FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2 - Java

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

Alumnos:	BIAUS, Ivo GARCIA, Ailen LOPEZ HIDALGO, Tomas TROTTA, Laura
Grupo:	2

Índice

1. Supuestos

1) Cada Jugador, dentro de un mismo turno, puede realizar múltiples acciones, dado que su turno se compone de cada acción de cada Pieza.

2) Los Edificios pueden realizar una sola acción por turno, la cual es crear una Unidad.

3) Un Aldeano, en un mismo turno, sólo puede mover, reparar, o construir, pero solo si realiza la acción de mover (o ninguna) puede generar oro.

4) Si un Aldeano está reparando/construyendo, no puede realizar ninguna otra acción o cancelar la actual tarea hasta que ésta se termine.

2. Diagramas de clase

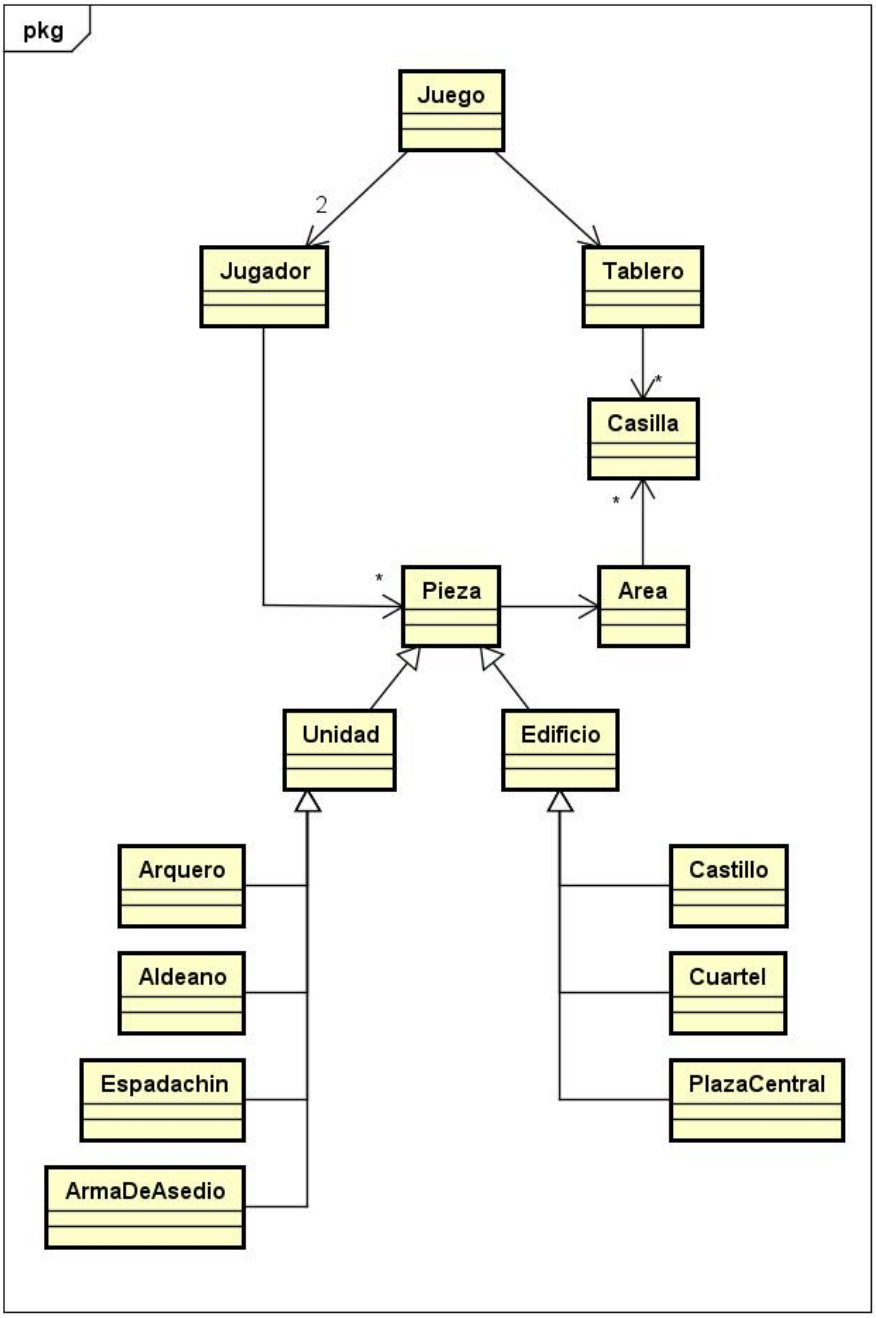

Figura 1: Diagrama del Algo Empires2.

