

Tercera entrega del Trabajo Practico 2

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

Alumno	Nro padron
Germán Dawidowski	101595
Brian Stanley	101753
Sebastián Tassone	99712
Yorick Vinesse	103118

Índice

1. Introducción	2
2. Supuestos para la segunda entrega	2
3. Modelo de dominio	2
4. Diagramas de clase	4
4.1. Estrategias de desgaste	4
4.2. Materiales	4
4.3. Herramientas	5
4.4. Juego	5
4.5. Navegación	6
4.6. Creación de herramientas	6
4.7. Mapa	8
5. Detalles de implementación	9
6. Excepciones	9
7. Diagramas de secuencia	9
7.1. uso de una herramienta	9
7.1.1. movimientoDelJugador	10
7.1.2. Creacion del Jugador	10
7.1.3. moverNorte	11
7.1.4. crearHerramienta	11
8. Casos de Ejecucion	12
8.0.1. Aplicación de interfaz gráfica inicial básica	12

1. Introducción

En esta serie de informes los integrantes del equipo detallaremos el progreso del segundo trabajo práctico de la materia Algoritmos y programación III.

En estas páginas se expondrán los detalles pertinentes del trabajo realizado a lo largo de la semana por todos nosotros, acompañados de los diagramas necesarios para su correcta comprensión.

2. Supuestos para la segunda entrega

- 1 Tomamos como supuesto que por cada vez que se use una herramienta en un material, a la durabilidad del material se le restará la fuerza de la herramienta en cuestión. Lo hicimos porque la consigna no especifica como decrece la durabilidad de los materiales.
- 2 Tomamos como supuesto que si una herramienta se usa sobre un material que no puede romper, se desgasta la herramienta pero no el material, a excepción del pico fino que solo se desgasta con el diamante.
- 3 Tomamos como supuesto que el usuario podrá crear herramientas de un mismo tipo en las cantidades que el desee.

3. Modelo de dominio

Dividimos el modelo del trabajo práctico en cuatro paquetes que contienen los elementos principales del modelo.

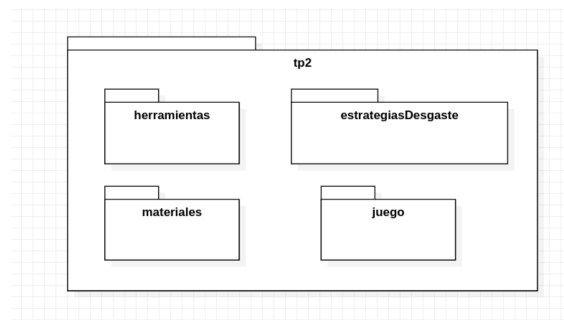


Figura 1: diagrama general del tp.

En el paquete herramientas están todas las clases que conforman las herramientas que el usuario puede usar en el juego con sus comportamientos básicos.

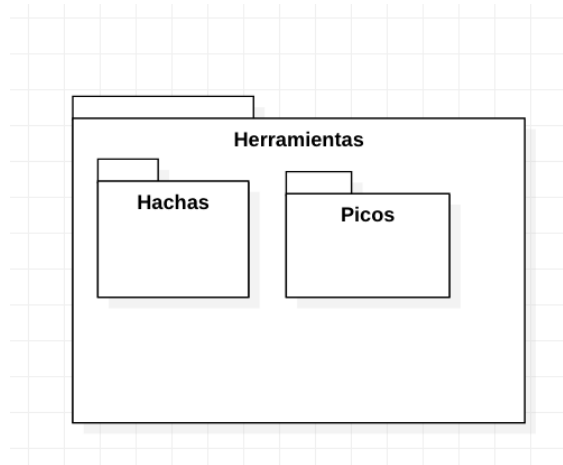


Figura 2: diagrama del paquete herramientas.

En el paquete `estrategiasDesgaste` están las clases que conforman las estrategias de desgaste que usan las herramientas para calcular el decremento de su durabilidad con el uso.

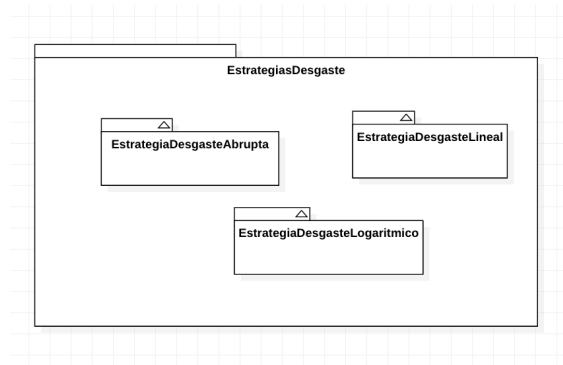


Figura 3: diagrama del paquete estrategiasDesgaste.

