

## 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

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

## 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.

-Supuesto 4: Funcionamiento del ataque del jinete: Si tiene algun soldado aliado cerca o no hay ninguna Unidad enemiga cerca, el arma de ataque será el arco y flecha. De otra forma que no sean estas 2, el jinete utilizará la espada para atacar.

-Supuesto 5:

## 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 y Curandero. Batallón no es una unidad en sí sino un conjunto de unidades de tipo soldado, este se crea, utiliza, y disuelve en un mismo turno a voluntad del jugador del mismo. 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 (también se podría considerar como un patrón Strategy). Cada Unidad también guarda el ejército al cual pertenece.

Las Unidades se comunican varias veces, con el Tablero, ya sea para moverse, para atacar, o curar.

Los Jugadores tienen un nombre y un Ejército, y el mismo tiene una cierta cantidad de puntos, el mismo nombre del jugador, y Unidades.

## 4. Diagramas de clase

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

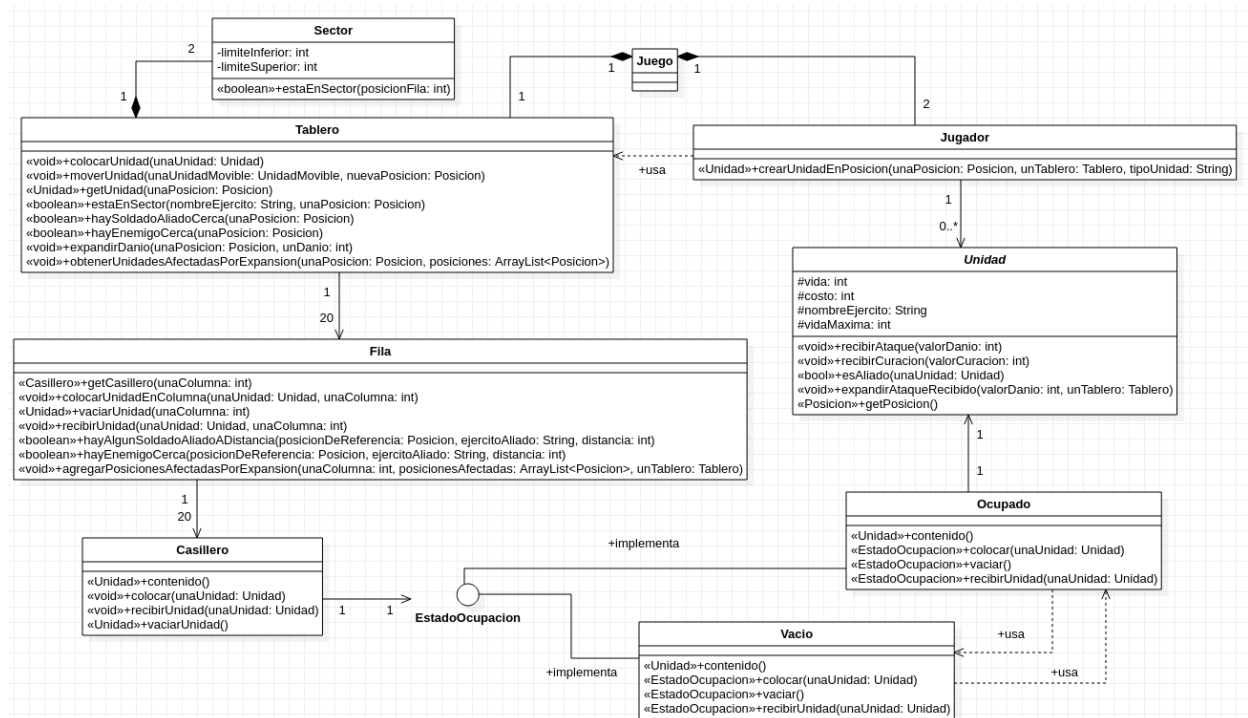


Figura 1: Diagrama de clases del diseño general del programa.



