

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III
Curso 1
Primer cuatrimestre de 2019

Alumnos:	BERTOLOTTO, Francisco FERNANDEZ, Andrés SANTONI, Mauro SCARFIELLO, Nicolás
Padrones:	102671 102220 102654 102090
Emails:	fbertolotto@fi.uba.ar andyfer91@gmail.com mausantoni@gmail.com nramirez@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Detalles de implementación	2
3.1. Paquete Jugador	3
3.2. Paquete Juego	3
3.3. Paquete Tablero	3
3.4. Paquete Posicionable	3
3.5. Paquete Material	3
3.6. Paquete Herramientas	3
3.7. Paquete Desgaste	4
3.8. Casos específicos	4
4. Diagramas de clase	4
5. Diagramas de secuencia	6
6. Diagramas de paquetes	13
7. Excepciones	13

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 un Minecraft simplificado en Java utilizando los conceptos del paradigma de la orientación a objetos.

2. Supuestos

En base a las pruebas dadas por la cátedra en el enunciado, que debimos implementar, surgieron casos de funcionamiento no abarcados por la consigna que debieron ser establecidos a nuestro criterio, entre los cuales destacan:

Desgaste de PicoFino: El material Piedra se daña con un pico genérico y el PicoFino es esencialmente un pico. Por lo tanto, cuando se golpea el material Piedra con un PicoFino, se gasta el material pero no el pico, que, según lo especificado en las pruebas a realizar, sólo se gasta al golpear Diamante .

Colisiones: Cuando el jugador intenta avanzar sobre una posición inválida, ya sea porque está fuera del mapa o hay un elemento en la nueva posición, no se realiza ningún movimiento por parte del jugador, simplemente se queda en su posición previa al intento de movimiento.

Distancia para obtener Material: Cuando el jugador intenta obtener un material del mapa, solo puede hacerlo de forma correcta si el material se encuentra a una distancia maxima de 1 (un) bloque del jugador. En caso de tener un solo material a su alrededor al accionar el boton `usar` se intentara obtener automaticamente, en caso de que haya mas de uno el usuario debera hacer click en el Material que desea obtener.

Equiparse con materiales denegado: El jugador no puede elegir un material como la herramienta a equiparse.

AutoMasoquismo denegado: El jugador no puede golpearse a si mismo.

3. Detalles de implementación

Para el desarrollo del modelo se decidió seguir varios principios SOLID.

Comenzamos con la idea de mantener el principio `open-closed`, en el cual nuestro modelo permitiera agregar fácilmente nuevas herramientas y nuevos materiales, pero que fuese cerrado a los cambios internos de cada uno de ellos. Sumado a esto, generamos varias clases abstractas, para que las diferentes Herramientas puedan heredar de ellas, siguiendo el Principio de Sustitución de Liskov, como por ejemplo: Hacha y Pico. Haciendo que cada una de éstas pueda ser reemplazada por una instancia "hija". Un ejemplo de esto, es que el Material recibe la clase Herramienta para calcular su desgaste, independientemente de si es un HachaDeMadera o un PicoDePiedra.

En relación a patrones de diseño, hasta el momento se utilizó el método "Template", el cual delega el comportamiento de una clase abstracta a sus herederas. El mejor ejemplo del uso del patrón, como mencionamos antes, podría ser el de la clase Herramienta, en el cual podemos darle un comportamiento específico a cada tipo de herramienta, manteniendo como invariante el comportamiento general que tiene una herramienta.

El patrón "State" también fue aplicado en cierta medida, ya que se abstrae el tipo de desgaste de cada herramienta de manera que si en un futuro se quisiera agregar la posibilidad de modificar de alguna forma el modo de desgaste de una herramienta (por ejemplo, templándola), no habría más que actualizar su tipo de desgaste (teniendo en cuenta que esto podría hacerse mediante encantamientos en el juego original), Además, se incluyó también, el método "Double Dispatch".^a a través de la sobrecarga de metodos en varias partes del TP, ya sea crear material en el mapa o la forma de usar una herramienta particular con un material particular.

3.1. Paquete Jugador

Clase Jugador : representa a un jugador de AlgoCraft. Posee un inventario en el que puede guardar herramientas y tiene la responsabilidad de jugar el juego y usar las herramientas.

Clase Inventario : representa al inventario de un jugador, en donde se guardan las herramientas y los materiales conseguidos.

Clase Zombie : representa a una entidad "Zombie" que se mueve a través del mapa (pequeño "easter egg").

3.2. Paquete Juego

Clase Juego : representa el juego en su totalidad, desde el mapa hasta el jugador y la interacción entre ambos.

Clase Craftero : representa la receta del craftero de una herramienta.

Clase MesaDeCraftero : representa a la mesa del juego original, encargada de poseer todas las recetas y validar los crafteros.

3.3. Paquete Tablero

Clase Tablero : representa al mapa de juego.

Clase InicializadorDeMateriales : se encarga de inicializar el mapa con materiales al azar.

Clase Posicion : representa una posición (X,Y).

Clase PosicionadorRandom : se encarga de posicionar un posicionable en una posición al azar válida.

3.4. Paquete Posicionable

Clase Abstracta Posicionable : Es la clase abstracta madre de todos los materiales y jugador (y próximamente herramientas). Permite que puedan ser ubicados en un tablero.

3.5. Paquete Material

Clase Abstracta Material : representa un material que existe en AlgoCraft. Todo material posee una durabilidad que se puede desgastar.

Clase Diamante : (hereda de Material). Representa al Diamante, con una durabilidad 100.

Clase Metal : (hereda de Material). Representa al Metal, con una durabilidad 50.

Clase Piedra : (hereda de Material). Representa a la Piedra, con una durabilidad 30.

Clase Madera : (hereda de Material). Representa a la Madera, con una durabilidad 10.

3.6. Paquete Herramientas

Clase Abstracta Herramienta : representa a una herramienta que existe en AlgoCraft. Toda herramienta tendrá una durabilidad, una fuerza y una forma de desgaste. Toda herramienta puede ser usada.