En el paquete `materiales` se encuentran las clases que definen el comportamiento de los distintos materiales que el usuario puede encontrar en el juego.

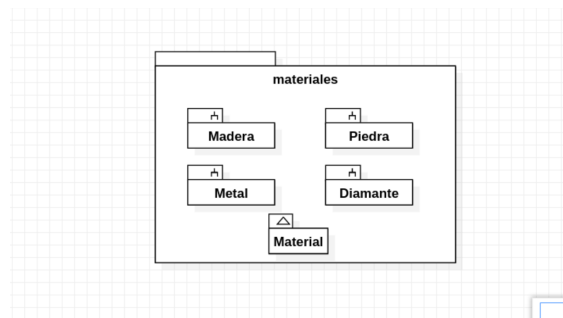


Figura 4: diagrama del paquete materiales.

En el paquete `juego` se encuentran los primeros rastros de lo que serían los elementos del modelo que harían posible que el juego se juegue, por lo pronto solo contiene una clase `Jugador` y una clase `Inventario`.

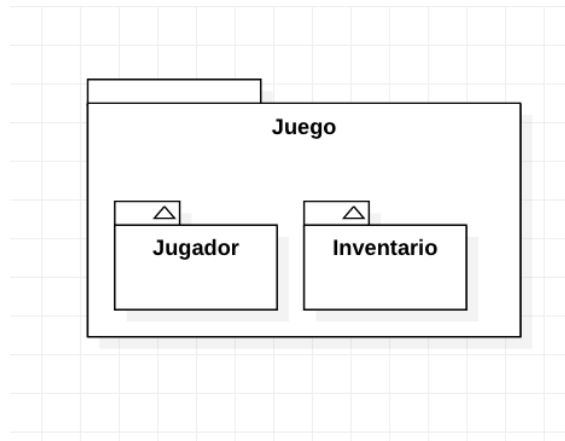


Figura 5: diagrama del paquete juego.

4. Diagramas de clase

A continuación veremos como están conformadas las clases más importantes del trabajo práctico.

4.1. Estrategias de desgaste

Estas clases se encargan de manejar el desgaste de las distintas herramientas que el jugador puede usar. Todo objeto de clase herramienta o de alguna clase hija tiene una estrategia de desgaste

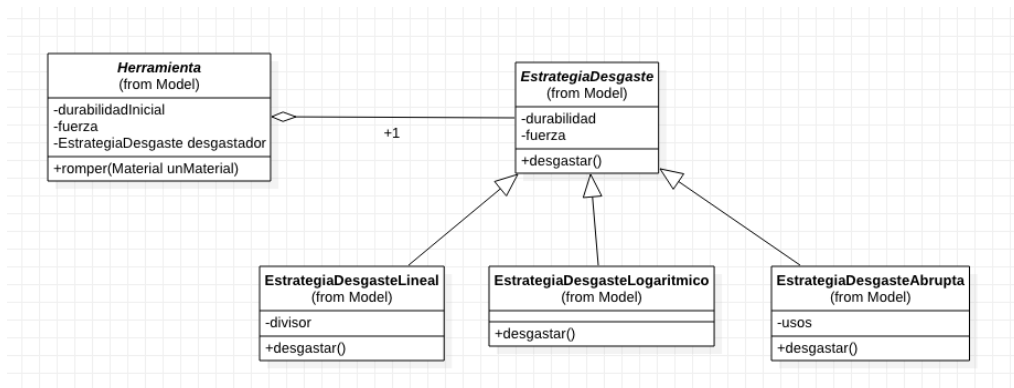


Figura 6: diagrama de clases de los tipos de desgaste.

4.2. Materiales

Estas clases modelan los distintos materiales que el jugador puede desgastar con sus herramientas durante el juego. Cada uno de los materiales sobrecarga el método `romper` de la clase `Material` para que funcione solamente con los tipos de herramientas que pueden romper a dicho material.