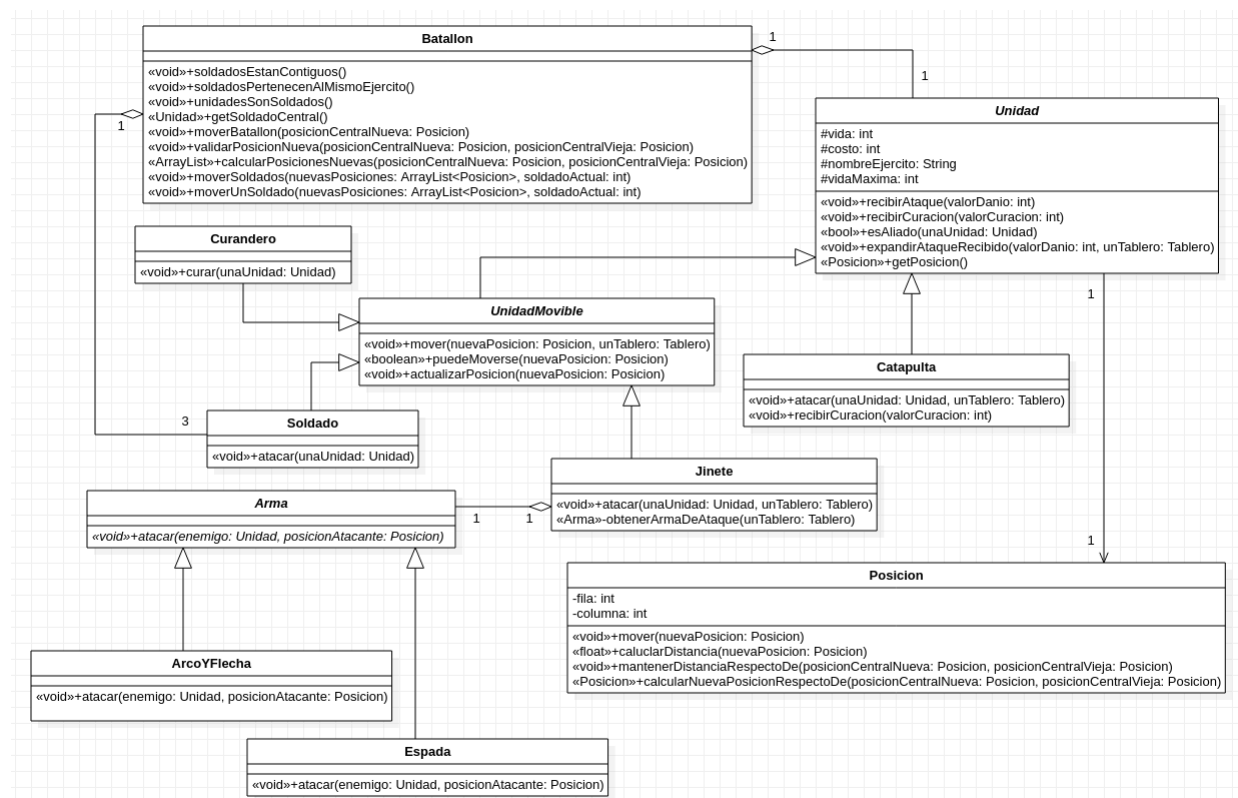


Figura 3: Diagrama de clases de la sección de Unidades.

## 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.

**ExcepcionCantidadIncorrectaDePosiciones** Esta excepción se lanza cuando se le envían mas, o menos, posiciones de las necesarias, tres, para crear un Batallon.

- 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.
- ExcepcionPuntosInsuficientes** Esta excepción se lanza cuando al jugador de turno no tiene los puntos suficientes para crear una Unidad determinada.
- ExcepcionAtaqueAAliado** Esta excepción se lanza cuando se intenta atacar a una Unidad Aliada (menos la catapulta).
- ExcepcionCatapultaNoAtacaAliados** Esta excepción se lanza cuando se intenta atacar a una Unidad Aliada con la catapulta.
- ExcepcionDistanciaAtaqueInvalida** Esta excepción se lanza cuando se intenta atacar a una distancia diferente a la que puede atacar una Unidad.
- ExcepcionDistanciaCuracionInvalida** Esta excepción se lanza cuando se intenta curar a una unidad con un curandero que se encuentra a una distancia mayor a la corta respecto al mismo.
- ExcepcionCuranderoNoAtaca** Esta excepción se lanza cuando se intenta atacar con un curandero en el turno.
- ExcepcionCasilleroVacio** Esta excepción se lanza cuando se intenta obtener una Unidad de un Casillero que se encuentra Vacío.
- ExcepcionCuracionACatapulta** Esta excepción se lanza cuando un Curandero intenta curar a una Catapulta.
- ExcepcionCuracionAEnemigo** Esta excepción se lanza cuando un Curandero intenta curar a una Unidad Enemiga.
- ExcepcionLasUnidadesEstanSeparadas** Esta excepción se lanza cuando se crear un Batallon con unidades que no son contiguas.
- ExcepcionMovimientoInvalido** Esta excepción se lanza cuando se intenta mover a una Unidad a una Distancia mayor a la que puede moverse.
- ExcepcionUnidadNoPerteneceATuEjercito** Esta excepción se lanza cuando el jugador de turno intenta crear utilizar una unidad que no le pertenece.
- ExcepcionTipoUnidadInvalida** Esta excepción se lanza cuando se intenta crear un Batallon con unidades que no son Soldados.
- ExcepcionFinDelTablero** Esta excepción se lanza cuando se intenta acceder a un Casillero que no se encuentra dentro del rango del Tablero.
- ExcepcionFinDeJuego** Esta excepción se lanza cuando uno de los jugadores se quedó sin unidades, por lo que el otro ganó.

