

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III

Curso 1

Segundo cuatrimestre de 2019

Alumno:	Penic, Federico
Número de padrón:	102501
Alumno:	Carretero, Andrés
Número de padrón:	101004
Alumno:	Serra, Juan Cruz Bautista
Número de padrón:	102792
Alumno:	García, Pablo
Número de padrón:	75322

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	2
5. Detalles de implementación	3
5.1. Interacción entre Jugador y Tablero	3
5.2. Creación de Unidades	3
6. Excepciones	3
7. Diagramas de secuencia	3

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 un Juego similar al Ajedrez utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

- Supuesto 1: Suponemos que las unidades que se pueden mover lo hacen en direcciones horizontales, verticales y diagonales.
- Supuesto 2: Suponemos que el tablero será siempre cuadrado.
- Supuesto 3: Los Jugadores no pueden tener el mismo nombre.

3. Modelo de dominio

El diseño general del programa por ahora consta de 3 partes, primero tenemos el tablero, luego las unidades y luego los jugadores.

El tablero consiste en una clase que contiene cierta cantidad de Filas, cada Fila contiene una cierta cantidad de Casilleros, los cuales están determinados por una Posición.

Los Casilleros pueden estar Vacíos u Ocupados, esto se implementó con el patrón de diseño state.

Por otra parte, las Unidades son: Catapulta, Soldado, Jinete, Curandero. Catapulta es una Unidad, y las demás Unidades son Unidades Movibles, para resolver esto utilizamos herencia.

Cada Unidad tiene un costo y una vida, y según si puede atacar o no puede tener un cierto daño. El jinete tiene un Arma, la cual cambia según si está Acompañado o Aislado, también resolvimos esto con un patrón de diseño state. Cada Unidad también tiene el nombre del Jugador al cual pertenece.

Los Jugadores tiene una cierta cantidad de puntos, un nombre y tienen Unidades.

Por ahora esto es lo que tenemos de la aplicación, pero nos falta implementar otra parte del diseño, que vendría a ser una clase Juego, la cual se encarga de implementar el flujo del programa y el correcto desenlace del juego.

4. Diagramas de clase

A continuación se van a ir mostrando los diferentes Diagramas de Clases

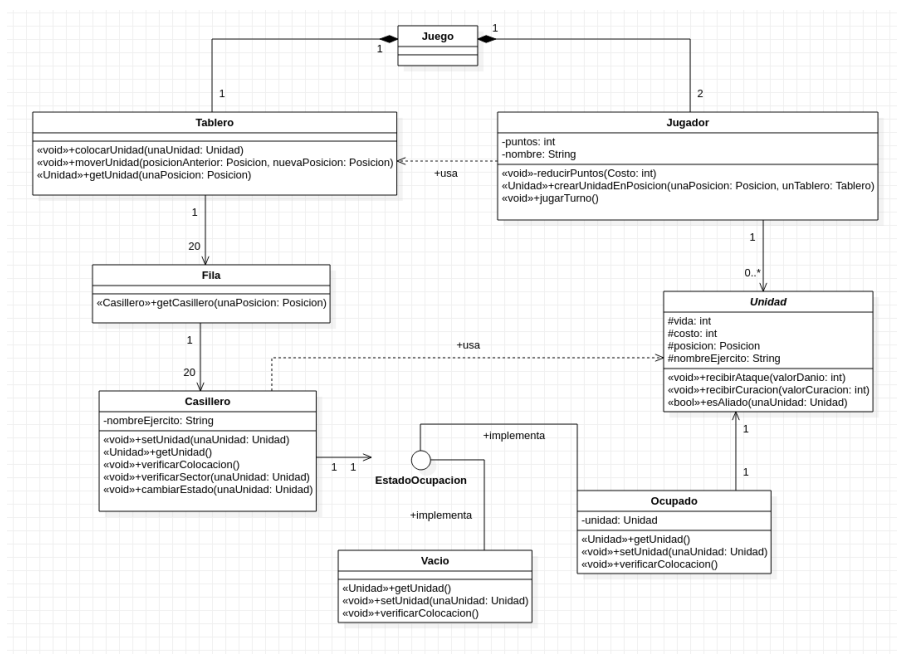


Figura 1: Diagrama de clases del diseño general del programa.

5. Detalles de implementación

5.1. Interacción entre Jugador y Tablero

Consideramos que el Jugador va a necesitar interactuar directamente con el Tablero, para ello, en ciertos mensajes recibe al Tablero como parámetro. Por ejemplo, cuando un Jugador crea una Unidad, es el Jugador el que se encarga de crear la Unidad, asignarle una Posicion y luego pedirle al Tablero que la posicione. Consideramos que el Juego va a ser el responsable de pasarle como parámetro al Jugador el Tablero en el cual posicionar la Unidad.

5.2. Creación de Unidades

Si bien por ahora el Jugador tiene un mensaje diferente para la creación de cada Unidad diferente, consideramos que a futuro para que sea más prolijo y menos repetitivo tendremos que implementar un patrón de diseño del tipo factory, es decir, cada Jugador tendrá su propia Fabrica De Unidades a la cual pedirle que cree cierta Unidad.

6. Excepciones

ExcepcionCasilleroOcupado Esta excepción se lanza cuando se intenta colocar una Unidad en un casillero Ocupado o mover una Unidad a un casillero Ocupado.

ExcepcionDeCambioDeEstado Esta excepción se lanza cuando se coloca una Unidad en un casillero, sería una forma de comunicar que el casillero pasa de estar Vacío a estar Ocupado.

ExcepcionSectorEnemigo Esta excepción se lanza cuando se intenta colocar una Unidad en el Sector Enemigo, lo cual no debería ser posible.

7. Diagramas de secuencia