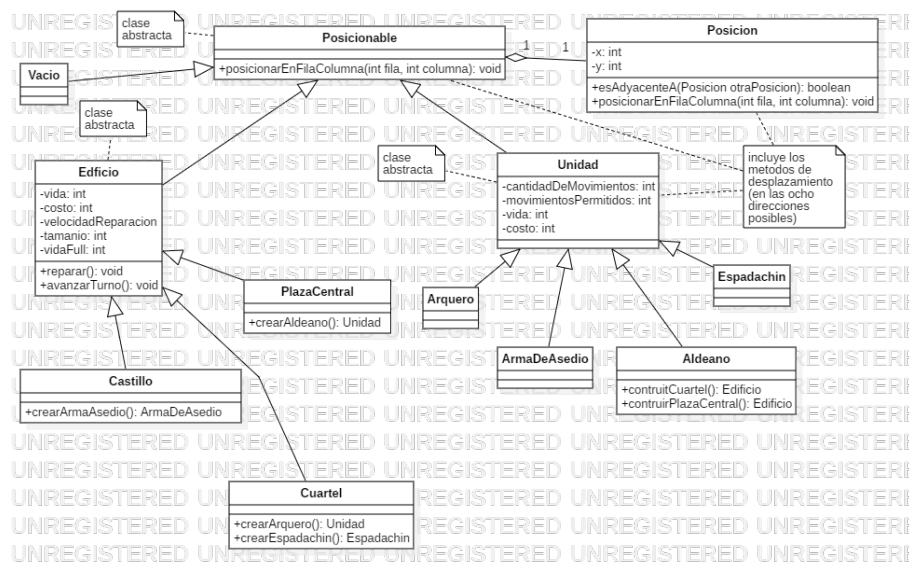


Figura 2: Diagrama que representa parte de la solución del trabajo práctico

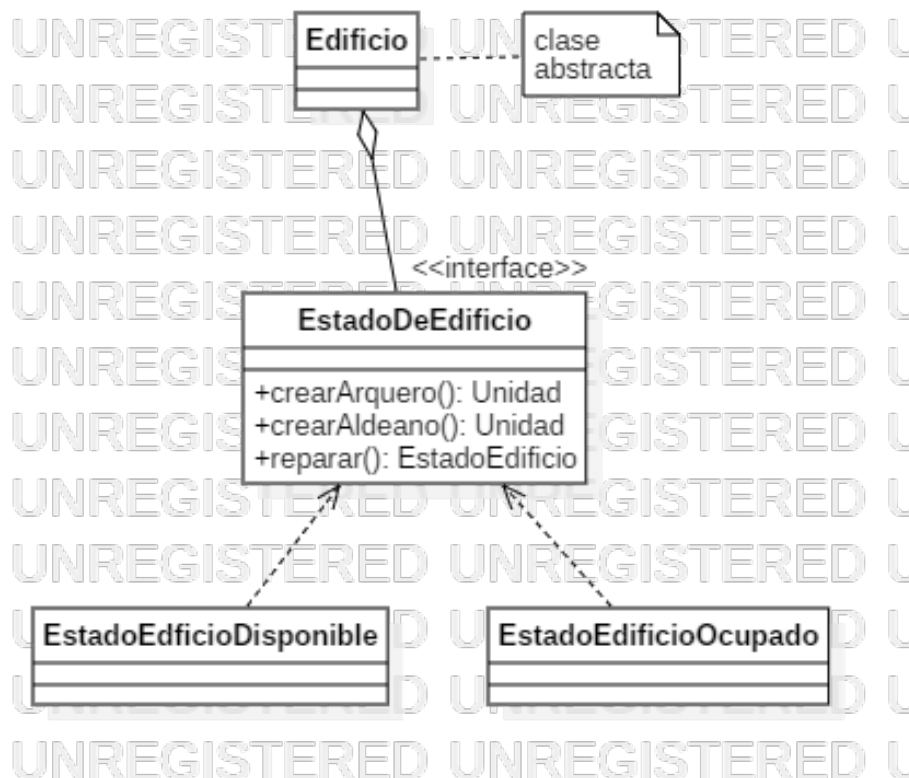


Figura 3: Diagrama que representa parte de la solución del trabajo práctico

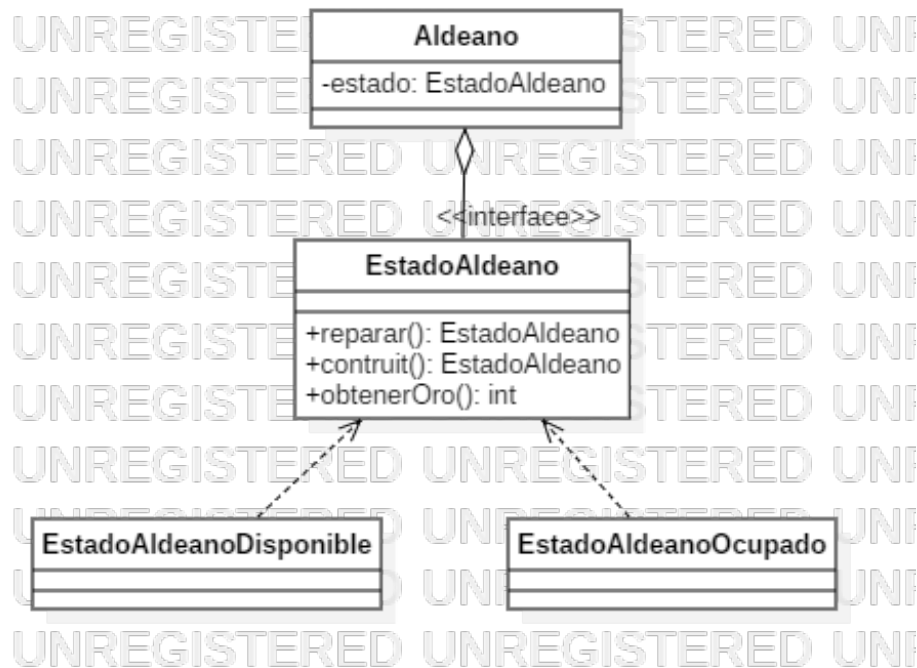