Clase Abstracta Pico : (hereda de Herramienta). Representa un Pico,

Clase PicoDeMadera : (hereda de Pico). Representa un Pico hecho con el material Madera, con una durabilidad 100, fuerza 2 y una forma de desgaste lineal cuya pendiente es 1.

Clase PicoDePiedra : (hereda de Pico). Representa un Pico hecho con el material Piedra, con una durabilidad 200, fuerza 4 y una forma de desgaste lineal cuya pendiente es $1/1.5$.

Clase PicoDeMetal : (hereda de Pico). Representa un Pico hecho con el material Metal, con una durabilidad 400, fuerza 12 y tiene una forma de desgaste con un limite de 10 usos.

Clase PicoFino : (hereda de Pico). Representa un Pico hecho con los materiales Metal y Piedra, con una durabilidad 1000, fuerza 20 y una forma de desgaste lineal cuya pendiente es $1/10$

Clase Abstracta Hacha : (hereda de Herramienta). Representa un Hacha,

Clase HachaDeMadera : (hereda de Hacha). Representa un Hacha hecho con el material Madera, con una durabilidad 100, fuerza 2 y una forma de desgaste lineal cuya pendiente es 1.

Clase HachaDePiedra : (hereda de Hacha). Representa un Hacha hecho con el material Madera, con una durabilidad 200, fuerza 5 y una forma de desgaste lineal cuya pendiente es 1.

Clase HachaDeMetal : (hereda de Hacha). Representa un Hacha hecho con el material Madera, con una durabilidad 400, fuerza 10 y una forma de desgaste lineal cuya pendiente es $1/2$.

3.7. Paquete Desgaste

Clase Abstracta Desgaste : Representa una forma de desgaste, tiene la responsabilidad de calcular como se va a desgastar la herramienta.

Clase DesgasteLineal : (hereda de Desgaste). Representa un desgaste en el cual la herramienta agotará su durabilidad según determinada pendiente.

Clase DesgastePorUso : (hereda de Desgaste). Representa un desgaste en el cual la herramienta agotará su durabilidad cuando llegue a su límite de usos.

3.8. Casos específicos

Prueba AgotarDurabilidad PicoFino: La prueba que verifica el lanzamiento de excepción al agotar la durabilidad de PicoFino no se realizó, ya que al reducirse la misma un 10 por ciento cada vez que es usada, nunca llega a 0, siendo ésta la condición para el lanzamiento de la excepción.

Colisiones: En base al supuesto de las colisiones, se implementó de manera que la excepcion levantada al intentar mover el jugador se atrape la excepcion en la clase Tablero y que ésta la maneje como deba (en este caso no realiza nada ya que el jugador se queda en su posición original).

4. Diagramas de clase

Los siguientes diagramas de clases abarcan la totalidad de las clases implementadas y sus relaciones. Las clases poseen los métodos y atributos más relevantes para una visión general del proyecto.

