

## Gwent - T.P. 2

[TB025] Paradigmas de Programación  
Primer cuatrimestre de 2025

**Grupo 07 - Estudiantes:**

Nombre	Padrón	Mail
Matias Domine Folco	112268	mdomine@fi.uba.ar
Enrique Heller	111605	eheller@fi.uba.ar
Andy Mayuri	111492	amayuri@fi.uba.ar
Franco Montanelli	111521	fmontanelli@fi.uba.ar
Andrés Moyano	110017	ammoyano@fi.uba.ar

**Tutor: Joaquín Pandolfi**

## Índice

<b>1. Supuestos.....</b>	<b>3</b>
1.1. Supuesto.....	3
1.2. Supuesto.....	3
1.3. Supuesto.....	3
<b>2. Diagramas de clase.....</b>	<b>3</b>
2.1. Diagramas del Modelo Tentativo:.....	3
<b>3. Diagramas de secuencia.....</b>	<b>6</b>
3.1. Diagrama de Secuencia de Modelo Tentativo(Entrega-0):.....	6
3.2. Entrega-1:.....	7
3.2.1. Verificar que un jugador posea cartas suficientes para empezar el juego en su mazo.....	7
3.2.2. Verificar que a un jugador se le reparten 10 cartas de su mazo.....	8
3.2.3. Verificar que un jugador pueda colocar una carta en una sección del tablero.....	9
3.2.4. Verificar que un jugador juegue una carta de su mazo y tenga un puntaje parcial.....	10
3.2.5. Verificar que las cartas pasen a la pila de descarte.....	10
3.2.6. Verificar que al modificar una carta con una carta unida se cambien sus puntos y se aplique el valor solo a la ronda.....	10
3.2.7. Verificar que al utilizar una carta especial de clima se reduzca el valor de las cartas de la sección correspondiente.....	10
<b>4. Diagramas de paquete.....</b>	<b>10</b>
<b>5. Detalles de implementación.....</b>	<b>10</b>
<b>6. Excepciones.....</b>	<b>10</b>

## 1. Supuestos

- 1.1. Supuesto
- 1.2. Supuesto
- 1.3. Supuesto

## 2. Diagramas de clase

### 2.1. Diagramas del Modelo Tentativo:

Como primera presentación de un modelo tentativo nos enfocamos en tratar de modelar clases un poco simples pero casi completas, teniendo en cuenta una visión de cómo podría afectar estas bases al futuro trabajo a completar.

