

# Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III

Curso 2

Primer cuatrimestre de 2020

Grupo 11	
Alumnos:	Coronel, Juan Ignacio - 104252 - jcoronel@fi.uba.ar
	Dituro, Celeste - 104011 - cdituro@fi.uba.ar
	Florit, Francisco - 104289 - fflorit@fi.uba.ar
	Lopez, Victoria Abril - 103927 - vlopez@fi.uba.ar
	Vargas, Alejandro - 104121 - avargas@fi.uba.ar
Corrector:	Pablo Massuh

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Supuestos</b>	<b>2</b>
<b>3. Diagrama de paquetes</b>	<b>3</b>
<b>4. Diagramas de clase</b>	<b>4</b>
4.1. Pregunta . . . . .	4
4.2. Vista . . . . .	8
4.3. Flujo Juego . . . . .	8
<b>5. Detalles de implementacion</b>	<b>8</b>
5.1. Patrones utilizados . . . . .	9
5.2. Clases en las que se utiliza herencia . . . . .	9
<b>6. Excepciones</b>	<b>9</b>
<b>7. Diagramas de secuencia</b>	<b>10</b>
<b>8. Diagramas de estado</b>	<b>12</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 una aplicación de Preguntas y Respuestas en Java utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

## 2. Supuestos

**Supuesto 1.** El valor de las opciones:correcta e incorrecta con y sin penalidad no cambia durante el transcurso del juego:

- Valor de opción correcta = 1;
- Valor de opción incorrecta sin penalidad = 0;
- Valor de opción incorrecta con penalidad = -1.

**Supuesto 2.** Las preguntas pueden tener hasta una máxima cantidad de opciones:

- MultipleChoice: 5 opciones;
- VerdaderoFalso: 2 opciones;
- OrderedChoice: 5 opciones;
- GroupChoice: 6 opciones.

**Supuesto 3.** Cada jugador puede hacer uso de dos multiplicadores (X2 y X3) a lo largo del juego.

**Supuesto 4.** Dos opciones son iguales cuando tienen un mismo título.

**Supuesto 5.** En MultipleChoice con penalidad se calcula el puntaje sumando las opciones correctas cuyo valor es 1 y las incorrectas con valor -1.

**Supuesto 6.** A una Respuesta siempre se le agregan las opciones de la pregunta asignada.

**Supuesto 7.** La Exclusividad no se asigna si hay dos respuestas correctas.

**Supuesto 8.** En GroupChoice la cantidad mínima de opciones por grupo es 1 y la máxima es 5.

**Supuesto 9.** En GroupChoice no podrá haber una misma opción correcta para ambos grupos.

**Supuesto 10.** En el método aplicarARespuestas() correspondiente a la clase Comodín, las respuestas recibidas por parámetro corresponden a distintos jugadores.