Diagrama General del Modelo

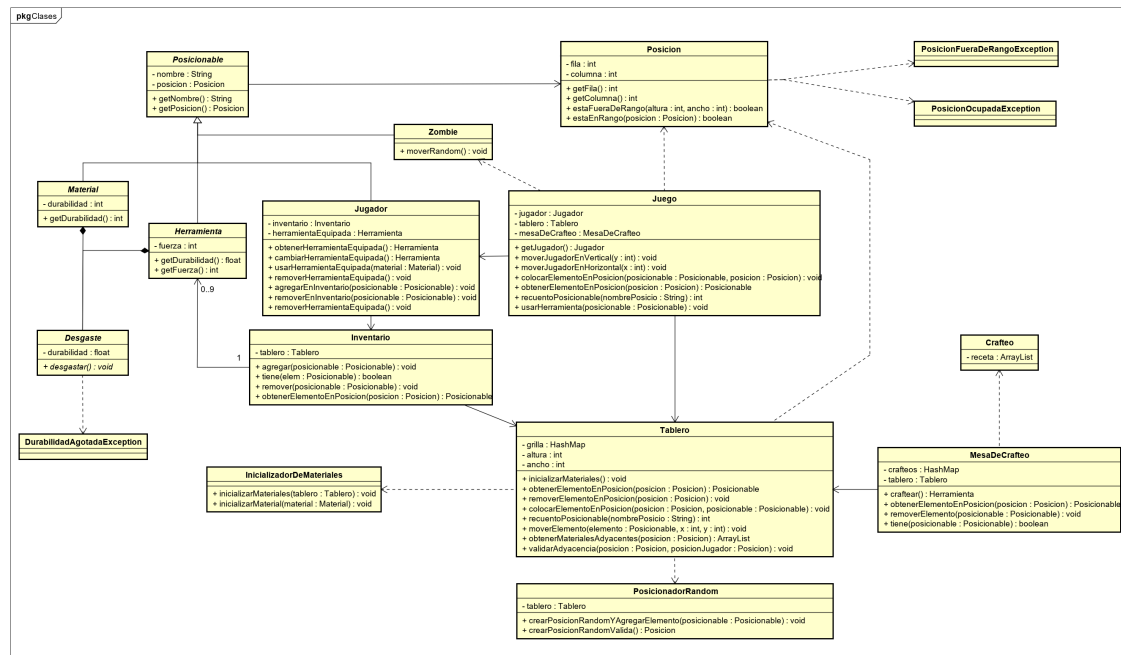


Diagrama de Herramientas y Materiales

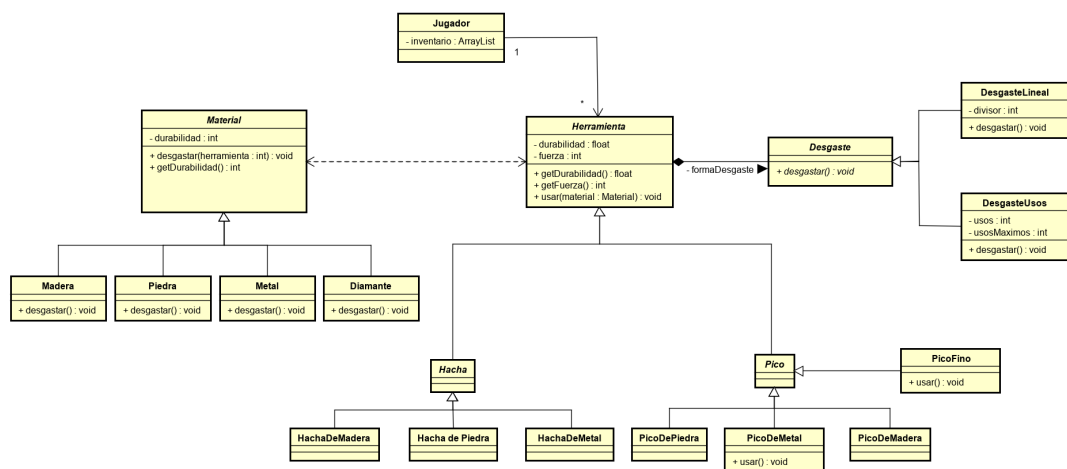
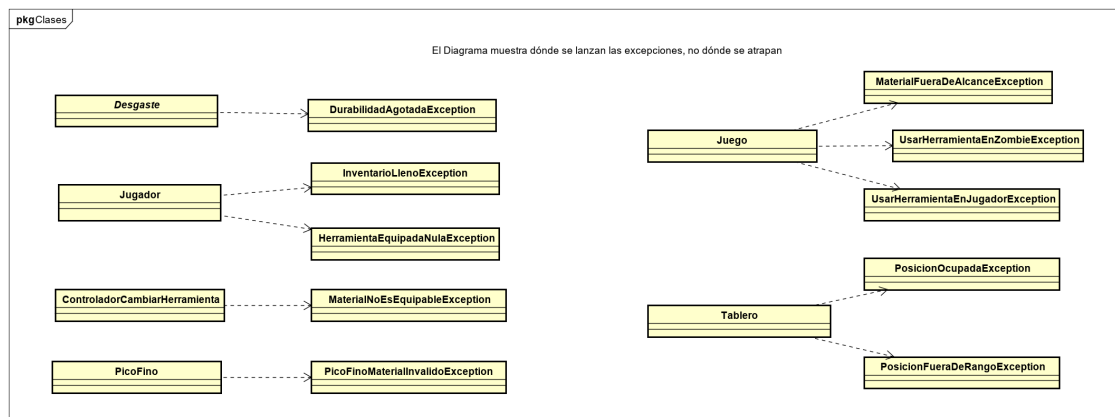


Diagrama de Excepciones



5. Diagramas de secuencia

Para el diseño de diagramas de secuencia, teniendo en cuenta que sólo se han implementado ciertas funciones básicas del sistema, se eligió la secuencia que consideramos más abarcativa y la cual pensamos que mejor reflejaba las primeras interacciones entre las diferentes clases del sistema.

EJEMPLOS DE SOBRECARGA DE MÉTODOS

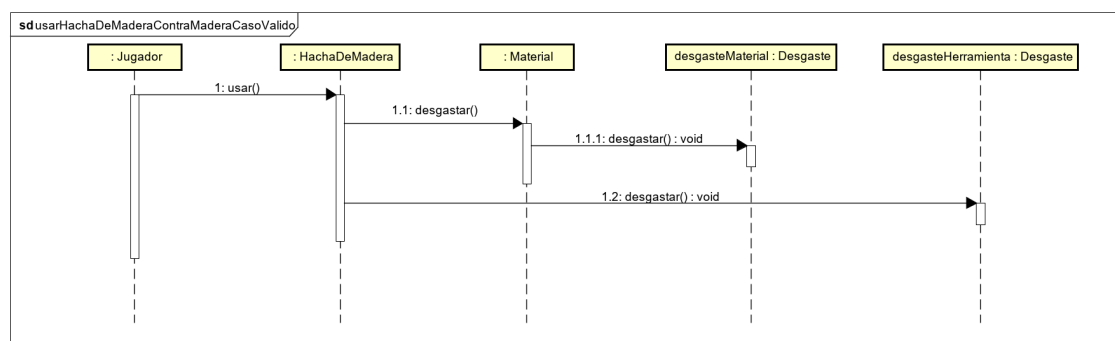


Figura 1: Diagrama de secuencia en el que el jugador utiliza un Hacha (no desgastada) contra Madera, se desgastan ambos.

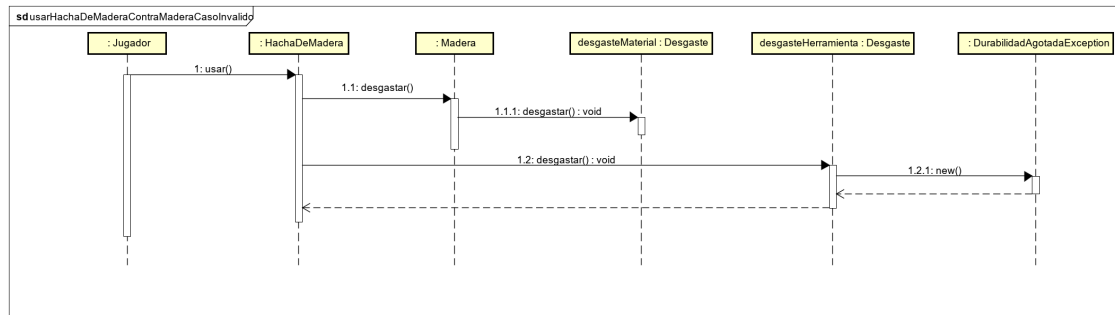


Figura 2: Diagrama de secuencia en el que el jugador utiliza un Hacha (desgastada) contra Madera, se desgastan ambos.