Figura 4: Diagrama que representa parte de la solución del trabajo práctico

4. Diagramas de secuencia

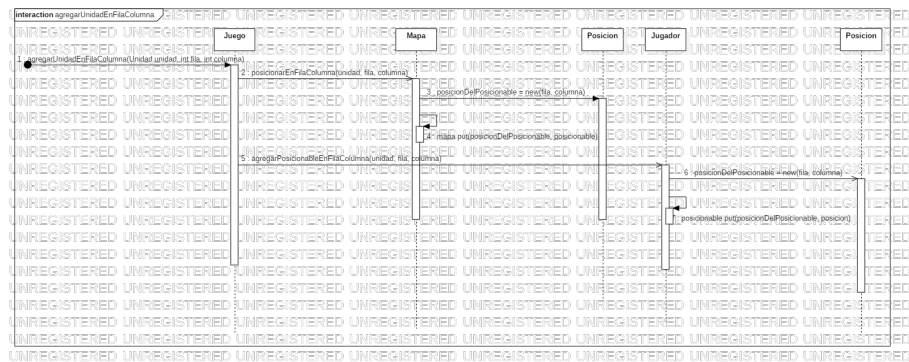


Figura 5: Diagrama de secuencias agregarUnidadEnFilaColumna

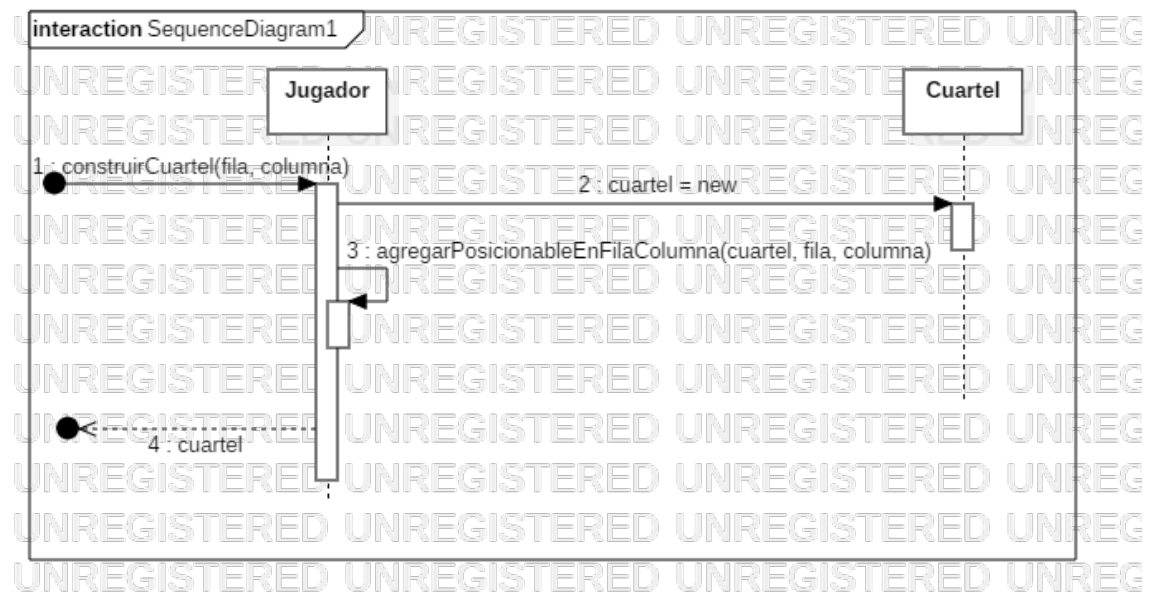


Figura 6: Diagrama de secuencias construirCuartel.

