**DESCRIPICION Y MANUAL DE USO DEL JUEGO**

Nada más comenzar el juego nos encontraremos con una pantalla de elección en la cual podrás optar por tres tipos de héroes, un caballero, una pistolera y un mago;

|  |  |  |  |
| --- | --- | --- | --- |
|  | Caballero | GunWoman | Wizard |
| 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% |

Una vez hayamos escogido nuestro héroe se nos otorgaran 5 puntos de habilidad para nosotros poder subir las estadísticas a nuestra elección.

|  |  |
| --- | --- |
| Vida | Multiplicador: Caballero: 75, GunWoman: 50, Wizard: 30 |
| Ataque | Daño = Math.log(Ataque)/Math.log(15) |
| Magia | Curación = Magia \* 7 |
| Defensa | Porcentaje de daño reducido = Math.log(Defensa)/Math.log(15) |
| Velocidad | Probabilidad de evitar un ataque |

A partir de aquí empezaran los combates.

Los combates consistirán en cuatro mundos, con mapas aleatorios, constituidos por cuatro enemigos cada uno, el esqueleto (Especializado en ataque), el chief (Especializado en magia), el lobo (Especializado en Velocidad) y el robot (Especializado en Defensa).

Se dispondrán de dos acciones, atacar y curar.

A su vez, después de cada turno, el enemigo decidirá si ataca o se cura dependiendo de la vida que le quede.

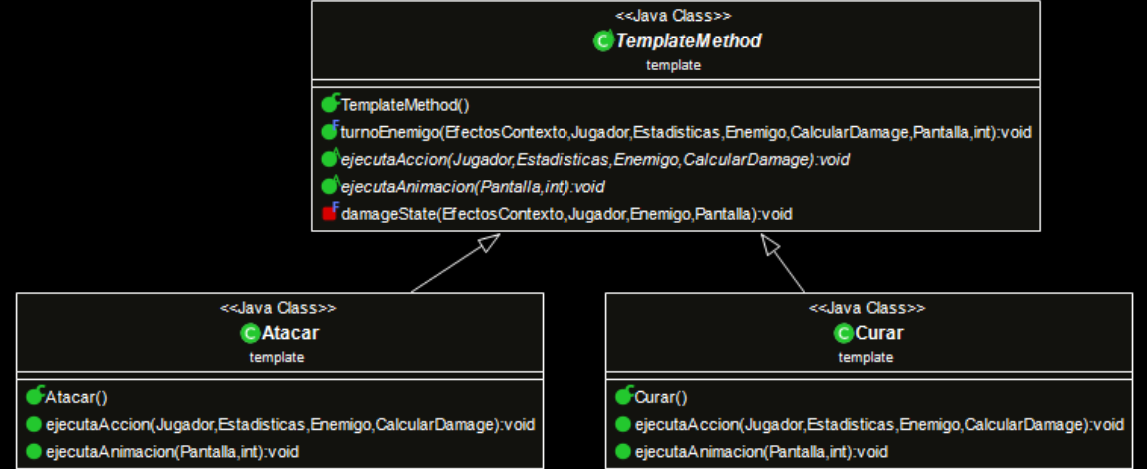
Cada vez que avances de mundo, todas tus estadísticas aumentaran en 1 punto y se te curará toda la vida, de la misma manera que escalaran las estadísticas de los enemigos según avances en la partida.

(ALEX BRO SI QUIERES EST PARTE RETOCALA UN POCO PARA QUE QUEDE MAS FLAMA Y TAL)