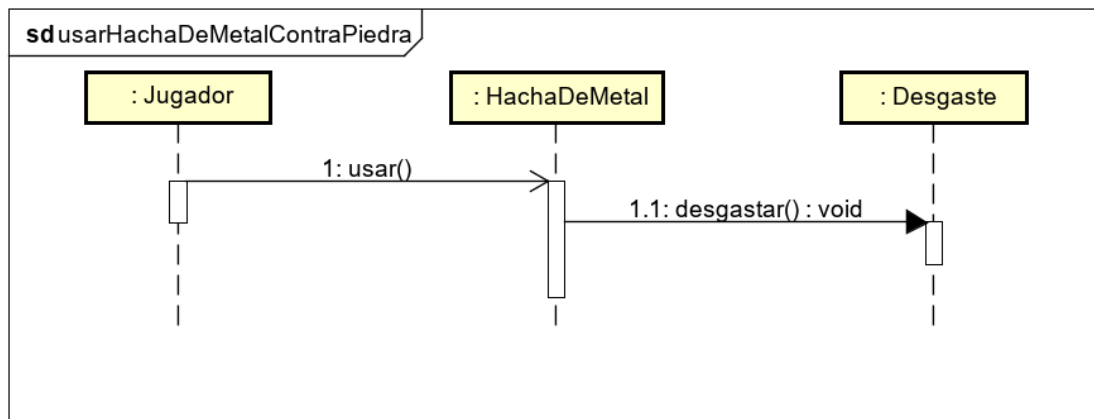


Figura 3: Diagrama de secuencia en el que el jugador utiliza un Hacha de Metal contra Piedra, sólo se desgasta el hacha.

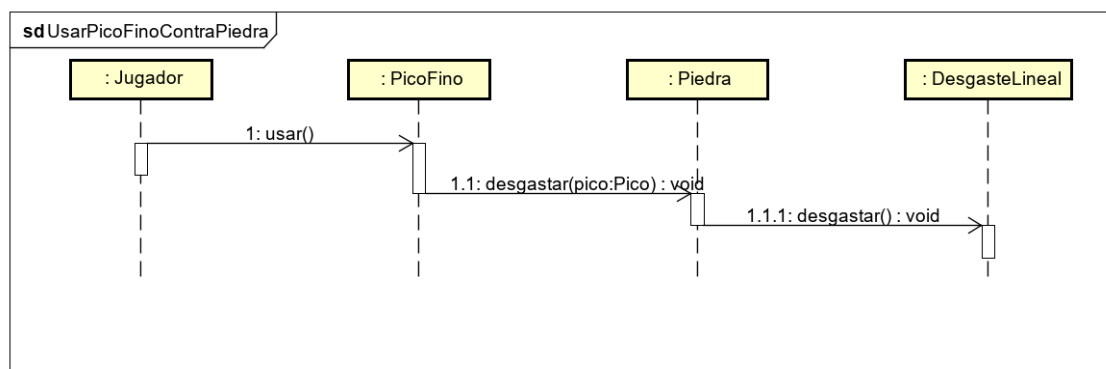


Figura 4: Diagrama de secuencia en el que el jugador utiliza un Pico Fino contra Piedra, sólo se desgasta la Piedra.

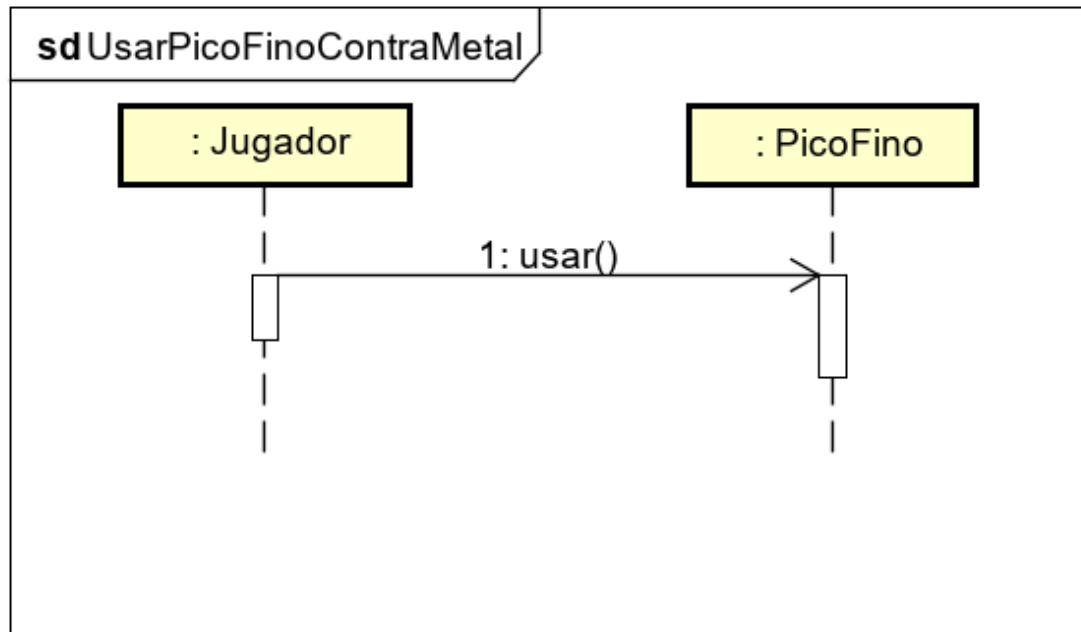


Figura 5: Diagrama de secuencia en el que el jugador utiliza un Pico Fino contra Metal, no se desgasta ninguno.