3. Diagramas de secuencia

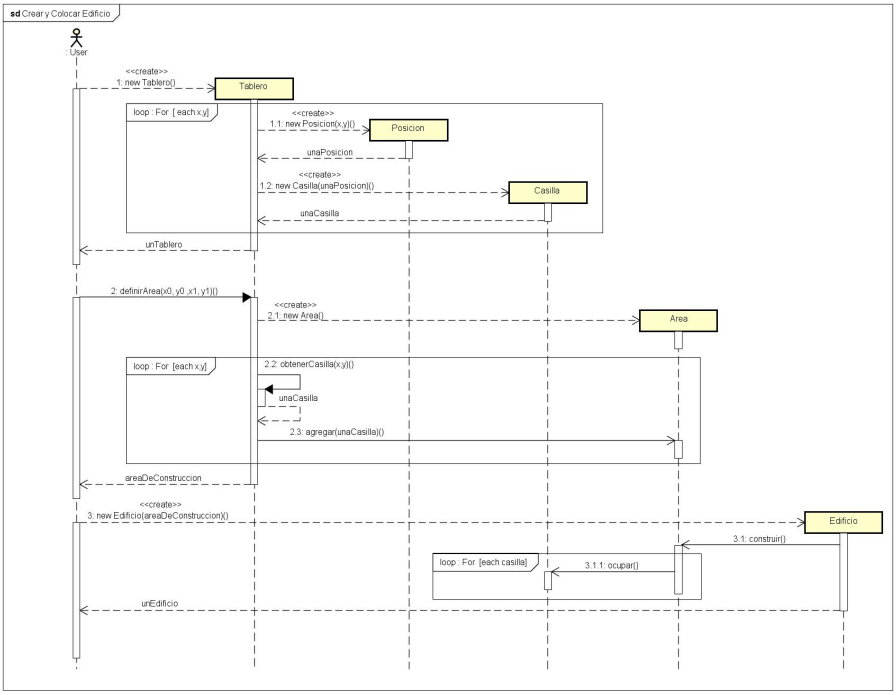

Figura 2: Diagrama Crear y Colocar Edificio

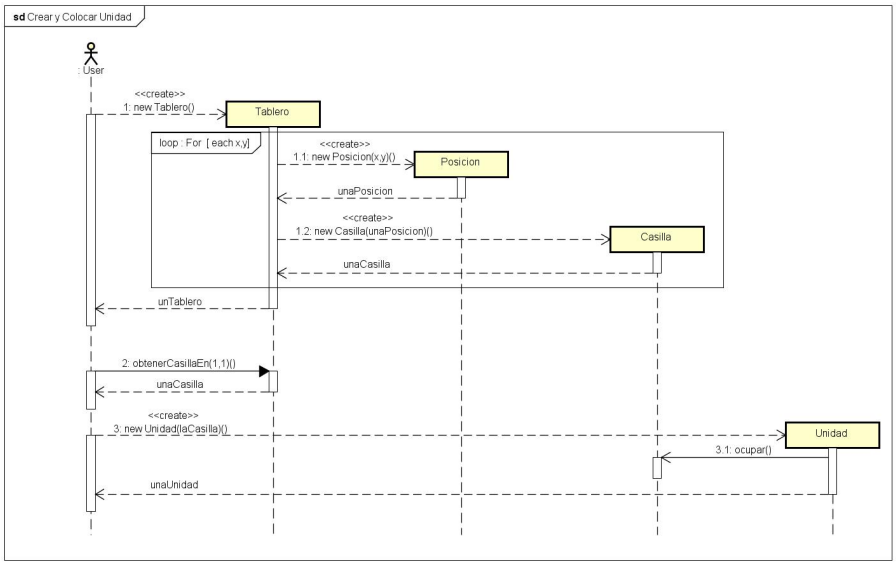

Figura 3: Diagrama Crear y Colocar Unidad

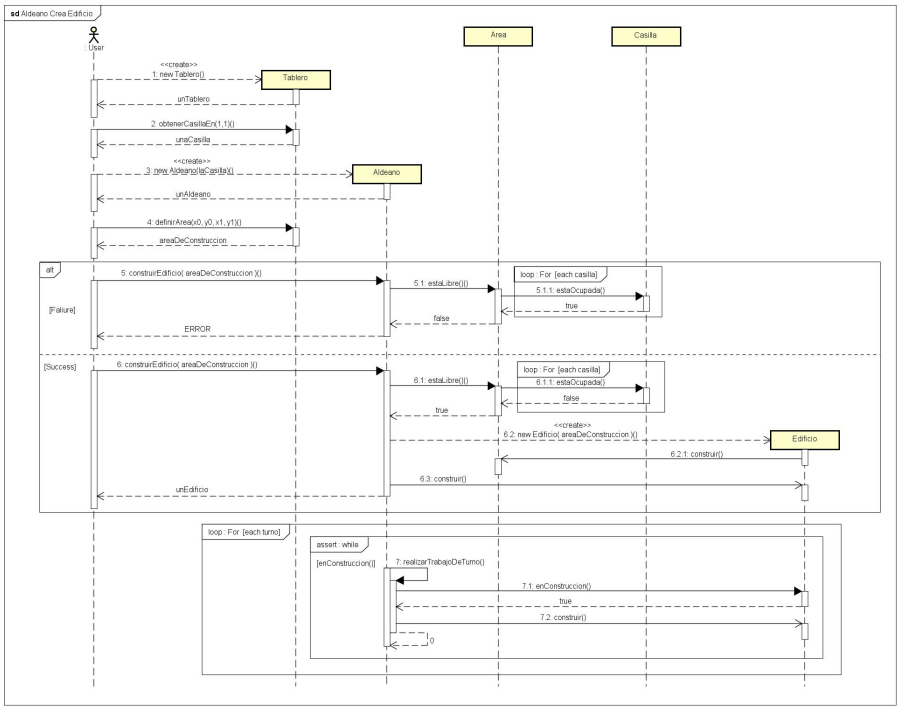

Figura 4: Diagrama Aldeano Crea Un Edificio

4. Diagramas de paquete

5. Diagramas de estado

6. Detalles de implementacion

Para la implementacion del trabajo practico utilizamos principalmente los patrones de MVC y State.

Usamos MVC en el aspecto general del juego, separando el codigo en paquetes de Modelo, Vista, y Controlador. Controlador lo dejamos para la ultima instancia, junto con Vista.

Luego tambien se uso Patron para organizar estados de las clases como por ej Aldeano cuando este Repara, Construye, o esta Libre; En breve tambien probablemente lo apliquemos para el Arma de Asedio en cuanto a que esta puede estar accionada para atacar o moverse.

Tambien se lo uso para los estados del Juego, los cuales pueden ser hasta el momento: NoComenzado, JuegaJugador1, JuegaJugador2 y Terminado.

7. Excepciones

Exception JuegoNoTerminadoError Esta excepción se lanza cuando se le pide al Juego que me muestre el ganador, si aún el juego no ha finalizado.

Exception NoExistenJugadoresActualesError Esta excepción fue creada para que, en el caso de que se le pida al Juego el jugador actual, y todavía la partida no ha comenzado.

Exception NoHayJuegoEnProcesoError Esta excepción se lanza en el caso de que se quiera terminar un juego que aún no ha comenzado.

Exception CasillaOcupadaError Esta excepción fue creada para evitar que un jugador ocupe una casilla que ya está siendo ocupada por una pieza.

Exception NullPointerCasillaError Esta excepción se lanza si, en algún caso, se pasa por parámetro una Casilla que no apunta a nada.

Exception PiezaOcupadaNoPuedeAccionarError Esta excepción se lanza en el caso de que, a una Pieza que está ocupada, le pido que realice una acción.

Exception PiezaYaJugoEnTurnoActualError Esta excepción se lanza en el caso de que una Pieza quiera jugar en un turno en donde ya había jugado.

Exception CasillaInvalidaError Esta excepción fue creada para evitar que un jugador quiera acceder a una Casilla que está fuera del mapa, es decir, que posea coordenadas inválidas.

Exception PoblacionLimiteSuperadaError Esta excepción se lanza en el caso de que a un jugador que ya posee la cantidad de población límite, se le quiere agregar más población.