



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
Año 2019 - 1.^{er} cuatrimestre

ALGORITMOS Y PROGRAMACIÓN III (75.07)

Informe de Trabajo Práctico 2

GRUPO N7:

Carrero Riveros, Maria Daniela <dcarrero@fi.uba.ar>	- #99316
Fullone, Josefina <fullonejose@hotmail.com>	- #98374
Torres Baivads, Jesus Enrique <Jesusvaivads@hotmail.com>	- #102909
Garcia, Matáis Ezequiel <megarcia@fi.uba.ar>	- #101274

Índice

1. Objetivo	2
2. Introducción	2
3. Supuestos	2
3.1. Tablero	2
3.2. Herramientas	2
3.3. Jugador	2
3.4. Materiales	3
4. Diagramas de Clases	3
5. Diagrama de paquetes	8
6. Detalles de Implementación	8
6.1. Herramientas	8
6.2. Materiales	8
6.3. Mapa	8
6.4. Jugador	9
6.5. Juego	9
7. Excepciones	9
7.1. DurabilidadTerminadaException	9
7.2. HerramientaRotaException	9
7.3. FormaInvalidaException	9
7.4. PosicionInvalidaException	9
7.5. PosicionOcupadaException	9
7.6. LugarOcupadoException	9
7.7. SinEquipoExcepcion	9
8. Diagramas de Secuencias	10
8.1. Secuencias Test Materiales	10
9. Diagramas de estado	14
10. Patrones de diseño	14
11. Conclusión	15

1. Objetivo

Desarrollar una aplicación de una versión simplificada del juego Minecraft utilizando el lenguaje de programación Java, con un diseño del modelo orientado a objetos y utilizando las técnicas de TDD e Integración Continua.

2. Introducción

El programa realizado consta de un jugador/personaje, que representa al usuario, el cual puede moverse libremente dentro de un tablero (2D) y recolectar materiales utilizando sus herramientas. Las herramientas y materiales que posee el jugador se encuentran en un inventario de herramientas y materiales respectivamente.

3. Supuestos

3.1. Tablero

Es el ambiente principal donde desarrolla el juego, está diseñado en base a una cuadrícula de tamaño 11x11, donde cada celda puede ser ocupado a lo sumo por un elemento, el cual corresponde al jugador o a un material. Al inicio del juego los materiales se encontrarán posicionados en la mitad del mapa respecto a los bordes superiores e inferiores y respecto a los bordes laterales. Los materiales son los siguientes:

- 10 maderas
- 5 piedras
- 5 metales
- 1 diamante

3.2. Herramientas

Inicialmente el jugador posee un hacha de madera en su inventario de herramientas. El resto, solo se pueden obtener construyéndolas, y para construirlas el jugador debe tener los materiales suficientes, para luego posicionarlas en el constructor de herramientas cumpliendo con el patrón de posiciones correspondiente a cada tipo de herramienta. Las herramientas pueden visualizarse en el inventario de herramientas del jugador. Este, a su vez, puede seleccionar en cualquier momento la herramienta que desea utilizar dependiendo el material que desea obtener.