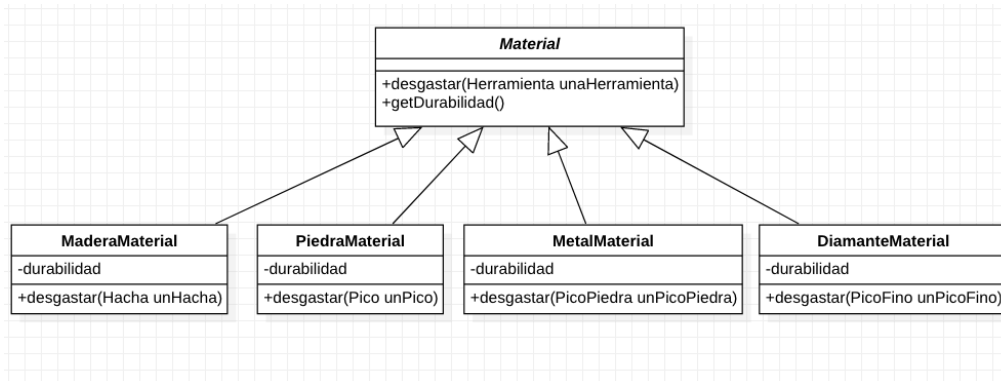


Figura 7: diagrama de clases de los tipos de material.

4.3. Herramientas

Estas clases son las que modelan las herramientas a disposicion del jugador. Estas estan divididas en dos subgrupos (Hacha y Pico). Dentro de esta subdivision tenemos un objeto por cada tipo de herramienta que el jugador puede crear. Hicimos esto para facilitar el proceso de creacion y simplificar los atributos de las clases de herramienta especificas, a demas tener clases especificas para cada herramienta particular facilita el proceso de desgaste de los materiales.

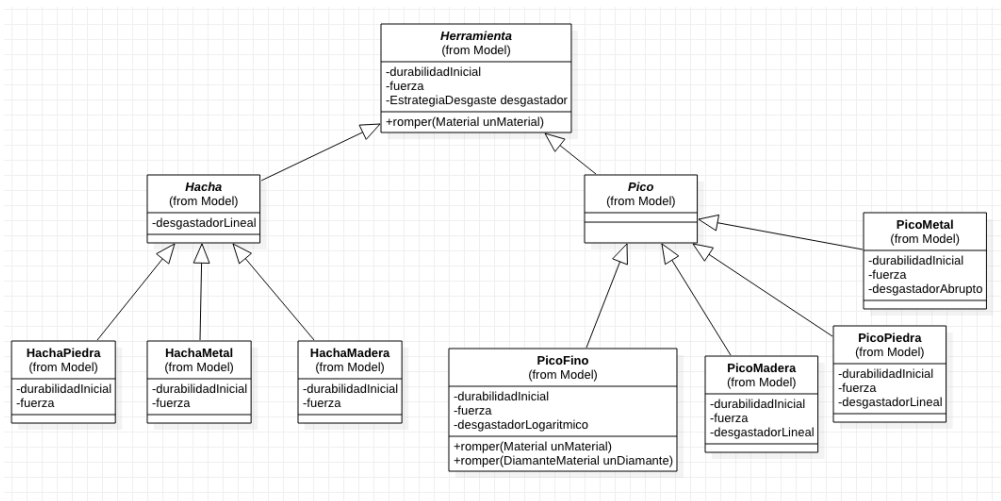


Figura 8: diagrama de clases de los tipos de herramientas.

4.4. Juego

La clase Juego es la que engloba la ejecución del modelo. Este inicializa un jugador con su navegador y un mapa de juego.

