FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2 — Balatro

[7507/9502] Algoritmos y Programación III
Paradigmas de la Programación
Curso 01 - Suárez
Segundo cuatrimestre de 2024

Alumno	Número de padrón	Email
BUSTOS, Franco	110759	fbustos@fi.uba.ar
HONDA, Federico	110766	fhonda@fi.uba.ar
KOTELCHUK, Camila	110289	ckotelchuk@fi.uba.ar
PEZMAN, Bruno	110457	bpezman@fi.uba.ar

Tutor:

Nota Final:

Índice

1. Introducción **2**

2. Supuestos **2**

3. Diagramas de clase **3**

4. Diagramas de Secuencia **7**

5. Diagrama de Paquetes **15**

6. Detalles de Implementación **20**
6.1. Principios S.O.L.I.D 20
6.1.1. Single Responsability 20
6.1.2. Liskov's Substitution Principle 21
6.2. Herencia y Delegación 21
6.3. Patrones de Diseño 21
6.3.1. Factory Method 21
6.3.2. Singleton 21
6.3.3. Decorator 21
6.3.4. Facade 21
6.3.5. Strategy 22
6.3.6. Chain of Responsability 22

1. Introducción

El presente informe reune la documentación de la solución del segundo trabajo práctico de la materia Algoritmos y Programación III, que consiste en desarrollar el juego **BALATRO** utilizando los conceptos del paradigma de la orientación a objetos vistos en el curso. Esto lo hacemos con el fin de mejorar su implementación interna, con la aplicación apropiada de los mismos.

2. Supuestos

- Como DESCARTE se entiende a la accion de seleccionar hasta 3 cartas de las 8 que posee el jugador y obtener del mazo la cantidad de cartas descartadas (ej: Tengo 8 cartas, selecciono 2 y las descarto, eso cuenta como UN DESCARTE de los que se pueda hacer en la ronda que se este).
- Como MANO JUGABLE entendemos a el conjunto de EXACTAMENTE 5 cartas de las 8 que posee el jugador (ej: si el jugador selecciona 4 cartas del mismo palo no sera tomado como mano Poker hasta que seleccione una quinta carta) .
- Si una RONDA tiene **n descartes totales** para hacer y **m** TURNOS, la suma de descartes en los **m** TURNOS no podrá superar los **n descartes totales** de la RONDA.
- Los COMODINES de COMBINACIÓN activan sus comodines por **separado**. Es decir, al cumplirse una de las **condiciones de activación**, el **comodin** asociado a dicha **condición** se **activará**.
- Los COMODINES elegidos al inicio de cada RONDA, se activarán durante la totalidad del juego en el orden que fueron elegidos. Es decir, sin importar la ronda en la que se esté, se podrán utilizar dichos comodines.
- Los COMODINES NO SON INTERCAMBIABLES. Una vez completado el cupo de 5 comodines no sera posible obtener mas y NO PODRAN SER REORDENADOS.
- Los TAROTS elegidos en el inicio de cada RONDA solo aplicara sus efectos al seleccionarse y aplicarse, valga la redundancia. Luego el tarot sera eliminado de la lista de tarots del jugador.
- Por turno, solo UN TAROT PODRA SER UTILIZADO. Si se desea aplicar otro, los efectos del primero seran perdidos. Esto es definido de esta manera para que el jugador pueda 'gastar' sus tarots mas rapidamente ya que solo puede contar con 2 en su posicion.
- Solo se podrán realizar hasta 3 elecciones (u omitir) de las 5 opciones posibles de la TIENDA al inicio de cada ronda.
- Al comprar una CARTA en la TIENDA, esa carta sera añadida al mazo (el mismo teniendo ahora 53 cartas totales) y se mantendra SOLO EN ESA RONDA, eliminandose en la siguiente ronda.
- Al selecionar cartas para jugar una **mano**, y esta selección cumple con el **criterio** de **dos o más tipos de manos distintas**, se tomará el **tipo de mano** con **mayor puntuación**. (ej: escalera con todas cartas del mismo color, si bien puede ser considerada una escalera simplemente, esta se tomará como una escalera de color (cumple requisitos) al representar esta una mano de mayor puntaje.
- La definición de los efectos, tanto para comodines como para tarots, que aplican efectos de suma al puntaje o multiplicador de manos y cartas tendrán valor de 0 en caso que no afecten a ese atributo, basandonos para ello en la descripción de los mismos (ej: descripción: '+100 al multiplicador si se juega mano par' $->$ En este caso el efecto al puntaje es +0).

- Los TAROT que necesitaban de una eleccion de carta para aplicarse se aplicar siempre a la carta mas alta de la mano jugada. Esto con el fin de que el jugador no desperdicie un tarot de ese estilo siendo los mas raros en el juego.