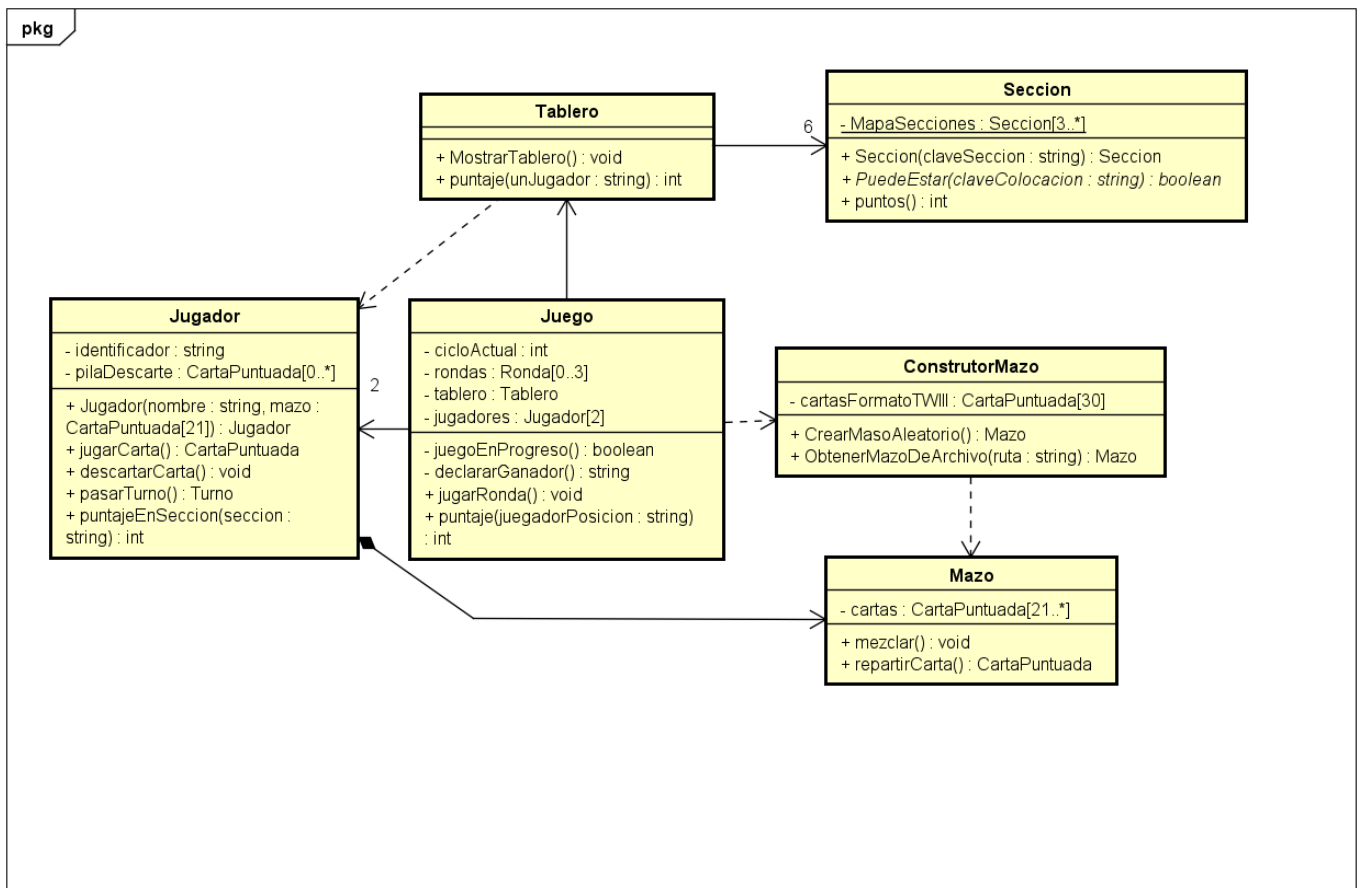


Figura 1: Diagrama de clases tentativo del modelo

Siguiendo un poco la consigna de lo que se pide, que es el modelado de un juego de cartas basado en “Gwent”, planteamos las siguientes clases que nos servirán como pilares para poder establecer todas las reglas del juego, por ello se contempla como clase principal el “Juego” que será el intermediario para poder averiguar puntajes, `declararGanador()`, y verifica si aun el juego sigue en progreso, también se encarga de ir designando a qué jugador

le tocaría jugar por Ronda y que a su vez como algo en general se encargará de delegar estas responsabilidades a otras clases.

Como parte del juego tenemos un “Tablero” encargado de tener 6 secciones de tipo “Seccion” que serán usadas para poder recibir las cartas que coloque el “Jugador”, también nos servirá para poder conocer el puntaje de las cartas de los jugadores que se encuentran en el tablero (demostrado más adelante en el diagrama de secuencia). Luego tenemos a la clase “Jugador” el cual toma las decisiones respecto al qué hacer en su Turno, teniendo 2 opciones en cuenta: “jugarCarta() y pasarTurno()”, el jugador también es el que está a cargo de manejar su mazo, poder elegirlo y también recibir las cartas con las que jugará, seleccionadas a partir de su “Mazo”.

