

# Trabajo Práctico 2 — A.L.T.E.G.O

[7507/9502] Algoritmos y Programación III  
Curso 2  
Primer cuatrimestre de 2021

Integrante 1:	Fernandez, Leonel
Integrante 2:	Higa, Axel
Integrante 3:	Pereira, Francisco
Integrante 4:	Valdovinos, Julia
Integrante 5:	Valdovinos, Lucia

## Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Fase Inicial . . . . .	2
1.2. Fase de Juego . . . . .	2
<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>10</b>
5.1. Método: atacarPaisDesdeA . . . . .	10
5.2. Clase: Juego . . . . .	10
5.3. Paquete: Cartas . . . . .	10
<b>6. Excepciones</b>	<b>10</b>
<b>7. Diagramas de secuencia</b>	<b>11</b>

## 1. Introducción

La aplicación consiste en una versión del juego T.E.G. que plantea un conflicto bélico que ocurre sobre un planisferio dividido en 50 países.

Consiste en dos fases:

### 1.1. Fase Inicial

Se selecciona la cantidad de jugadores en la partida (de 2 a 6) y se les asigna colores. Se decide cuál es el primer jugador de forma aleatoria, luego los turnos continúan en forma ascendente según los números de los jugadores.

A cada jugador se le asigna un objetivo secreto que debe cumplir para ganar el juego. (también puede cumplir el objetivo general de dominar 30 o más países). Para evitar que el objetivo secreto sea divulgado con todos los jugadores, los objetivos deberán ser revelados de a uno a la vez a medida que cada jugador lo indique. Mientras el objetivo secreto de uno de los jugadores se muestra en la pantalla los otros deberán mirar para otro lado.

Se reparten los 50 países de forma equitativa y aleatoria entre todos los jugadores y en cada país se coloca un ejército del jugador al que fue asignado.

Cada jugador coloca 5 ejércitos en los países que desee.

Cada jugador coloca 3 ejércitos en los países que desee.

### 1.2. Fase de Juego

Luego comienza la fase de juego, la cual está compuesta por rondas en las que cada jugador puede:

- a) Atacar (Las veces que quiera, solo a países limítrofes/conectados).
- b) Reagrupar (Transferir ejércitos entre países limítrofes/conectados).
- c) Colocar ejércitos (Puede colocar tantos ejércitos como países domine, dividido 2).

## 2. Supuestos

Asumimos que cuando se refiere al 'jugador de la derecha', es aquel que tiene su numero de jugador mas grande en uno que el jugador actual. Cuando se refiere, por ejemplo, al jugador a la derecha del jugador2, se refiere al jugador3. En caso del ultimo jugador, el jugador6, su jugador a la derecha es el jugador1.

En la fase de reagrupación, solo se pueden pasar ejércitos a países limítrofes.

El atacante atacara con toda la cantidad de dados que pueda, no puede elegir con cuantos ataca.

## 3. Modelo de dominio

Comenzamos modelando las clases iniciales basándonos en las [reglas](#) clásicas del T.E.G y el enunciado. A medida que pasaban las entregas nuestra modalidad fue dividir la entrega a comienzo de semana juntándonos cada 1 o 2 días para ver como iba quedando todo, al llegar la entrega tomamos notas de las devoluciones para ir corrigiendo los errores que fueron surgiendo y ver posibles cambios y mejoras en el diseño. Mientras esto sucedía fuimos actualizando los diagramas de clases y los de secuencia. Luego diseñamos los distintos paquetes, agrupando todas las clases que se relacionaban fuertemente entre si, como por ejemplo, el paquete FlujoDeJuego solo tiene las clases que controlan el flujo del juego, las fases. Una vez terminado el modelo general de la aplicación, comenzamos a trabajar en la interfaz gráfica. Esta diseñada con el mapa del juego como centro y a un costado las acciones de cada jugador. Al comienzo se da a elegir la cantidad de jugadores y luego comienza la fase inicial. La interfaz gráfica también cuenta con una barra de menús arriba, donde se puede acceder a un botón de ayuda que redirige al usuario a las reglas del juego y opciones para la musica.