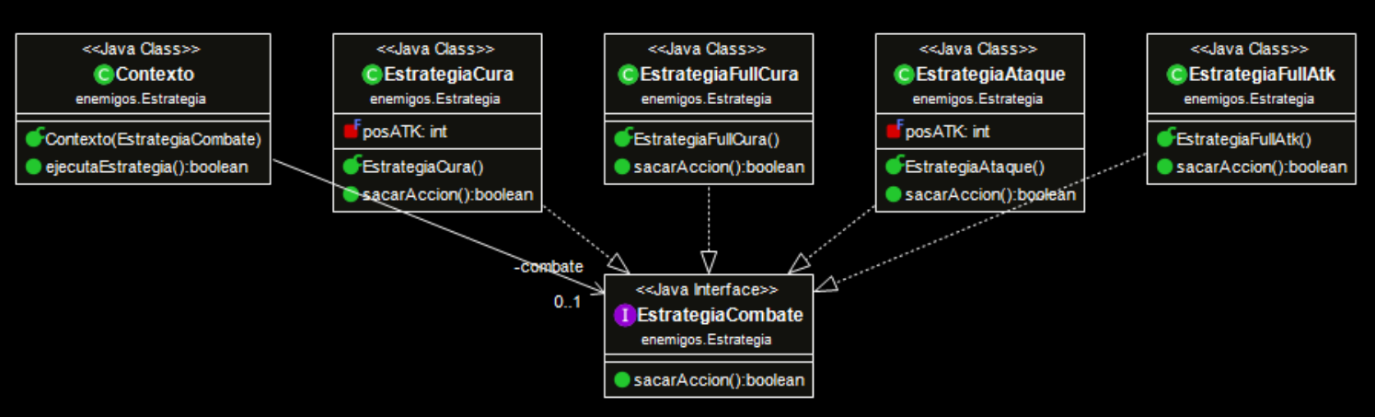
**DIAGRAMA DE FUNCIONES**

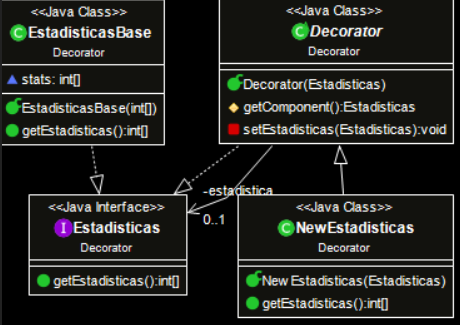
A continuación, se explicará brevemente el uso que se le ha dado a cada patrón y la manera de la que se ha implementado gracias al diagrama de clases.

**PATRON TEMPLATE METHOD**

Hemos usado el patrón template para implementar el algoritmo de la acción que efectuara el enemigo. Esta acción se decide desde el patrón Strategy.

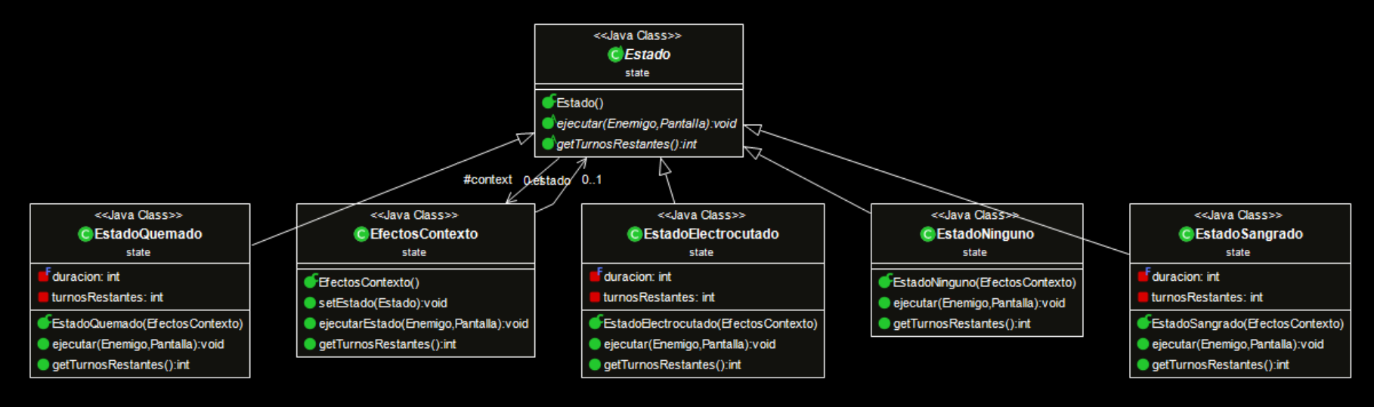
**PATRON STRATEGY**

El patron Strategy esta implementado de tal forma, que devuelve un booleano, true cuando ataca y false cuando se cura, esto se decide dependiendo de la estrategia a seguir, en FullAtaque siempre devolverá true, en Ataque, habrá una posibilidad del 70% de devolver true. Por otro lado, FullCura devuelve siempre false, y Cura tiene una posibilidad del 70% de devolver false. La estrategia se decidirá dependiendo de la vida que le quede al enemigo, 100% - FullAtaque, 100%-50% - Ataque, 50%-25% - Cura, 25%-0% FullCura.