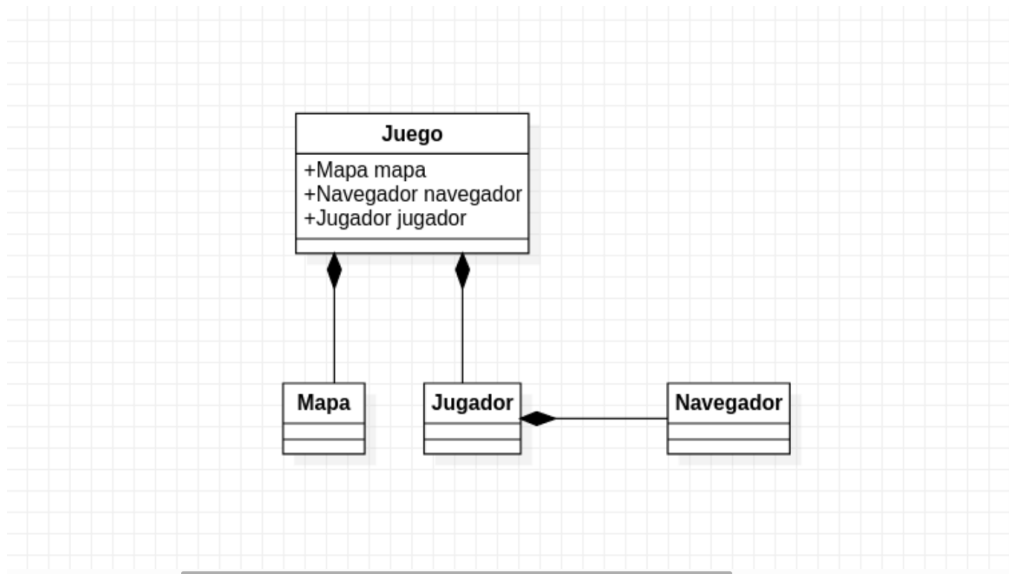


Figura 9: diagrama de clases de los elementos inicializados al comenzar el juego

4.5. Navegación

Estas son las clases que permiten al jugador moverse por el mapa.

La clase **Navegador** se comunica con el mapa y guarda la posición en la que está parado el jugador. También administra todos los movimientos del jugador por el mapa.

La clase **Dirección** extrapola del navegador la responsabilidad de decidir hacia adonde se debe mover; cada dirección tiene métodos para moverse en la misma.

La clase **Posición** tiene un par de coordenadas `x` y `y` que representan una ubicación en el mapa. La usa el navegador para registrar en qué posición del mapa se encuentra el jugador.

4.6. Creación de herramientas

Estas son las clases que intervienen en la creación de herramientas

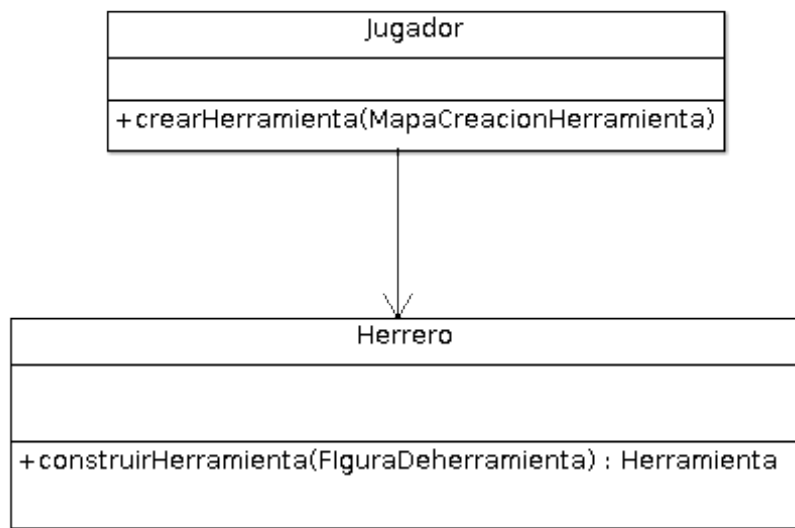


Figura 10: diagrama de clases relacion entre jugador y herero

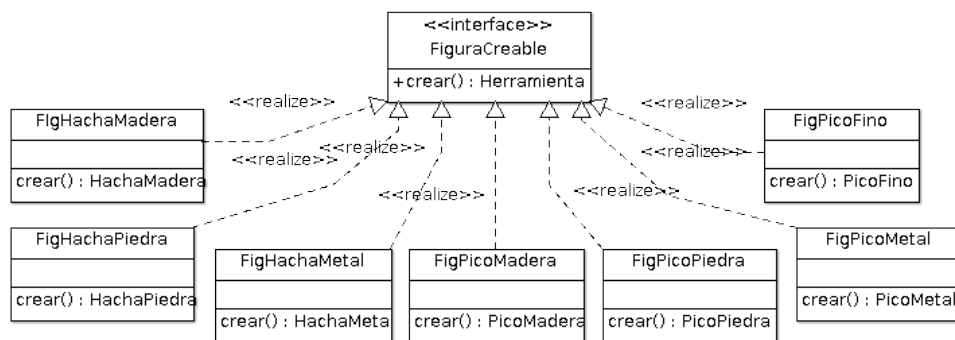


Figura 11: diagrama de clases de jerarquia de las figuras que puede crear el jugador

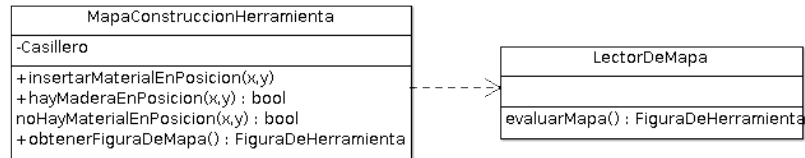


Figura 12: diagrama de clases de relacion entre los mapas de construccion de herramienta y el lector de mapas