3. Diagramas de clase

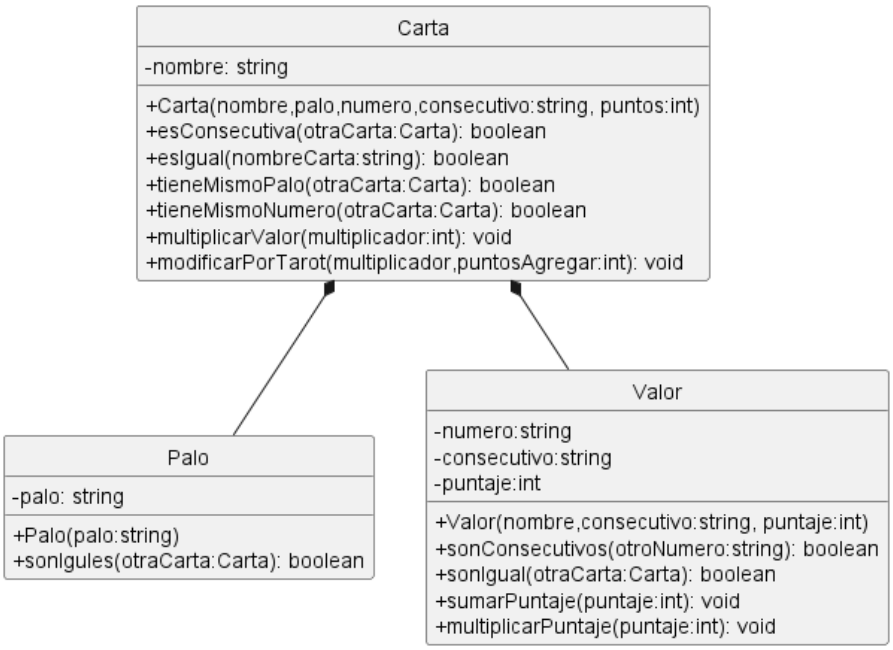

Figura 1: Diagrama de clases CARTA

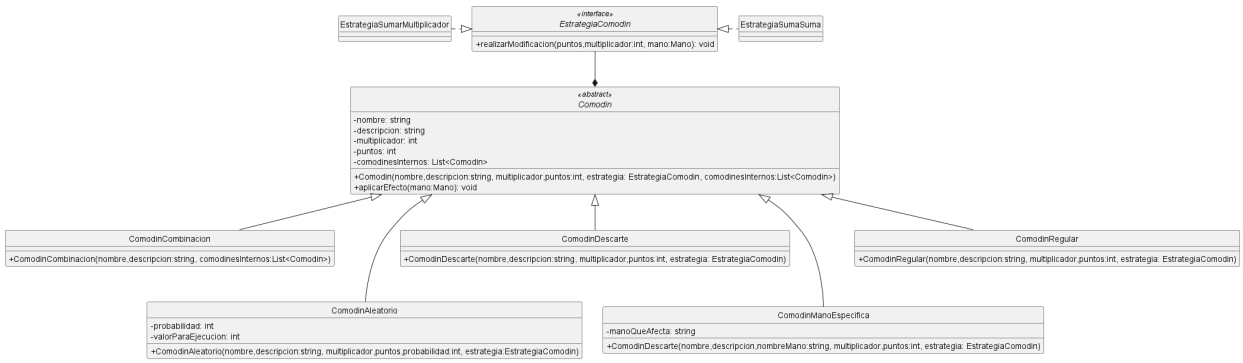

Figura 2: Diagrama de clases COMODIN

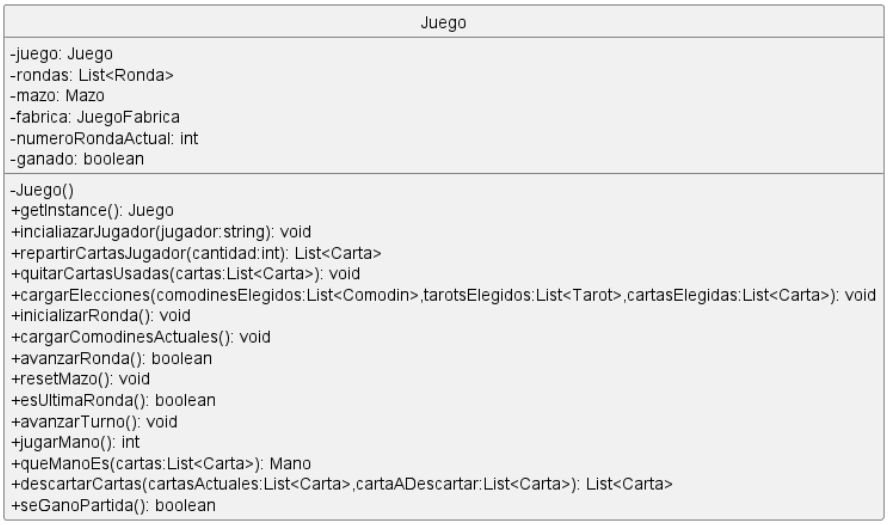

Figura 3: Diagrama de clases JUEGO

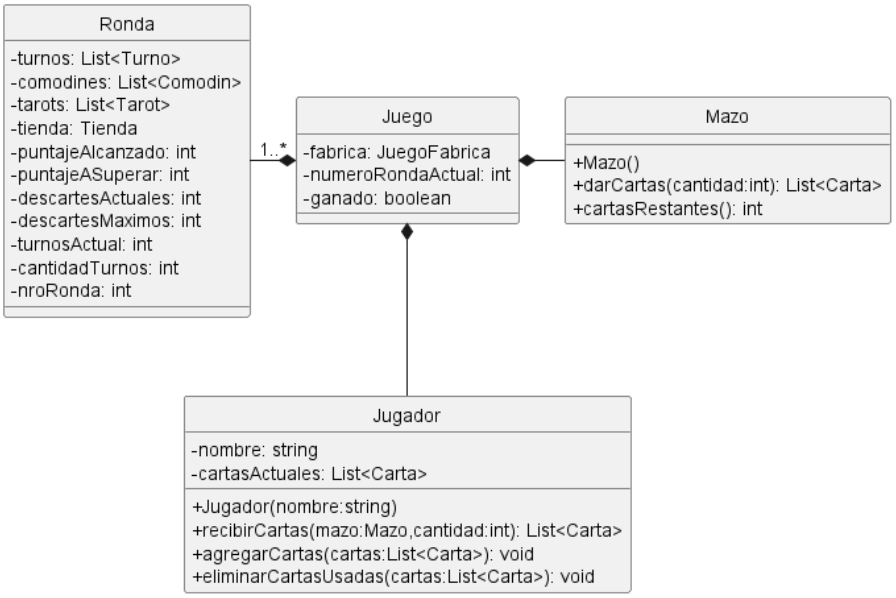

Figura 4: Diagrama de clases JUEGO - Relaciones

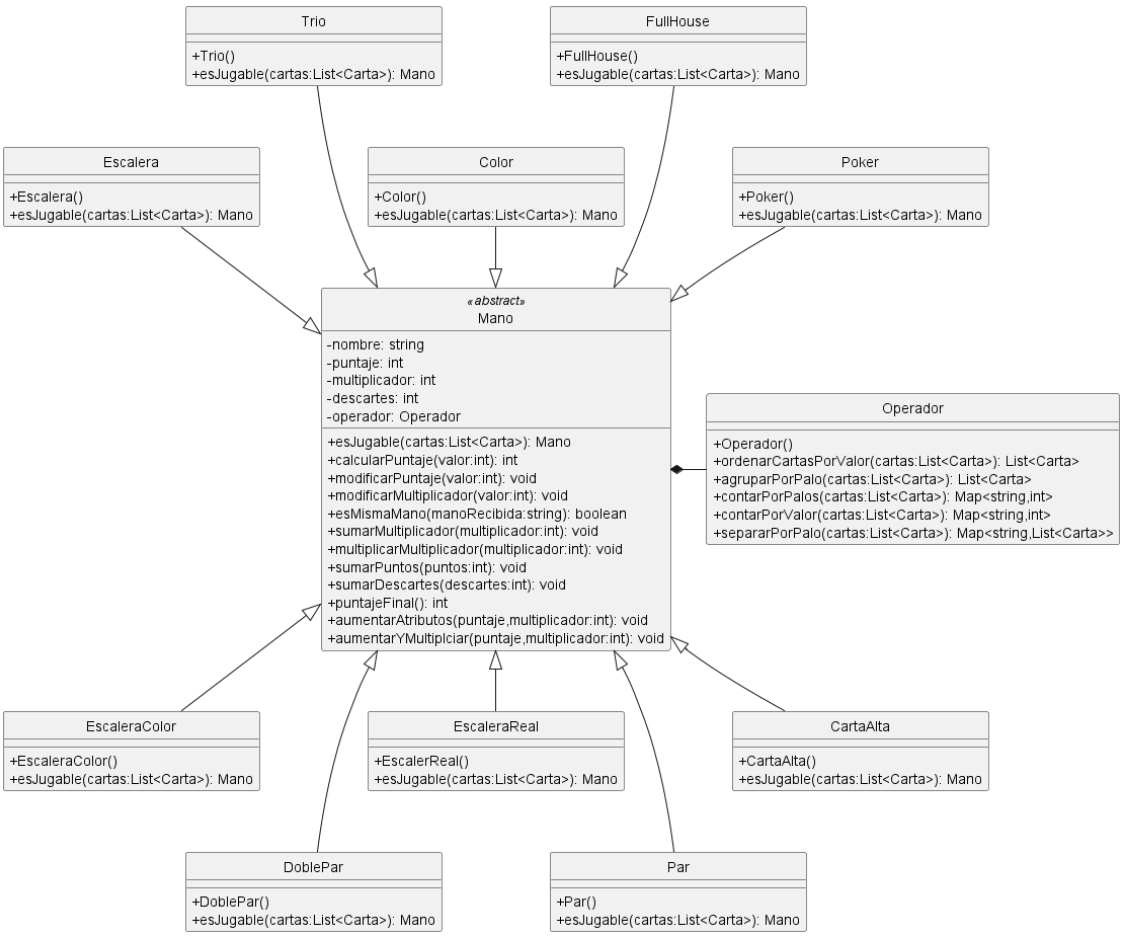

Figura 5: Diagrama de clases MANO

Figura 6: Diagrama de clases RONDA

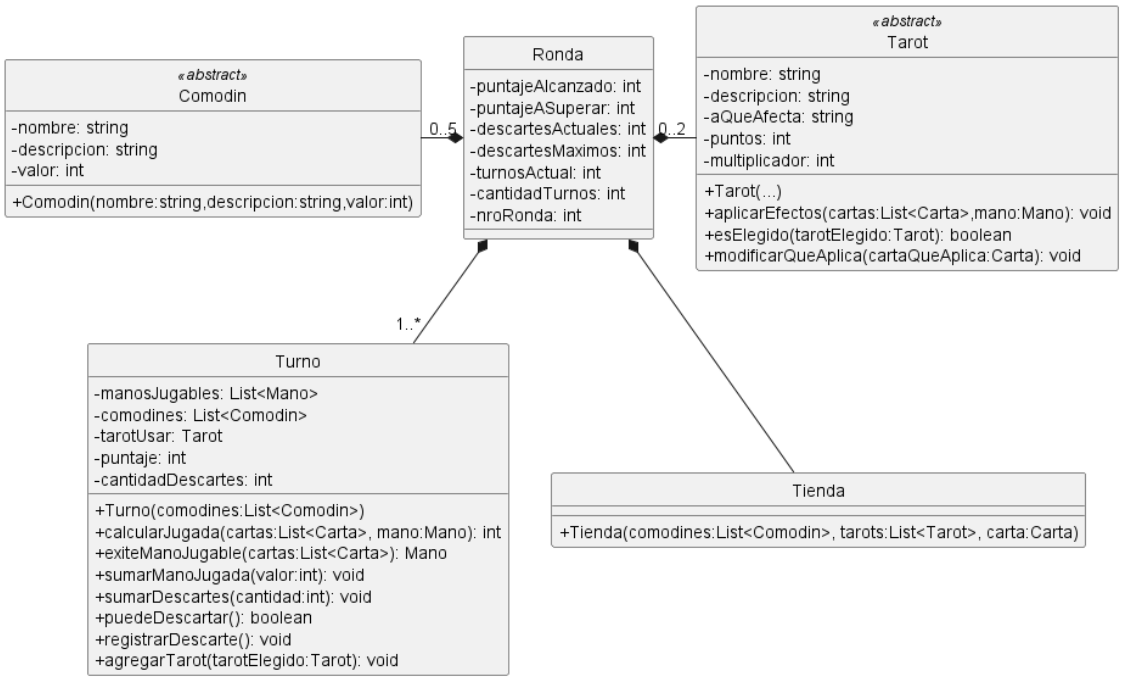