DIAGRAMAS DE PRUEBAS

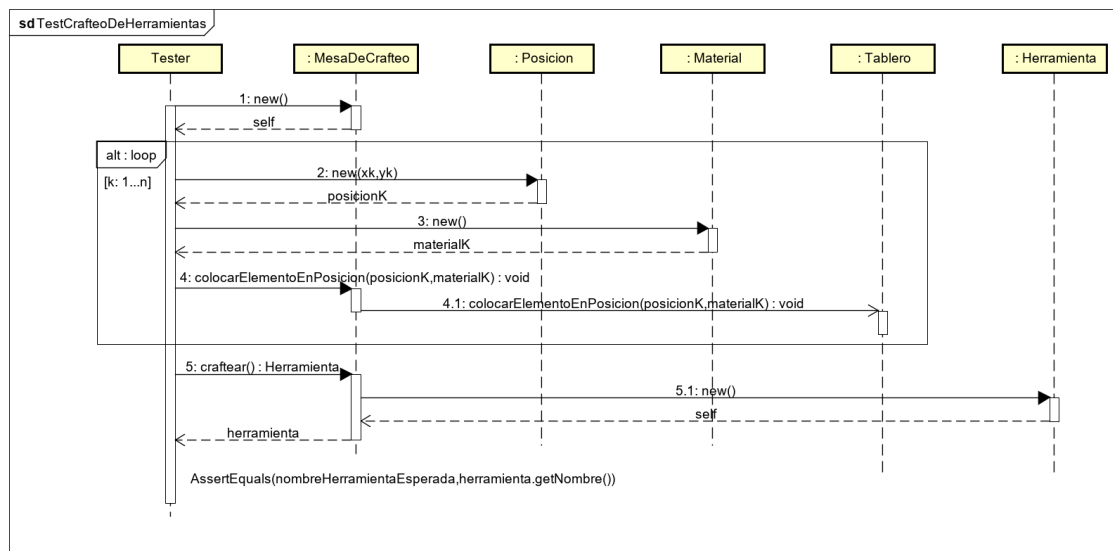


Figura 6: Diagrama de secuencia que muestra la prueba del crafeo de una Herramienta.

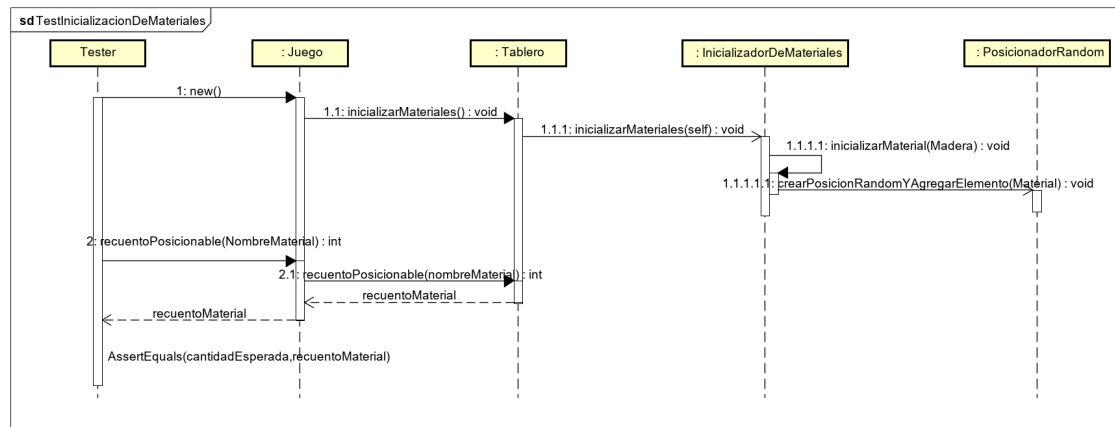


Figura 7: Diagrama de secuencia que muestra la prueba de inicialización de materiales.

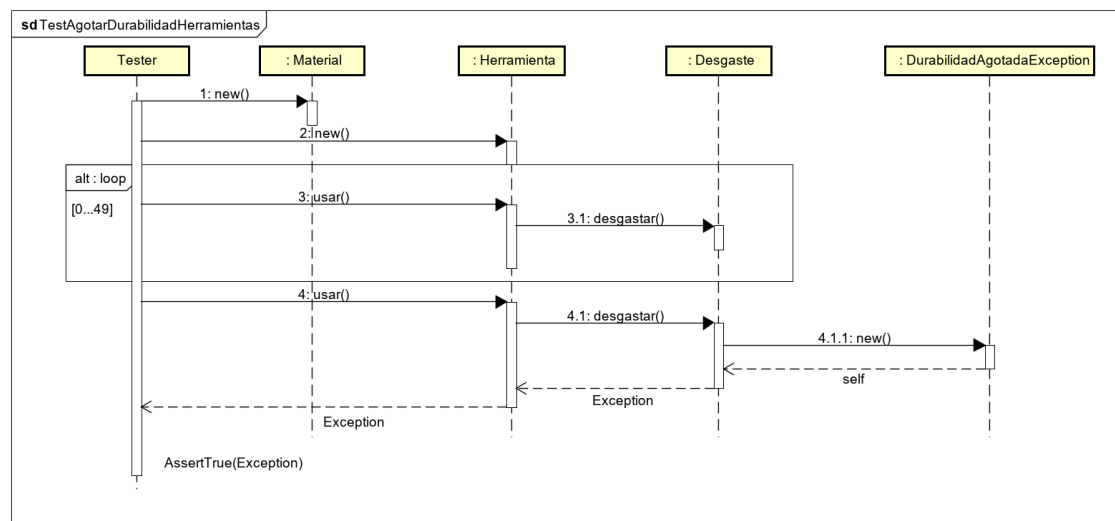


Figura 8: Diagrama de secuencia que muestra la prueba en el que se ahota la durabilidad de una herramienta y se lanza una excepción.

