

# Trabajo Práctico 2

## Al-Go-Oh!

[7507/9502] Algoritmos y Programación III

Curso 1

Primer cuatrimestre de 2018

Alumno	Padrón	E-mail
Souto, Rodrigo Nicolás	97649	rnsoutob@gmail.com
García Villamor, Delfina	101154	delfinagarciav@gmail.com
Schischlo, Franco Daniel	100615	
Silvestri, María Carolina	99175	silconito@hotmail.com

Fecha entrega final:

Tutor: Fontela, Carlos

Comentarios:

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Supuestos</b>	<b>2</b>
<b>3. Modelo de dominio</b>	<b>2</b>
<b>4. Diagramas de clase</b>	<b>3</b>
<b>5. Detalles de implementación</b>	<b>3</b>
5.1. Aliquam vel eros id magna vestibulum rhoncus . . . . .	3
5.2. Proin sodales leo dapibus sapien fermentum . . . . .	4
<b>6. Excepciones</b>	<b>4</b>
<b>7. Diagramas de secuencia</b>	<b>4</b>

## 1. Introducción

El presente informe reúne la documentación de la solución del segundo trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar el juego de cartas conocido como Yu-Gi-Oh!. Utilizando los conceptos del paradigma orientado a objetos vistos en el curso.

## 2. Supuestos

El trabajo inicialmente se diseñó pensando en la futura existencia de una interfaz gráfica que integre la totalidad de las clases. De este modo simplificamos el hecho de realizar una refactorización a gran escala. Aún así, se trabajó con las excepciones correspondientes para poder simular lo que la interfaz gráfica no nos hubiera permitido hacer.

El juego a pesar de que tenga 2 jugadores, los turnos van a ser jugados por una sola persona.

## 3. Modelo de dominio

Las clases principales de nuestra implementación son las siguientes:

- **Juego:** Es la clase principal. Contiene a los dos jugadores del juego. Establece el inicio y el final del juego.
- **Jugador:** Representa al jugador con sus puntos de vida, se encarga de todo lo que este puede realizar y contiene una referencia a su propio tablero.
- **Turno:** Se encarga de controlar las distintas fases de los turnos de los jugadores.
- **TableroJugador:** Es el tablero propio de cada jugador. Posee las distintas zonas del tablero: el cementerio, el mazo y el campo. Se encarga de la interacción entre las cartas y el campo durante el duelo.
- **Campo:** Zona del tablero donde se posicionan las distintas cartas del campo estas son: Mágicas, Trampa y Monstruos. Se encarga colocarlas en el lugar correcto y quitarlas en caso de ser necesario.

Además se implementó la clase Carta con sus :

- **Carta:** Es una clase abstracta reúne los atributos pertenecientes a cada carta (nombre, efecto, estado, en turno y jugador). Se encarga de todo lo que puede realizar una carta en general. Es la clase madre de las siguientes: Monstruo y NoMonstruo, este último incluye las cartas mágicas y de trampa.
- **Monstruo y NoMonstruo:** Estas clases se encargan de las tareas específicas de cada tipo de carta. Como se mencionó anteriormente, NoMonstruo incluye a las cartas mágicas y de trampa. Ambas contemplan los modos (estado) en los que se puede encontrar la carta.

Dentro de carta encontramos interfaces y clases de estado que nos permiten saber la disposición de las cartas:

- **EstadoCarta:** Es una interfaz que se encarga de informar si la carta se encuentra en juego o no.
- **EnJuego y FueraDeJuego:** una carta está en juego cuando se encuentra en el campo, lo que implica que la carta puede ser utilizada. En cambio, cuando la carta esta fuera de juego esta se encuentra en el cementerio, en el mazo o en la mano y queda inhabilitada para su uso hasta que se la revive o se la coloca en el campo. Estas clases abstractas que implementan EstadoCarta se encargan de informar y verificar que las cartas que interactúan estén en juego.

- BocaArriba y BocaAbajo: Implementan EnJuego y son las encargadas de la lógica de las cartas en ese estado.
- ModoMonstuo: clase abstracta encargada de derivar a sus hijas el modo de resolver la logica del ataque entre monstruos estas son: ModoAtaque y ModoDefensa.