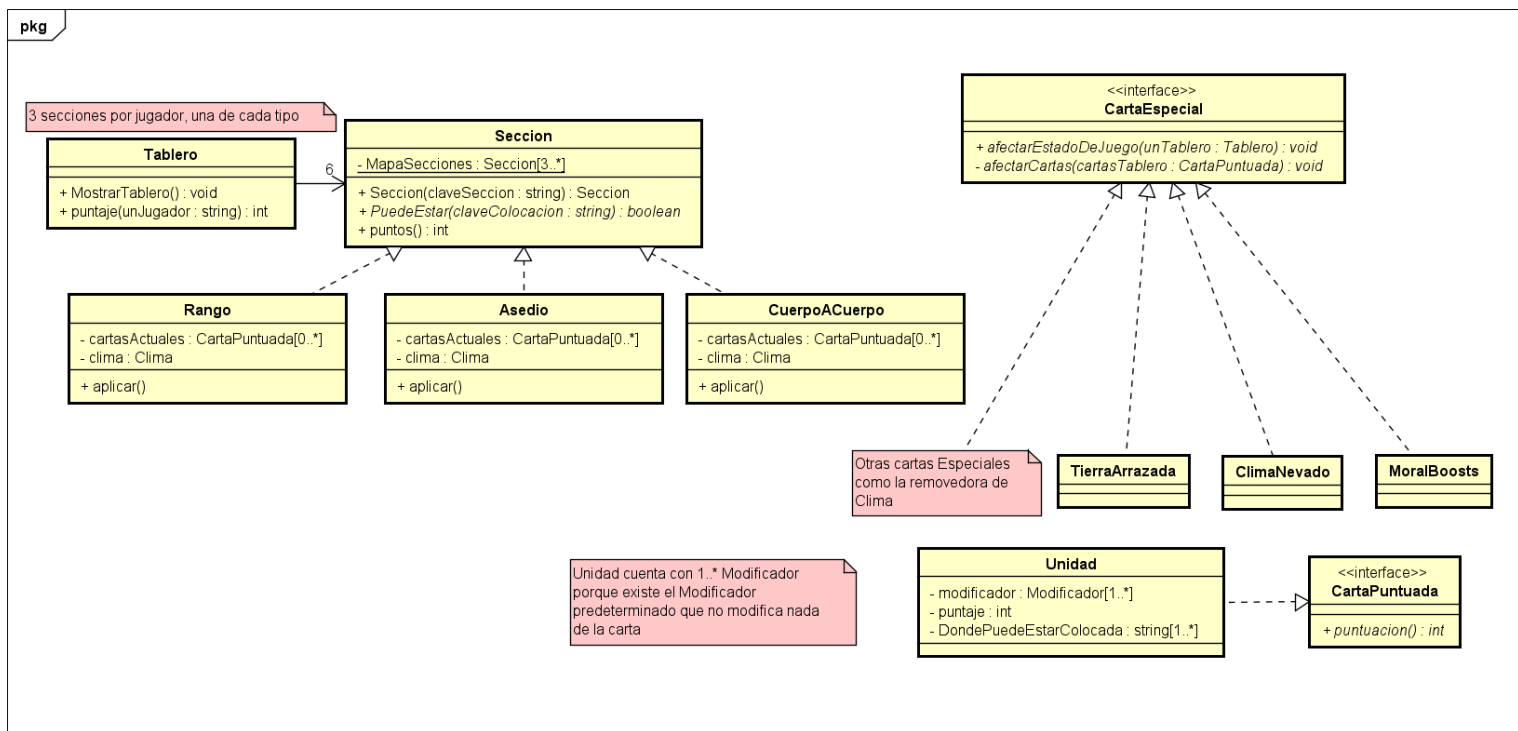


Figura 2: Diagrama de clases tentativo del modelo

Para poder mostrar un poco más del modelado de las clases que usaremos para el juego, se nos menciona en las especificaciones, que hay diferentes tipos de cartas, como se muestra en el diagrama tendremos las cartas “Unidad” que tendrán una puntuación (las clasificaremos como una “CartaPuntuada”), tendrán o no Modificadores y que además se puede colocar en una de las 3 secciones que tiene el jugador en el “Tablero”, estas secciones serán “Rango”, “Asedio” y “CuerpoACuerpo”, clases que fueron pensadas también para los efectos que tienen algunas de las cartas especiales sobre algunas secciones del tablero, como por ejemplo la “CartaEspecial” - “Clima”. Como las cartas especiales no cuentan con puntos y solo afectan a las unidades no serán una “CartaPuntuada”, será más bien una interfaz que dependiendo el tipo de “CartaEspecial” tendrá métodos que implementan esas cartas de acuerdo a sus características, generando diferentes efectos.