5. Diagramas de paquetes

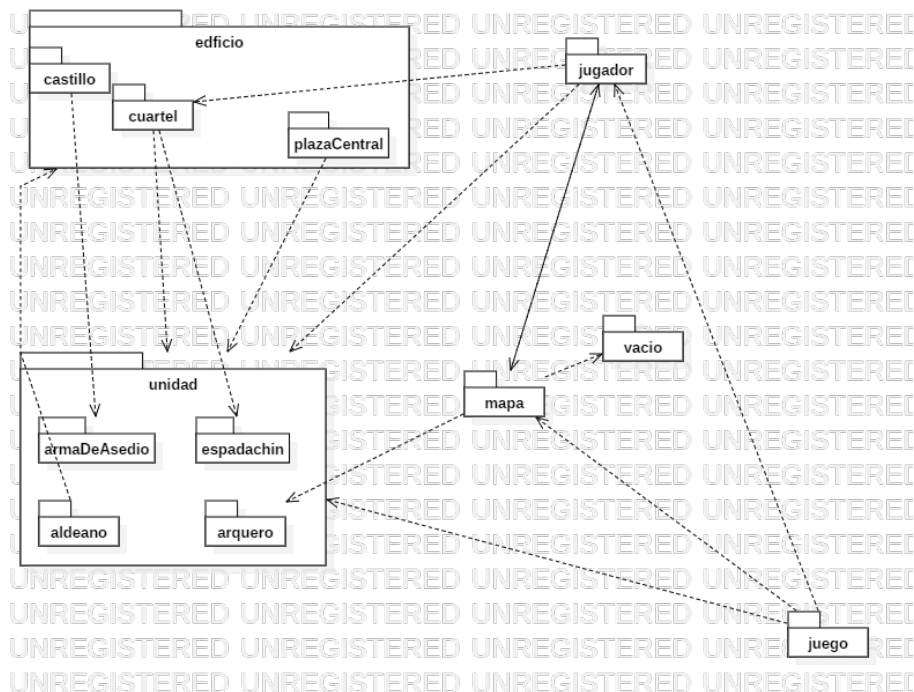


Figura 7: Diagrama de paquetes del trabajo práctico

6. Diagramas de estado



Figura 8: Diagrama de estado para un aldeano

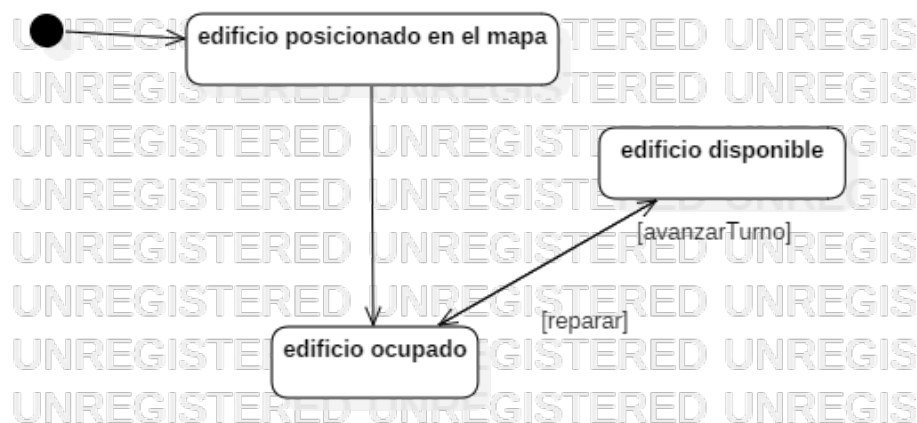


Figura 9: Diagrama de estado para un edificio

7. Detalles de implementación

7.1. Mapa

La clase Mapa tiene un atributo mapa. Este mapa es un diccionario del tipo Map <Posicion, Posicionable> siendo Posicion la clave y Posicionable el valor. Almacena todas las posiciones disponibles en el terreno de juego como clave, asignándole como valor una clase del tipo Unidad, Edificio o Vacio. Estas clases heredan de Posicionable. Inicialmente, el mapa tiene todas sus claves como Vacio, a lo largo del juego se cambiarán por Unidad y Edificio.

7.2. Juego

La clase Juego almacena al mapa que representa al terreno de juego, será del tipo Mapa. Además, también almacena al Jugador que se encuentra en juego actualmente, por lo que éste

atributo irá cambiando de acuerdo a qué instancia de Jugador se encuentra accionando al momento. De esta forma se manejarán los cambios de turnos durante el juego.

7.3. Jugador

La clase Jugador va a tener como atributo posicionables, un diccionario que contiene los mismas clases que en la clase Mapa. A diferencia almacenará únicamente las Unidades y Edificios que pertenezcan a la instancia de Jugador, junto con sus posiciones. Por lo tanto, no tendrá ninguna clase del tipo Vacio.