Además las cartas posen efectos por lo tanto las clases que manejan esto son:

- EfectoCarta: Es una interfaz, corresponde al efecto de una carta.
- EfectoNulo: le corresponde a una carta sin efecto.

Clase EfectoAgujeroNegro: Implementa EfectoCarta, corresponde a la carta mágica “Agujero Negro”. Clase ResultadoCombate: Es una interfaz, se encarga de resolver la problemática de dañar a quien sea el jugador con su respectivo monstruo del combate, que se vio afectado en batalla de monstruo contra monstruo.

EN CONSTRUCCIÓN

## 4. Diagramas de clase

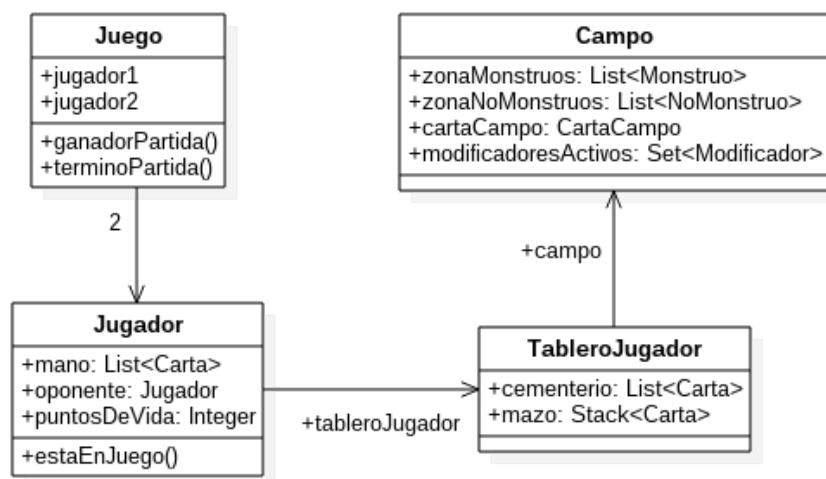


Figura 1: Diagrama de clases a nivel del juego.

## 5. Detalles de implementación

### 5.1. Aliquam vel eros id magna vestibulum rhoncus

.

## 5.2. Proin sodales leo dapibus sapien fermentum

## 6. Excepciones

## 7. Diagramas de secuencia

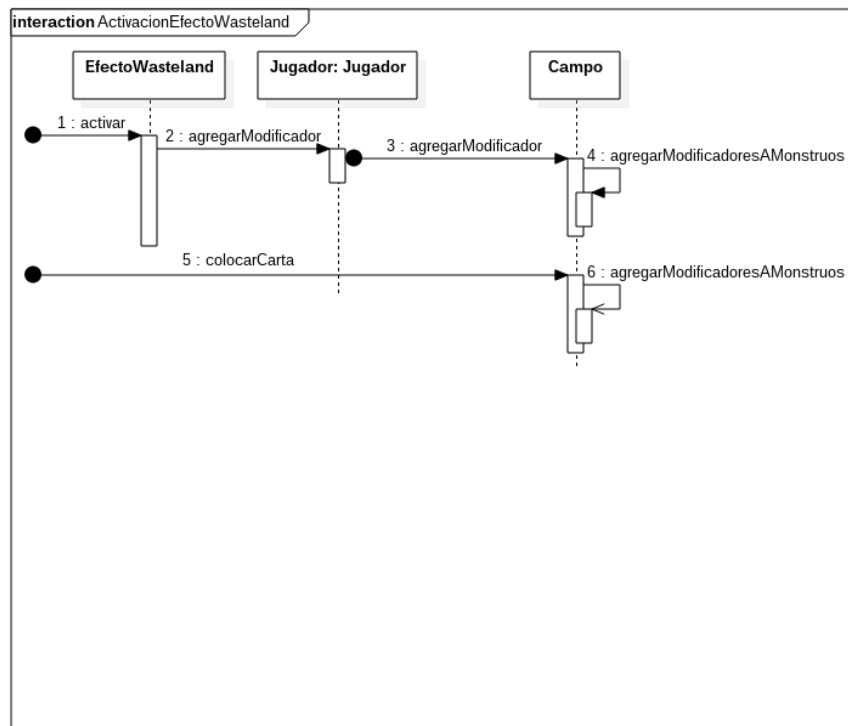


Figura 2: Diagrama que representa la activación del efecto Wasteland.