3.3. Jugador

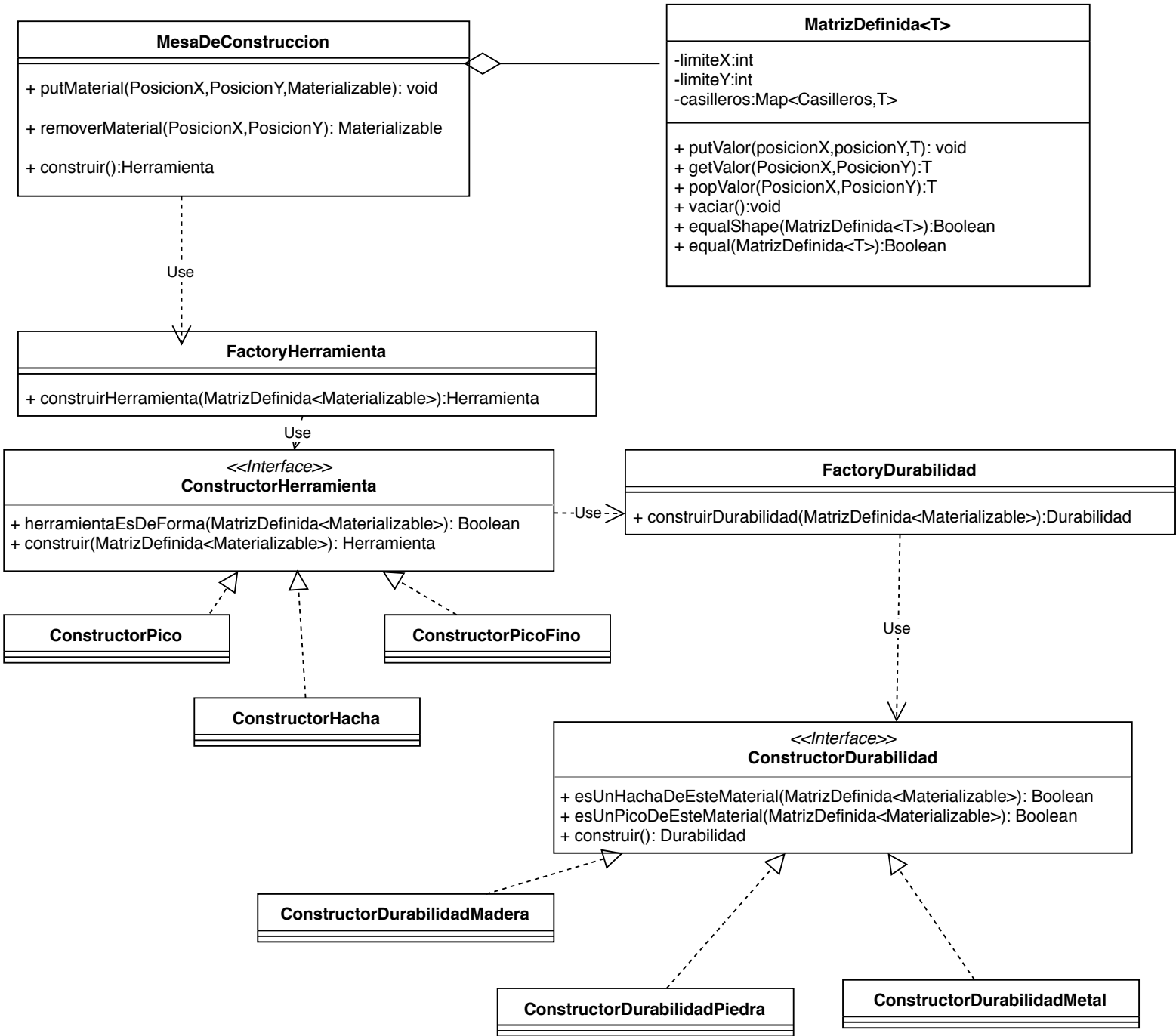
El jugador siempre, al iniciarse un juego nuevo comienza ubicado en el extremo superior izquierdo del mapa. El mismo se inicializa con un hacha de madera en su inventario de herramientas. Para recolectar materiales, el jugador debe posicionarse en una celda adyacente al material y debe estar orientado en la dirección de este último (y así podrá usar la herramienta contra el material). Luego de desgastar completamente al material entonces el jugador podrá posicionarse sobre este y de forma automática pasará a formar parte del inventario de materiales.

3.4. Materiales

Para el comienzo del desarrollo de la aplicación suponemos que los materiales que presenta el juego se ubican en el tablero siempre de igual manera y no se regeneran una vez que el jugador logra conseguirlos. Es decir, para comenzar suponemos que el juego comienza con 10 Maderas, 5 Piedras, 5 Metales y un Diamante. A medida que el jugador va a obteniendo los materiales los mismos no vuelven a generarse por lo que llegara un momento en el que tablero se quede sin materiales para recolectar. Llegado este punto se considera que el jugador ha "ganado", teniendo que reiniciar el juego.

Los materiales se ubican en forma de cruz a la mitad del tablero.