## 4. Diagramas de clase

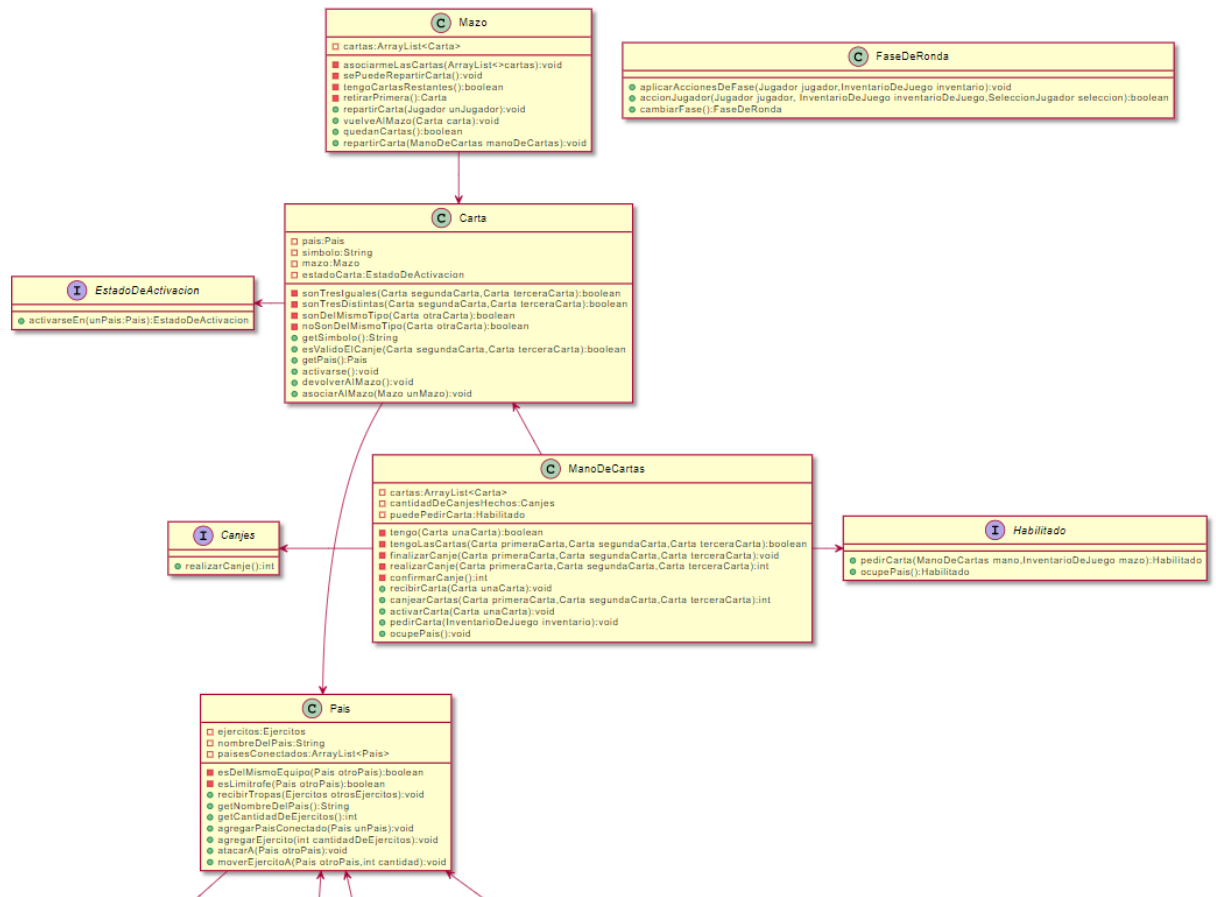


Figura 1: Diagrama de Clases, parte 1.

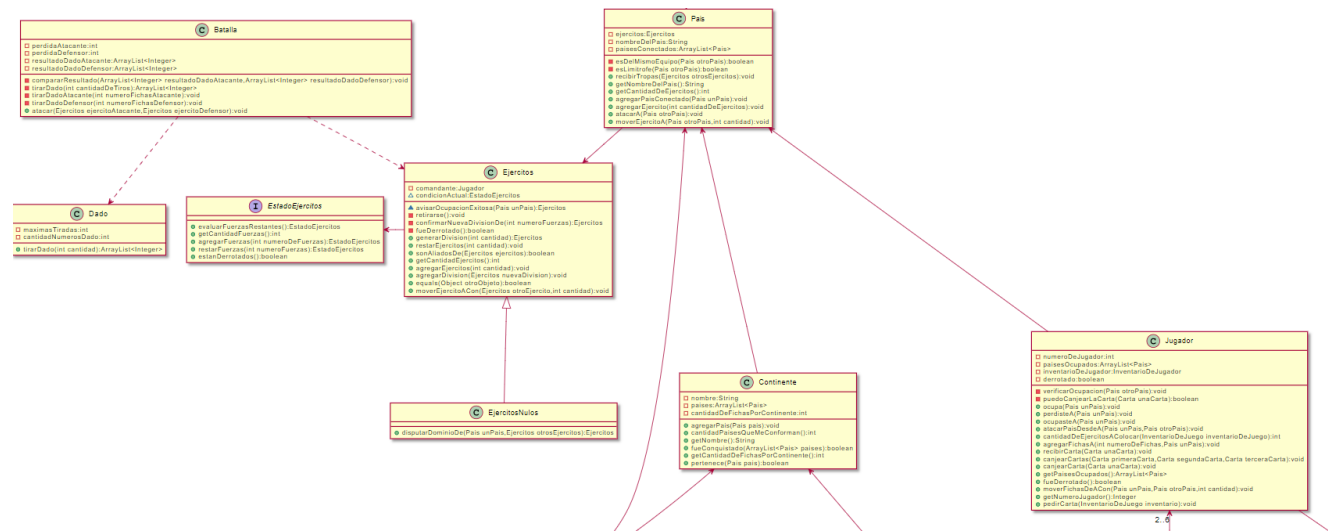


Figura 2: Diagrama de Clases, parte 2.