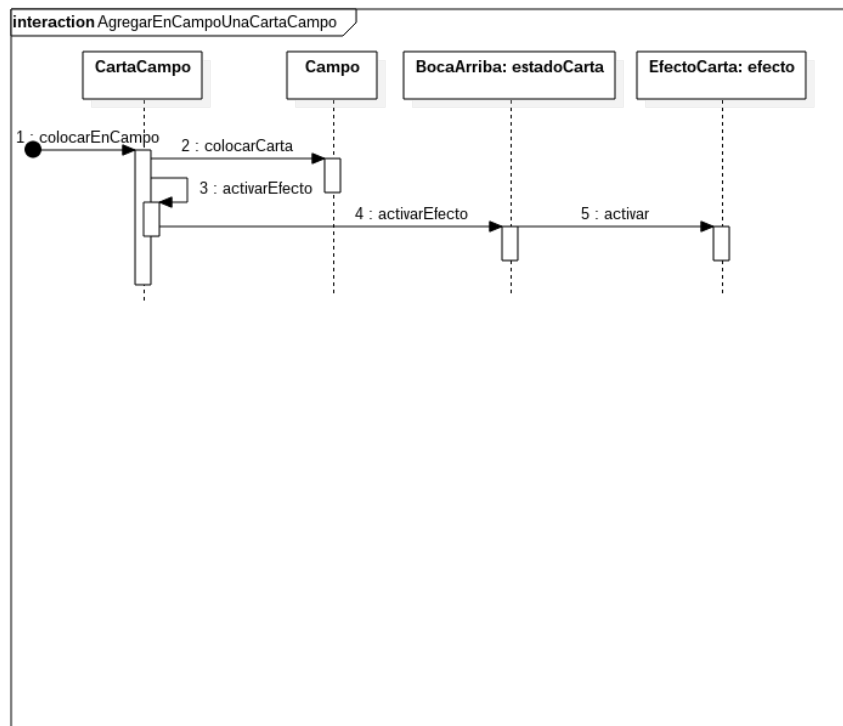


Figura 3: Diagrama de secuencia: colocación de cartas en campo.

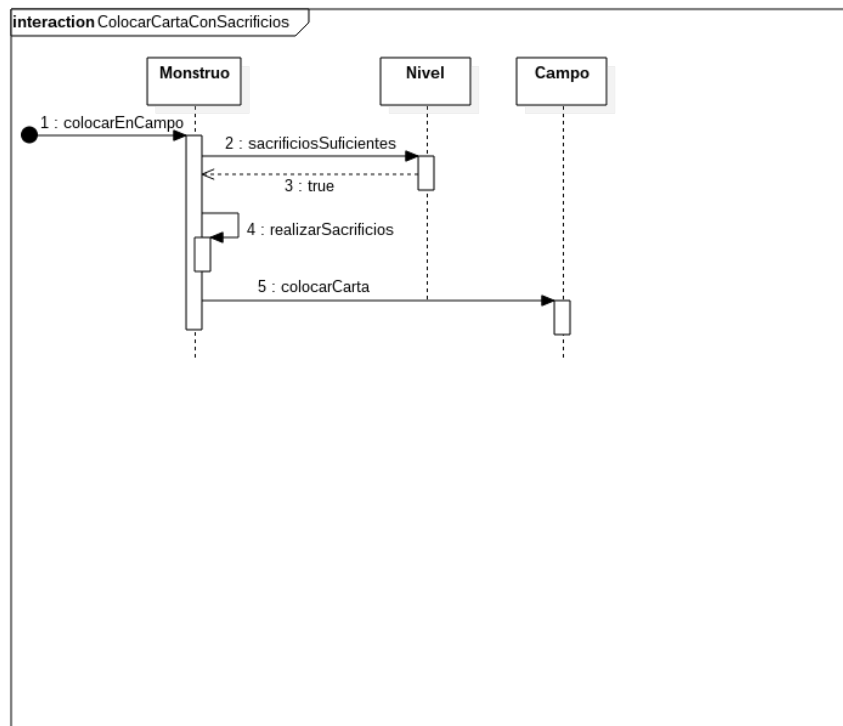


Figura 4: Diagrama de secuencia: colocación de monstruos que requieren sacrificios.