## 7. Diagramas de secuencia

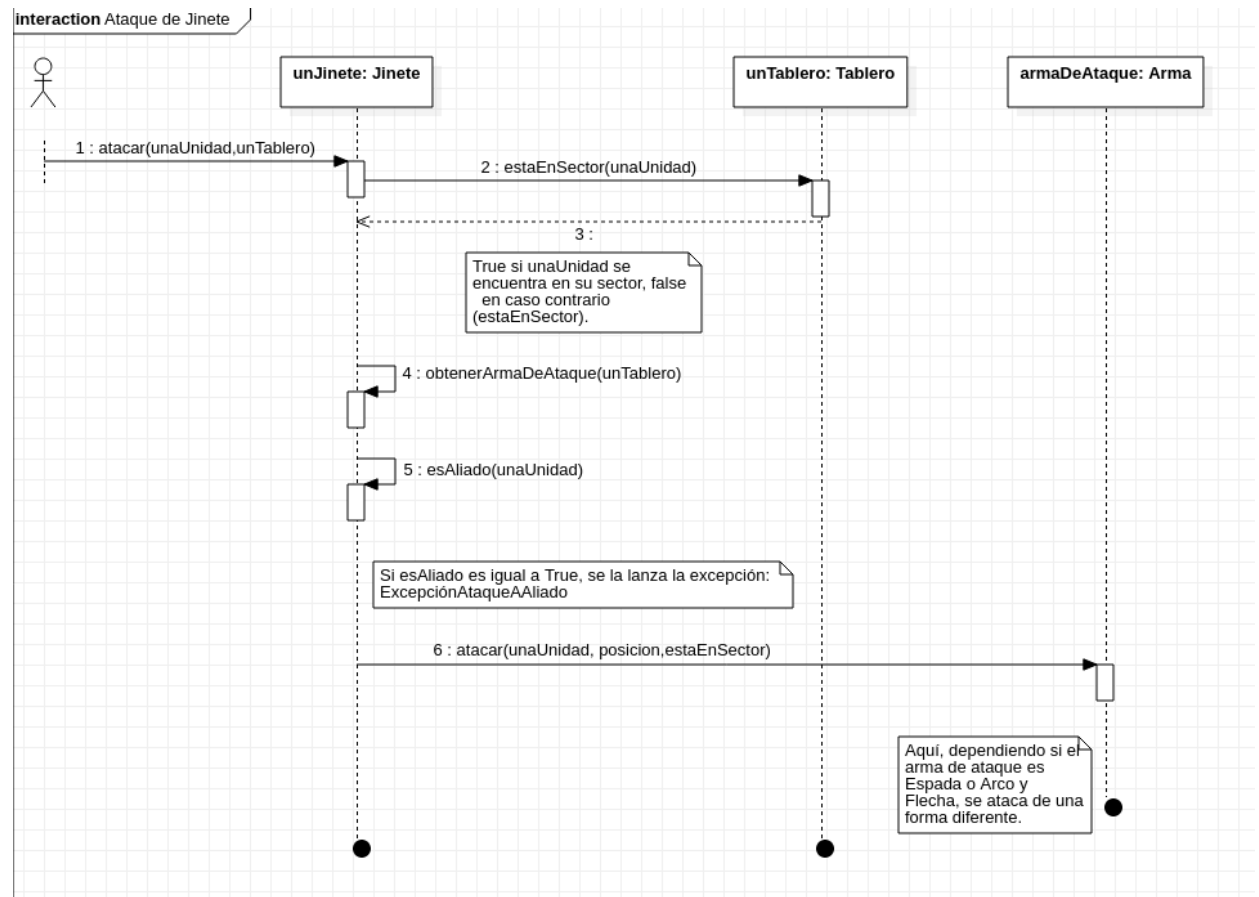


Figura 4: Ataque de un jinete.