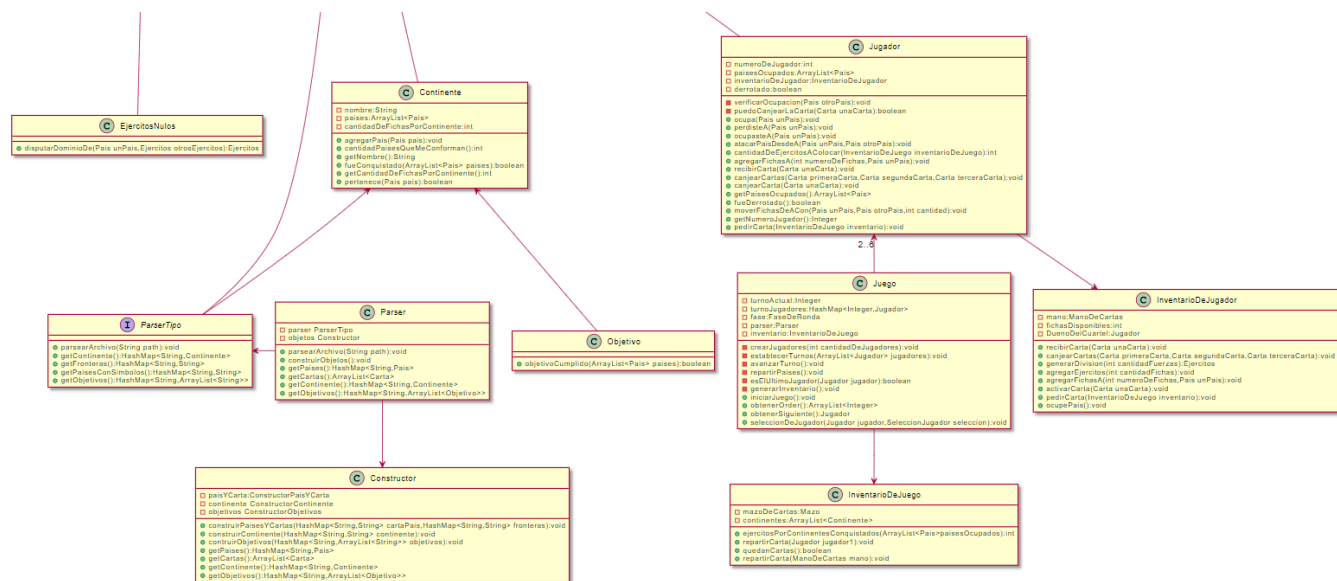
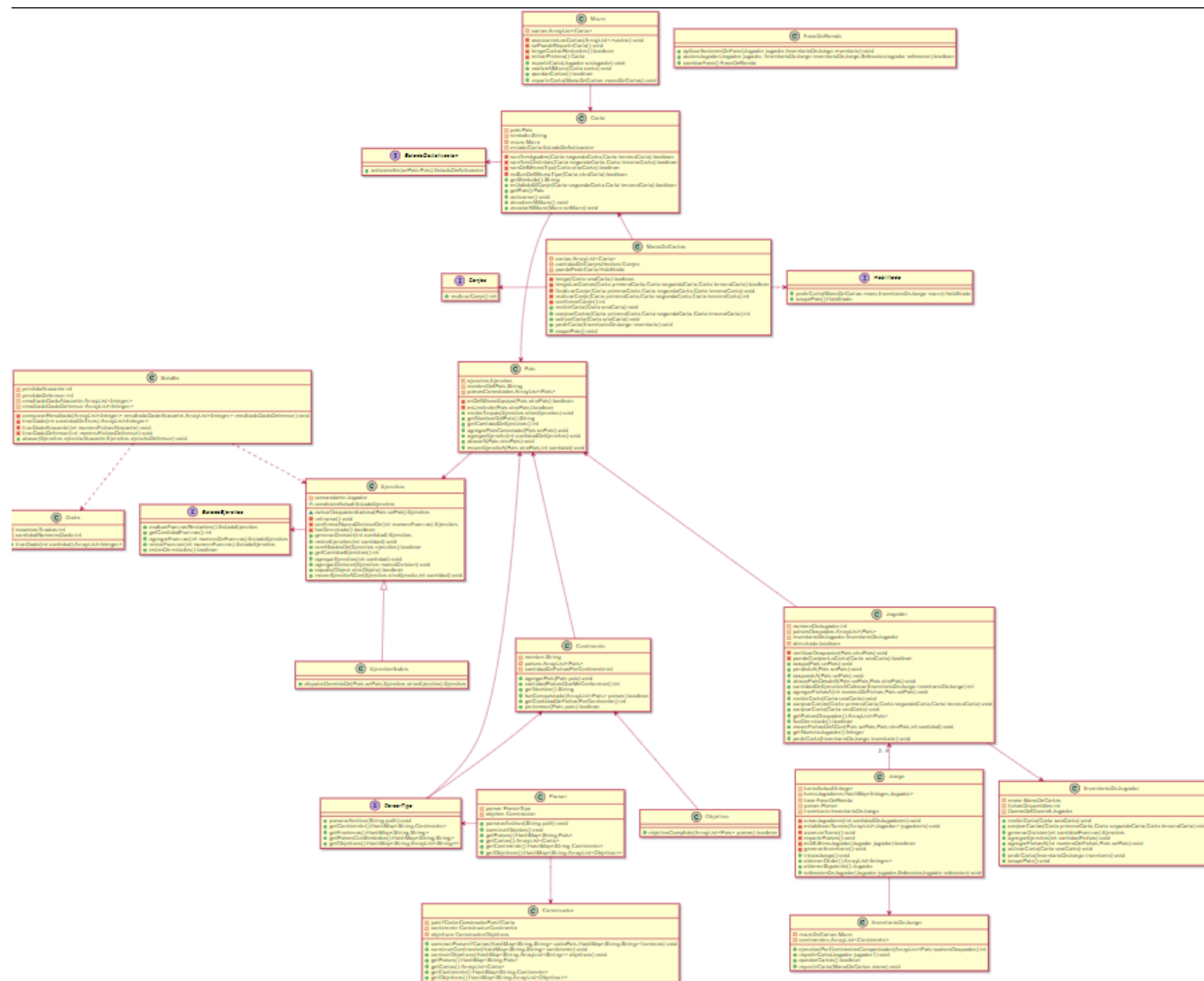