Figura 7: Diagrama de clases RONDA - Relaciones

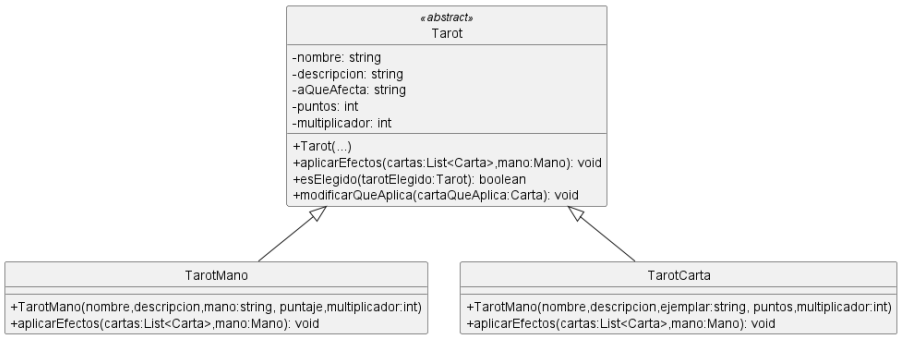

Figura 8: Diagrama de clases TAROT

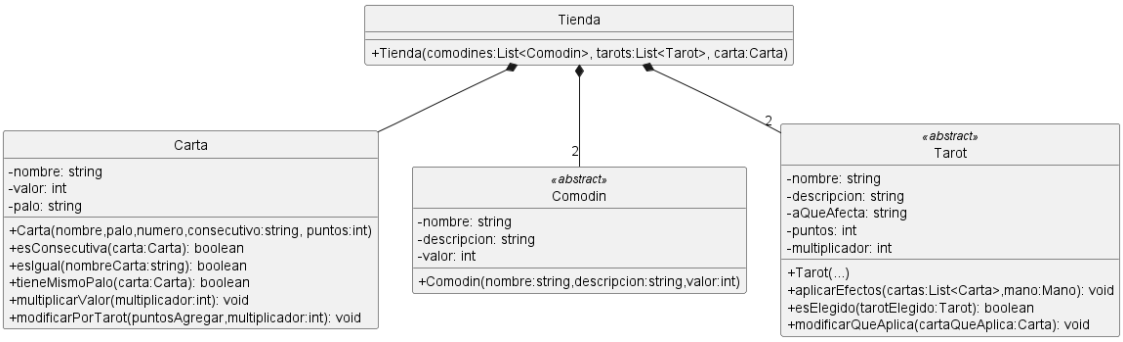

Figura 9: Diagrama de clases TIENDA

Aclaración: En los diagramas por temas de legibilidad y prolijidad optamos por 'simplificar' algunas clases, con el fin de que los diagramas no sean excesivamente grandes. Por ello, brindamos al principio de esta sección, los diagramas correspondientes a todas las clases que participan o no son desarrolladas en posteriores diagramas.