Además, tendrá un atributo mapa que representa al mismo mapa que Juego ya que cada Jugador conoce el terreno de juego y, como puede desplazar a sus unidades, el Mapa necesita conocer esos desplazamientos para reubicar las unidades en él.

8. Excepciones

EdificioOcupadoException Un edificio no puede funcionar normalmente en caso de encontrarse en construcción o en reparación.

PosicionDesocupadaError No se puede desplazar una Unidad si ésta no se encuentra en la posición indicada. Es decir, no se puede realizar ningún tipo de desplazamiento si la posición indicada se encuentra vacía.

PosicionOcupadaError Esta excepción se lanzará en caso de querer agregar un elemento en una posición que ya se encuentra ocupada por otro actualmente o, en caso de querer desplazar a una Unidad hacia una posición que se encuentra ocupada. Teniendo en cuenta el supuesto mencionado sobre las posiciones que almacenan un único elemento.

MovimientoPorTurnoExcedidosError Cada Unidad puede realizar una cierta cantidad de movimientos en cada turno. En este trabajo práctico esa cantidad es uno. Por lo tanto, si se quiere realizar más de un desplazamiento con la misma unidad se lanzará esta excepción.

PosicionFueraDelMapaError Tanto los desplazamientos como la inclusión de objetos al juego sólo pueden ser en posiciones pertenecientes al mapa. Por lo tanto, en caso de violar esto se lanzará esta excepción.

AldeanoOcupadoException un Aldeano no puede realizar una tarea en caso de estar Ocupado.