**Supuesto 11.** En OrderedChoice() el orden en que se agregan las opciones es el correcto.

### 3. Diagrama de paquetes

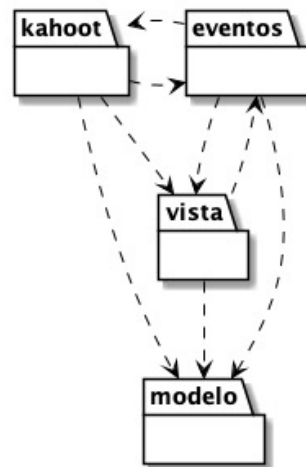


Figura 1: Paquetes

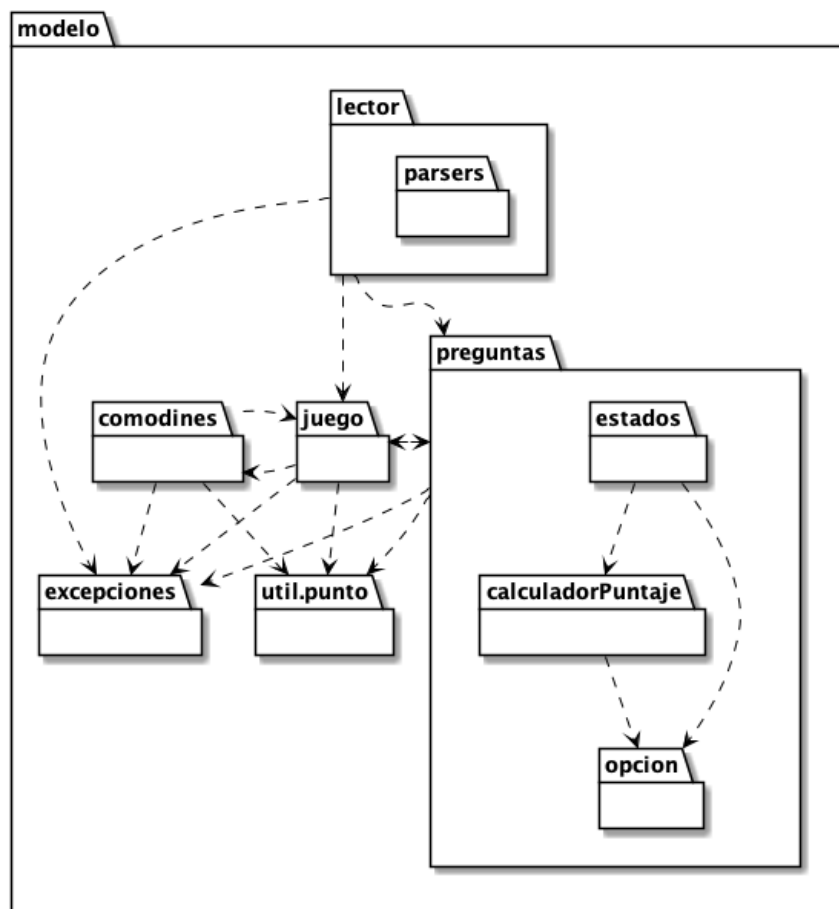


Figura 2: Modelo

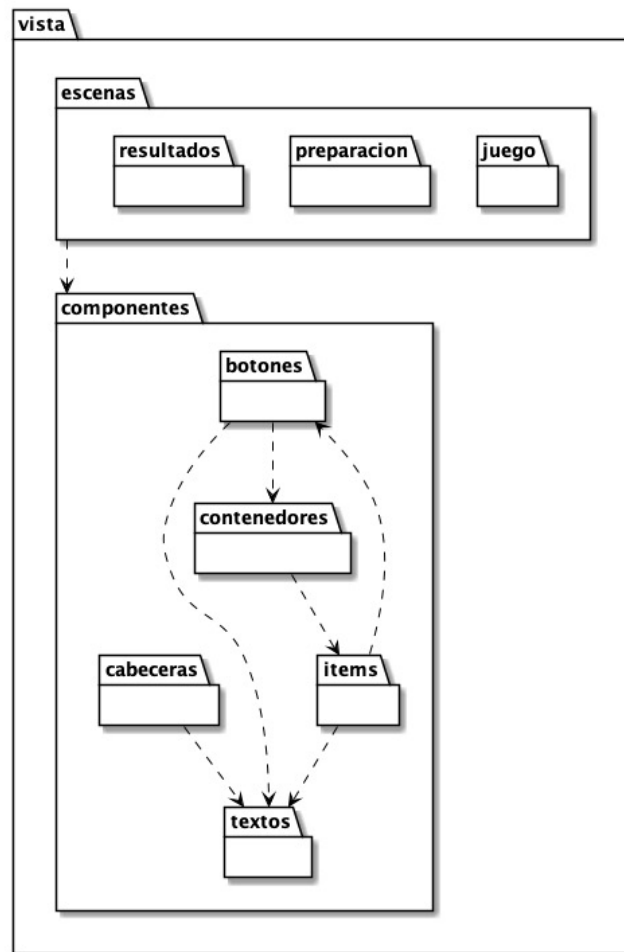


Figura 3: Vista

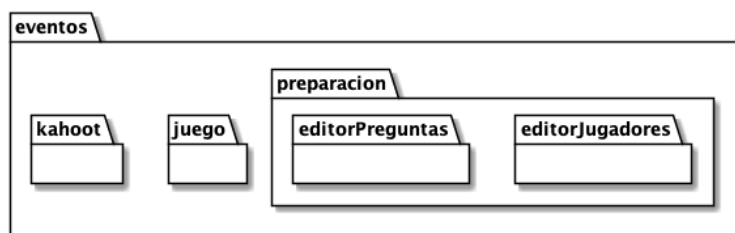


Figura 4: Eventos

## 4. Diagramas de clase

### 4.1. Pregunta

Diagramas del modelo relacionadas con la pregunta, en las Figuras 5, 6 y 7 se detalla como estan implementada la estructura de cada pregunta y en las 8, 9 y 10 las clases concretas de cada pregunta y como agregan las opciones.