4. Diagramas de Clases



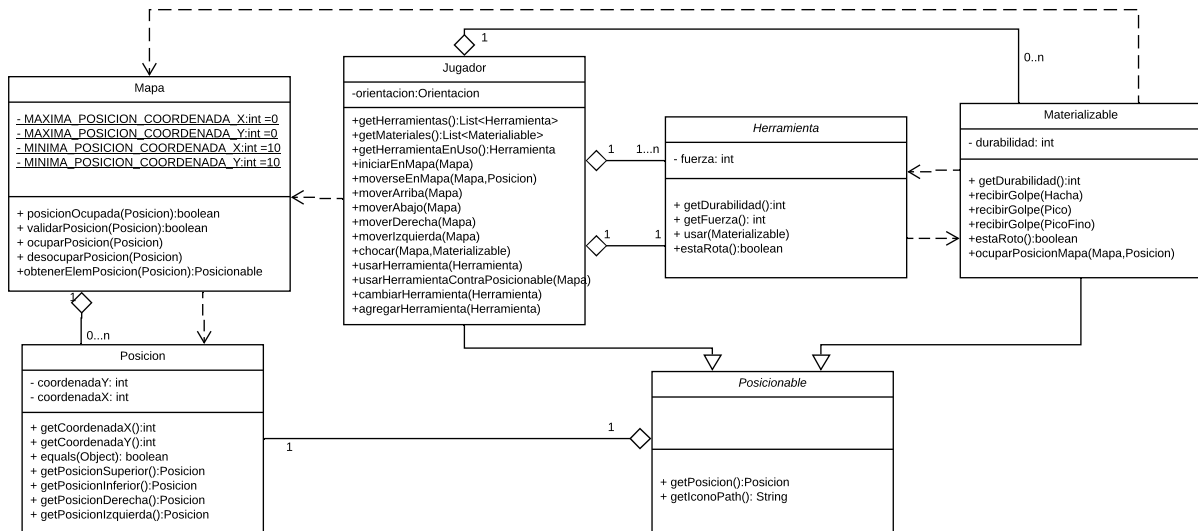


Figura 1. Diagrama de clases: Herramientas y Materiales

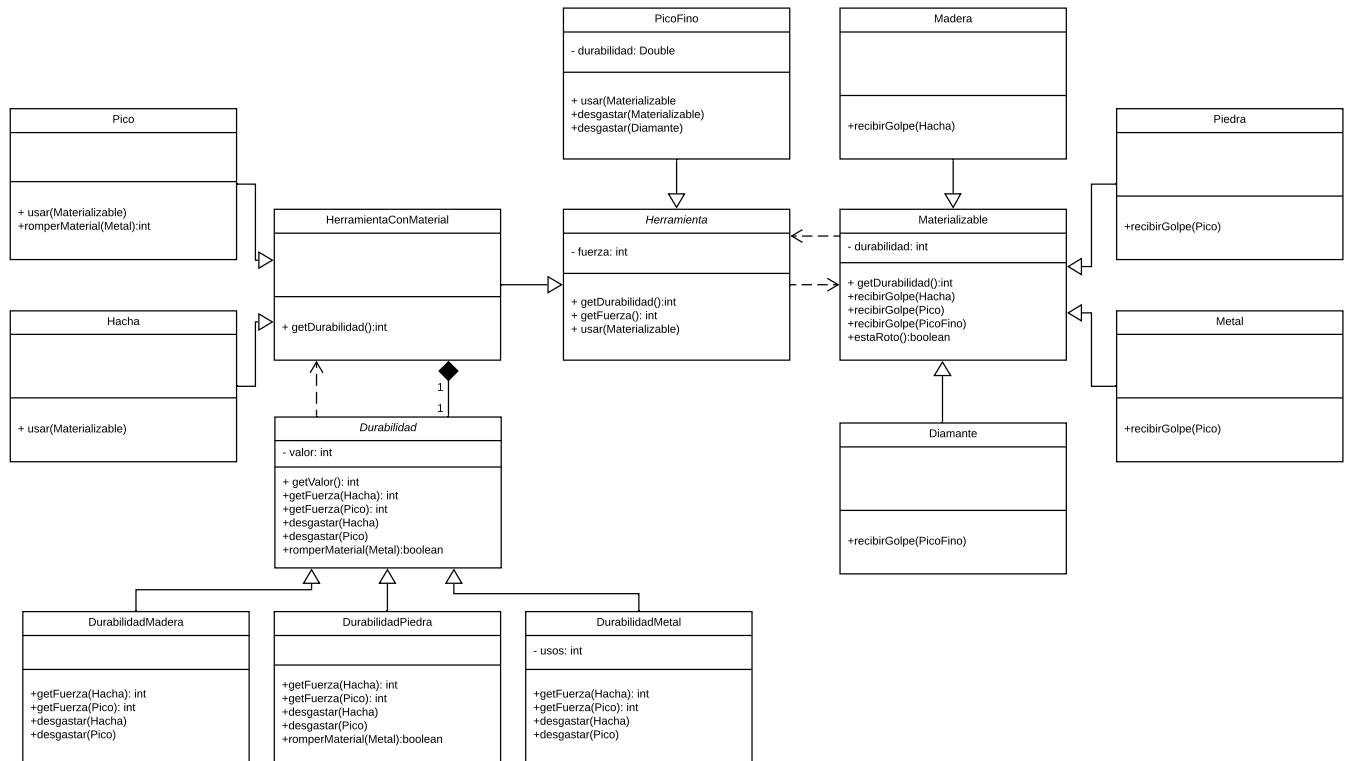


Figura 2. Diagrama de Clases: Jugador y Mapa

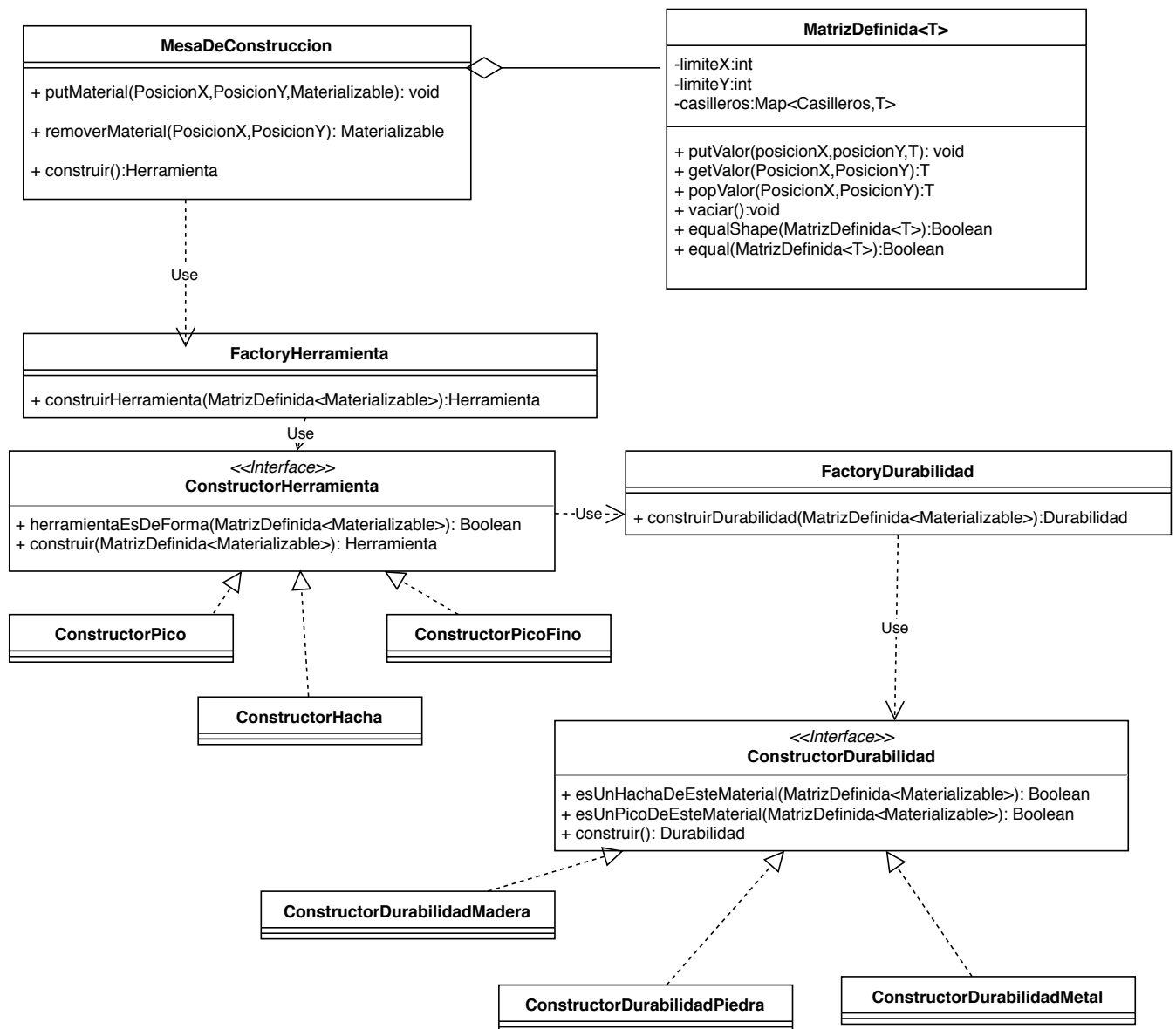


Figura 3. Diagrama de Clases: Mesa de construcción

5. Diagrama de paquetes