4. Diagramas de Secuencia

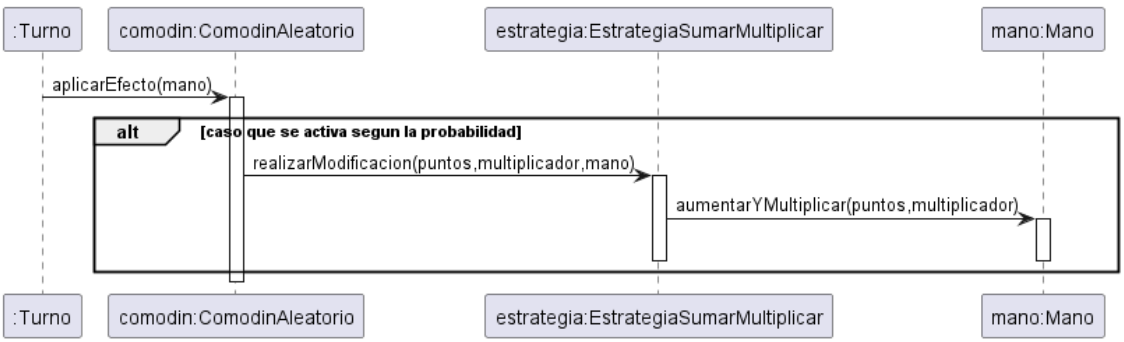

Figura 10: Diagrama de secuencia COMODIN ALEATORIO - AplicarEfecto

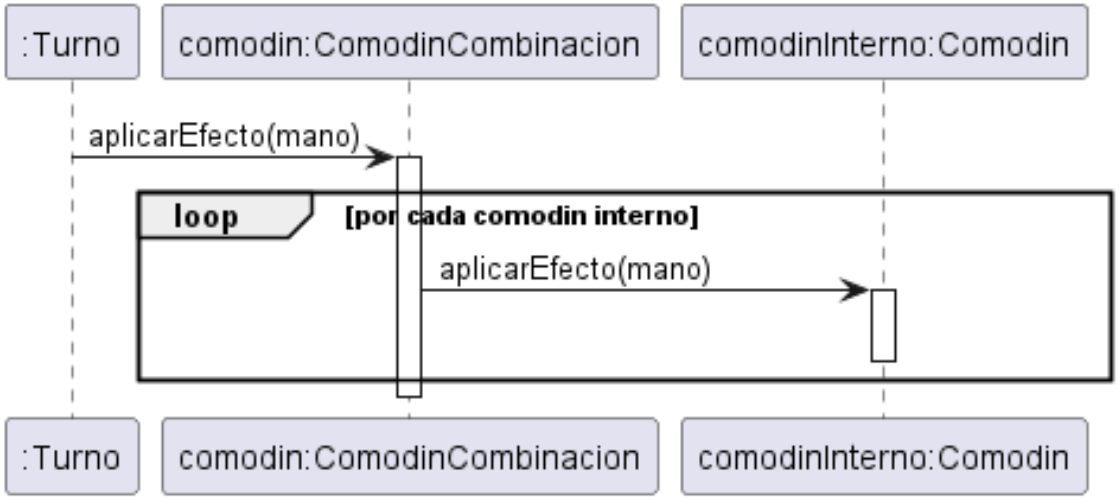

Figura 11: Diagrama de secuencia COMODIN COMBINACION - AplicarEfecto

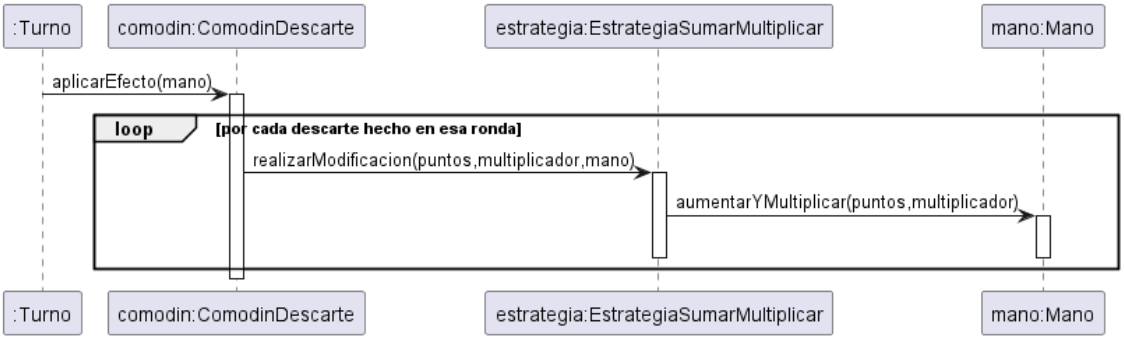

Figura 12: Diagrama de secuencia COMODIN DESCARTE - AplicarEfecto

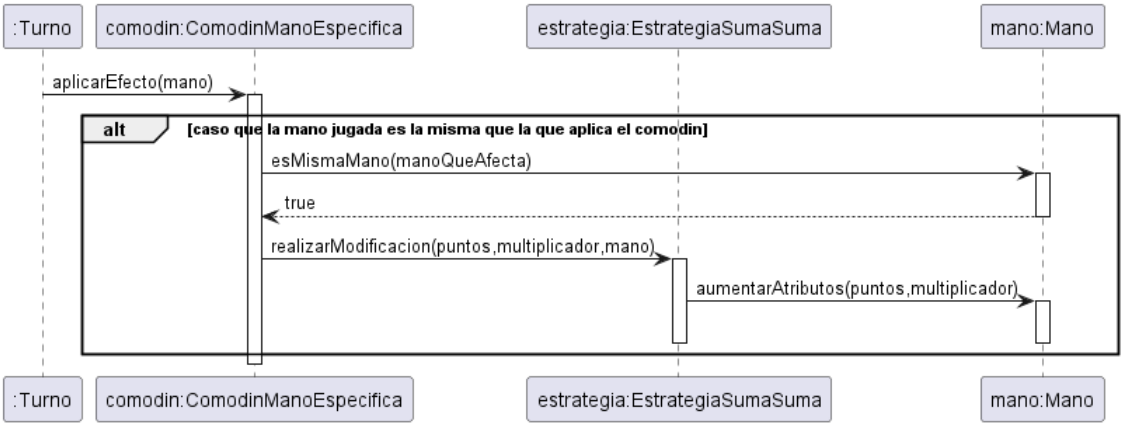

Figura 13: Diagrama de secuencia COMODIN MANO ESPECIFICA - AplicarEfecto

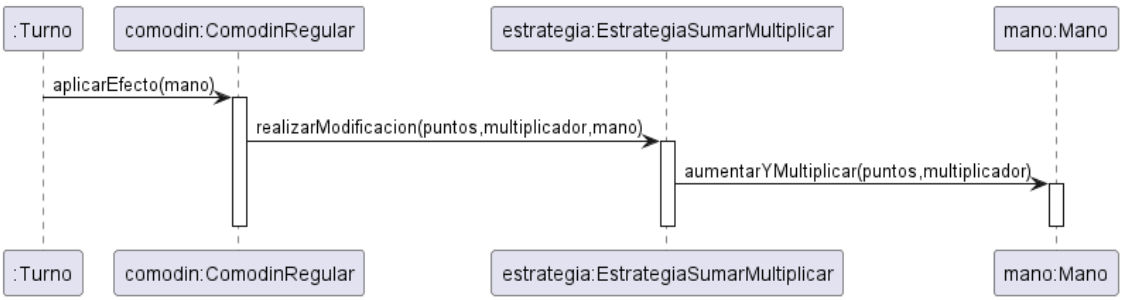

Figura 14: Diagrama de secuencia COMODIN REGULAR - AplicarEfecto