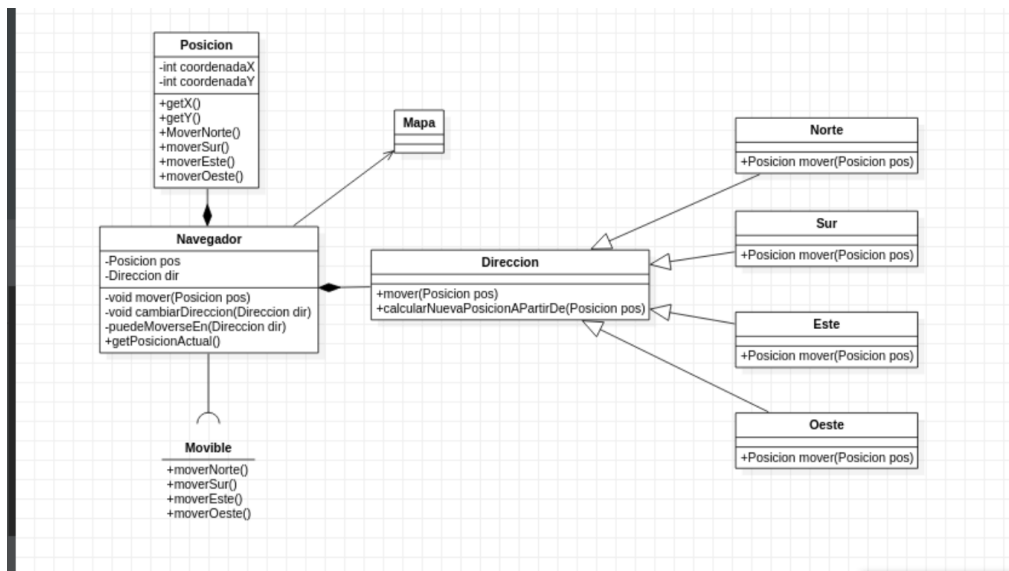


Figura 13: diagrama de clases de los elementos de navegación

4.7. Mapa

Estas son las clases que conforman al mapa.

La clase **Casillero** representa cada uno de los casilleros individuales que conforman al mapa. Estos pueden contener un material o no contener nada, si contienen un material, se consideran vacíos y disponibles para que el jugador se pare en ellos.

La clase **mapa** tiene una matriz que representa al mapa del juego, cuyos campos son objetos de tipo **Casillero**.

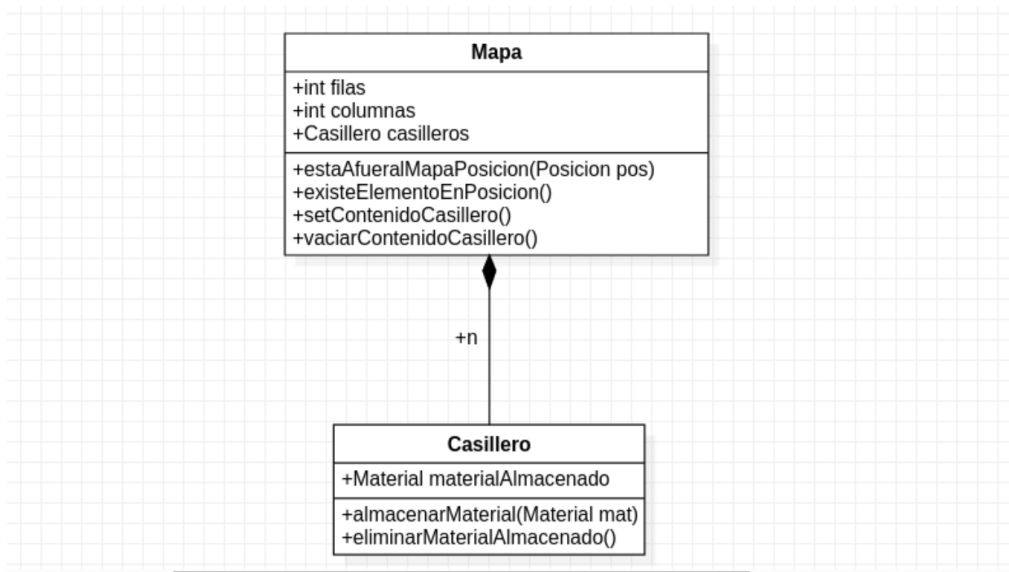


Figura 14: diagrama de clases de los elementos que conforman el mapa

5. Detalles de implementación

6. Excepciones

■ Excepciones de Mapa

1. *CasilleroOcupadoExcepcion* es lanzada por un *Casillero* cuando se intenta ocupar un objeto *Casillero* que fue ocupado con anterioridad.