**Patron Decorator**

El patron decorator será el encargado de, como su nombre indica, decorar las estadísticas del jugador, aumentando en 1 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 NewEstadisticas, que se encarga de sumar ese punto a las estadísticas.

**PATRON STATE**

Hemos decidido darle uso a este patrón en forma de un estado que se le puede influir al enemigo en cada ataque dependiendo de una probabilidad, hay 3 tipos de estado, uno por cada personaje, Caballero tiene sangrado, pistolera quemado y el mago electrocutar. También existe el EstadoNinguno, que es el por defecto y al cual vuelven los enemigos pasados un numero de turnos.

Diagrama, Texto

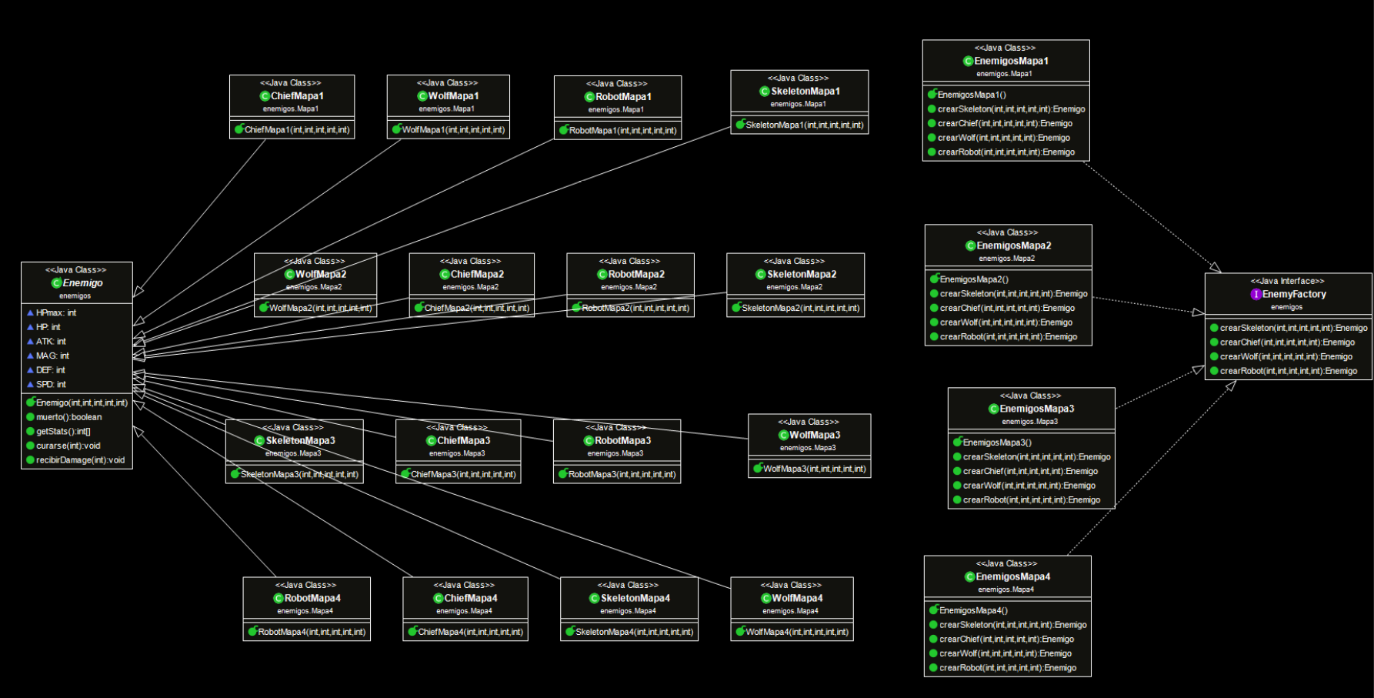
Descripción generada automáticamente

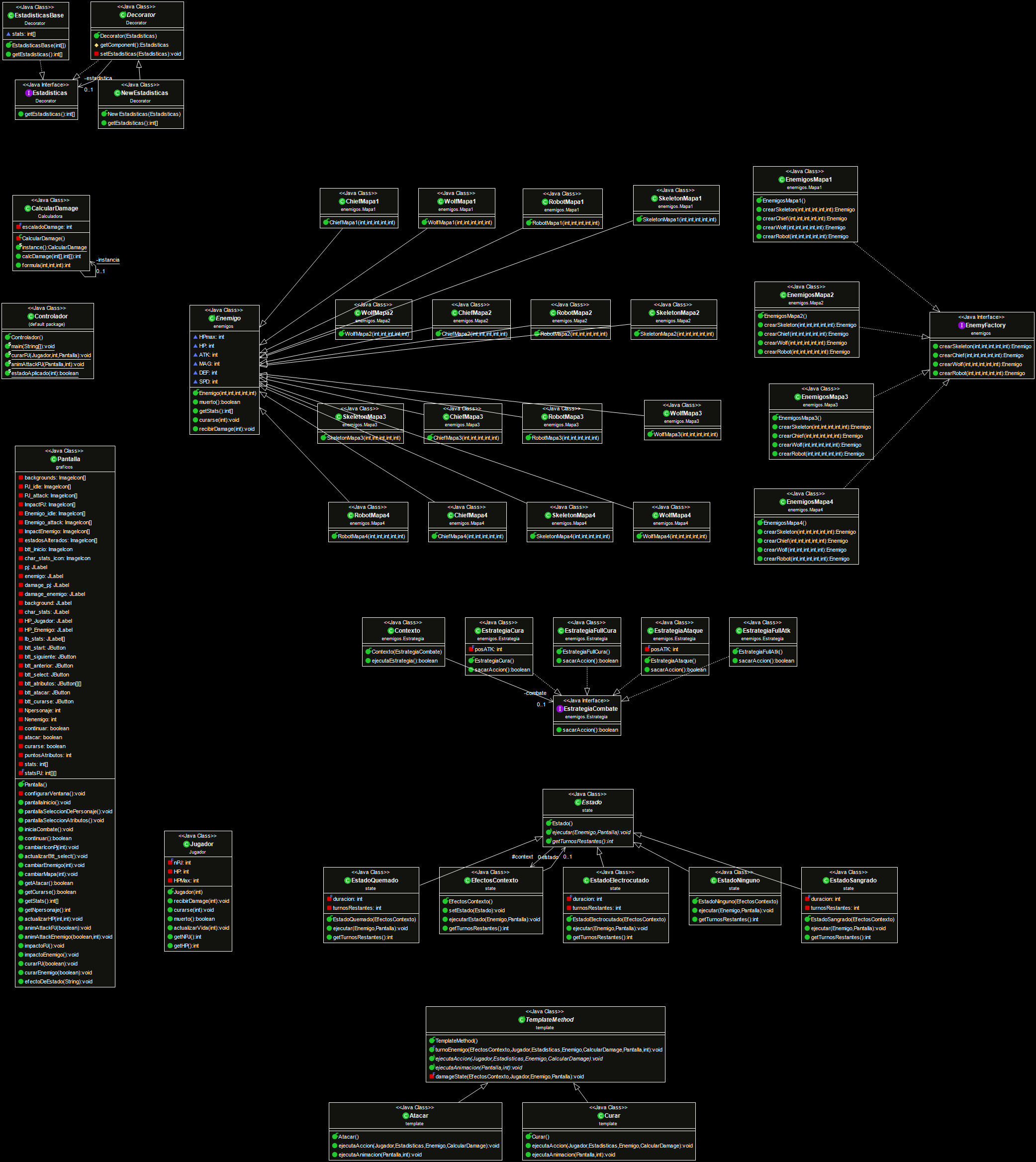
**Patrón Singleton**

Se compone de una única clase la cual 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, hecho para que únicamente se pueda crear una instancia de dicha clase.

**PATRON 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 \*Falta musica\***