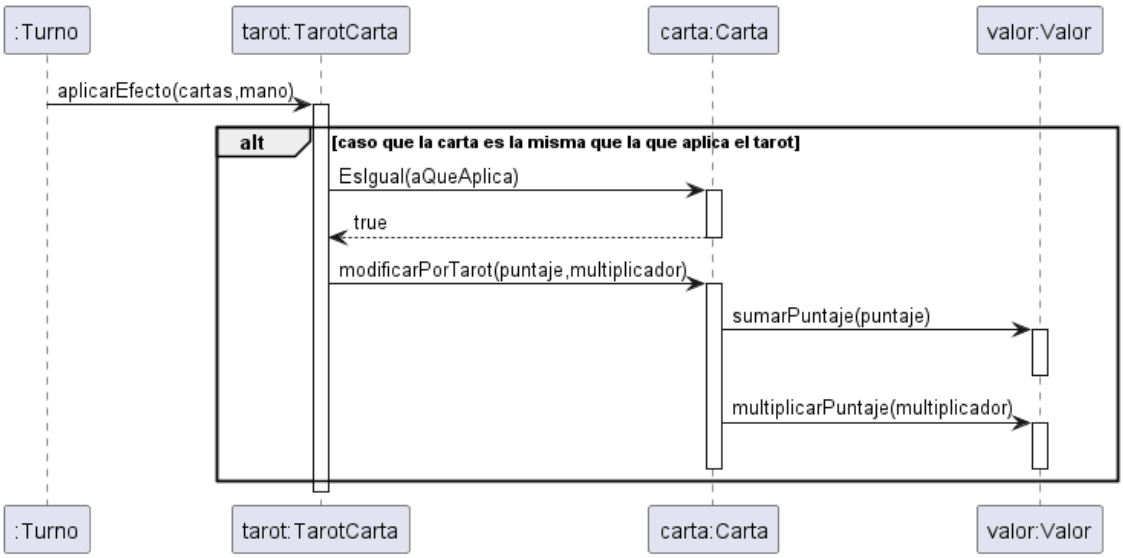

Figura 15: Diagrama de secuencia TAROT CARTA - AplicarEfectos

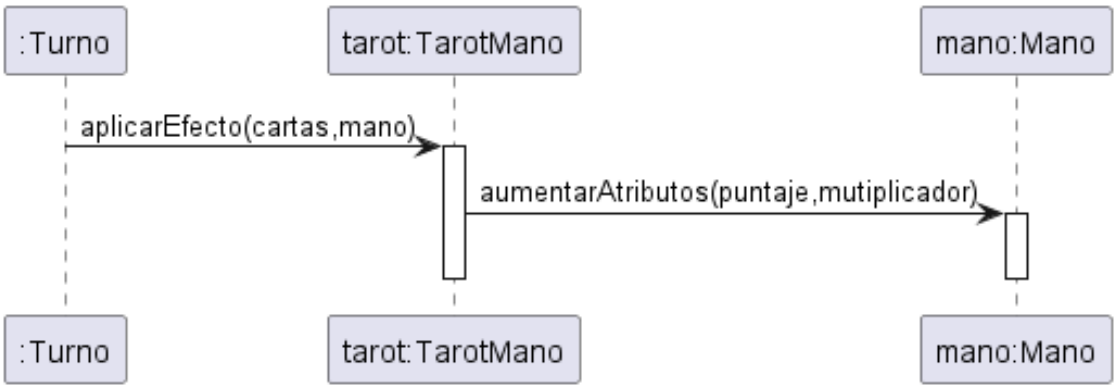

Figura 16: Diagrama de secuencia TAROT MANO - AplicarEfectos

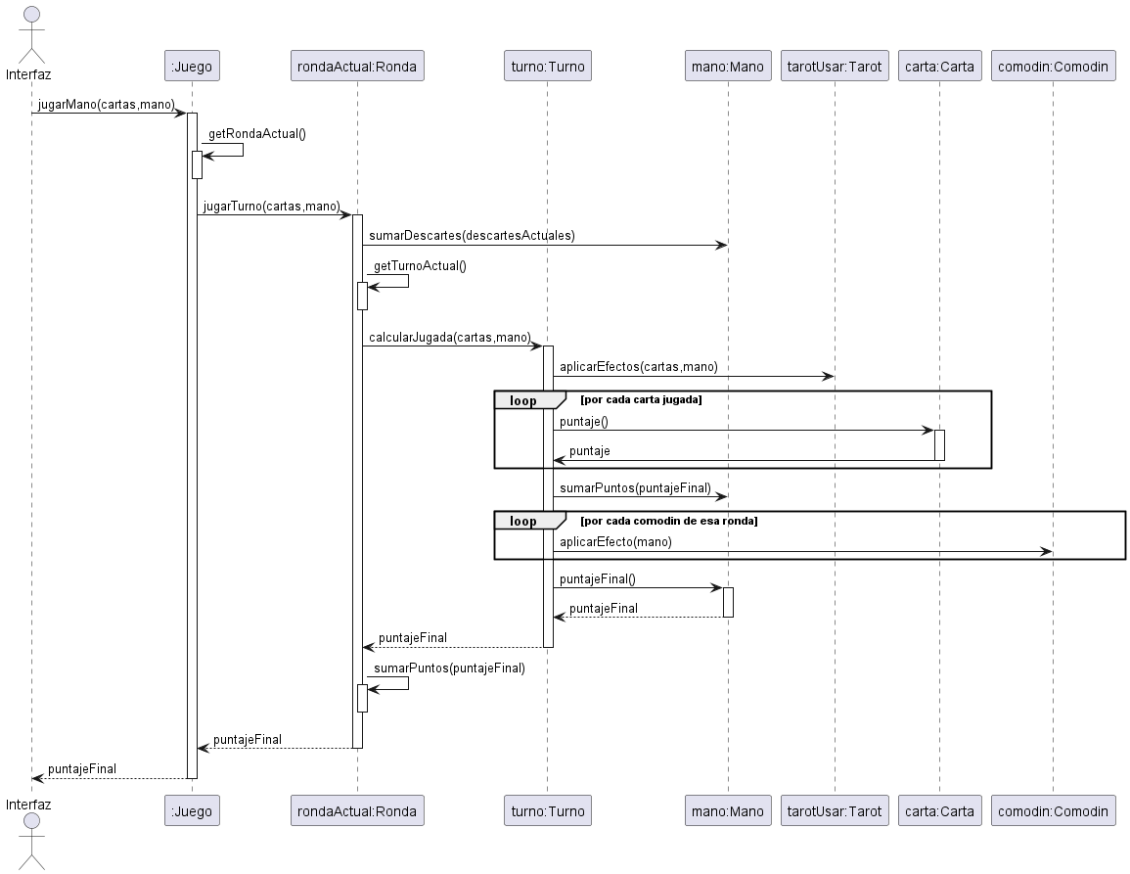

Figura 17: Diagrama de secuencia JUGAR MANO

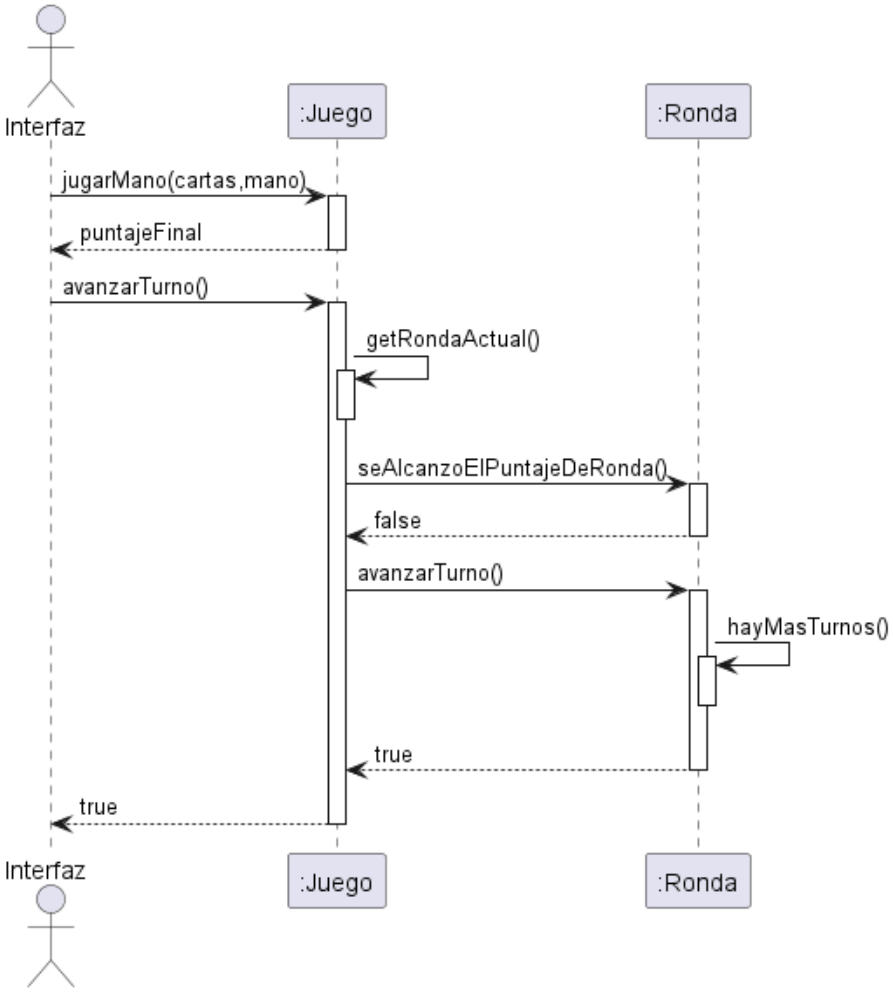

Figura 18: Diagrama de secuencia JUGAR MANO - Avanza Turno

Figura 19: Diagrama de secuencia JUGAR MANO - Avanza Ronda Directo

Figura 20: Diagrama de secuencia JUGAR MANO - Avanza Ronda y Gana Juego

