Trabajo Práctico 2 — Grupo 10

[7507/9502] Algoritmos y Programación III
Curso 1 - Entrega final
Segundo cuatrimestre de 2019

Nombre	Padrón	E-mail
CORONEL, Dalma	92.257	coronel.dg@gmail.com
CONTE GRAND, Joaquin	102868	jocontegrand@gmail.com
PITA, Joel	10166	joelJPita91@gmail.com
RESNIZKY, Nicolás	102.512	nres98@gmail.com

Índice

1. Introducción

El presente informe reune la documentación de la solución del Trabajo Práctico 2 de Algoritmos y Programación III que consiste en desarrollar un videojuego por turnos, de dos jugadores conformado de un tablero y distintas unidades sobre él. El objetivo del juego es destruir todas las unidades enemigas y gana el jugador que lo consigue primero. Se utiliza Java, técnicas de TDD e Integración Continua.

2. Supuestos

Consideramos que cada jugador está obligado a gastar todos los puntos para comprar piezas. En la fase media los jugadores tienen la opción de pasar de turno sin realizar ninguna acción. Además, los jugadores no pueden tener el mismo nombre. Con relación a los movimientos de un batallón, consideramos que los miembros del batallón siguen al Infante que se selecciona para mover. Asimismo, si un Infante se suma a un nuevo batallón, rompe con su batallón actual. Para que el juego sea más dinámico se decidió que en cada turno el jugador pueda realizar dos movimientos y tres ataques diferentes.