SECUENCIAS QUE SE INICIAN DESDE EL USUARIO

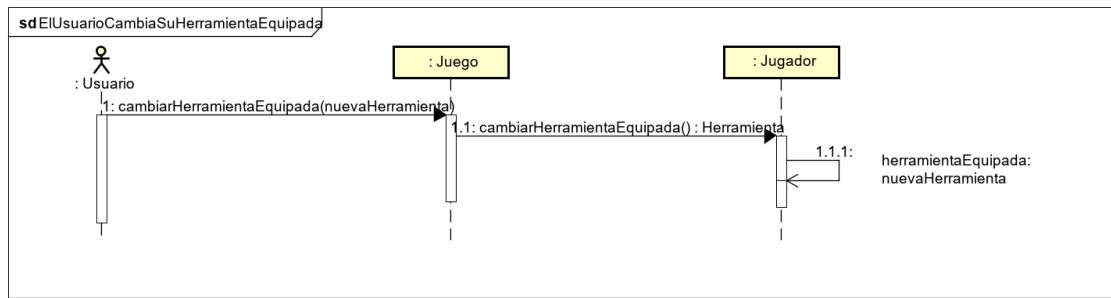


Figura 9: Diagrama de secuencia en el que el jugador cambia su herramienta equipada.

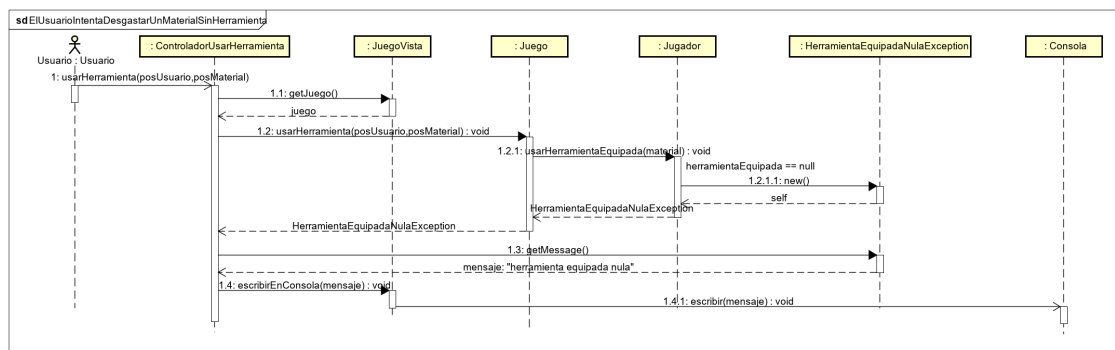


Figura 10: Diagrama de secuencia en que el usuario intenta desgastar un material, pero no tiene una herramienta equipada (levanta excepción).

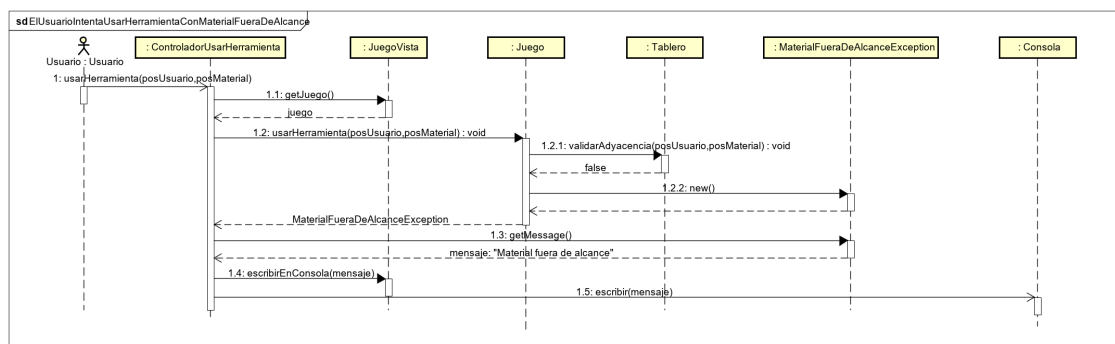


Figura 11: Diagrama de secuencia en que el usuario intenta desgastar un material fuera de su rango (levanta excepción).

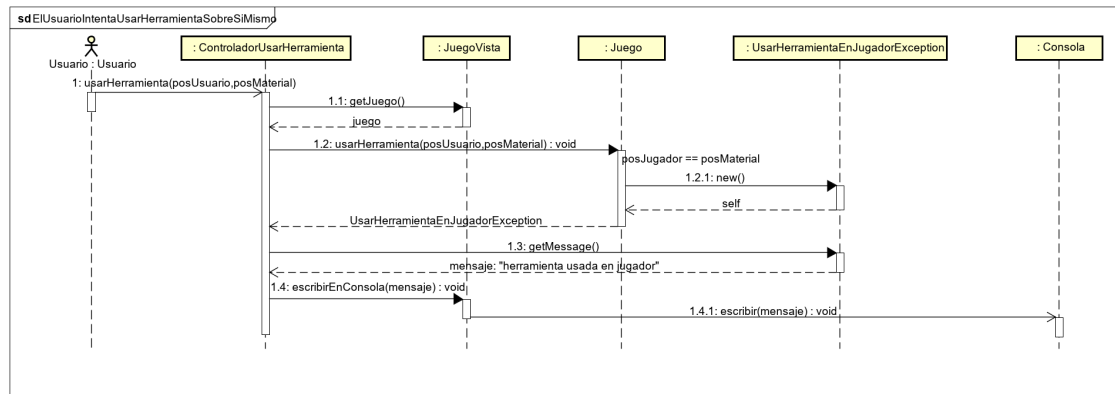


Figura 12: Diagrama de secuencia en que el usuario intenta usar una herramienta sobre sí mismo (levanta excepción).