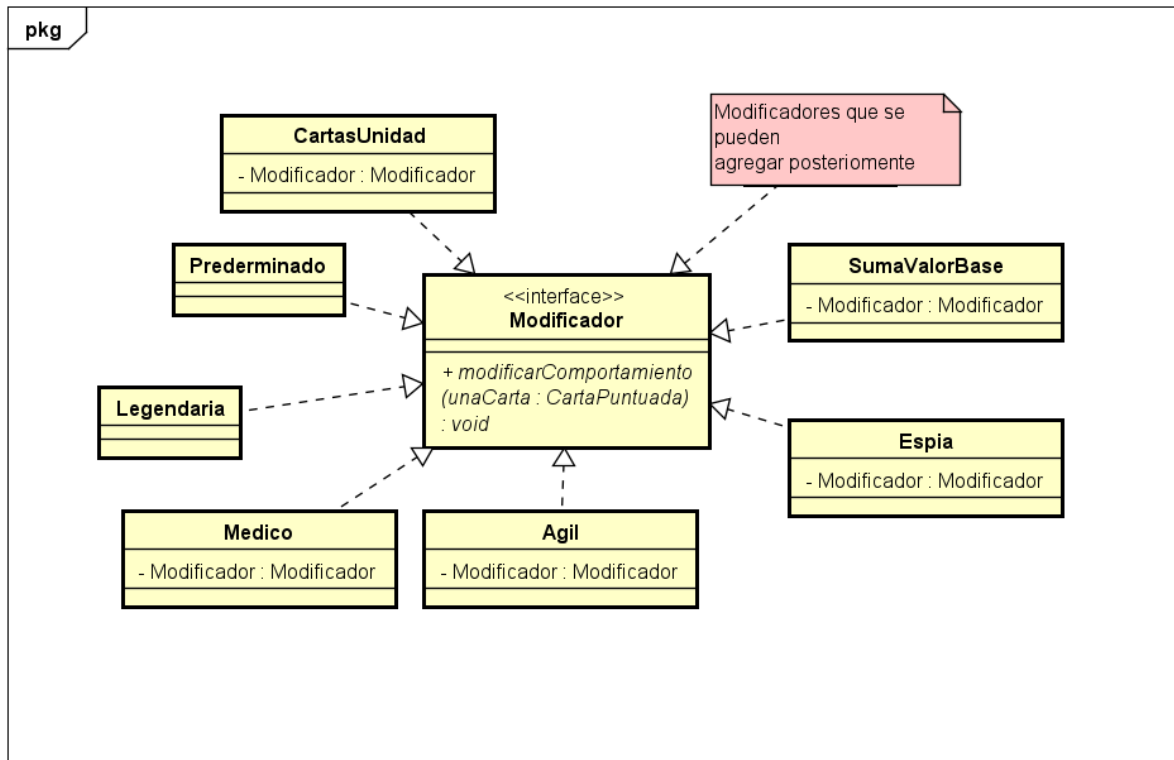


Figura 3: Diagrama de clases tentativo del modelo

El “Modificador” se pensó como una interfaz que dependiendo quien la implemente tendrá efectos diferentes, a su vez las clases que implementen el “Modificador”, tendrán o no un “Modificador” propio que depende del diseño y tipo de la carta. El Modificador “Predeterminado” será tomado como el base, este tiene un comportamiento que simulara el no tener modificador el cual será usado en las cartas “Unidad” en primera instancia ya que no se sabe si tendrá o no un Modificador. Puede haber un comportamiento de “lista enlazada” en el sentido de cómo se encadenan los  $n$  modificadores que pueden llegar a afectar el puntaje final.

### 3. Diagramas de secuencia

#### 3.1. Diagrama de Secuencia de Modelo Tentativo(Entrega-0):

Como primera entrega de modelo tentativo se nos propuso diseñar un diagrama de secuencia mostrando el caso donde se calcula el puntaje de un jugador basado en las cartas que se encuentran en el tablero, para nuestro diagrama se consideró la siguiente secuencia:

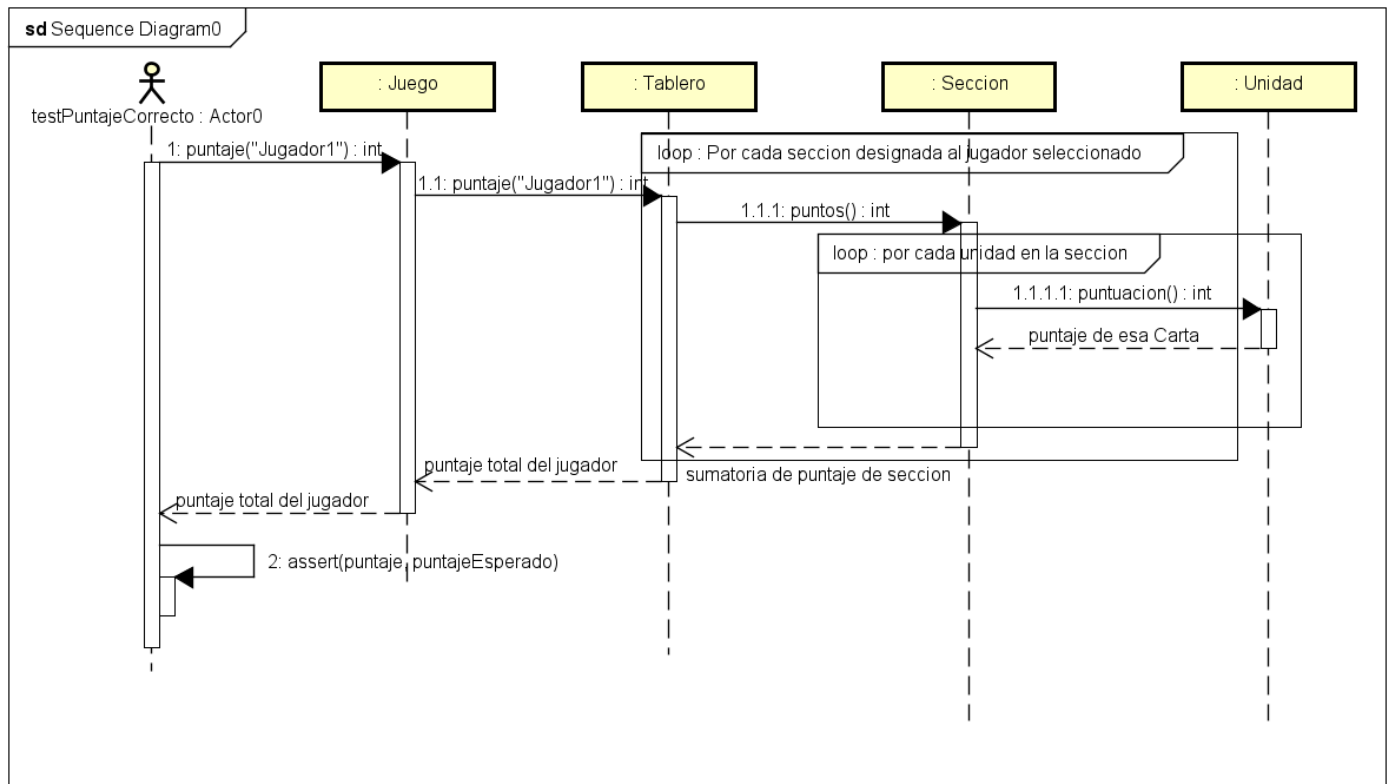


Figura 4: Diagrama de secuencia, cálculo del puntaje de un jugador basado en las cartas que se encuentran en el tablero.