■ Excepciones de Herrero

1. *HerreroNoConoceFiguraACrearExcepcion* es lanzada por el *Herrero* cuando se le pide que construya una herramienta de *FiguraDesconocida*.

■ Excepciones de Materiales

1. *MaterialRotoExcepcion* es lanzada por el *Material* en el momento que se checkea su durabilidad.

■ Excepciones de Estrategias De Desgaste

1. *DurabilidadCeroExcepcion* es lanzada por la *EstrategiaDeDesgaste* cuando se le solicita que desgaste y la durabilidad es igual o menor a cero.

7. Diagramas de secuencia

7.1. uso de una herramienta

El proceso mas complejo del trabajo practico a la fecha es el de usar una heramienta, por eso es que se incluye el diagrama de secuencia de dicha accion en el informe.

El actor del diagrama (en el programa seria el jugador) da la orden a la herramienta de romper, se pasa el material que el usuario quiere romper por parametro al metodo. Este metodo primero da la orden al material en si de desgastarse, de las clases de desgastar de

los materiales se decide que tipo de herramienta puede romper a cada material. Seguidamente se le da la orden al desgastador, clase que maneja la durabilidad y la forma de desgastarse de cada herramienta, que ejecute su metodo desgastar. Entonces termina el metodo disminuyendo la durabilidad de la herramienta del usuario y la del material, en caso de que la herramienta usada sea compatible.

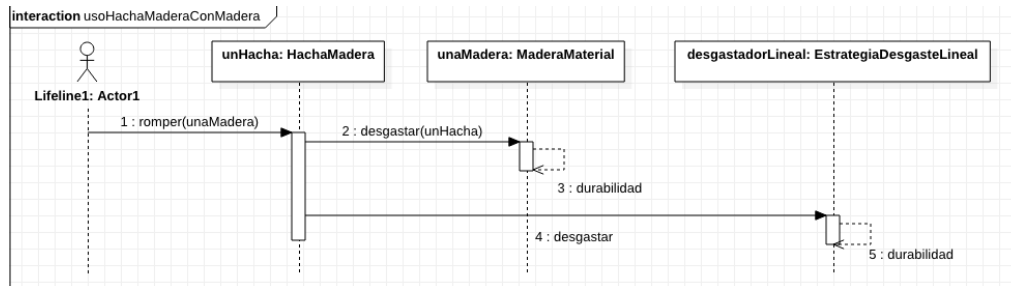


Figura 15: diagrama de secuencia de el uso de una herramienta.

7.1.1. movimientoDelJugador

El proceso de movimiento del jugador involucra muchas clases que tienen que ver con la navegación por el mapa.

Primero el jugador manda el mensaje al navegador de moverse en alguna de las cuatro direcciones cardinales (norte, sur, este, oeste). El navegador, antes de dar la orden de moverse a su posición actual, chequea que la posición a la que se le está ordenando moverse sea válida. Si no lo es el navegador no ejecuta la operación.