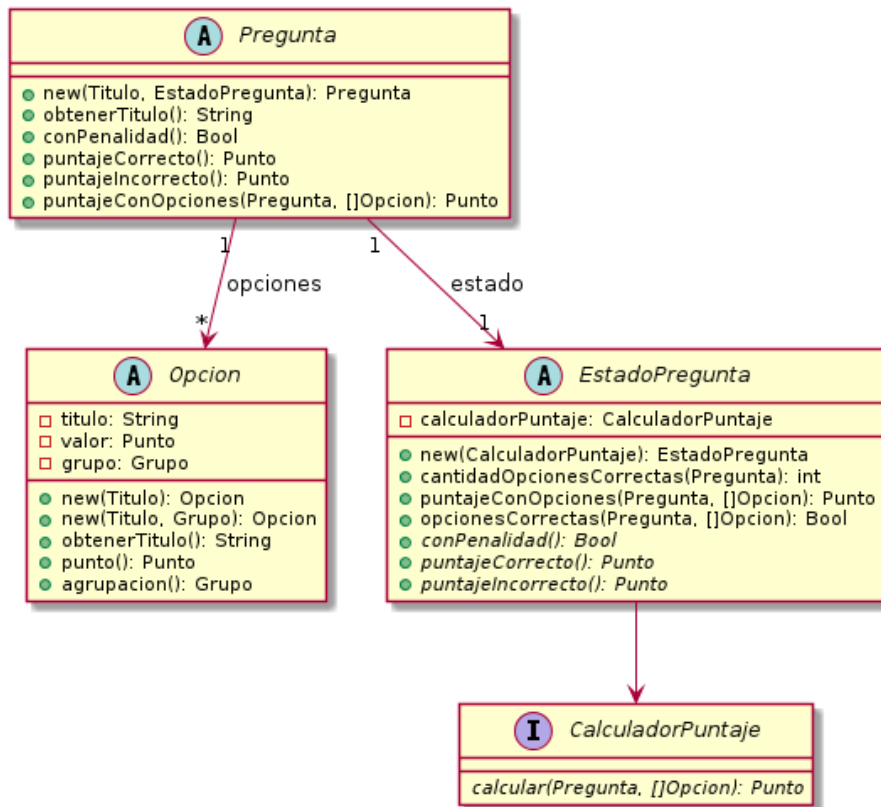


Figura 5: Estructura de la pregunta

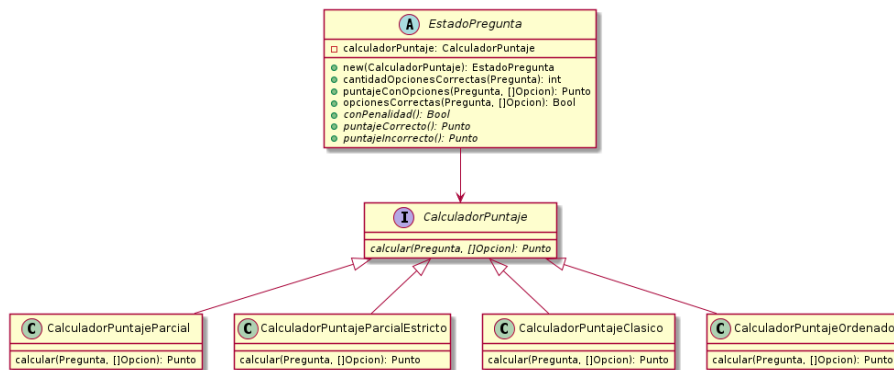


Figura 6: Tipos de calculadores de puntaje

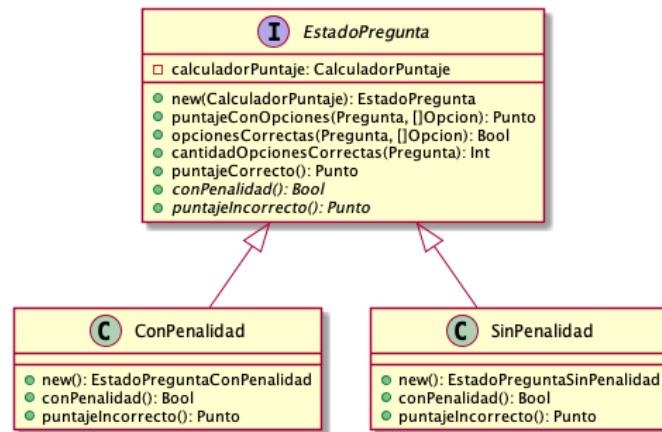


Figura 7: Estados de Pregunta

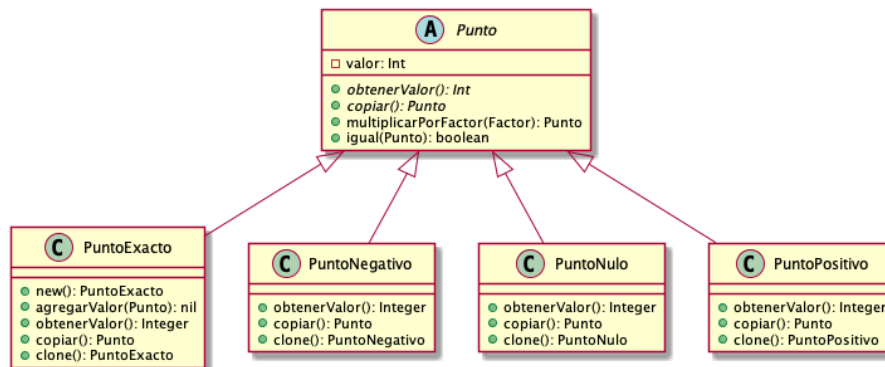