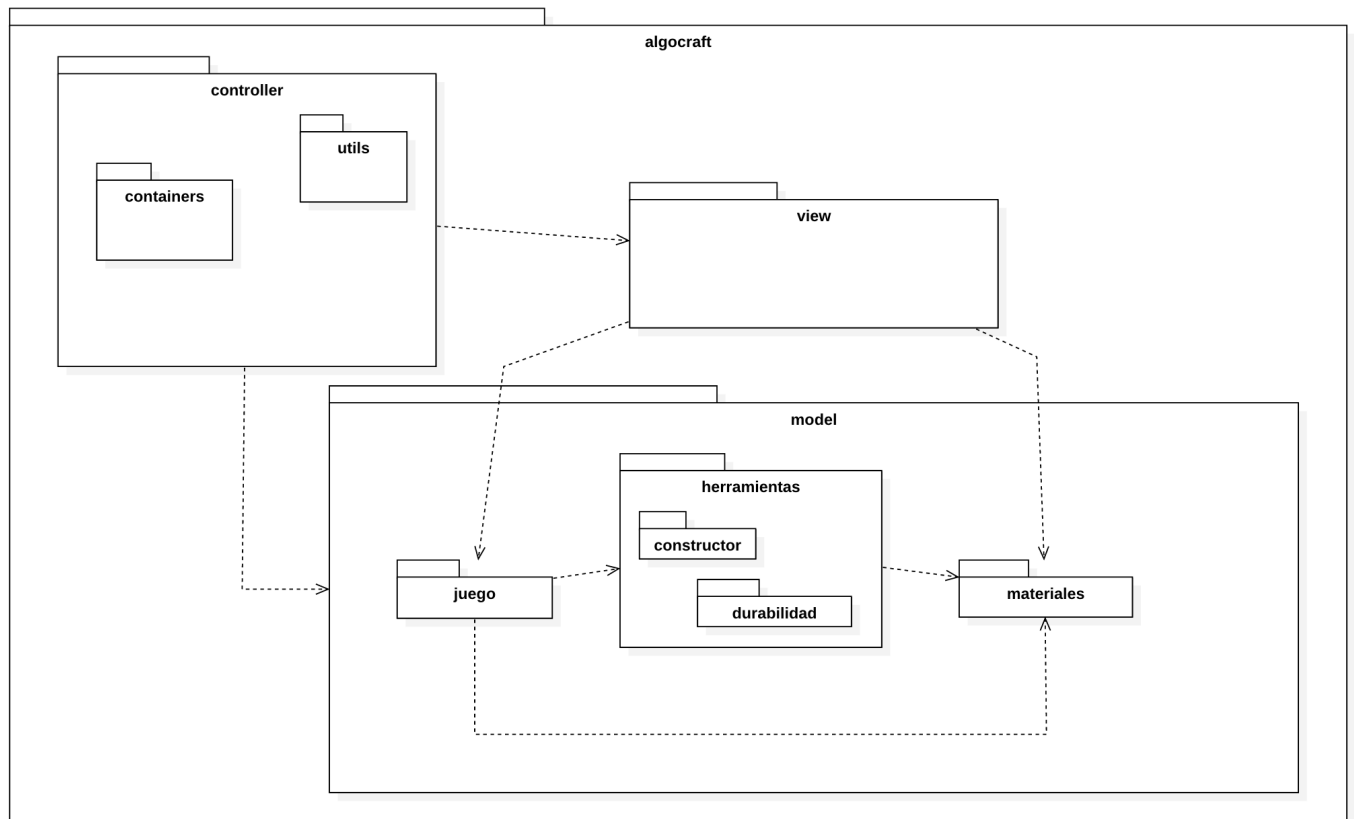


Figura 4. Diagrama de Paquetes

6. Detalles de Implementación

6.1. Herramientas

Con excepción del pico fino, todas las herramientas poseen un tipo de durabilidad que es caracterizada por el tipo de material con el cual fueron elaboradas, de modo que la mayor parte de los mensajes que son enviados a una herramienta, son delegados a su durabilidad. Para instanciar un pico o una hacha es necesario añadir al parámetro del constructor un tipo de durabilidad correspondiente al tipo de material de la herramienta.

6.2. Materiales

Cada tipo de material es un Materializable, y el comportamiento de cada tipo específico se caracteriza por sobrescribir un método llamado recibirGolpe.