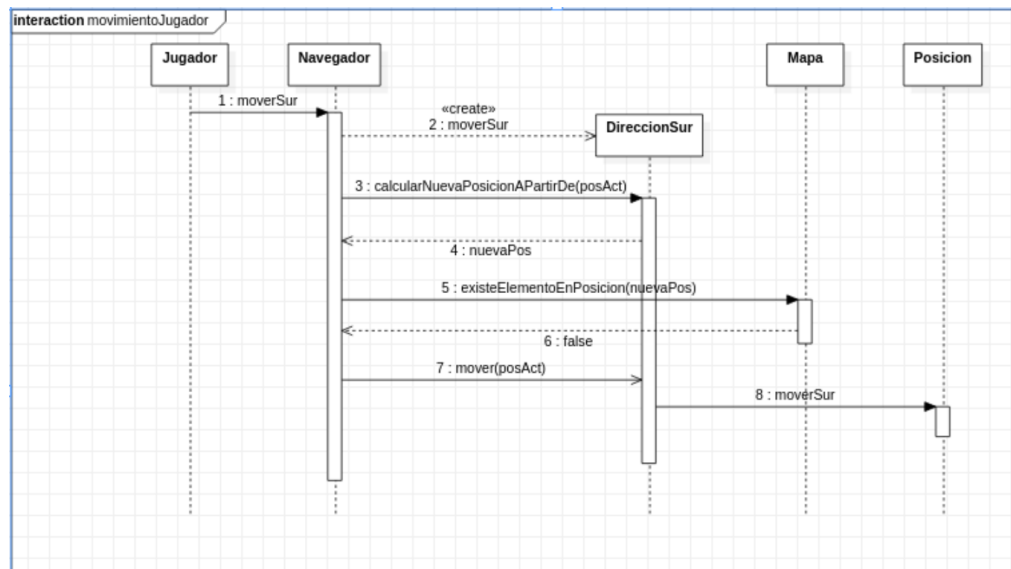


Figura 16: diagrama de secuencia de el movimiento de un jugador

7.1.2. Creacion del Jugador

Aquí vemos la creación correcta del jugador. Antes de crear al jugador en cuestión se crea un Mapa, que será la plataforma por la cual el jugador ejecutará sus acciones. Además se crea

un navegador, quien navega por el mapa mencionado anteriormente y se lo asocia el Mapa. Finalmente se crea el jugador y este utiliza el Navegador creado para circular por el Mapa

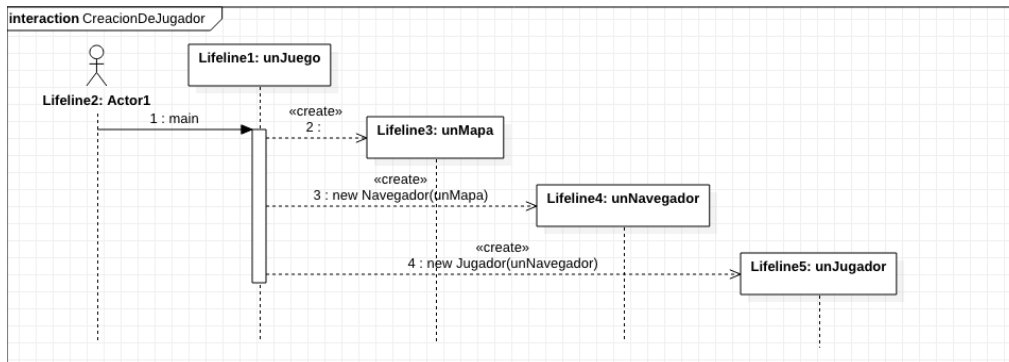


Figura 17: diagrama de secuencia de el movimiento de un jugador

7.1.3. moverNorte

El Jugador utiliza a su Navegador para circular por el Mapa. El jugador le dice al Navegador hacia donde quiere dirigirse. El Navegador analiza su posición actual y la posición solicitada. Para esto se comunica con el Mapa.

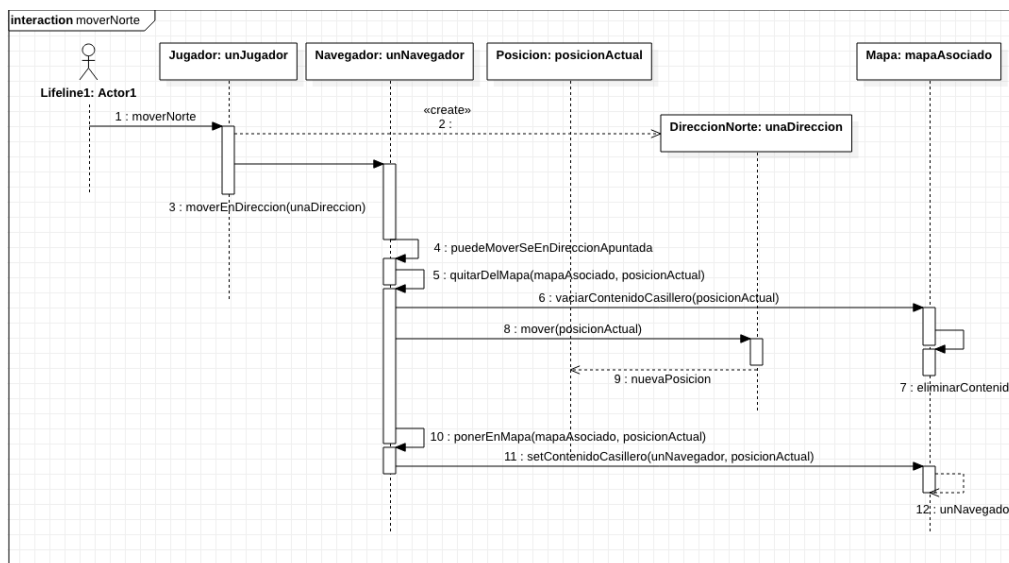


Figura 18: diagrama de secuencia de el movimiento de un jugador

7.1.4. crearHerramienta

El jugador recibe un Plano de Construcción, aquí llamado Mapa de Construcción, con materiales cargados por el usuario. El Jugador se comunica con su Herrero y le envía el Mapa de Construcción recibido anteriormente. El Herrero le solicita al Mapa de Construcción que le indique cual es la figura plasmada en el. Para esto el Mapa hace uso de su LectorDeMapa. Una vez analizado el Mapa por el Lector, el Herrero recibe una figura concreta y según cual sea la figura recibida crea una Herramienta específica y se la envía al Jugador, quien la almacena en su Inventario