Figura 3: Diagrama de Clases, parte 3.



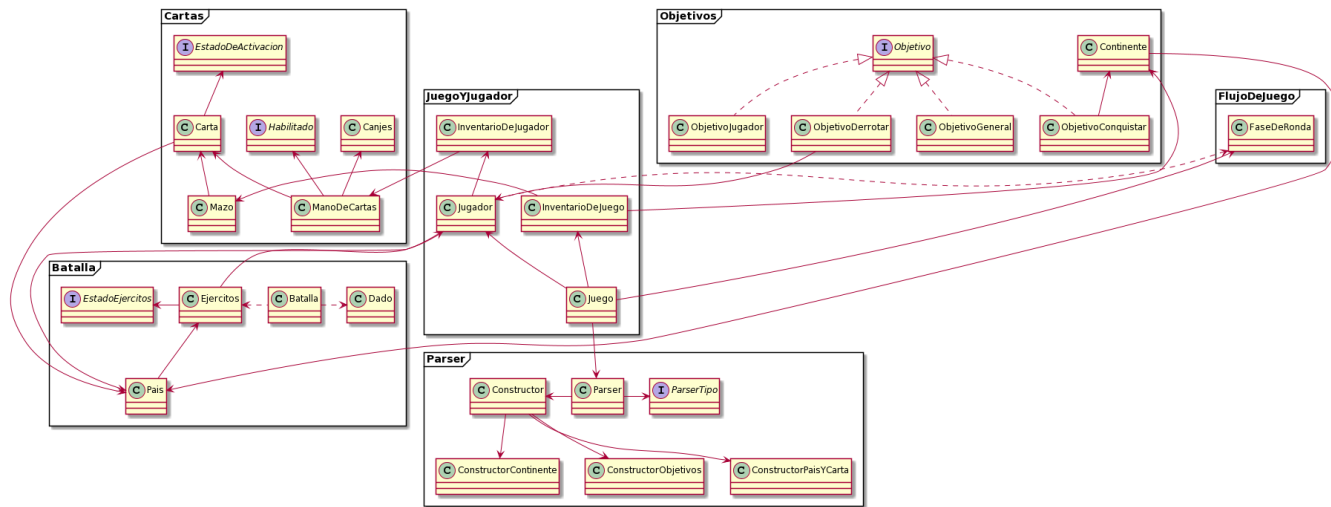


Figura 5: Diagrama de Paquetes.