6.3. Mapa

Se implementó con un hash que utiliza como clave posiciones y solo puede guardar elementos posicionables (jugador o un material), es decir, que las posiciones existentes en este hash corresponde a las posiciones ocupadas en el mapa, por lo tanto no pueden ser usados como clave de

nuevo. también define 4 valores que representan los límites de las coordenadas de las posiciones, los cuales usa para validar posiciones correctas del tablero.

6.4. Jugador

Utiliza listas para guardar los materiales y las herramientas. adicionalmente tiene una herramienta en uso y posee una orientación la cual está implementada en función de cuatro posibles valores asociados a los puntos cardinales. Para instanciar un jugador es necesario pasarle por parámetro al constructor la herramienta inicial con la que contará al iniciar.

6.5. Juego

Este se inicializa pasando por los parámetros del constructor el mapa y el jugador deseado.

7. Excepciones

7.1. DurabilidadTerminadaException

Al intentar usar una herramienta cuya durabilidad está desgastada (es cero), entonces se lanza esta excepción (es lanzada por la durabilidad)

7.2. HerramientaRotaException

La forma de manejar la excepción anterior es precisamente diciéndole a la herramienta con durabilidad desgastada que lance esta excepción cuando se intenta usar.

7.3. FormaInvalidaException

La construcción de herramientas en la mesa de construcción implica posicionar correctamente los materiales correspondientes en las celdas del recuadro 3x3, por lo tanto cualquier otra posibilidad de combinación no coincidente con las ya preestablecidas genera esta excepción.

7.4. PosicionInvalidaException

El mapa valida posiciones que se supone que pueden ser parte del mapa, cuando estas posiciones no están dentro de los límites que establece el mapa entonces es lanzada esta excepción

7.5. PosicionOcupadaException

El jugador puede desplazarse por todas las posiciones adyacentes a la suya siempre y cuando estas se encuentren vacías. cuando intenta ocupar una no vacía entonces se lanza esta excepción.

7.6. LugarOcupadoException

al intentar ocupar una celda de la mesa de construcción que ya está ocupada se lanza esta excepción

7.7. SinEquipoExcepcion

Decirle al jugador que use una herramienta sin tener ninguna en uso genera esta excepción

8. Diagramas de Secuencias

8.1. Secuencias Test Materiales

En las siguientes secuencia se muestra como funciona la implementación de materiales al recibir un golpe de una herramienta. En los ejemplos se puede ver una Madera golpeada por un hacha, un pico y un pico fino, y en otro un diamante golpeado por las mismas herramientas