3. Diagramas de clases

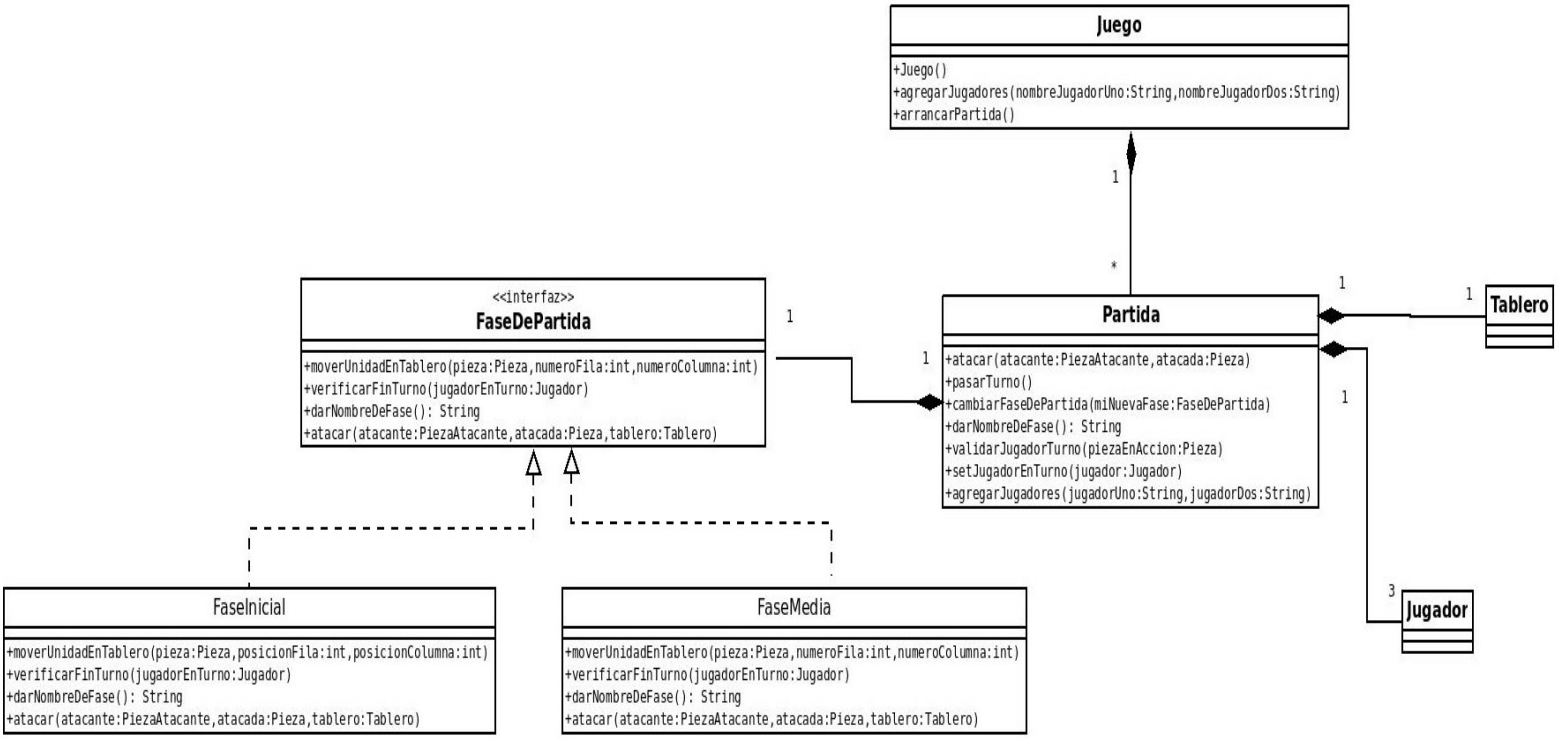

Figura 1: Diagrama de clases Juego.

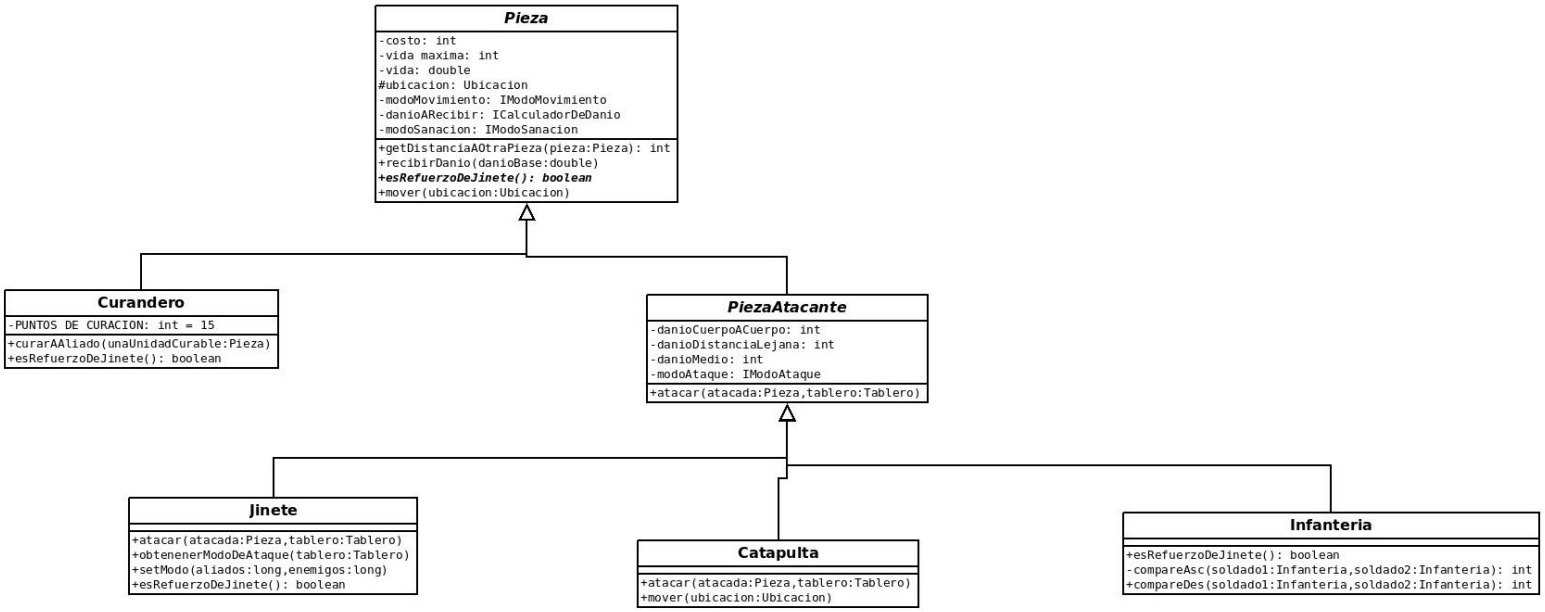

Figura 2: Diagrama de clases Pieza.

