

Trabajo Práctico 2 — AlgoCraft

[7507/9502] Algoritmos y Programación III

Curso 1

Primer cuatrimestre de 2019

Alumnos:	PALAZON, ALDREY, PAGURA, FRITZ
Números de padrón:	102679, 102731, 102649, 102320
Email:	aaajoaco@gmail.com, sebastianpagura@gmail.com, mpalazon3

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	2
4. Detalles de implementación	2
4.1. Desgaste de las Herramientas	2
4.2. usarContra	3
5. Excepciones	3
6. Diagramas de secuencia	3

1. Introducción

El presente informe reúne la documentación de la solución de AlgoCraft, trabajo práctico de la materia Algoritmos y Programación III, que consiste en desarrollar una aplicación de un juego parecido al Minecraft en Java utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

Ya que la consigna no especificaba el comportamiento de las herramientas en algunos casos particulares como usar el hacha contra piedra o usar el pico contra madera, decidimos que en estos casos, en los que la herramienta no es capaz de romper al material, la herramienta sea desgastada, pero el material no. Otro supuesto que tuvimos fue que el personaje tenga una herramienta de su inventario equipada, ya que en el caso de tener más de una herramienta, se debe especificar de alguna manera cuál es la herramienta que el personaje va a usar.

3. Diagramas de clase

El diagrama de clases muestra todas las relaciones de nuestro modelo implementadas hasta el momento.

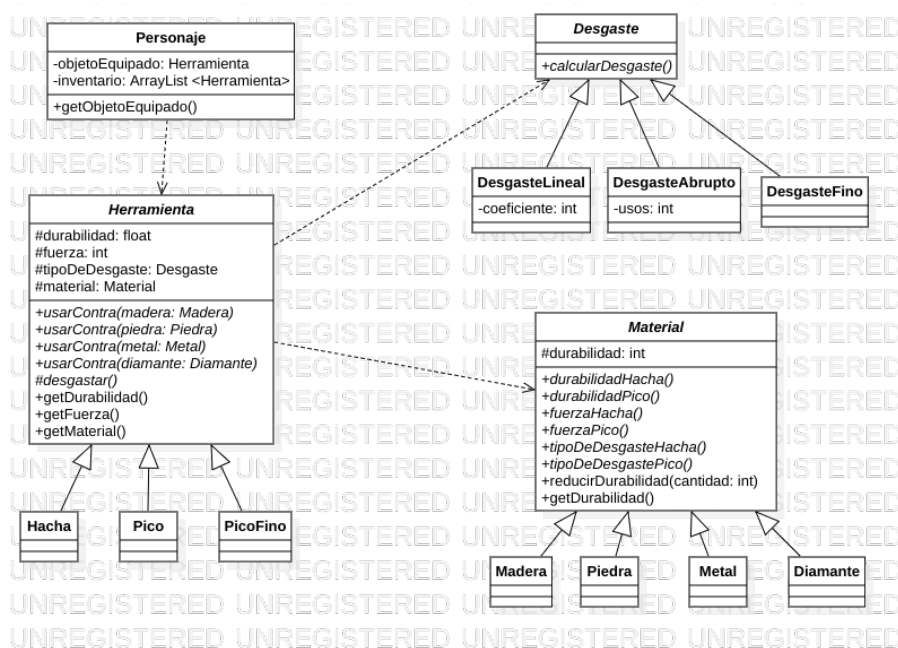


Figura 1: Diagrama del modelo.

4. Detalles de implementación

4.1. Desgaste de las Herramientas

Debido a que el desgaste de las herramientas dependía no sólo del tipo de herramienta, sino que también del material de dicha herramienta, decidimos que la herramienta tenga un atributo `tipoDesgaste`, una instancia de la clase **Desgaste**, de la cual heredan distintas clases representando las distintas maneras en las que las herramientas se pueden desgastar. El tipo de desgaste lo recibe

la herramienta de parte del material, al ser inicializado. De esta manera la herramienta delega la responsabilidad de calcular su desgaste a su atributo `tipoDesgaste`. En este caso, estamos usando el patrón Strategy, ya que estamos delegando el comportamiento de la clase Herramienta a la clase Desgaste.

4.2. `usarContra`

Otro punto importante de nuestra implementación fue como controlar el uso de las herramientas evitando condicionales. Para ello decidimos sobrecargar el método `usarContra` de cada herramienta, lo que permite que cada una tenga una manera distinta de usarse contra cada material. Esto nos permitió lograr implementar el uso de la herramienta con polimorfismo.

5. Excepciones

NoSePuedeUsarMaterialContraMaterialError Esta excepción saltaría cuando un jugador intente romper un material con un material equipado.

HerramientaInexistenteError Esta excepción saltaría en el caso de que el usuario intente crear una herramienta pero dibuje alguna forma inválida.

HerramientaRotaExcepcion Al llegar la durabilidad de las herramientas a cero, estas no podrán seguir siendo usadas.

MaterialNoEncontradoExcepcion Aparecería dado el caso de usarse una herramienta sin un bloque en frente.

6. Diagramas de secuencia

Estos diagramas muestran cómo se crea un hacha y cómo se usa contra el material madera.