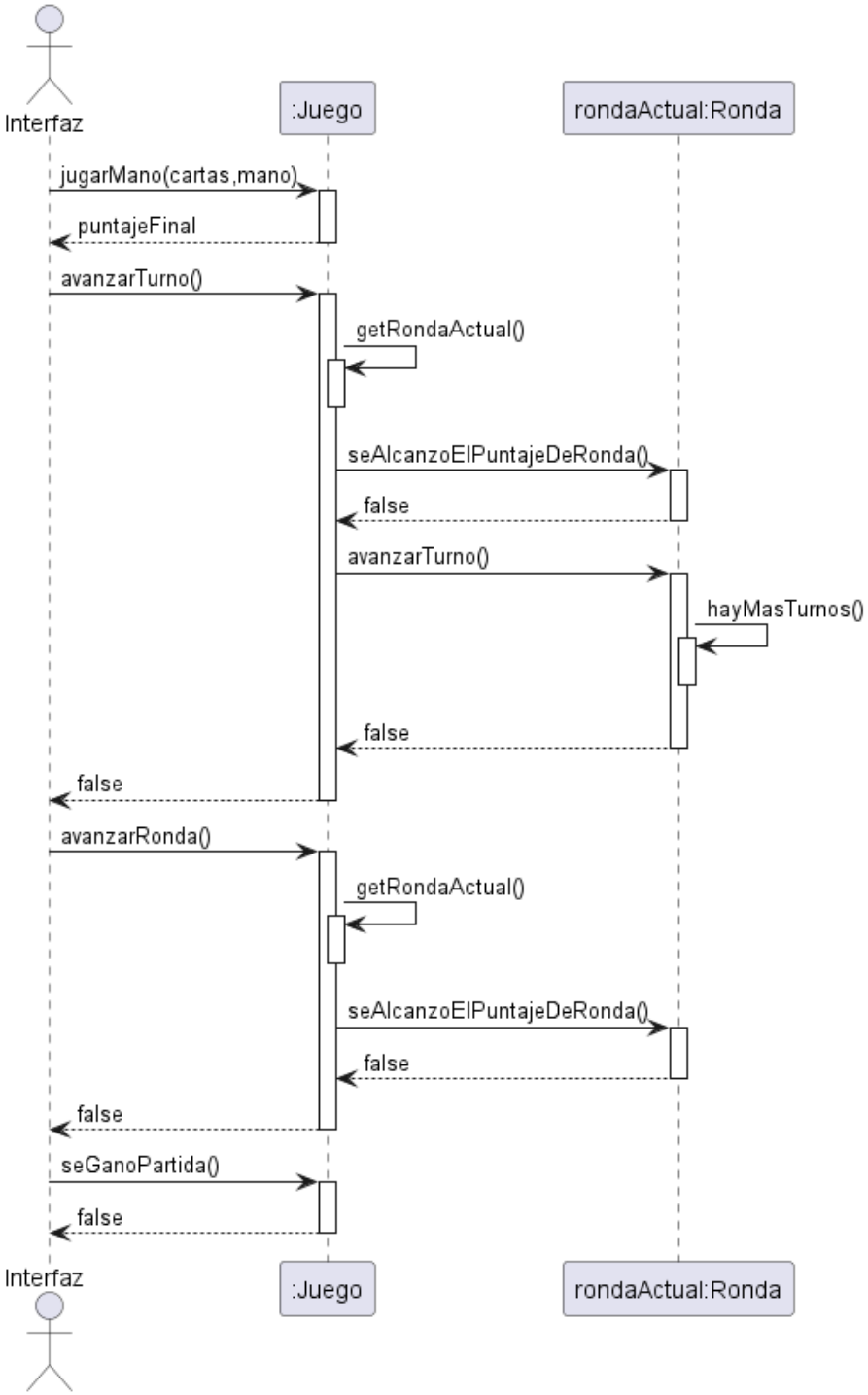

Figura 21: Diagrama de secuencia JUGAR MANO - Pierde Juego

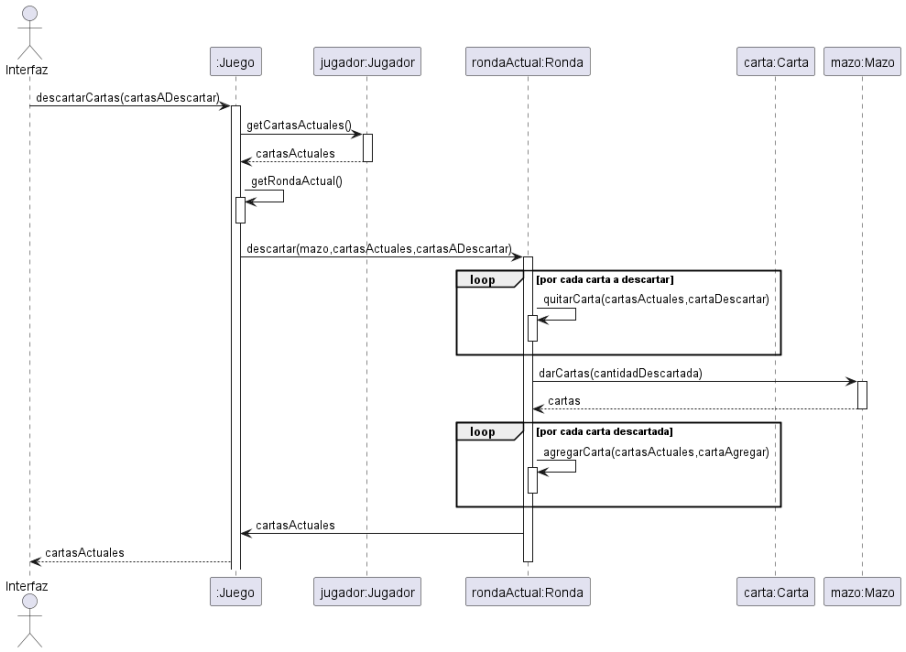

Figura 22: Diagrama de secuencia DESCARTAR

Aclaración: En los diagramas por temas de legibilidad y prolijidad optamos por 'simplificar' algunos métodos, con el fin de que los diagramas no sean excesivamente grandes. Por ello, brindamos al principio de esta sección, los diagramas correspondientes a todas las métodos que participan y no son desarrollados en posteriores diagramas.