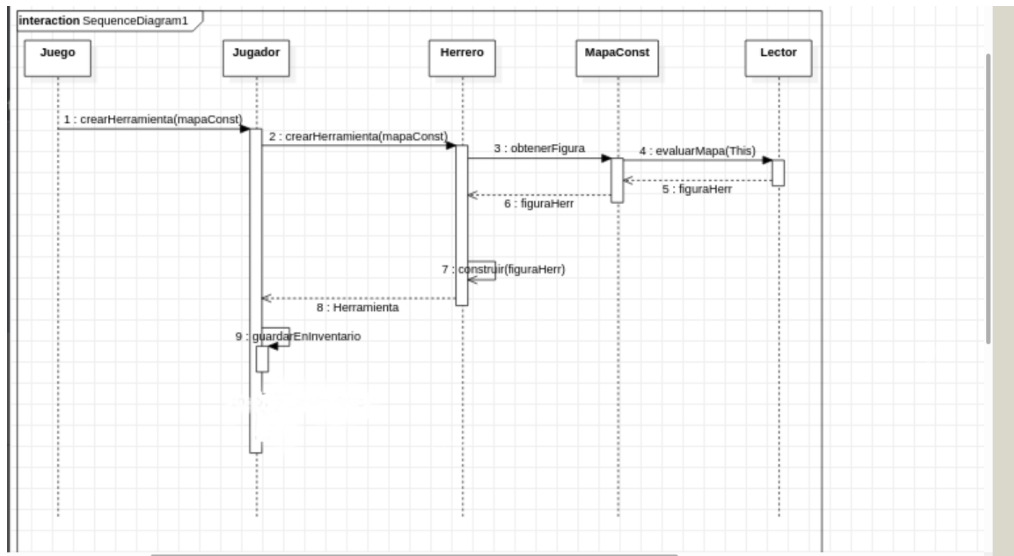
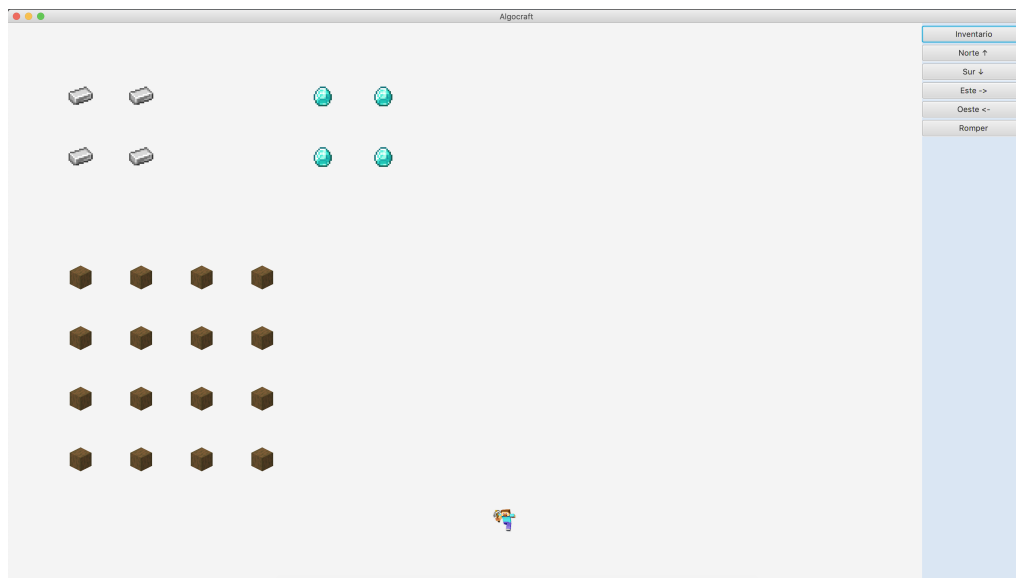


Figura 19: diagrama de secuencia de creacion de herramienta

8. Casos de Ejecucion

8.0.1. Aplicación de interfaz gráfica inicial básica

Situación: Comienzo del juego, disposición de materiales en el tablero y menú inicial.



Situación: Click en botón del panel.