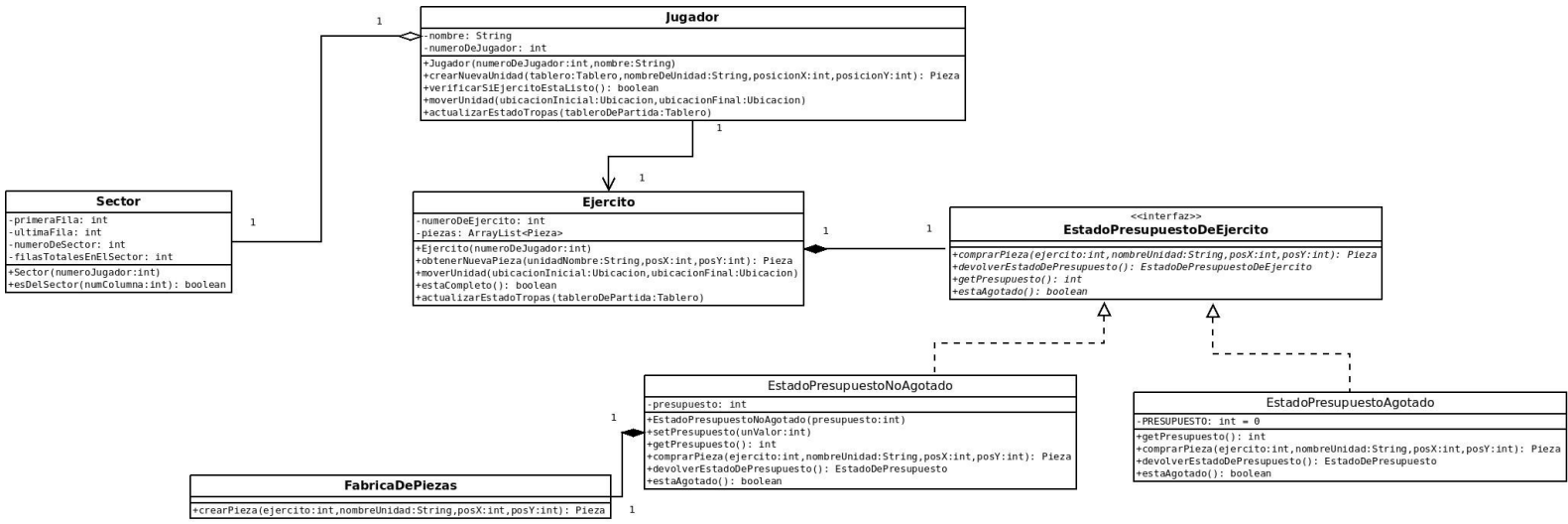

Figura 3: Diagrama de clases Jugador.