5. Diagrama de Paquetes

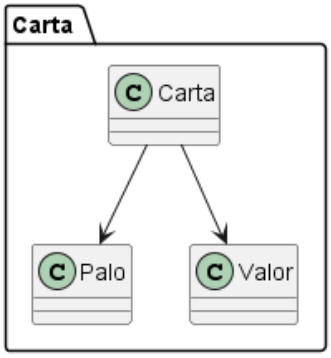

Figura 23: Diagrama de paquetes CARTA

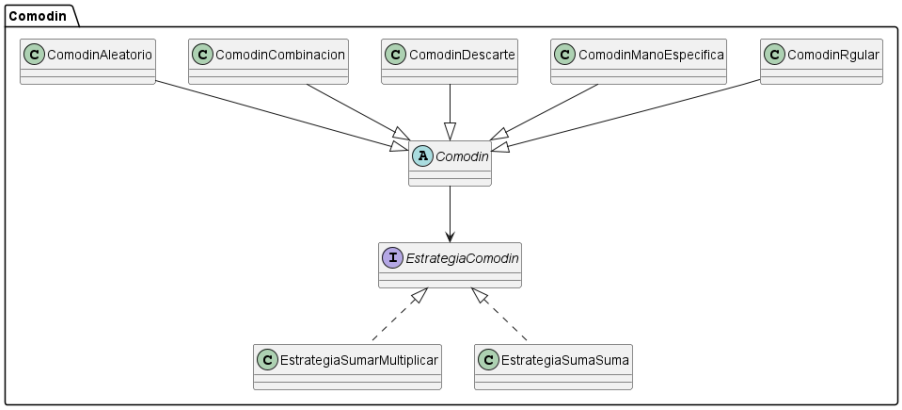

Figura 24: Diagrama de paquetes COMODIN

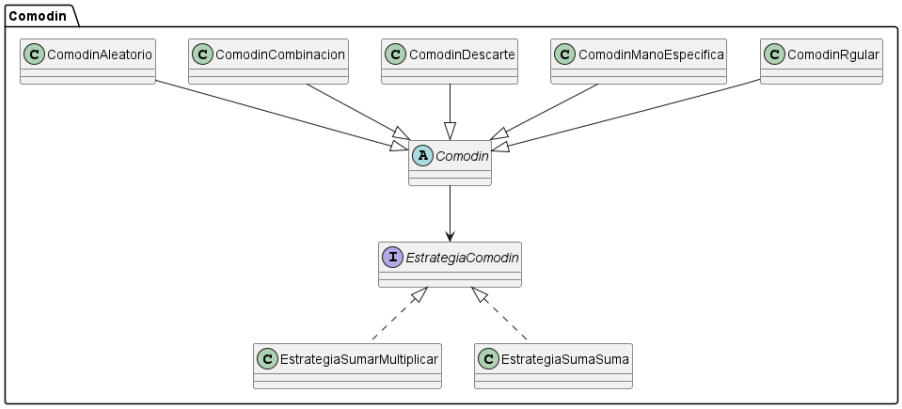

Figura 25: Diagrama de paquetes COMODIN

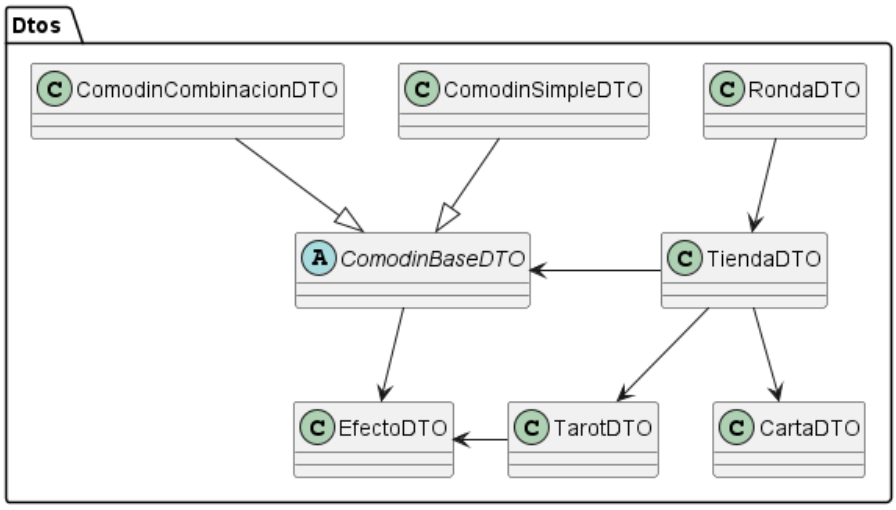

Figura 26: Diagrama de paquetes DTOS

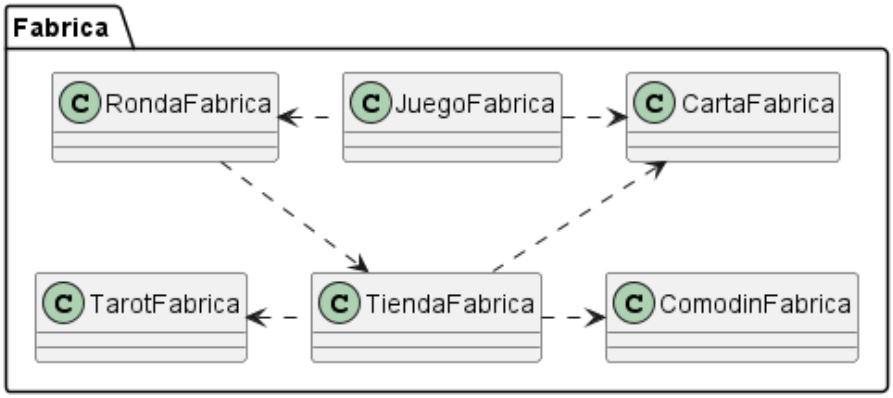

Figura 27: Diagrama de paquetes FABRICA

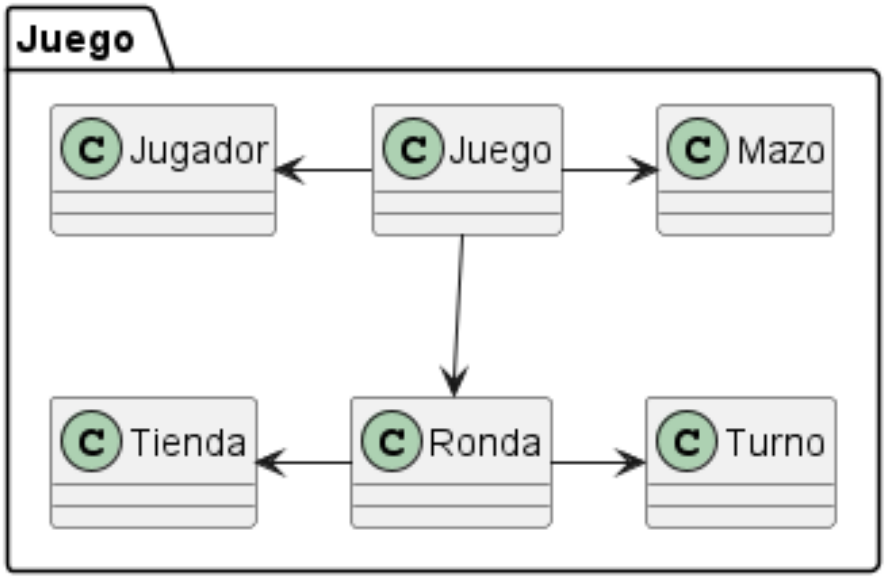

Figura 28: Diagrama de paquetes JUEGO

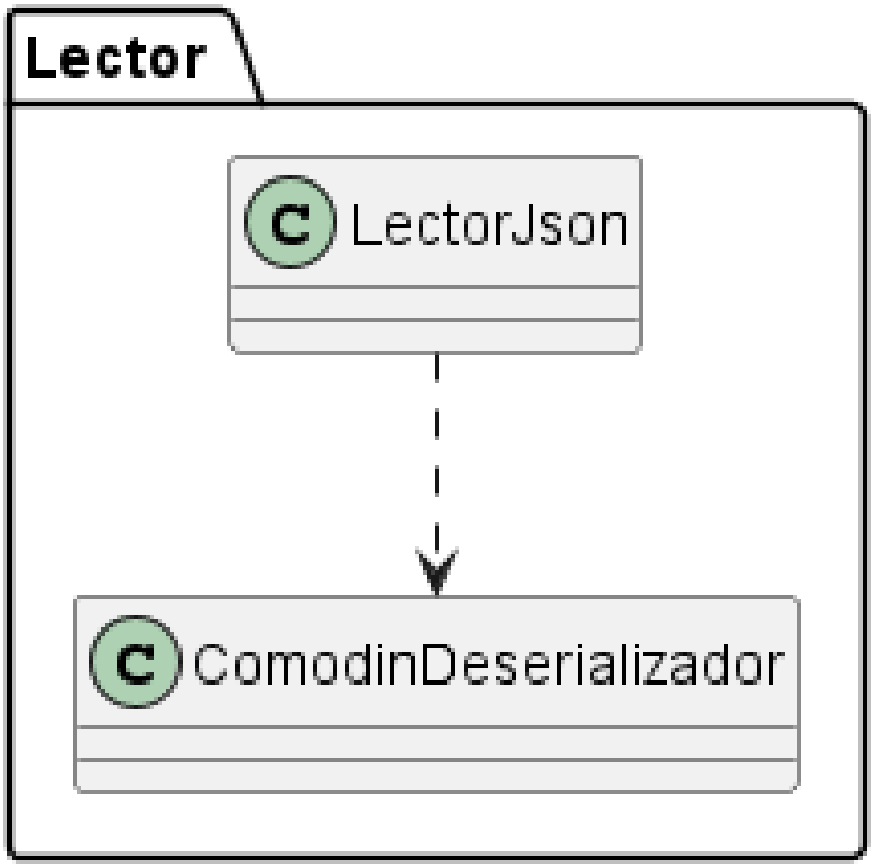

Figura 29: Diagrama de paquetes LECTOR

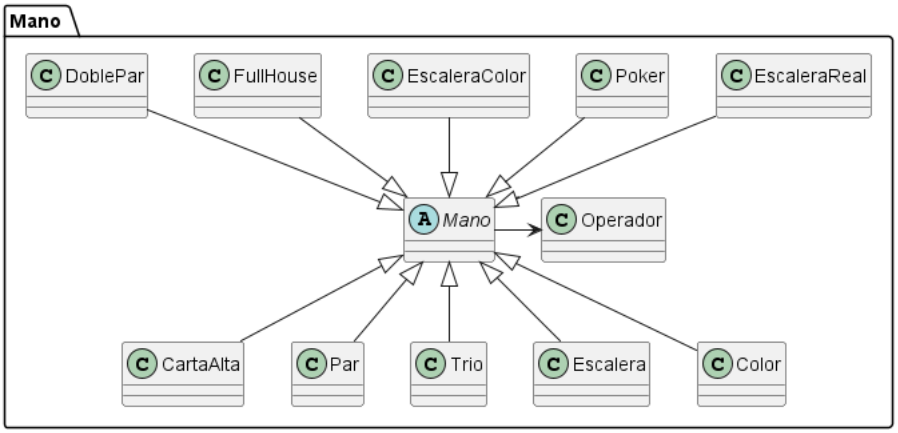

Figura 30: Diagrama de paquetes MANO

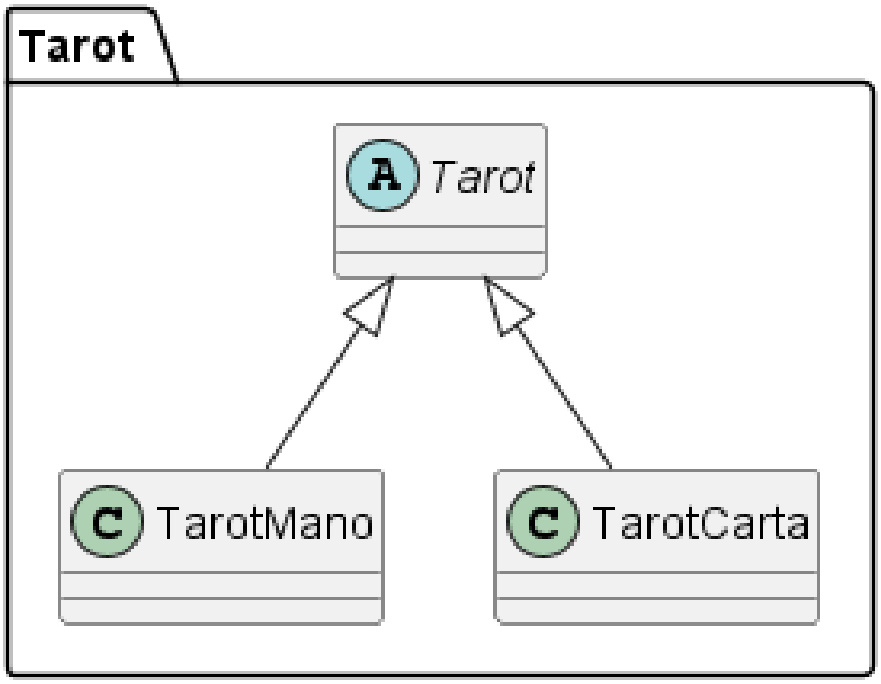

Figura 31: Diagrama de paquetes TAROT

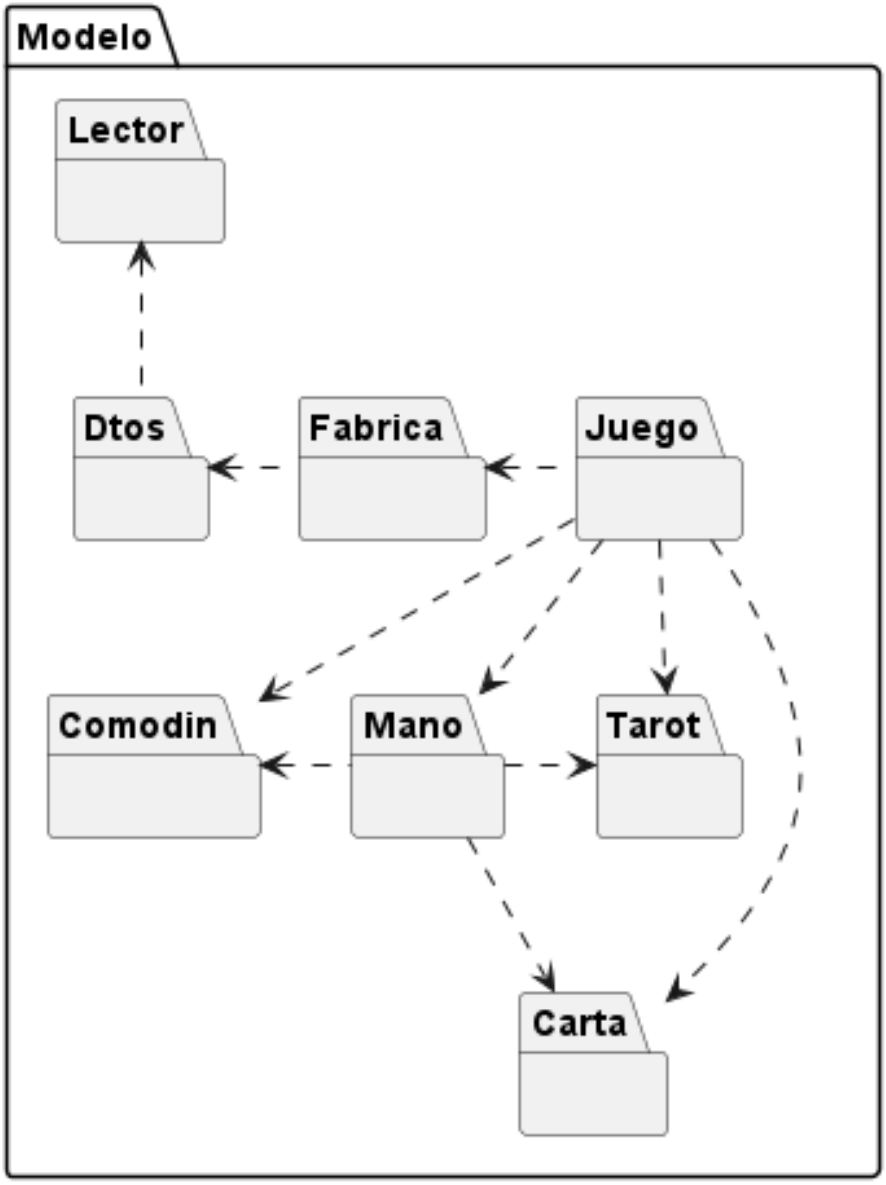

Figura 32: Diagrama de paquetes MODELO COMPLETO

6. Detalles de Implementación

6.1. Principios S.O.L.I.D

6.1.1. Single Responsability

En este modelado, seguimos el principio de 'Responsabilidad Única' al mantener siempre nuestras clases no sobrecargadas con información que no precisaban/utilizaban, de las cuales se podían encargar otras al contenerlas. Esto lo hicimos con el fin d eque si una clase debía ser modificada, solo tuviesen una única razón de cambio.

6.1.2. Liskov's Substitution Principle

En este modelado, seguimos este principio, al mantener complatible la relación entre las clases madres y sus clases hijas. Siendo así que los comportamientos comprendidos por las clases madres, por ejemplo 'Mano' o 'Tarot', también son compatibles con sus clases hijas como 'Color', 'Escalera', 'Poker, etc o 'TarotMano', 'TarotCarta'.

6.2. Herencia y Delegación

En este modelado utilizamos herencia en las clases 'Comodin', 'Mano' y 'Tarot'. Cada una de estas clases madres contenían atributos/comportamientos necesarios y/o pertenientes a sus clases hijas. Haciendo así que mediante un único mensaje (o mensajes) cada una lo reinterpretace o utilice (si ya está implementado por la clase madre) para su necesidad, por ejemplo en 'Tarot' con el método 'aplicarEfectos' o en 'Mano' con 'esJugable' o en 'Comodin' con 'aplicarEfectos' siendo todos estos métodos abstractos.

6.3. Patrones de Diseño

6.3.1. Factory Method

Este patrón creacional lo utilizamos para superar el problema de la creación de las distintas entidades que componen al modelado del juego. Ya que las mismas debían ser creadas a partir de un .json, lo cual nos traía esta necesidad de que su creación sea más dinámica y flexible, para la unión finalmente de las mismas. Para ello también, tuvimos que sumar, la creación de los Dtos que fueron de gran ayuda al contener la información que luego iba a ser cargada/procesada por las distintas fábricas. Por dar ejemplo de alguna ellas contamos con 'CartaFabrica' y su respectivo 'CartaDTO', las cuales se complementan para la creación de las distintas cartas.

6.3.2. Singleton

