



DESIGN PATTERNS

David Villaverde | Hugo Román | Alejandro García



DESCRIPCIÓN

Este videojuego consiste en que el jugador libre una serie de combates contra distintos enemigos en una variedad de mapas.

Al iniciar el videojuego, se tendrá la posibilidad de elegir entre tres personajes diferentes, cuyas estadísticas difieren y por lo tanto requerirán de una estrategia diferente para afrontar las peleas.

Los mapas se ordenarán al azar durante cada partida, y para superar el videojuego, deberás vencer a los enemigos de los cuatro mapas.

En los combates el jugador tendrá la posibilidad de elegir entre si desea atacar o desea curar su vida en cada turno. Para finalizar un combate tendrá que reducir la vida del enemigo a cero, en el caso de que la propia se reduzca a cero el jugador habrá perdido la partida.

MANUAL DE JUEGO

ELECCIÓN DE PERSONAJE

Al comenzar la partida se podrá elegir entre tres tipos de personajes: un caballero, una pistolera y un mago. La tabla muestra las diferencias y similitudes entre las características de estos:

	Caballero	Pistolera	Mago
Vida	5	3	3
Ataque	4	3	1
Magia	1	3	5
Defensa	4	1	3
Velocidad	1	5	3
Efecto	Sangrado – 20%	Quemado – 40%	Electrocutado – 50%

ASIGNACIÓN DE PUNTOS

Una vez hayamos escogido a uno de los diferentes héroes, tendremos la posibilidad de asignarle cinco puntos a nuestra elección, para mejorar sus estadísticas base. La siguiente tabla muestra estas características y como influirán en el combate:

Vida	Multiplicador: - Caballero: 75 - Pistolera: 50 - Mago: 30
Ataque	$\text{Daño} = \text{Math.log}(\text{Ataque}) / \text{Math.log}(15)$
Magia	$\text{Curación} = \text{Magia} * 7$
Defensa	$\text{Porcentaje de daño reducido} = \text{Math.log}(\text{Defensa}) / \text{Math.log}(15)$
Velocidad	$\text{Probabilidad de evitar un ataque} = \text{Velocidad} / 24$

COMBATES

Una vez seleccionado el personaje y asignados los puntos, comenzará el primer combate.

Los combates se darán en cuatro mapas diferentes, cuyo orden es aleatorio, contra cuatro enemigos en cada uno: el esqueleto (Especializado en ataque), el chief (Especializado en magia), el lobo (Especializado en Velocidad) y el robot (Especializado en Defensa). Además, las características de estos enemigos dependerán del mapa en el que se encuentren.

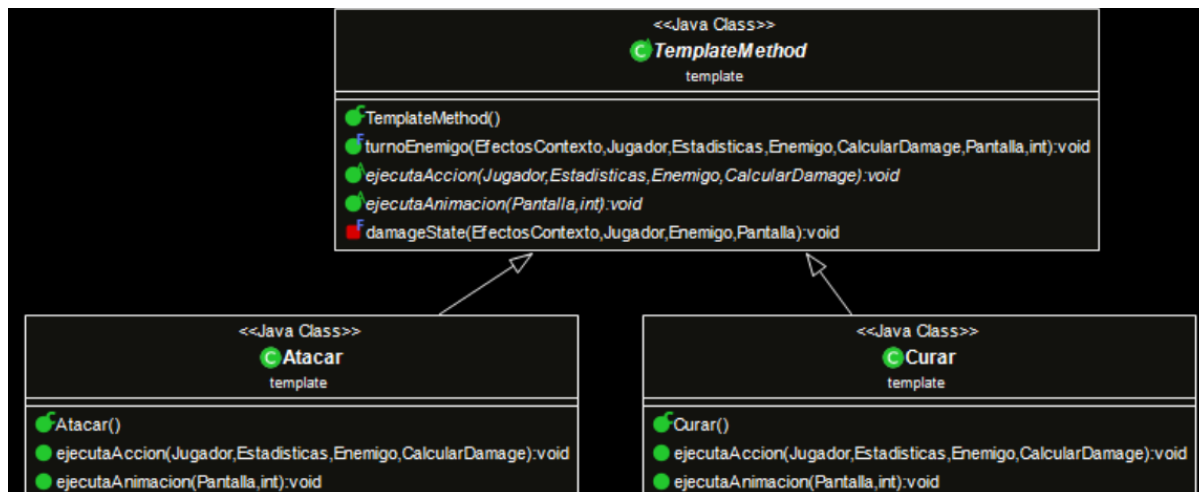
Tanto el jugador como el enemigo podrán realizar dos únicas acciones, atacar o curar, el enemigo basará su elección en la cantidad de vida que le quede. A su vez, ambos contrincantes tendrán la posibilidad de esquivar los ataques del otro, movimiento que dependerá de su velocidad.

Al ir avanzando de mundo, las estadísticas del personaje aumentarán en un punto y regenerará toda su vida, del mismo modo, los enemigos de los mapas superiores escalarán sus características según avances en la partida.