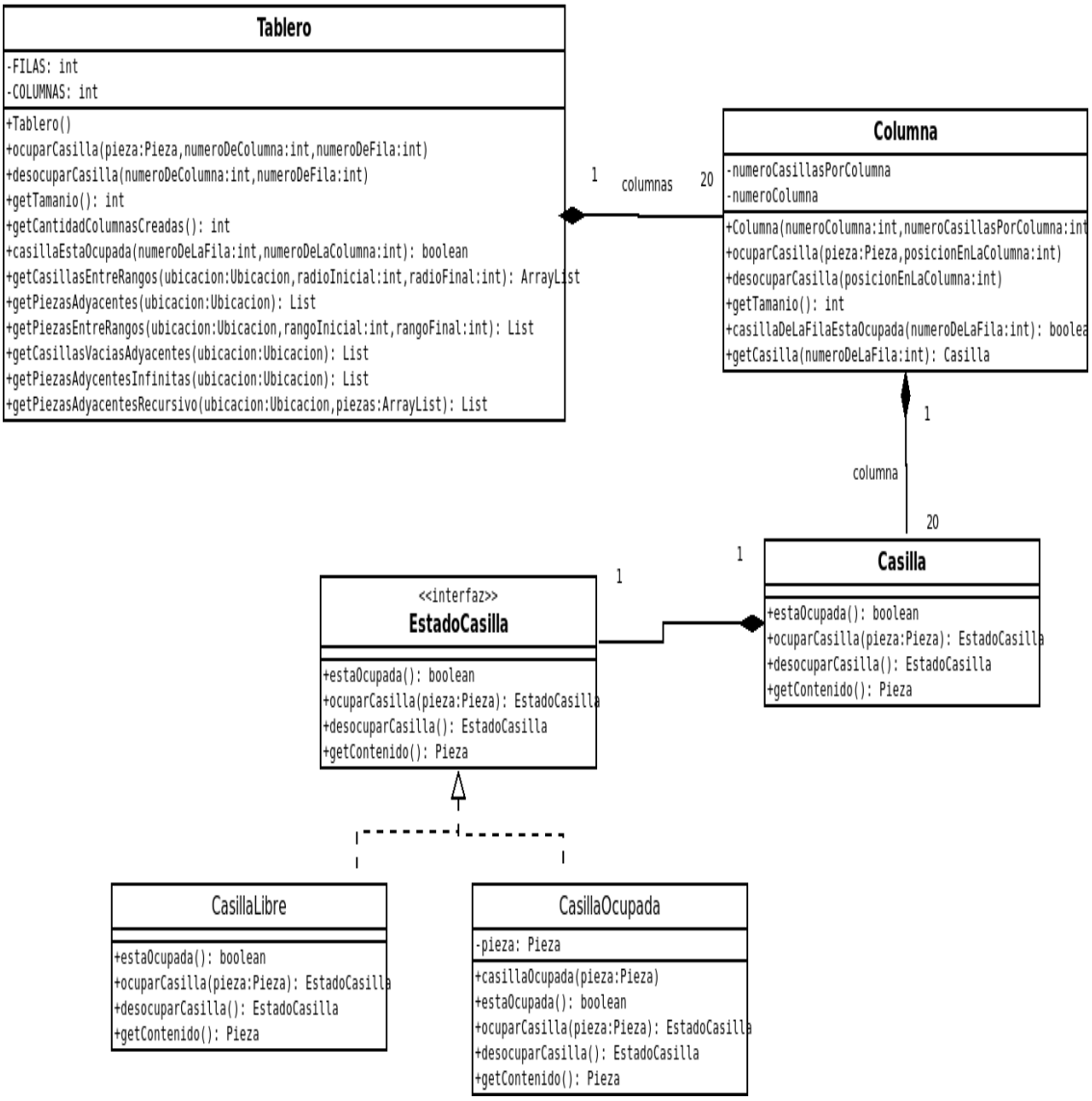

Figura 4: Diagrama de clases Tablero.

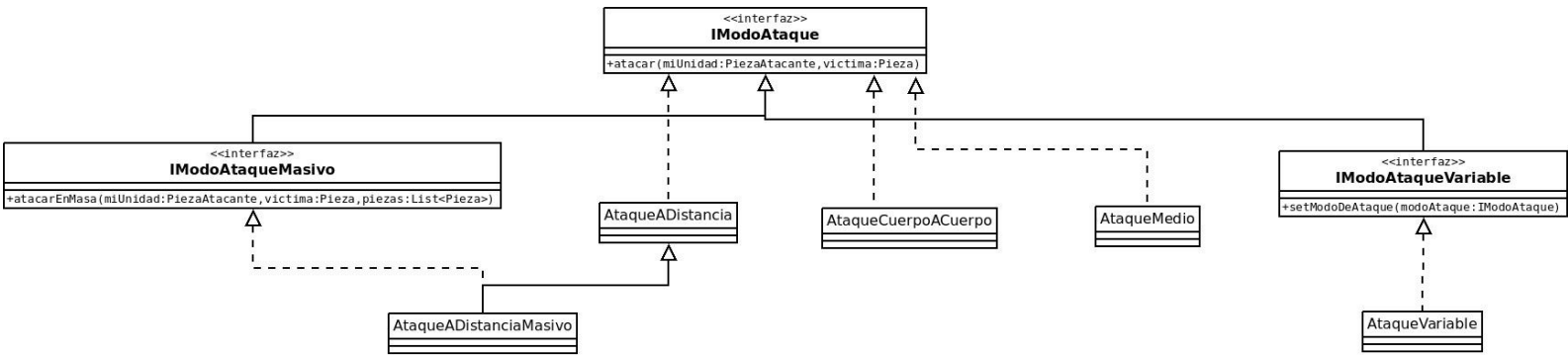

Figura 5: Diagrama de clases Ataques.