Este patrón creacional nos solucionó el gran problema de traer y llevar un único juego entre distintas clases. Esto hacía que fuera engorroso depender de la clase juego donde, en algunos casos, no se hacía uso de ella en diversas instancias. Dandonos así una manera más versátil y flexible de trabajar con un mismo juego, y que cuando una entidad lo precise, pueda acceder a él sin tener que depender de una cadena que se lo haga llegar o recibir demasiados argumentos cuando se puede acceder a ellos directamente desde la instacia 'Juego'.

6.3.3. Decorator

Este patrón de diseño estructural se repite varias veces en nuestro modelo, ya que muchas clases, a veces necesitaban funcionalidades que mantenían otras tipos de clases. Esto nos permitía, al encapsular clases con esas funcionalidades específicas dentro de otra, reutilzar dichos comportamientos de manera eficiente y dinámica.

Por ejemplo esto se da con las clases 'Escalera' que se encuentra contenida dentro de 'EscaleraColor' donde al utilizar 'EscaleraColor' la misma idea de 'Escalera' para la formación o no de su mano (sumado a su lógica propia), ésta es reutilizada por medio de esta última. Otro caso destacable de esto es la clase 'Operador', que se encuentra contenida dentro de la clase abstracta 'Mano', ya que dicho operador contiene la distintas funcionalidades que ayuda a las manos a verificar si las cartas que reciben cumplen con sus condiciones. Esta clase que se encuentra contenida en Mano, la heredan todas sus subclases, dando así a cada mano el uso de la misma para sus respectivas lógicas de verificación.

6.3.4. Facade

Este patrón de diseño estructural se hace presente principalmente en nuestra clase 'Juego', ya que él es la clase que contiene todas entidades que componen al mismo en su totalidad. Es decir

que a través de él se da el flujo/funcionamiento del juego, donde internamente todas entidades funcionan mediante los mensajes que él les envía según la orden enviada. Es así como el avance del juego se da por los mensajes internos que el juego envía (por ejemplo a rondas) y estas interactúan entre sí, actualizando sus valores internos, y en algunos casos devolviendolos para determinar alguna acción.

6.3.5. Strategy

Este patrón de diseño de comportamiento lo utilizamos para modelar los distintos comportamientos en los comodines. Ya que en algunos casos modificaban distintos atributos y había que buscar la forma de diversificar eso para todos los comodines existentes. Por ejemplo tenemos la interfaz 'EstrategiaComodin', que luego la implementaban las clases 'EstrategiaSumaSuma' y 'EstrategiaSumarMultiplicar' donde hacen su propia lógica de modificación para los atributos que reciben.

6.3.6. Chain of Responsability

Este patrón de comportamiento se da como herramienta principal a la hora del flujo del juego como nombramos anteriormente. Esto se da por ejemplo con las clases Juego ->Ronda ->Turno ->Mano , donde 'Juego' le delega a 'Ronda' el jugar una mano, donde luego la 'Ronda' le delega al 'Turno' el calcular la jugada y el turno que ya contiene los comodines y tarots a aplicar a las cartas y/o mano (delega dicha modificación a la entiedades 'Tarot' y 'Comodin'), para que finalmente la mano misma calcule el puntaje final de la mano jugada con sus modificaciones pertinentes. Así se pueden encontrar más casos, siendo éste el más significativo.