DIAGRAMA DE FUNCIONES

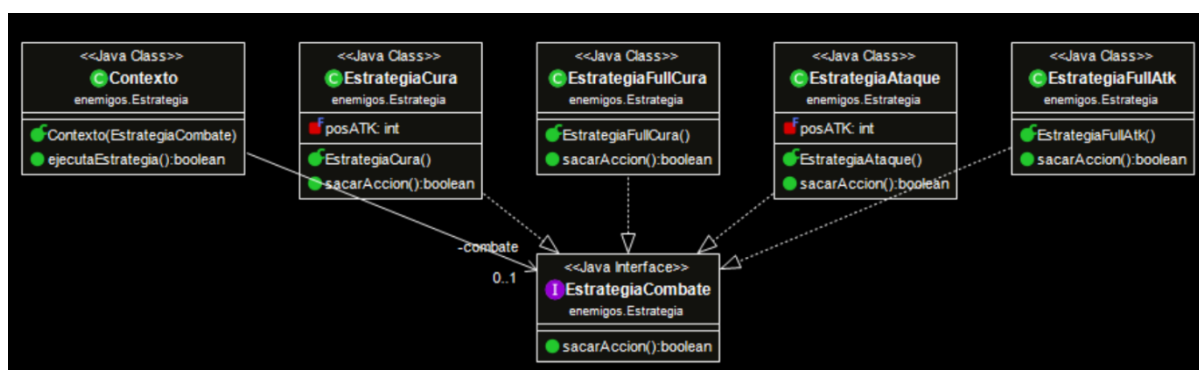
TEMPLATE METHOD

Utilizamos este patrón para implementar el algoritmo de la acción que realizará el enemigo. Esta acción se decide desde el patrón Strategy.



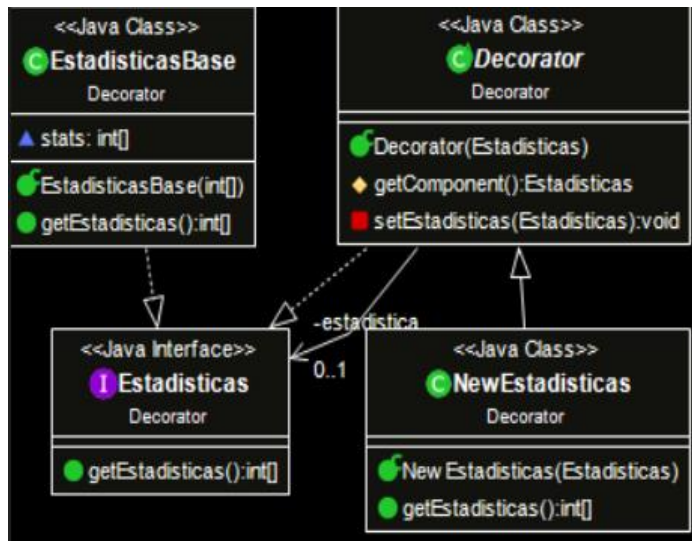
STRATEGY

El patrón Strategy está devolviendo un booleano, "true" cuando ataca y "false" al curarse, esto se decidirá en función de la estrategia, FullAtaque siempre devolverá "true", Ataque calculará aleatoriamente, con una posibilidad del 70% la devolución de "true". Por otro lado, FullCura devolverá siempre "false", y Cura tendrá una posibilidad del 70% de devolver "false". La estrategia se decidirá dependiendo de la vida que le quede al enemigo, 100% => FullAtaque, 75%-50% => Ataque, 50%-25% => Cura y 25%-0% => FullCura.



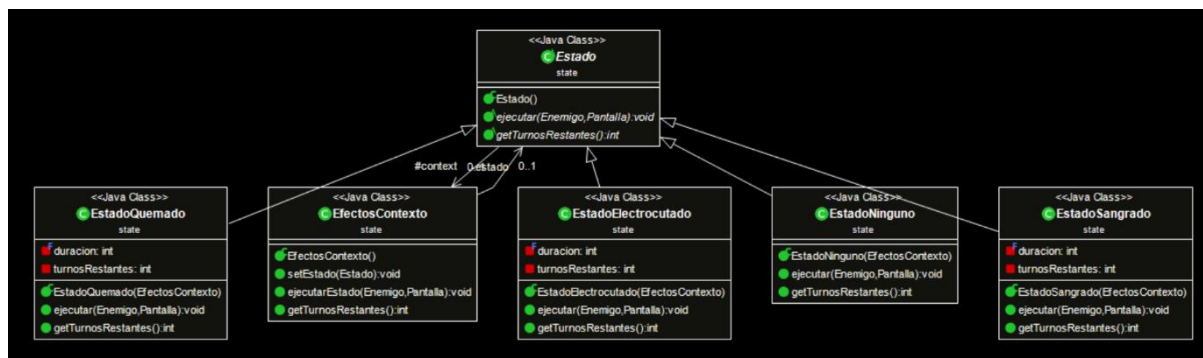
DECORATOR

Será el encargado de “decorar” las estadísticas del jugador, aumentando en uno el valor de cada estadística al pasar de mundo, como si fueran las capas de una cebolla. Se ha diseñado la clase de EstadísticasBase, que recoge el valor de las estadísticas al principio y NewEstadísticas, que se encarga de sumar ese punto a las estadísticas.