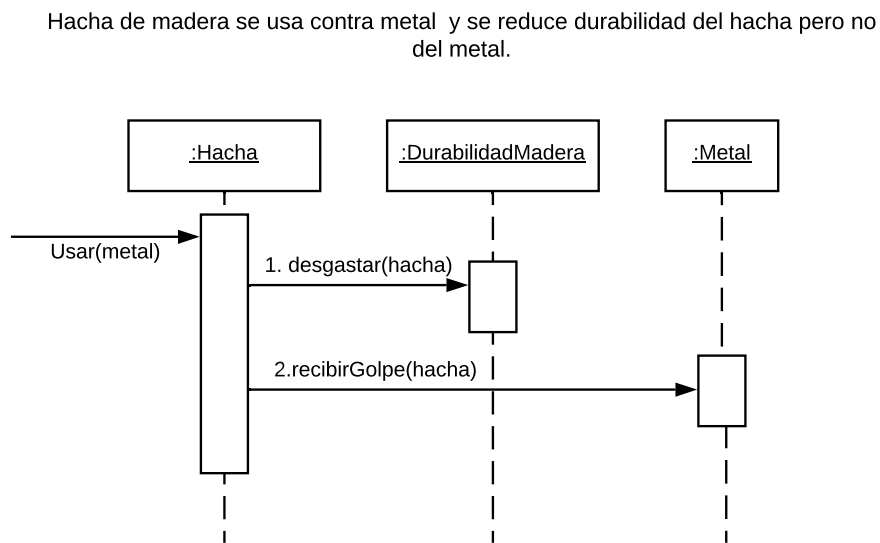


Figura 5. Diagrama de Secuencia: Hacha golpea metal

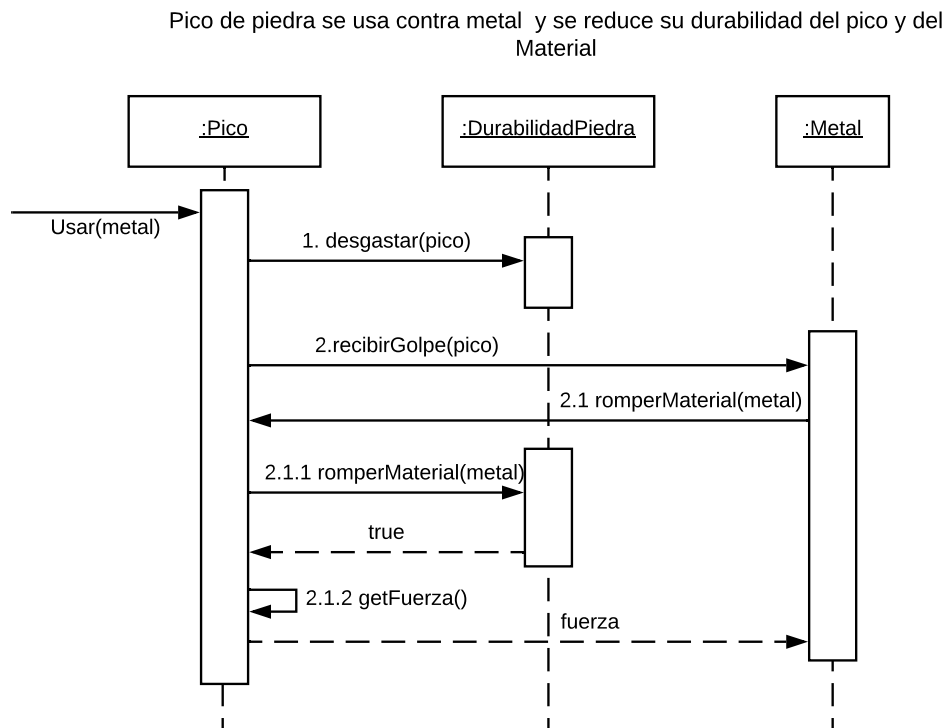


Figura 6. Diagrama de Secuencia: Pico de piedra se usa contra metal

Jugador se mueve hacia arriba estando esa posicion vacia en el mapa.