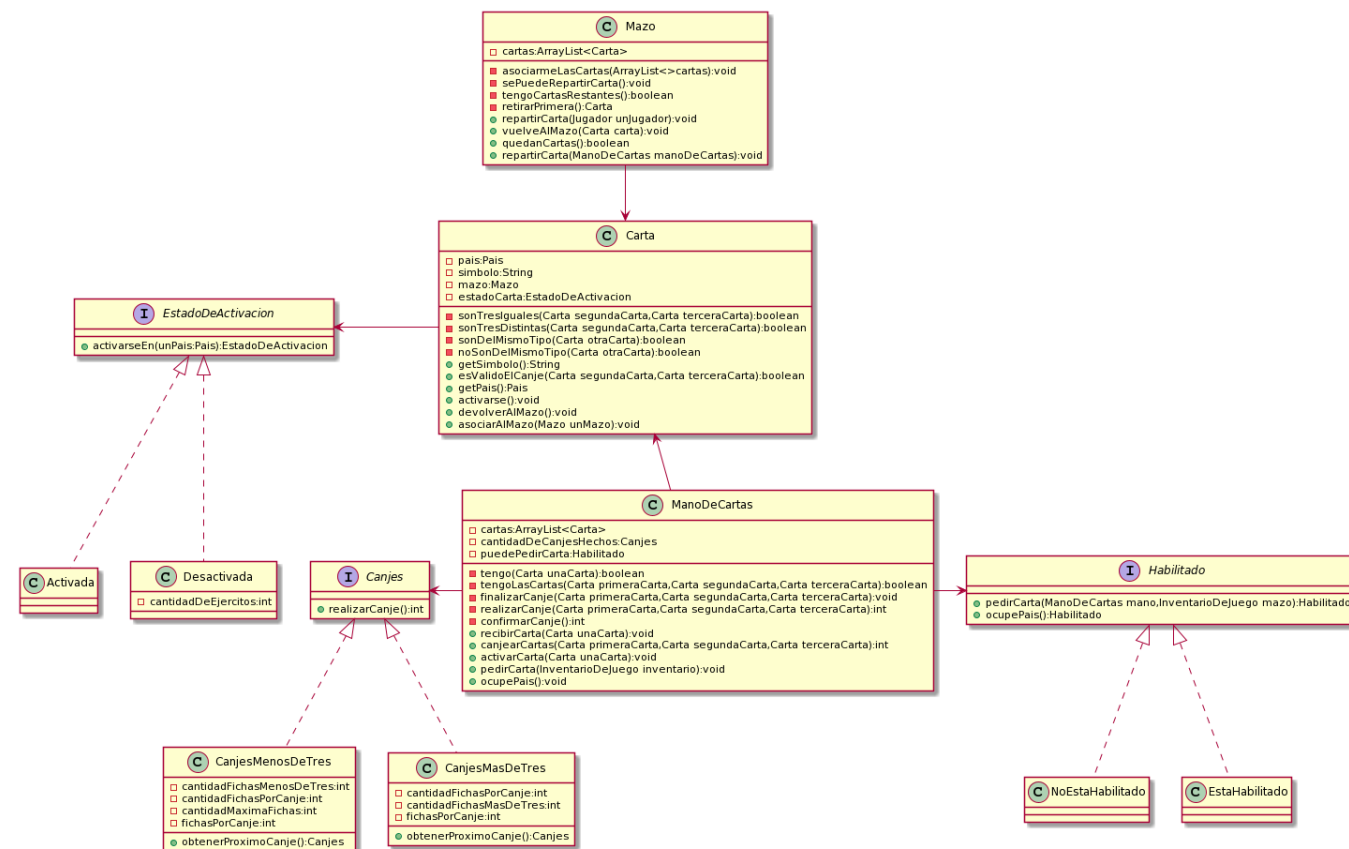


Figura 6: Diagrama del paquete Cartas.

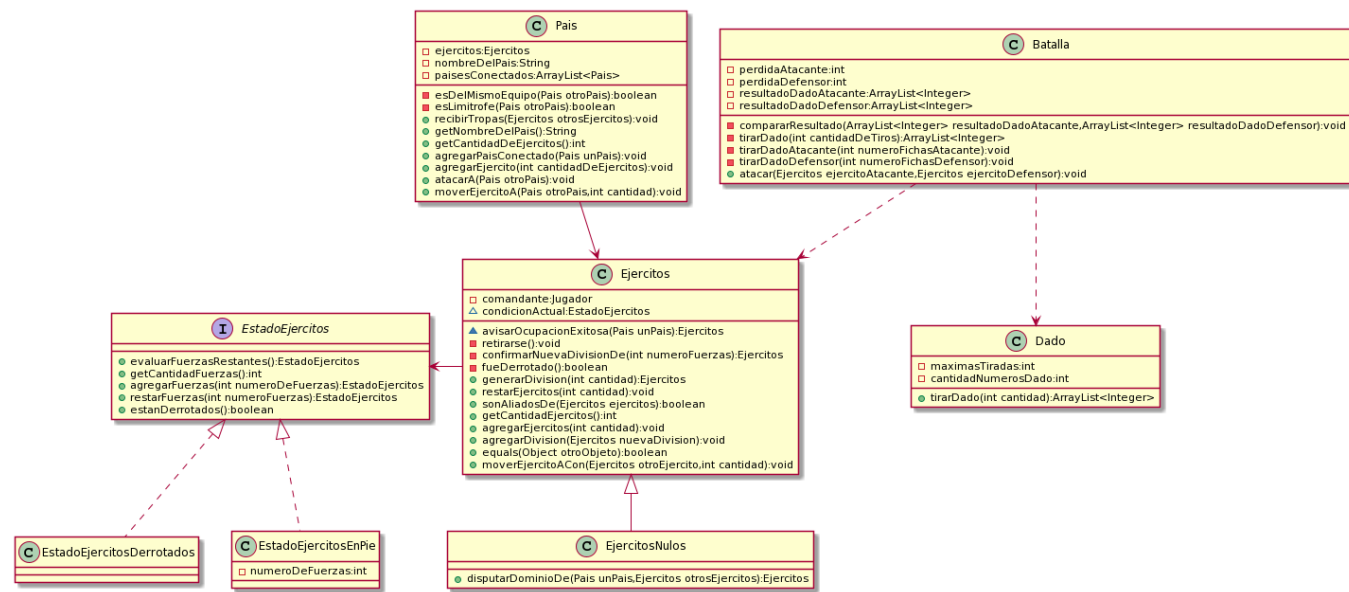


Figura 7: Diagrama del paquete Batalla.

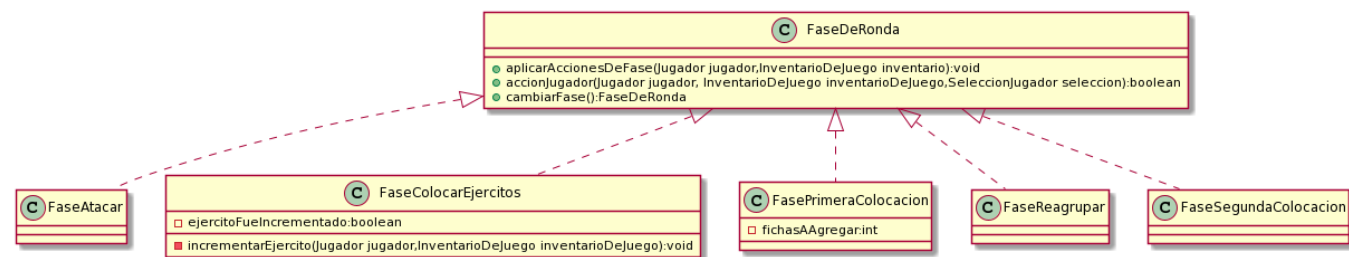


Figura 8: Diagrama del paquete FlujoDeJuego.