4. Diagramas de secuencia

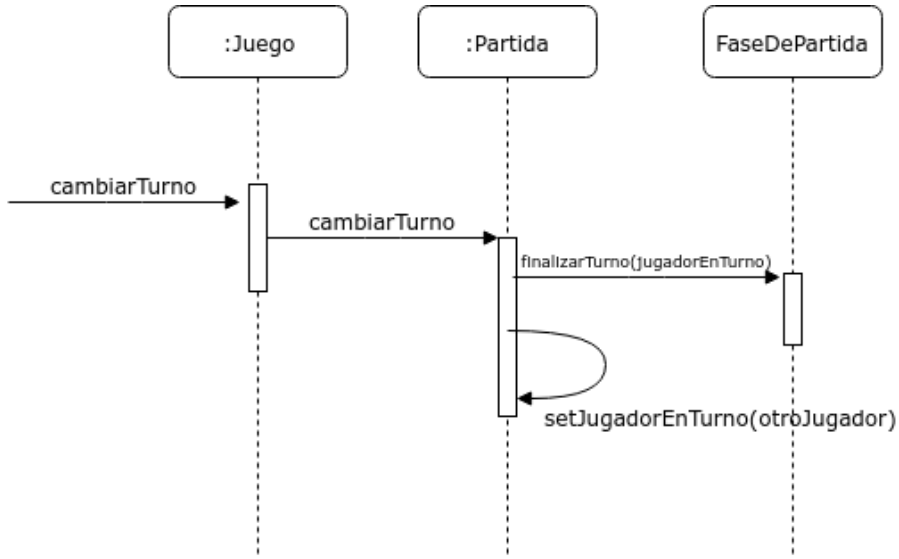

Figura 6: Diagrama de secuencia para cambiar el turno

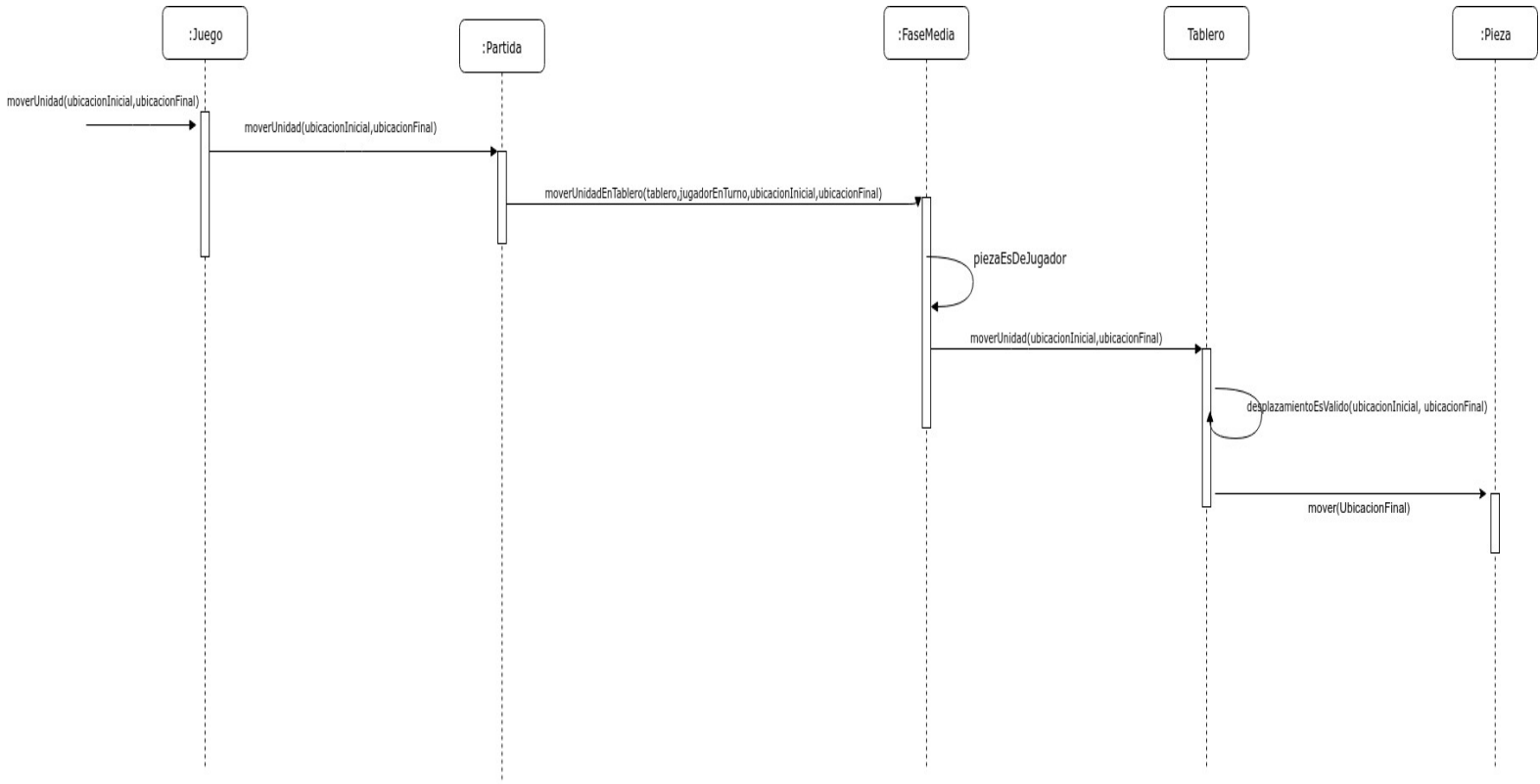

Figura 7: Diagrama de secuencia para mover una unidad

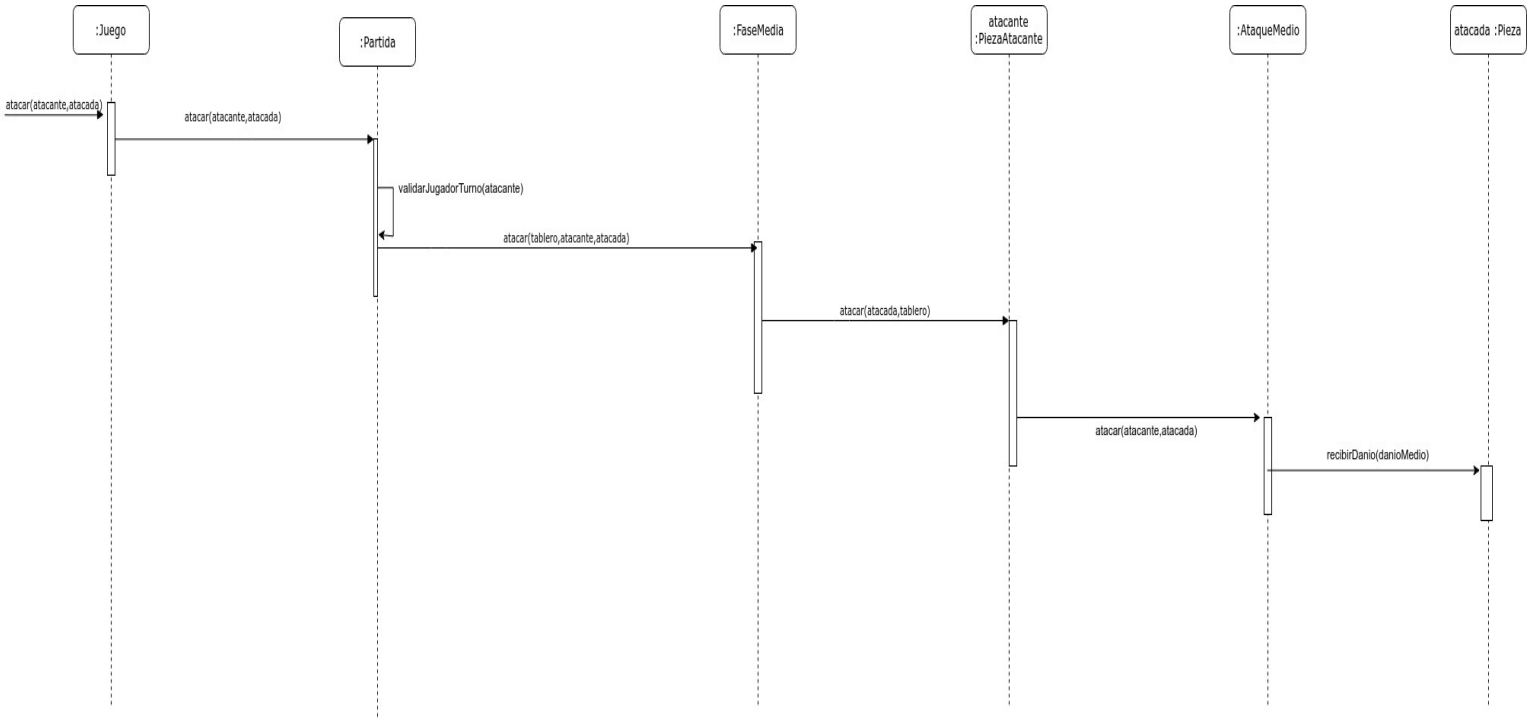

Figura 8: Diagrama de secuencia para un ataque medio

5. Diagrama de paquetes

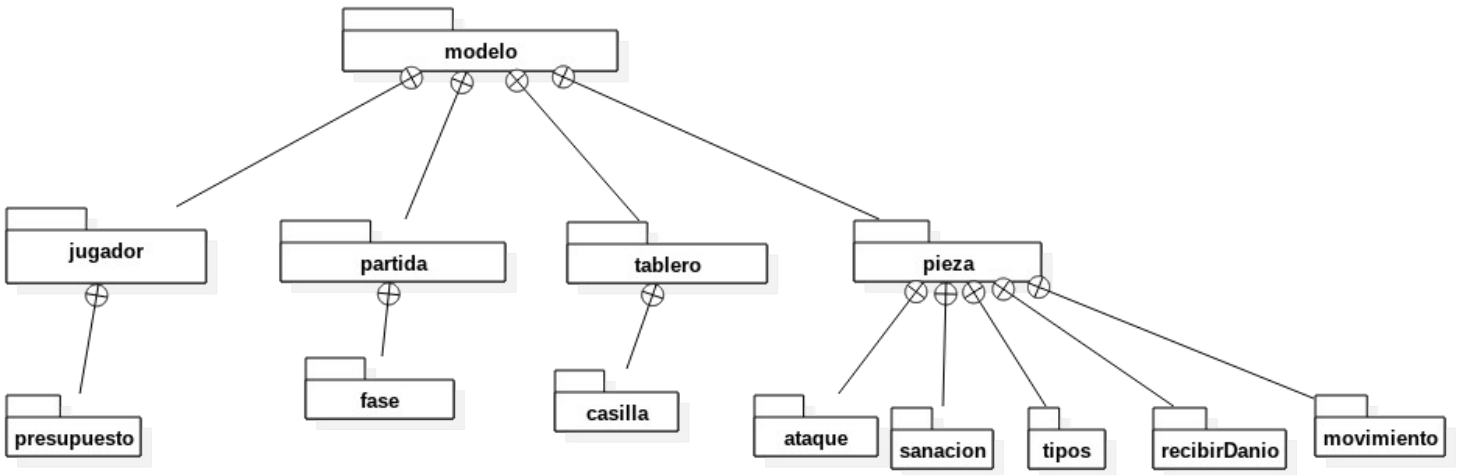

Figura 9: Diagrama de paquetes del modelo

6. Diagramas de estado

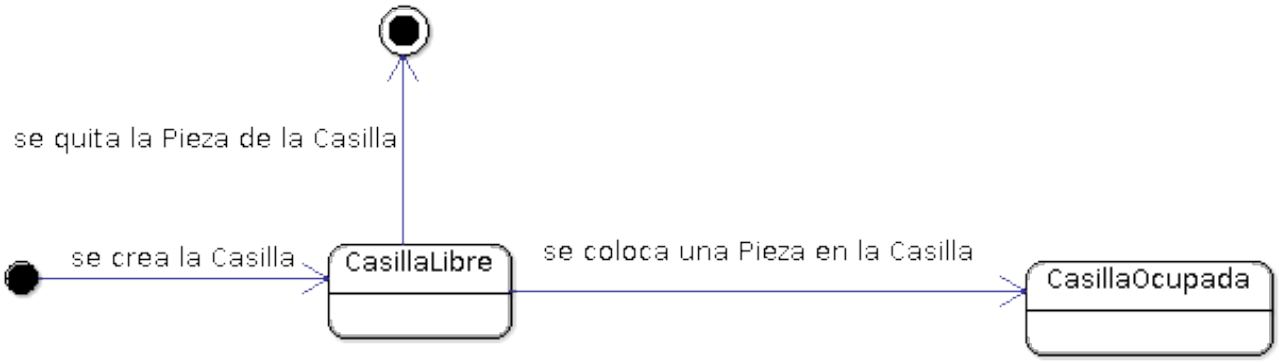

Figura 10: Diagrama de Estados de Casilla

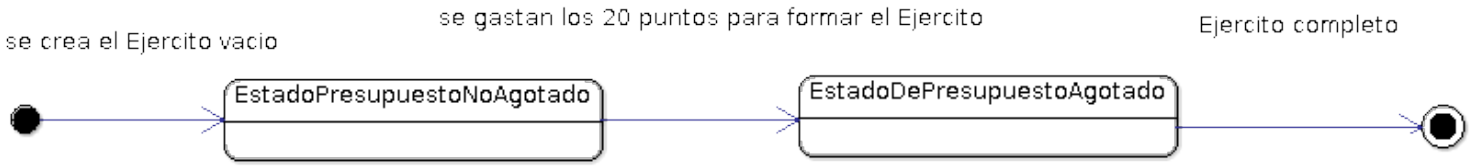

Figura 11: Diagrama de Estados de Presupuesto

7. Detalles de implementación

7.1. Juego