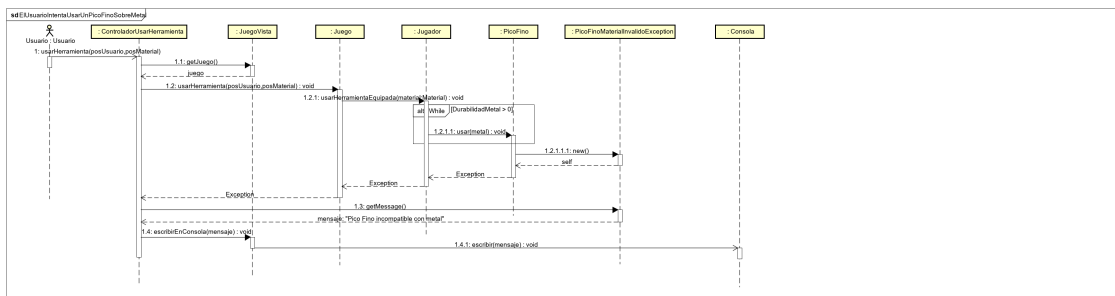


Figura 13: Diagrama de secuencia en que el usuario intenta usar un pico fino sobre metal (levanta excepción para evitar loop infinito).

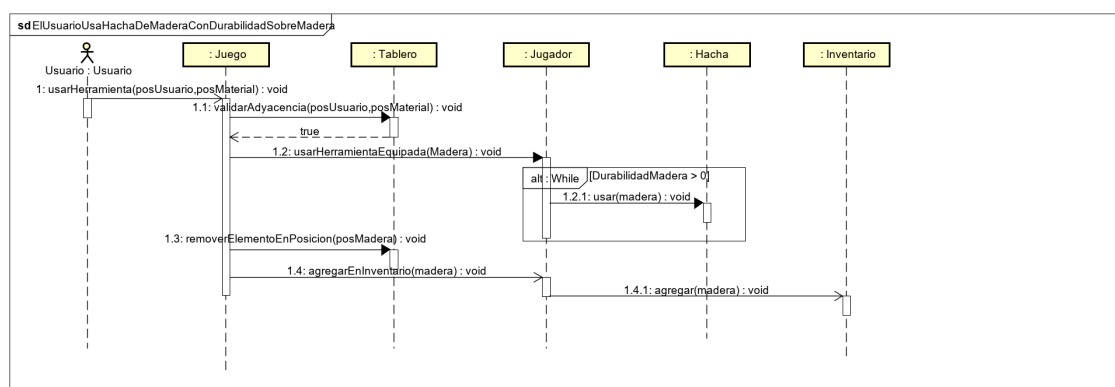


Figura 14: Diagrama de secuencia en que el usuario desgasta madera con un hacha, y agrega el bloque obtenido a su inventario.

OTRAS SECUENCIAS

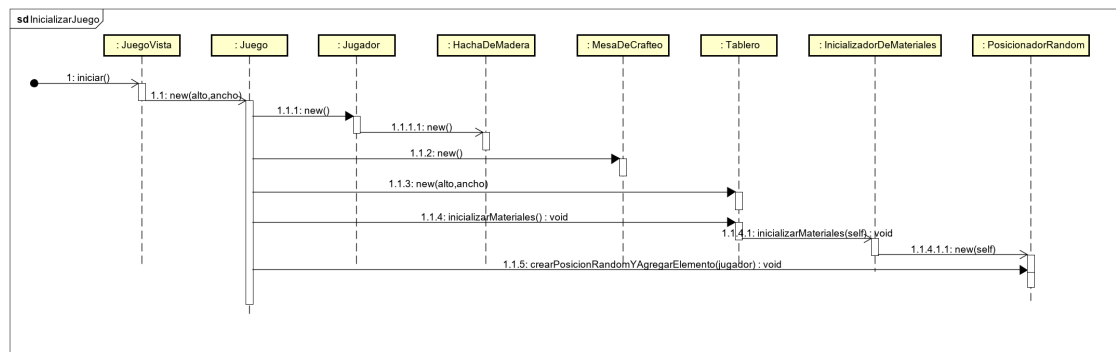


Figura 15: Diagrama de secuencia en el que se inicia el juego.

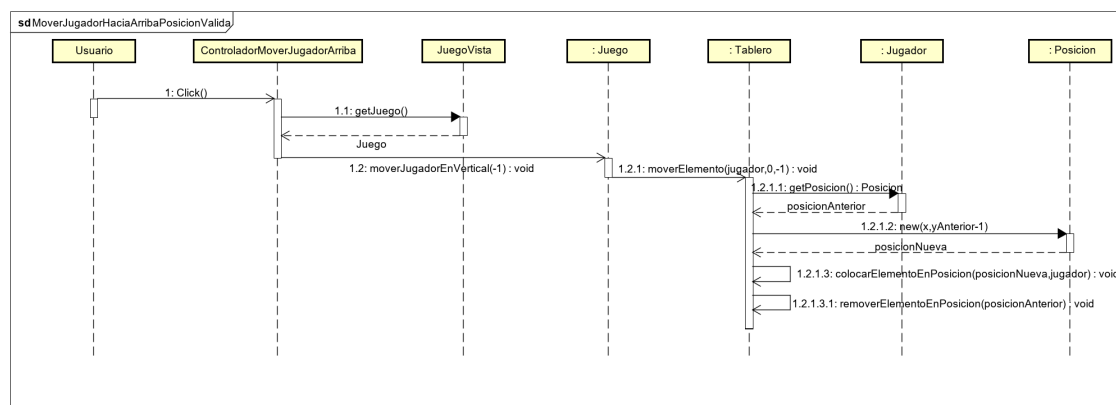


Figura 16: Diagrama de secuencia en el que el usuario mueve al jugador hacia arriba, a una posición válida.