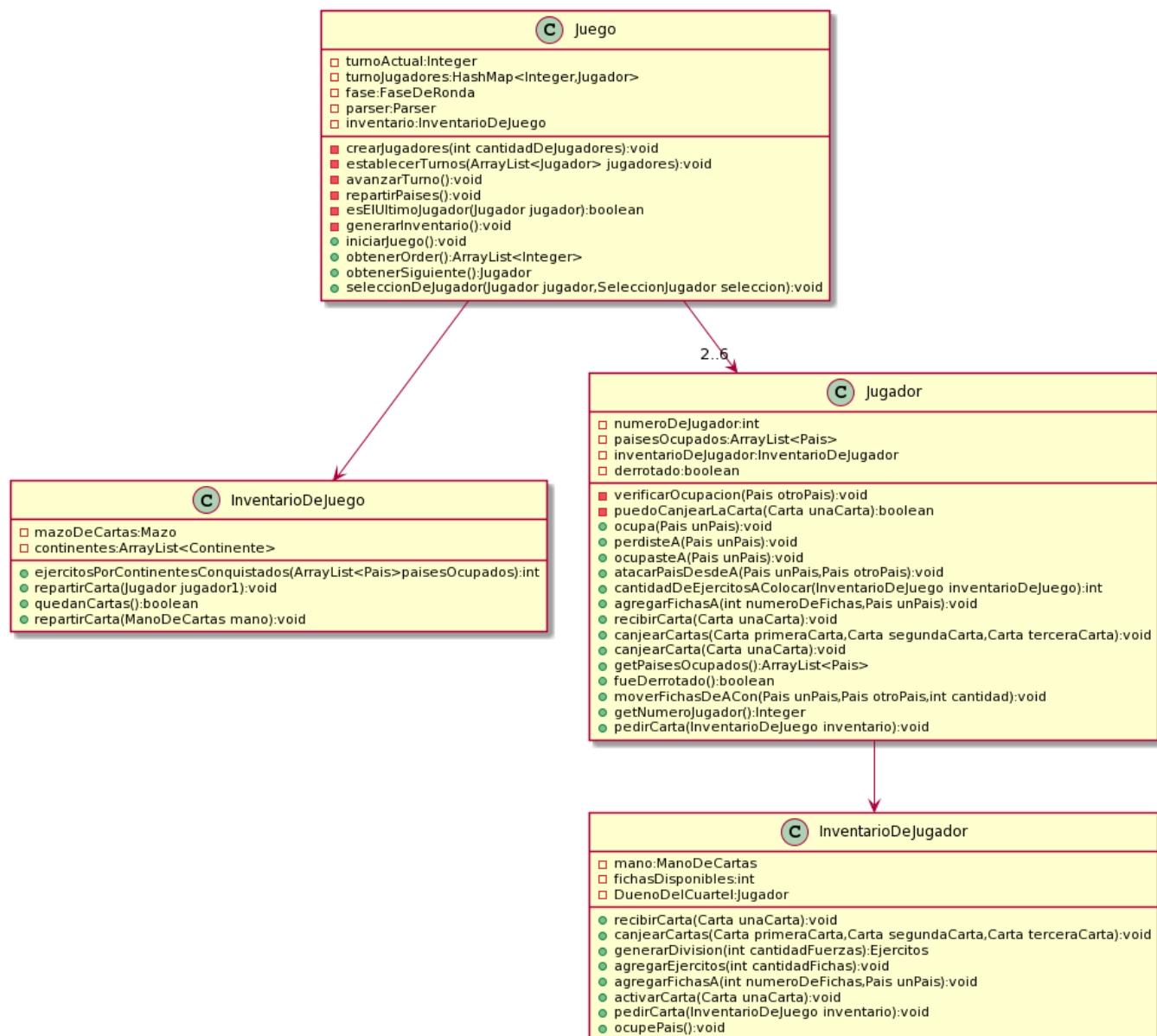


Figura 9: Diagrama del paquete JuegoYJugador.