Es la clase que interactúa con la interfaz gráfica para plasmar en el modelo las distintas acciones del juego. Tiene como atributo privado una partida, en la que delega las diferentes acciones.

7.2. Tablero, Columnas y Casillas

Tablero contiene una lista de 20 columnas. A su vez, cada columna contiene una lista de 20 casillas. Para solucionar la dificultad de evaluar si una casilla está vacía u ocupada, se decidió implementar el patrón State. Cada casilla cuenta con un atributo EstadoCasilla. EstadoCasilla es una interfaz, de la que se implementan CasillaLibre y CasillaOcupada. Por defecto, cuando se crea una casilla nueva, la misma comienza con EstadoCasilla = CasillaLibre.

7.3. Jugador y Ejército

Cada jugador tiene un nombre, número, sector al que pertenece y ejército. El ejército está implementado como una lista de piezas. Para que en la fase inicial se agregue una pieza al ejército de un jugador, cada ejército tiene un EstadoDePresupuesto. Este es representado por una interfaz, y puede ser EstadoPresupuestoAgotado y EstadoPresupuestoNoAgotado. En el único caso en el que se le va a permitir a un jugador agregar una pieza a su ejército es cuando sea EstadoPresupuestoNoAgotado. En este caso será la clase FabricaDePiezas, atributo de EstadoPresupuestoNoAgotado, la que se encargue de crear la nueva pieza.