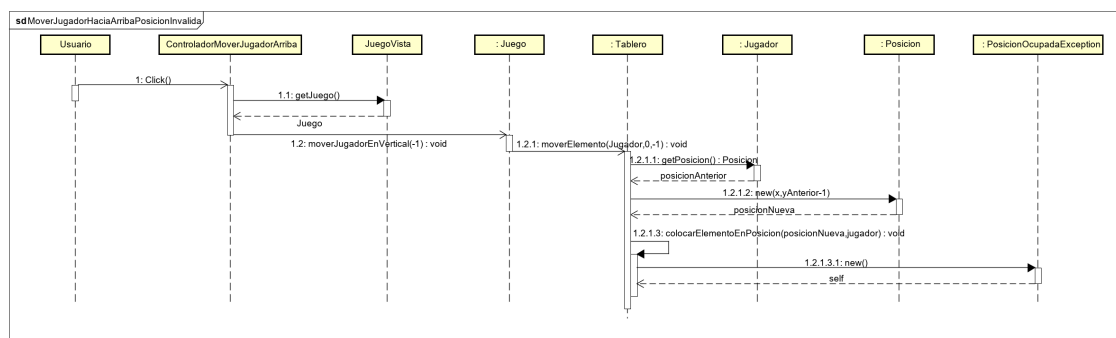


Figura 17: Diagrama de secuencia en el que el usuario mueve al jugador hacia arriba, a una posición inválida.

6. Diagramas de paquetes

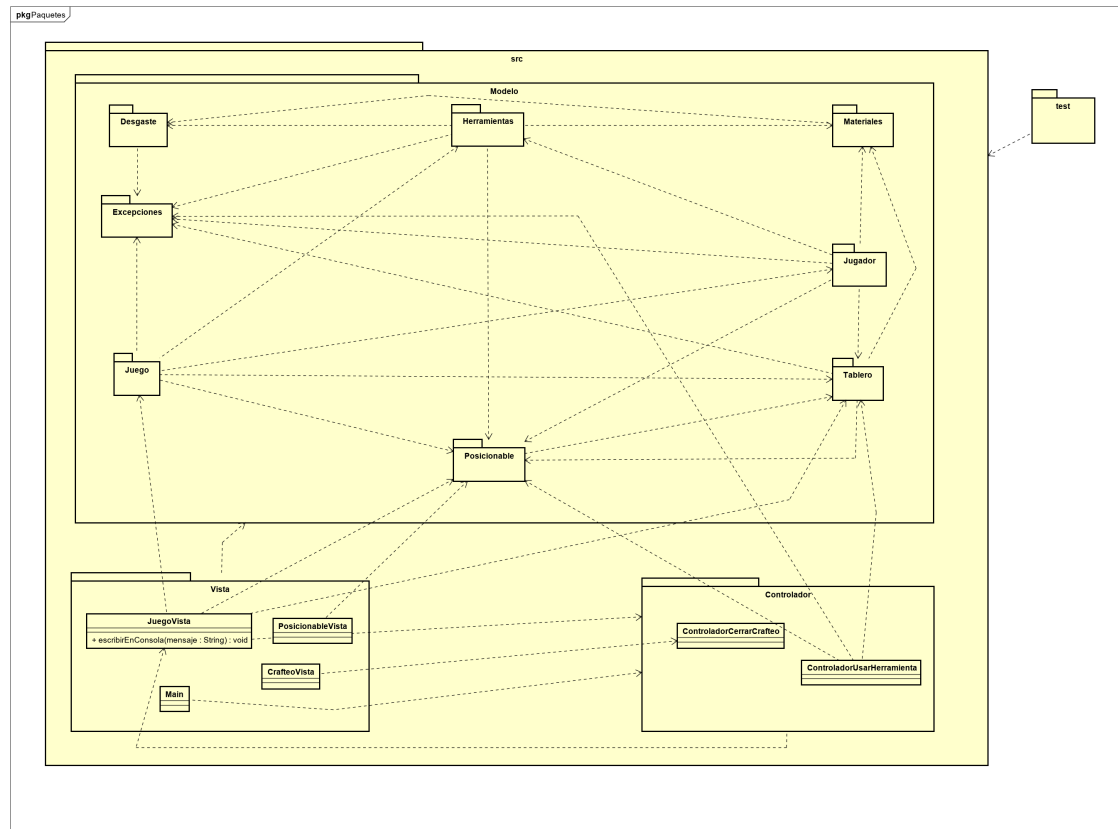


Figura 18: Diagrama de Paquetes.

7. Excepciones

DurabilidadAgotadaException Esta excepción tiene como finalidad lanzar un error cuando se intente utilizar una herramienta luego de que su durabilidad haya llegado a 0 (cero). De esta manera la herramienta tendrá un límite en su durabilidad, no pasará a valores negativos y funcionará como una herramienta análoga al Minecraft al agotar su vida útil.

HerramientaEquipadaNulaException Esta excepción tiene como finalidad lanzar un error cuando se intente usar la herramienta equipada y no haya ninguna (cuando se rompe y no se reemplaza por otra).

InventarioLlenoException Esta excepción tiene como finalidad lanzar un error cuando se intente colocar un material en el inventario, cuando ya este lleno.

MaterialFueraDeAlcanceException Esta excepción tiene como finalidad lanzar un error cuando se intente obtener un Material del mapa que este a mas de 1 (un) bloque de distancia del jugador.

MaterialNoEsEquipableException Esta excepción tiene como finalidad lanzar un error cuando se intente equipar un Material como la herramienta equipada que utiliza el jugador para obtener Materiales del mapa.

PicoFinoMaterialInvalidoException Esta excepción tiene como finalidad lanzar un error cuando se intente usar el Pico Fino contra un Material invalido, es necesaria ya que al tener durabilidad infinita nunca terminaria de desgastarse si se golpea el Material incorrecto.

PosicionFueraDeRangoException Esta excepción tiene como finalidad lanzar un error cuando se intente acceder a una posicion por fuera de los limites del tablero.

PosicionOcupadaException Esta excepción tiene como finalidad lanzar un error cuando se intente ocupar una posicon en la cual ya se encuentra un elemento.

UsarHerramientaEnJugadorException Esta excepción tiene como finalidad lanzar un error cuando se intente usar la herramienta equipada por el jugador contra si mismo.

UsarHerramientaEnZombieException Esta excepción tiene como finalidad lanzar un error cuando se intente usar la herramienta equipada por el jugador contra el "bonus"zombie.