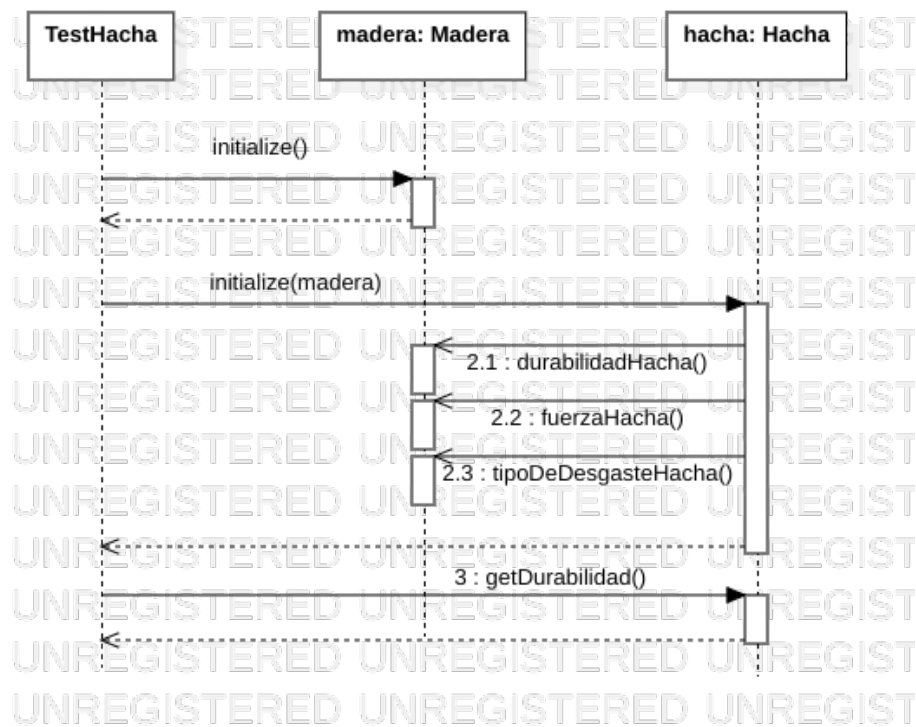


Figura 2: TestHachaDeMaderaSeCreaConDurabilidadCorrespondiente.

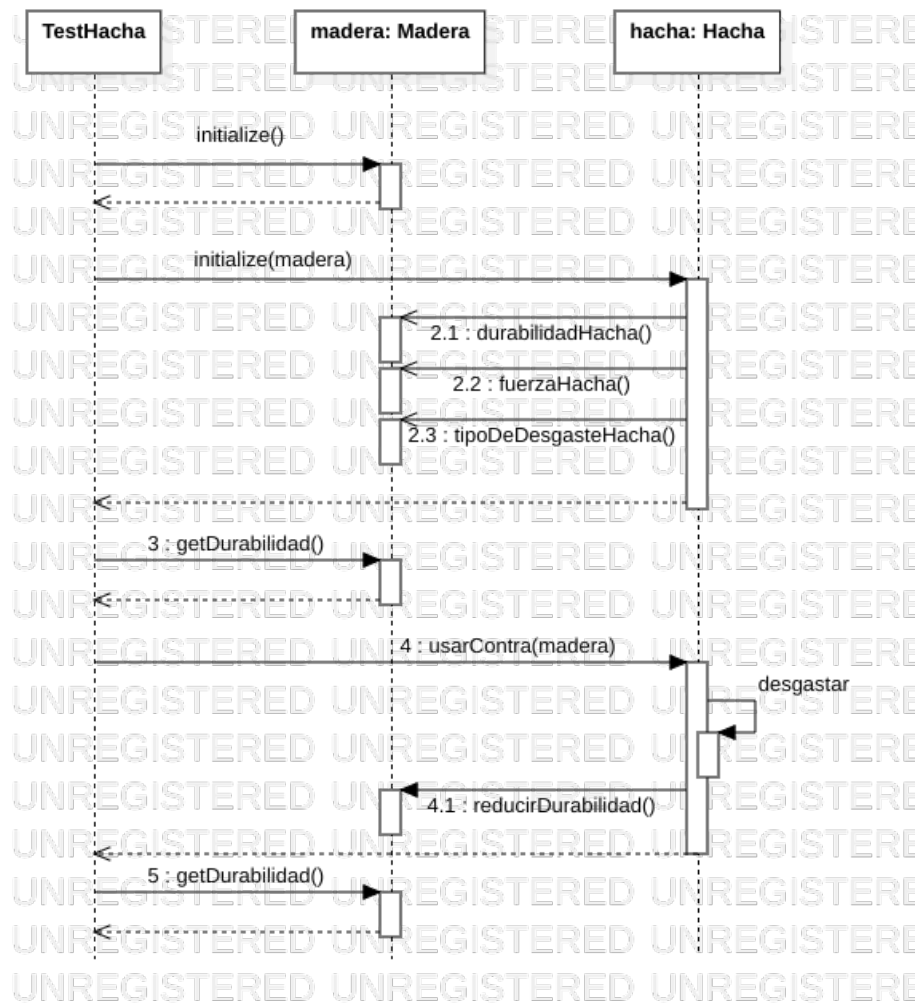


Figura 3: TestMaderaGolpeadaPorHachaDeMaderaReduceDurabilidad.