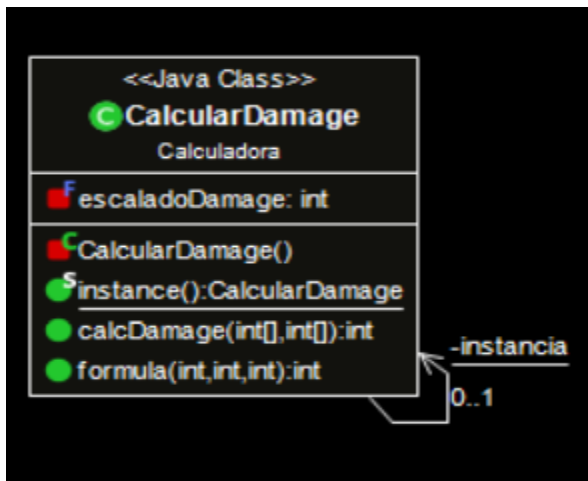
STATE

A este patrón le hemos atribuido la función de otorgar una habilidad extra a cada personaje, de forma que cada uno de ellos puede influir al enemigo un estado al atacar, que dependerá de una probabilidad y se calculará de manera aleatoria. De esta forma, el caballero podrá “hacer sangrar” a los enemigos, la pistolera les “quemará” y el mago les “electrocutará”. La clase EstadoNinguno, se utilizará como por defecto y será a la que volverán los enemigos al finalizar los turnos de cada efecto.



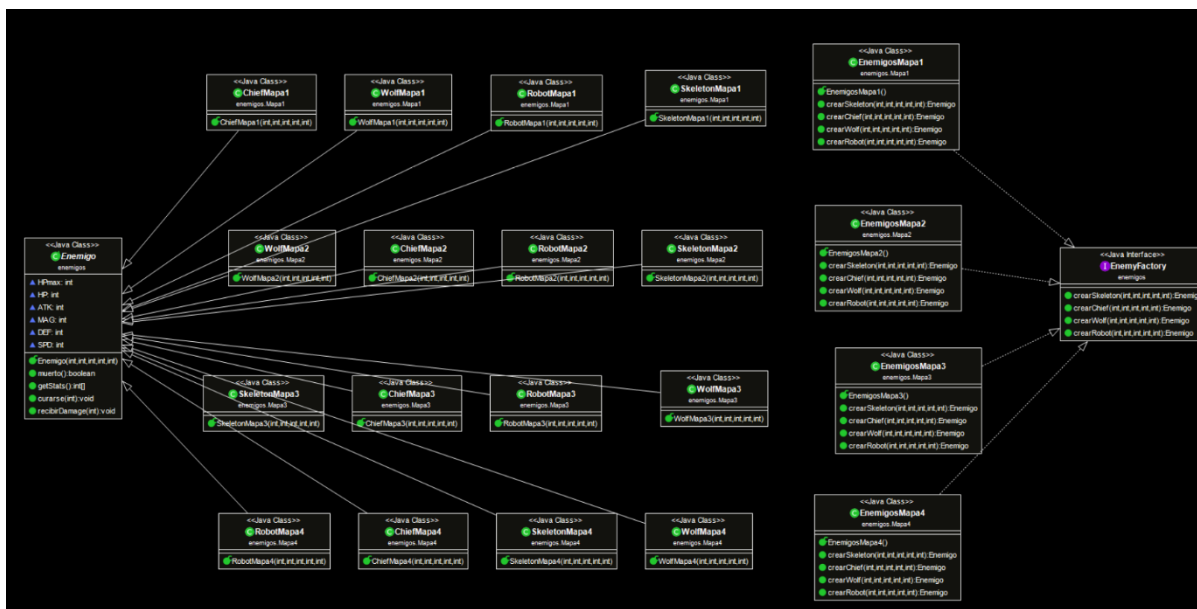
SINGLETON

Se compone de una única clase que contiene los cálculos con los cuales se gestiona el daño realizado por los ataques, dependiendo del ataque y defensa que posean el atacante y defensor, de esta manera únicamente se podrá crear una instancia de dicha clase.



ABSTRACT FACTORY

Este patrón es el encargado de organizar todo lo respectivo a los enemigos y los mundos, consiste en una clase **Enemigo**, en la cual se recogen las estadísticas y métodos que pueden tener los enemigos, a continuación, tenemos 4 clases por mundo, con la cual controlaremos después el escalado de las estadísticas dependiendo del número de mundo al que pertenezcan, en la cual se crea dicho enemigo conectadas las clases de **EnemigoMundo**, que gestiona los diferentes mundos.



UML COMPLETO