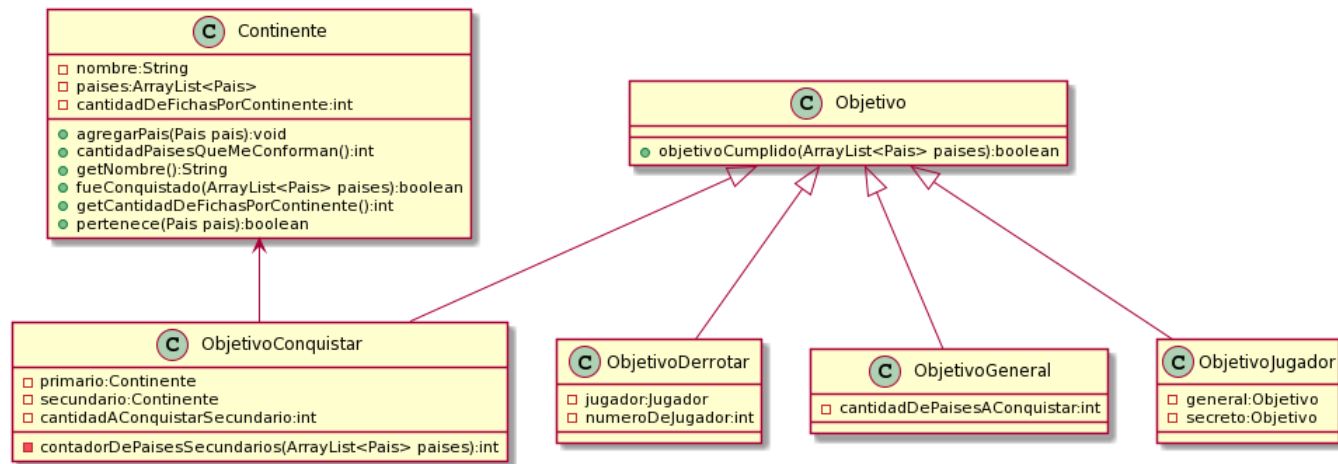


Figura 10: Diagrama del paquete Objetivos.

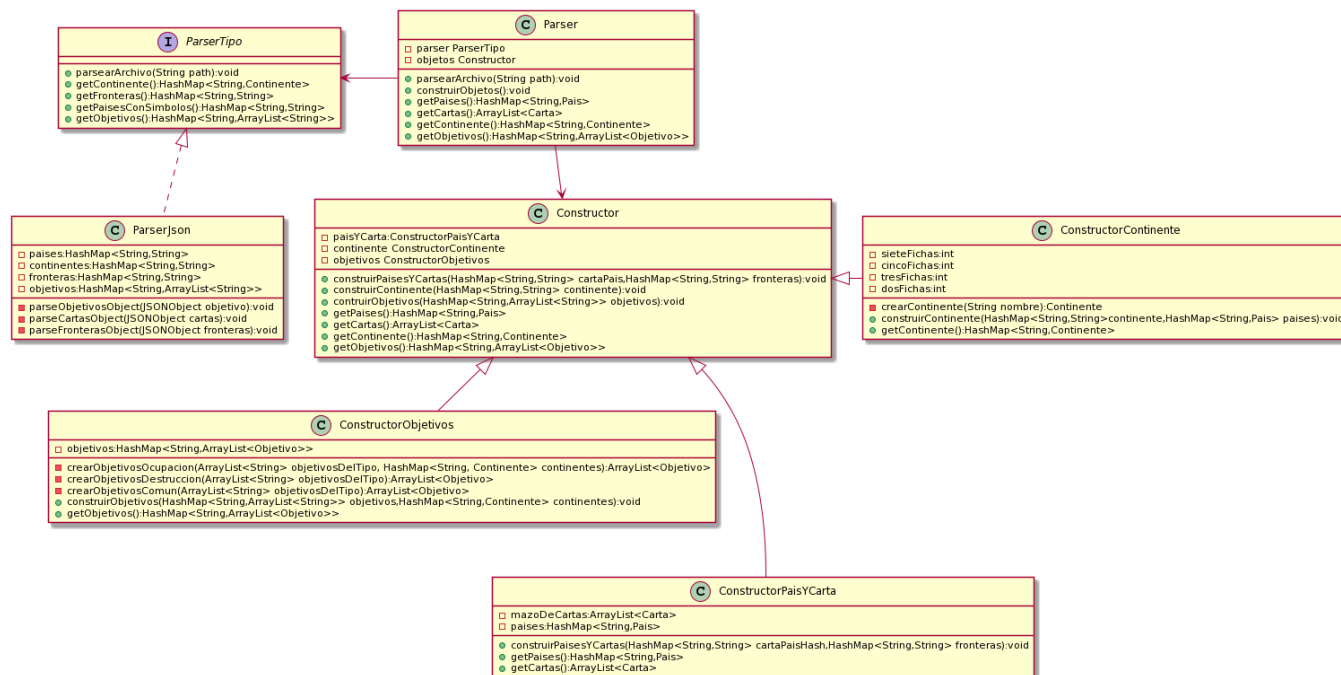


Figura 11: Diagrama del paquete Parser.

## 5. Detalles de implementación

A continuación, detalles de algunos métodos y clases sobre su implementación.

### 5.1. Método: atacarPaisDesdeA

Este es un método de la clase Jugador que se ejecuta durante la fase de atacar. Recibe como parámetro un país del jugador desde donde desea atacar y un país de un jugador contrario. Dentro de este método se invoca al método de país atacarA que recibirá el país del jugador enemigo. Dentro de este método, primero se verifica que el país recibido no sea uno del mismo jugador (que no sea un país aliado) y que el país recibido por parámetro sea limítrofe. En caso de que alguna de estas dos condiciones no se cumpla, se lanza una excepción y el jugador tendrá la posibilidad de elegir otro país para atacar. Una vez que se cumpla con la validez del otro país, se creará una batalla y esta se encargará de hacer el ataque. En la batalla, ambos países tirarán dados para ver quien es el ganador, luego se compara el resultado de estos dados y se restará la cantidad debida a cada ejército de cada país según el resultado. Una vez finalizada la batalla, se actualizará la cantidad correcta de ejércitos del país enemigo y, con esto, finalizando el proceso de un jugador ataca al país de otro jugador.