## 3.2. Entrega-1:

## 3.2.1. Verificar que un jugador posea cartas suficientes para empezar el juego en su mazo.

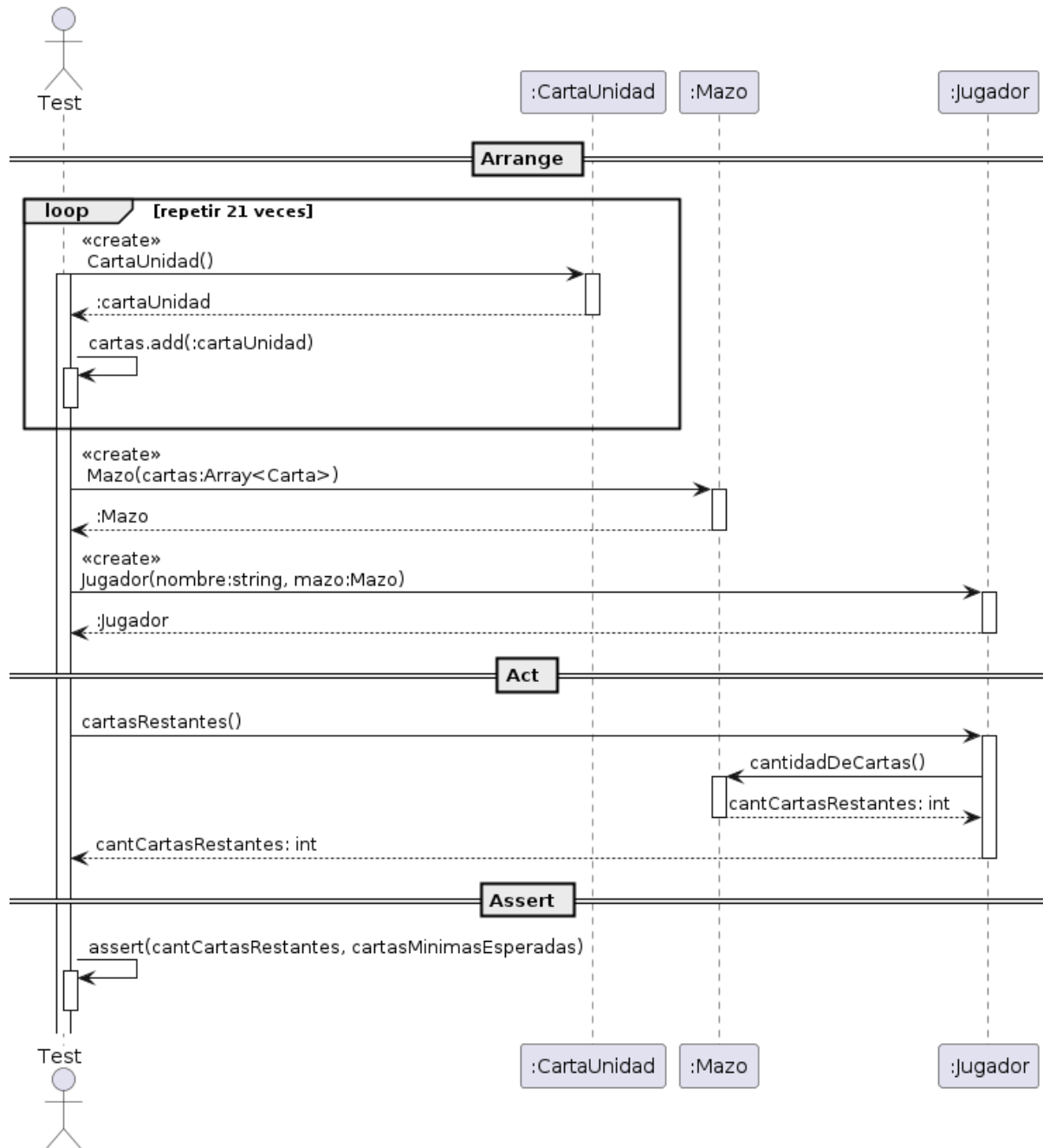


Figura 5: Test-01 Jugador Con Cartas Suficientes Al Empezar

## 3.2.2. Verificar que a un jugador se le reparten 10 cartas de su mazo.

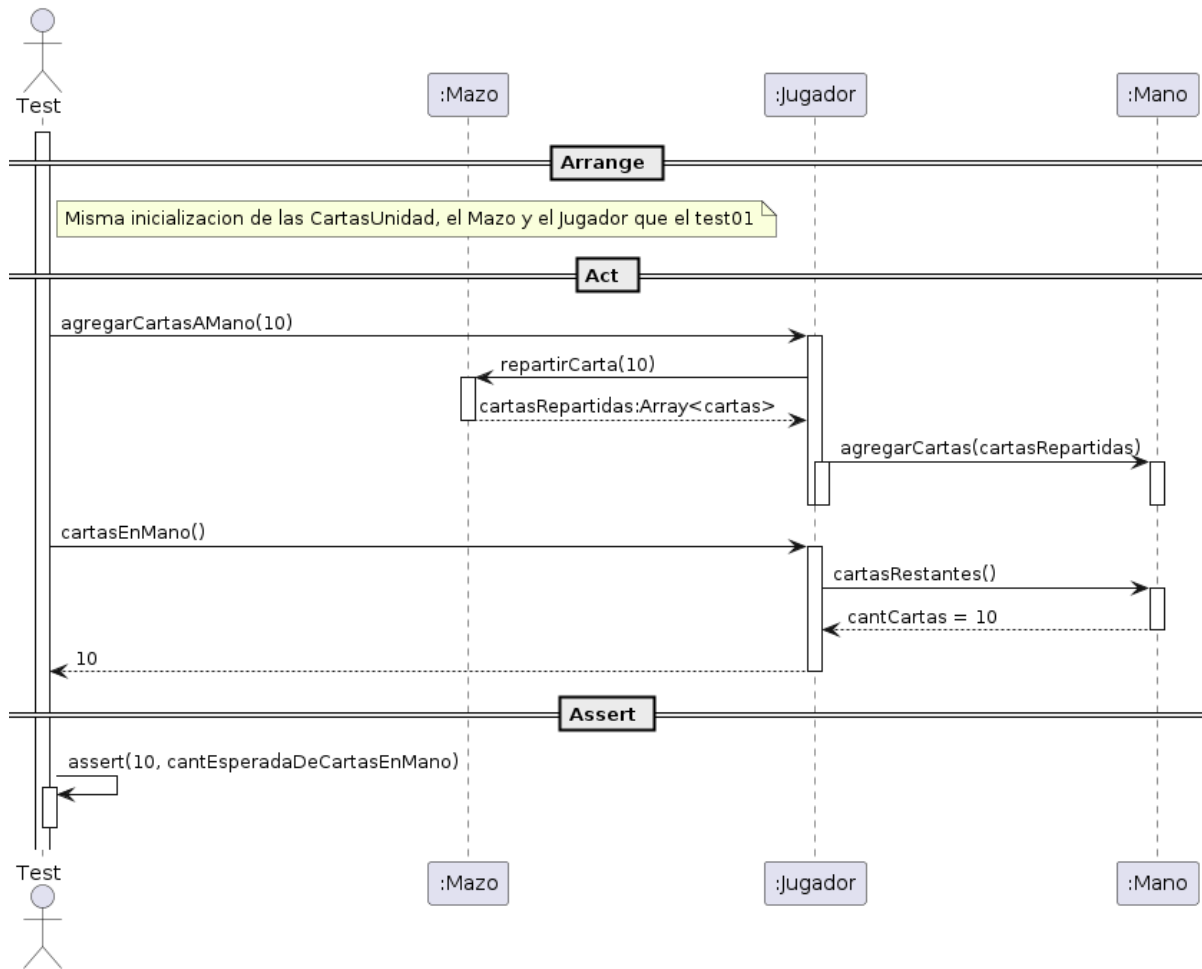


Figura 6: Test-02 Jugador Se Le Reparten 10 Cartas De Su Mazo