7.4. Ubicacion

Tiene coordenadas X e Y, que permiten ubicar a la pieza en el tablero.

7.5. Pieza

Pieza es una clase abstracta, de la que heredan las entidades que aparecen en el juego. Todas tienen un costo, puntos de vida, ubicación y pertenecen a un equipo. Se usó el patrón Strategy para modelar el comportamiento de las piezas, su movimiento(SeMuevenEnTodasDirecciones o SinMovimientos), cuánto daño reciben(DanioZonaEnemiga o DanioZonaPropia), cómo se curan(SanaciónNormal o SinSanación).

7.6. PiezaAtacante

Clase abstracta. De ésta heredan Jinete, Catapulta e Infantería.Utilizan una Interfaz ModoAtaque para saber si deben usar el daño cuerpo a cuerpo o daño a distancia).

7.7. Batallón

Es una clase que se crea para que cuando 3 Infanterías estén contiguas se puedan mover todas juntas. Contiene un AnalizadorDeBatallon, donde se evalúa que efectivamente se trate de 3 Infanterías, que estas sean aliadas y que estén contiguas. También contiene un objeto de la clase Ordenamientos, en la que se realizan los cálculos necesarios para determinar en qué orden se tienen que mover las piezas que componen el Batallón.

7.8. IModoRecibirDanio

Interfaz para calcular el daño que debe recibir la pieza. DanioZonaPropia devuelve el danioBase que le pasan por parámetro, DanioZonaEnemiga devuelve el danioBase múltiplicado por un valor.

7.9. IModoSanacion

Interfaz para calcular cuántos puntos de vida se debe sumar a las piezas. Si esa pieza no se puede curar, caso de la Catapulta, salta una exception.

7.10. FaseDePartida

Cada partida tiene almacenada la fase en la que se encuentra actualmente. FaseDePartida es una interfaz, de la que implementan FaseInicial y FaseMedia. En la primera cada jugador configura su ejército y en la segunda se desarrollan los movimientos y ataques entre ambos jugadores, hasta que uno resulte vencedor.

8. Excepciones

CompraInvalidaException Sucede cuando en la fase inicial se pretende comprar una pieza que es más cara que el presupuesto actual del jugador. Se la atrapa cuando en el controlador se quiere comprar una nueva pieza.

CurandoAEnemigoException Cuando un curandero intenta curar a una pieza enemiga.

CurandoCuraADistanciaCortaException Cuando un curandero intenta curar a una pieza a una distancia que no es corta. Se notifica que la distancia de curación es inválida.

DesplazamientoInvalidoException Sucede cuando se pretende mover a una pieza en un rango mayor a una casilla. Se notifica que las piezas sólo se pueden mover de a un casillero.

DistanciaDeAtaqueInvalidaException Cuando se le pide a una pieza atacar a una distancia fuera del rango de ataque de la misma.

EjercitoIncompletoException Cuando en la fase de armado del ejército, el jugador quiere terminar su turno sin haber gastado todos los puntos de su presupuesto. En esta situación, se le notifica al jugador que tiene que seguir comprando piezas hasta agotar su presupuesto.

JugadorNoPuedeManipularEsaPiezaException Cuando un jugador selecciona una pieza del otro jugador. Se le informa que no puede manipular esa pieza.

JugadorYaRealizoLaAccionException Sucede cuando un jugador ya realizó todas las acciones posibles en su turno.

NoHayUnidadEnPosicionException Cuando se clickea en una posición donde no hay ninguna pieza. Se informa que no se ha seleccionado ninguna unidad.

NoSePuedeMoverException Cuando el jugador intenta mover una pieza que no se puede mover.

NoSirvenParaBatallonException Cuando las piezas seleccionadas no cumplen los requisitos para formar un batallón.

PiezaAliadaNoAtacableException Cuando se intenta atacar a una pieza del mismo ejército. Se notifica que no está permitido.

PiezaFueraDeSectorException Cuando en la fase inicial se quiere colocar una pieza en el sector enemigo. Se notifica que se tiene que ubicar en el sector propio.

PresupuestoAgotadoException Cuando el jugador quiere seguir agregando piezas y su presupuesto está agotado. Se le avisa que no puede sumar más piezas.

UbicacionInvalidaException Cuando se quiere mover una pieza a una ubicación que ya está ocupada.

UnidadEstaMuertaException Cuando el jugador quiere manipular una pieza que murió. Se le informa que es imposible utilizarla.