### 5.2. Clase: Juego

Esta sería la clase "principal" de la aplicación, es a la que se accede primero y va delegando métodos a el resto de las clases. Contiene un contador para saber en que turno se encuentra, un hash con los jugadores y sus turnos, un parser para leer los archivos de inicialización, una instancia de la interfaz FaseDeJuego, un inventario y una lista con las fichas que se utilizara para la interfaz gráfica. La instancia de esta clase se crea recibiendo como parámetro la cantidad total de jugadores. Esta clase se encarga de inicializar el resto de las clases necesarias para el correcto funcionamiento de la aplicación, como los jugadores, los países, los objetivos, las fases, etc.

### 5.3. Paquete: Cartas

Este paquete contiene la clase Carta, la interfaz Canjes, la interfaz EstadoDeActivacion, la interfaz Habilitado, la clase Mazo y la clase ManoDeCartas. Están relacionadas de la forma en que se indica en el diagrama de mas abajo. De acuerdo a estas relaciones es que decidimos crear el paquete. Dentro del mismo, es donde se maneja toda la funcionalidad de una carta, que es la responsable de contener a los países junto con sus símbolos. Esto se utiliza para los canjes del jugador, donde este puede cambiar países por fichas siempre que se cumpla con las condiciones de la carta. La cantidad de fichas devueltas depende de cuantos canjes ya hizo el jugador.

## 6. Excepciones

**NoHayFuerzasRestantesError** Se lanza una excepción cuando no haya ejércitos disponibles.

**EjercitosDerrotadosError** Se lanza una excepción cuando un jugador intenta atacar un país propio o un país no limítrofe.

**AtaqueNoPermitidoError** Se lanza una excepción cuando se intenta agregar fuerzas a un ejército ya derrotado.

**MovimientoDeEjercitoError** Se lanza una excepción cuando no haya fuerzas suficientes para realizar el movimiento o el movimiento sea invalido.

**NoQuedanCartasError** Se lanza una excepción cuando se quieran sacar cartas del mazo y este se encuentre vacío.

## 7. Diagramas de secuencia

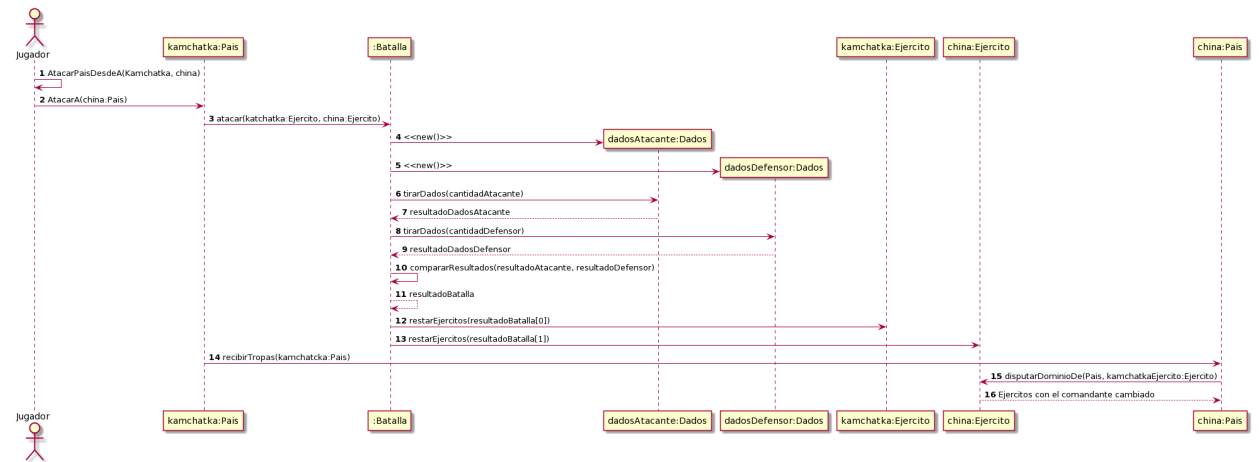


Figura 12: Diagrama de secuencia: Un País ataca a otro.

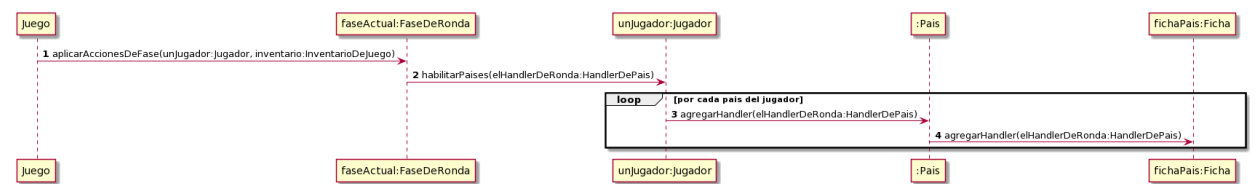


Figura 13: Diagrama de secuencia: Fases del Juego.