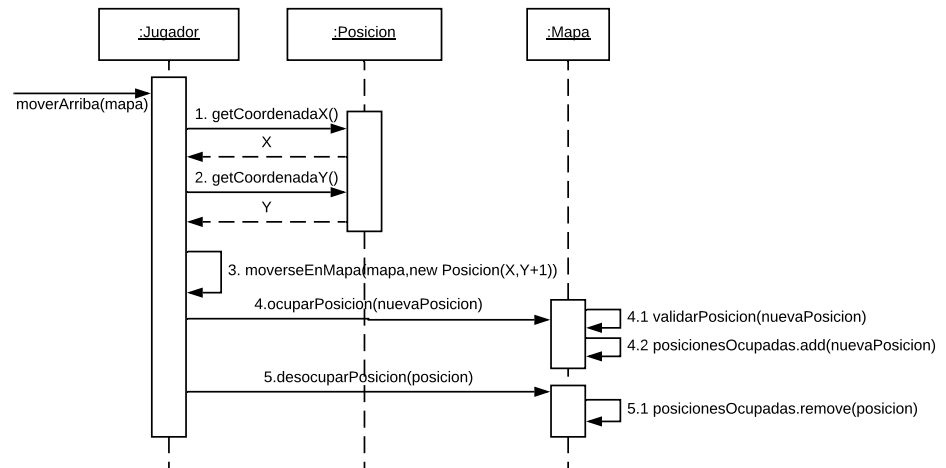


Figura 7. Diagrama de Secuencia: Jugador se mueve hacia arriba

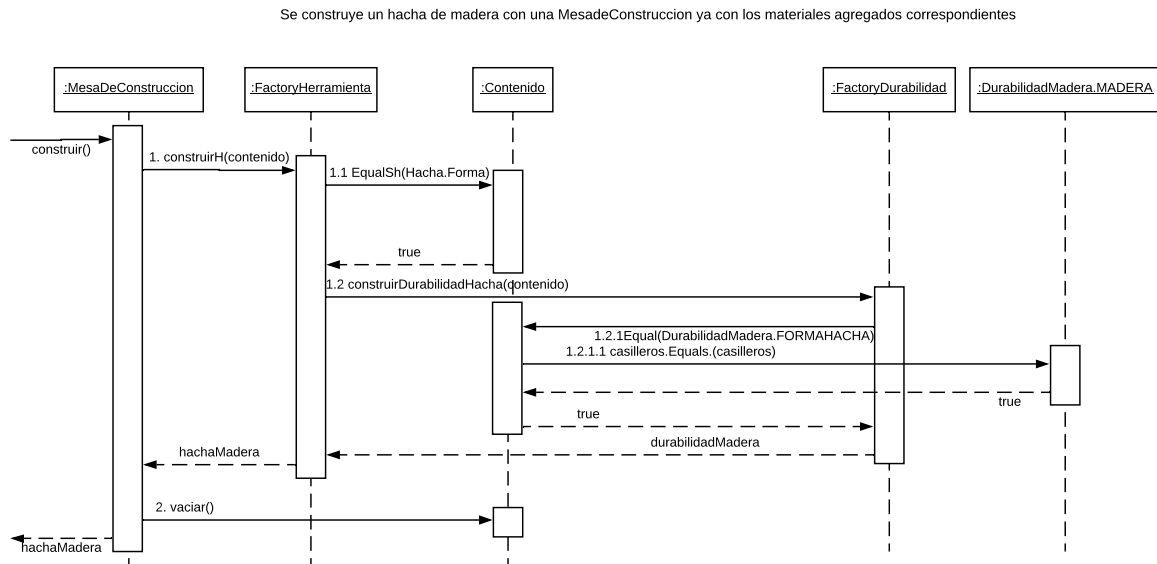


Figura 8. Diagrama de Secuencia: Construcción de una hacha de madera

9. Diagramas de estado

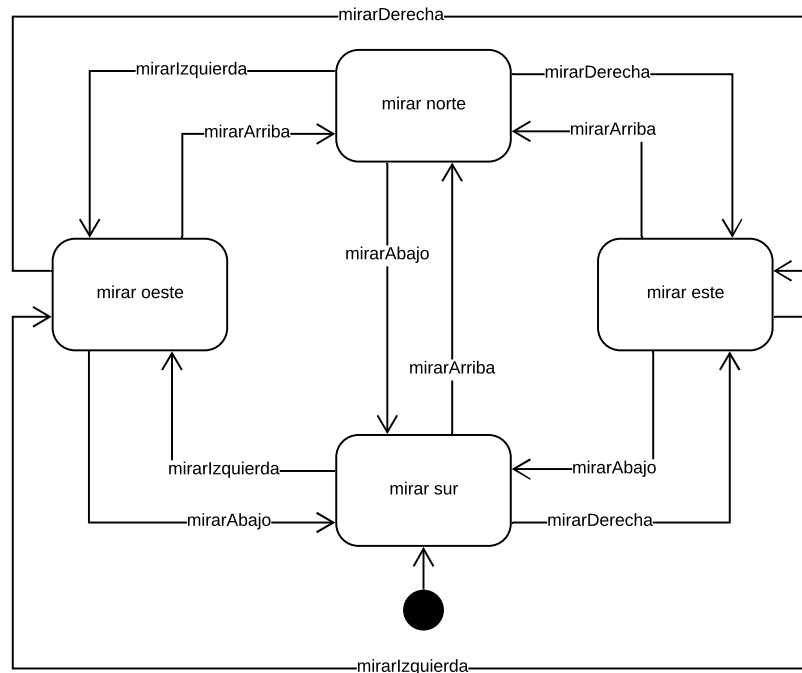


Figura 9. Diagrama de estado: Posibles estados de la orientación del jugador

10. Patrones de diseño

A continuación se detallan los distintos patrones de diseño utilizados en el programa y la razón por la cual se los escogió.

- **Visitor:** este patrón se utiliza para relacionar las herramientas con sus respectivas durabilidades (excepto el pico fino). Dado a que una herramienta tiene un tipo de durabilidad, y las propiedades de la herramienta dependen del tipo de durabilidad, entonces se decidió utilizar este patrón. Por ejemplo, la fuerza de una herramienta depende del tipo de herramienta y del tipo de durabilidad, por eso cuando se instancia una herramienta, dentro del constructor se le dice a la durabilidad que genere el valor de la fuerza pasándole por parámetro la misma herramienta (aquí se hace la visita), entonces de esta forma la durabilidad sabe que responder en función del tipo de herramienta.
- **Factory:** la mesa de construcción genera herramientas a partir de un contenido, pero debido a que no es conocido la forma del contenido se decidió implementar una fábrica de herramientas que a partir del contenido sepa que herramienta debe construir.

11. Conclusión