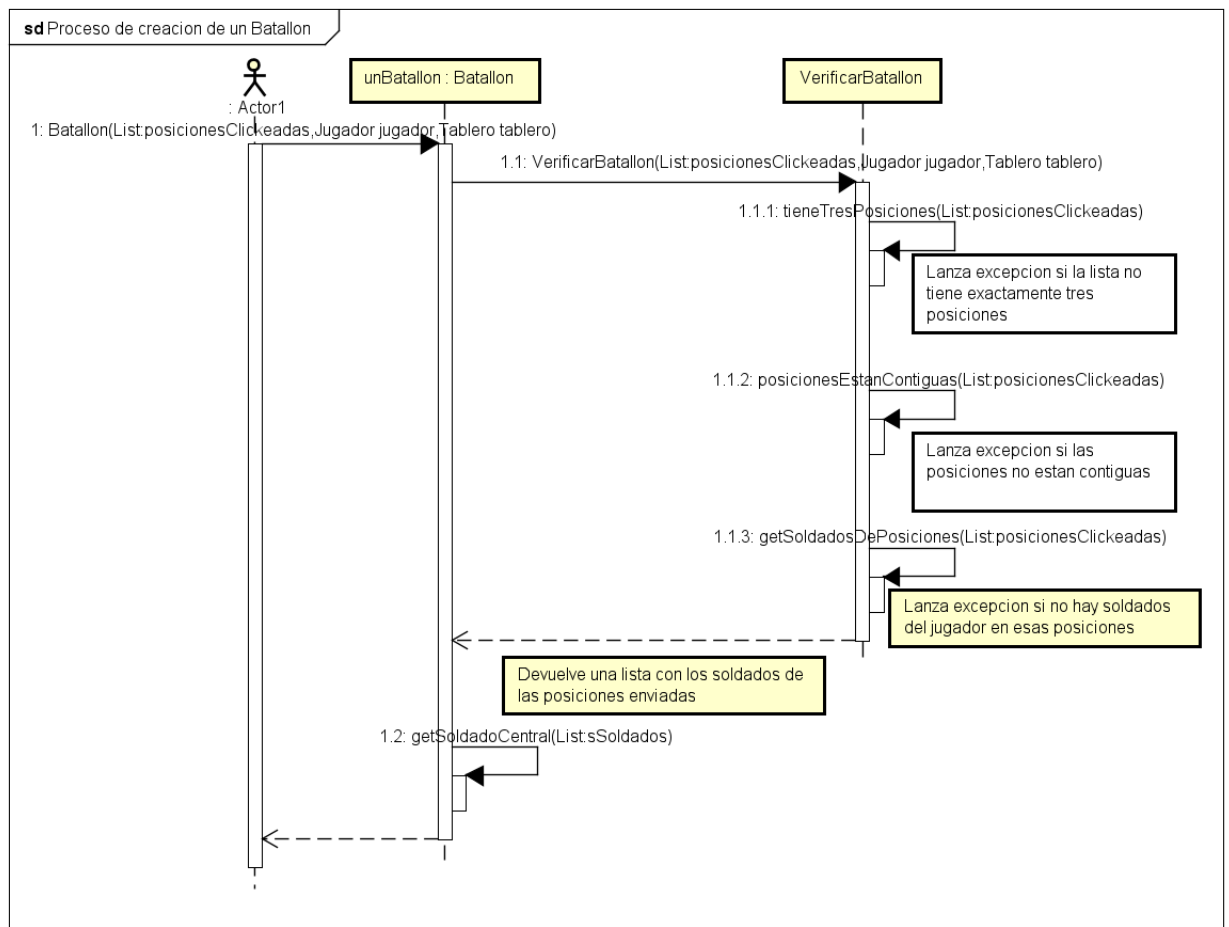


Figura 5: Creacion de un Batallon.