## 3.2.3. Verificar que un jugador pueda colocar una carta en una sección del tablero

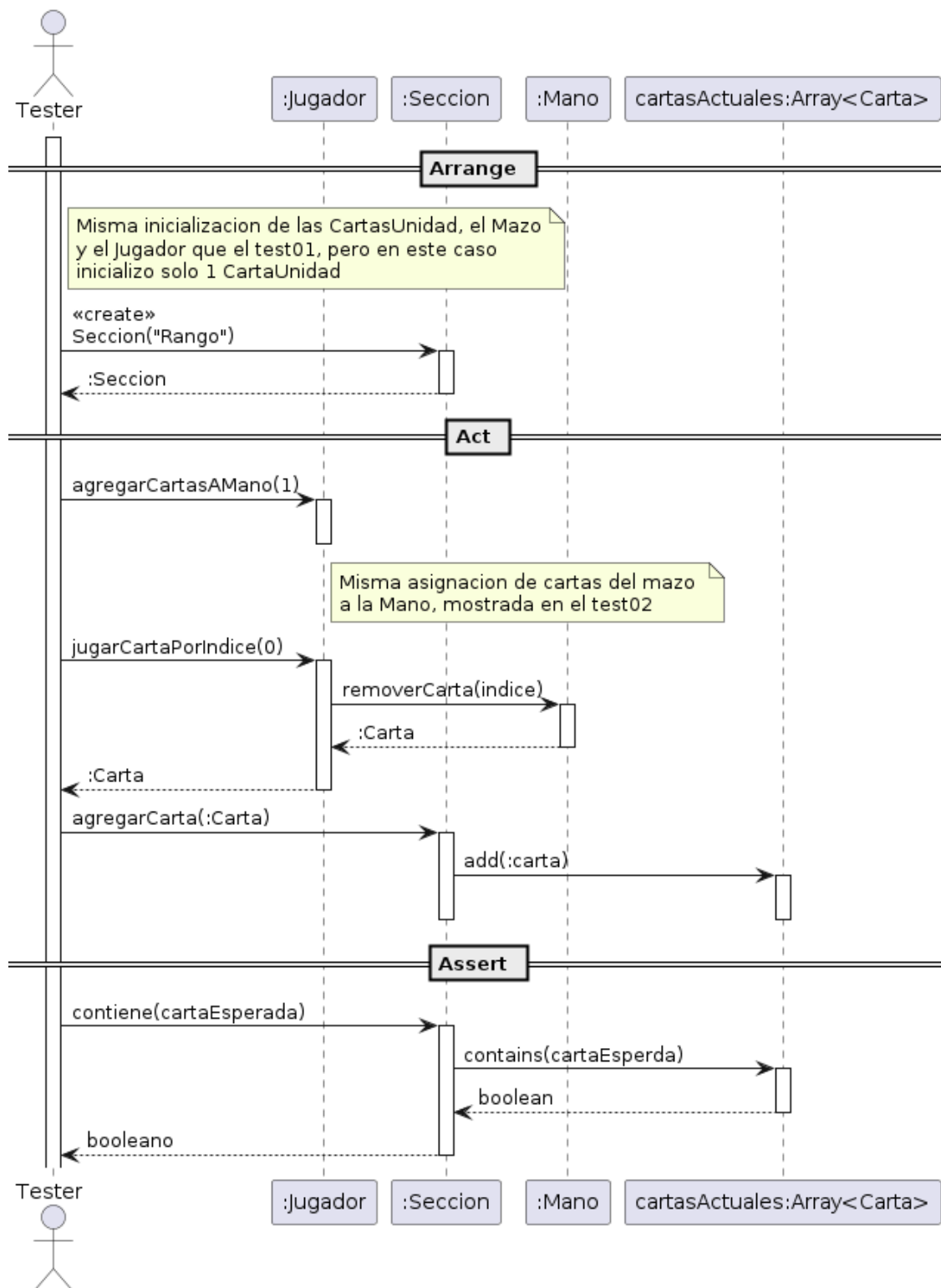


Figura 7: Test-03 Jugador Puede Colocar Carta En Sección Del Tablero

### 3.2.4. Verificar que un jugador juegue una carta de su mazo y tenga un puntaje parcial.

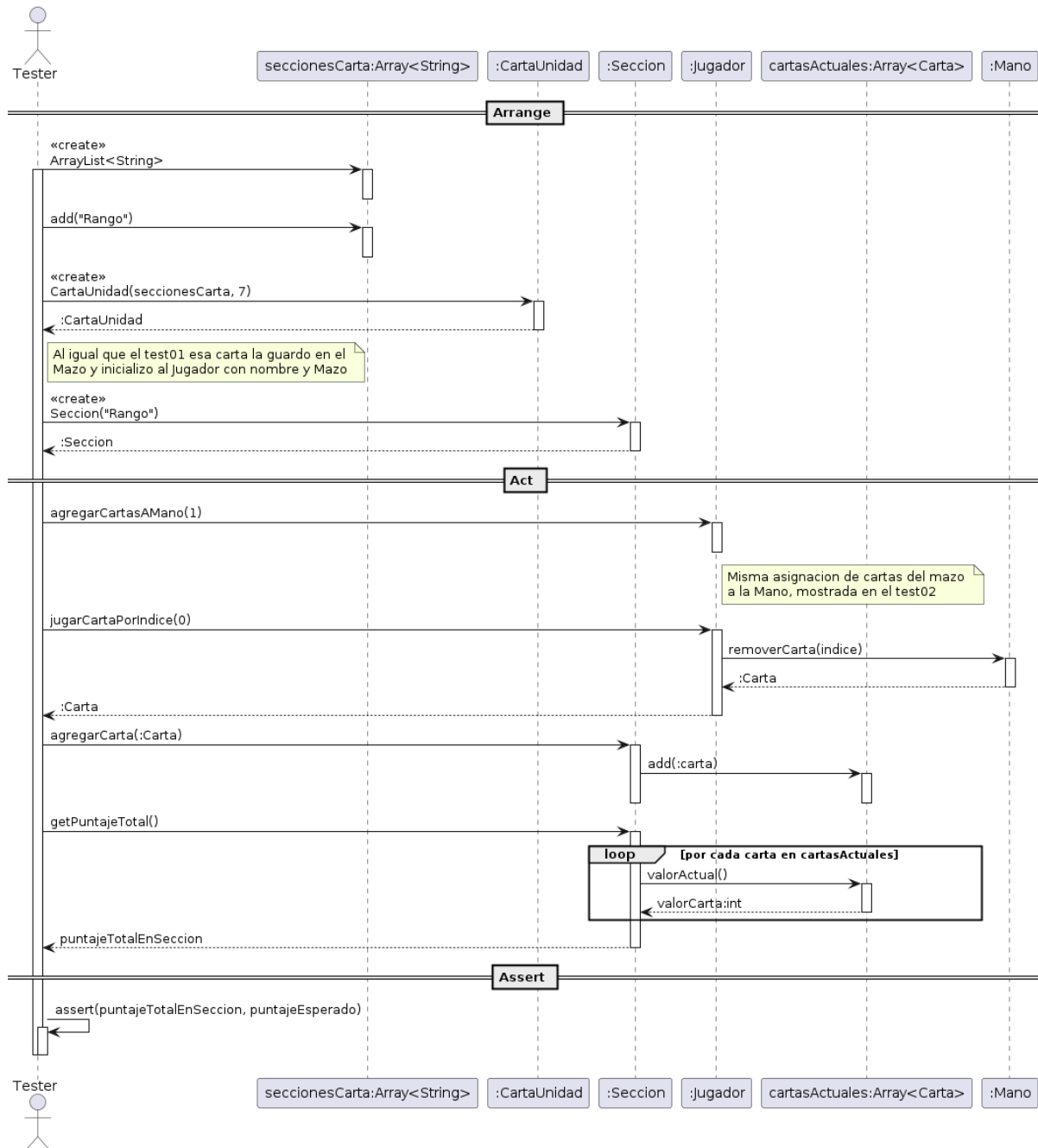


Figura 8: Test-04 Jugador Juega Carta Y Tiene Puntaje Parcial

- 3.2.5. Verificar que las cartas pasen a la pila de descarte
- 3.2.6. Verificar que al modificar una carta con una carta unida se cambien sus puntos y se aplique el valor solo a la ronda.
- 3.2.7. Verificar que al utilizar una carta especial de clima se reduzca el valor de las cartas de la sección correspondiente.

3.3. Entrega-2:

3.4. Entrega-3:

3.5. Entrega-4:

3.6. Entrega-5:

## **4. Diagramas de paquete**

## **5. Detalles de implementación**

## **6. Excepciones**