Figura 8: Puntos

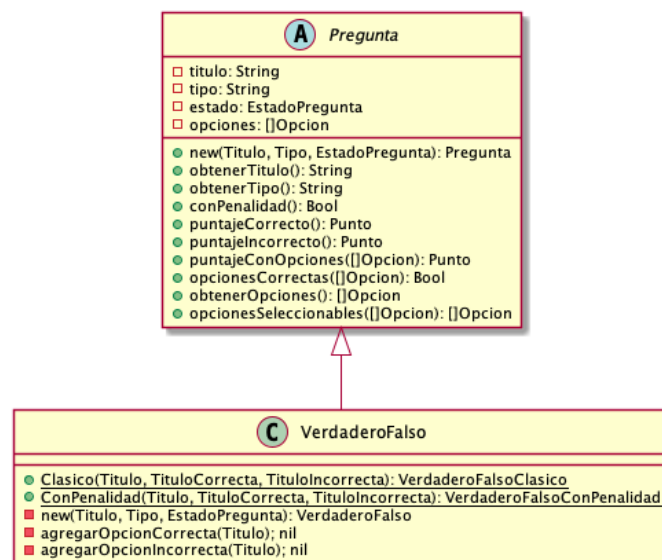


Figura 9: VerdaderoFalso

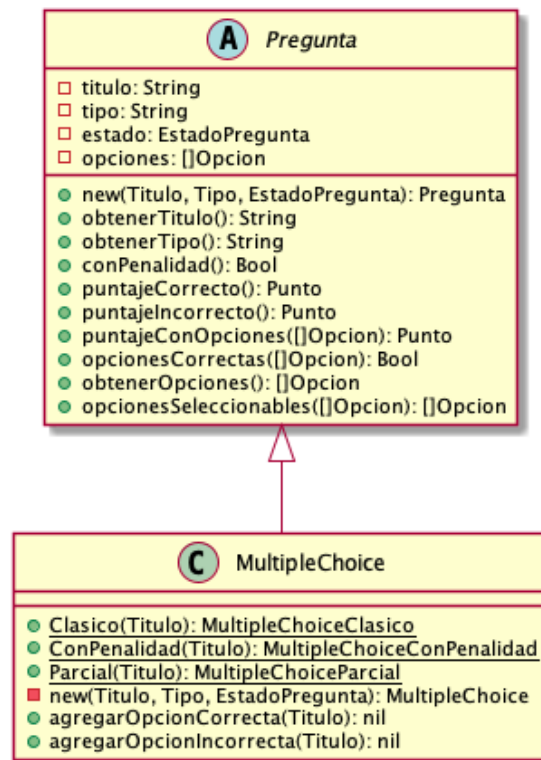


Figura 10: MultipleChoice

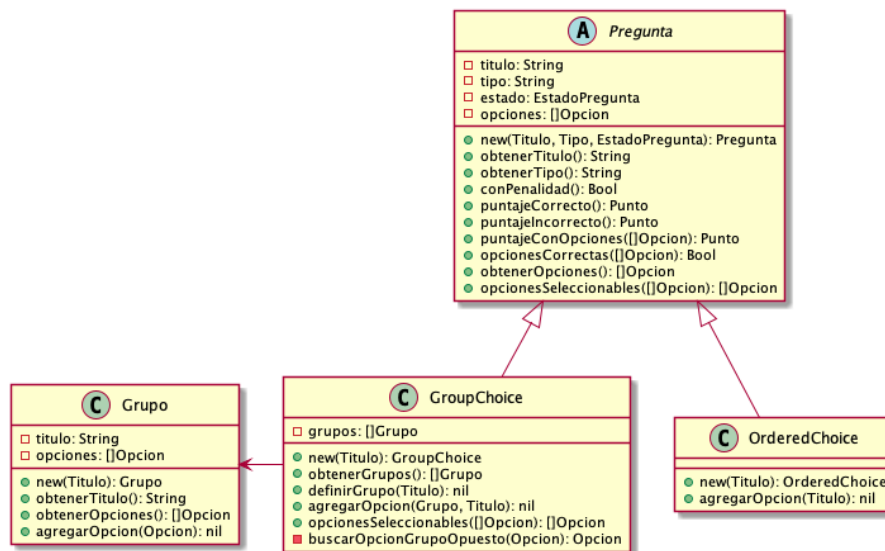


Figura 11: OrderedChoice y GroupChoice



## 4.2. Vista

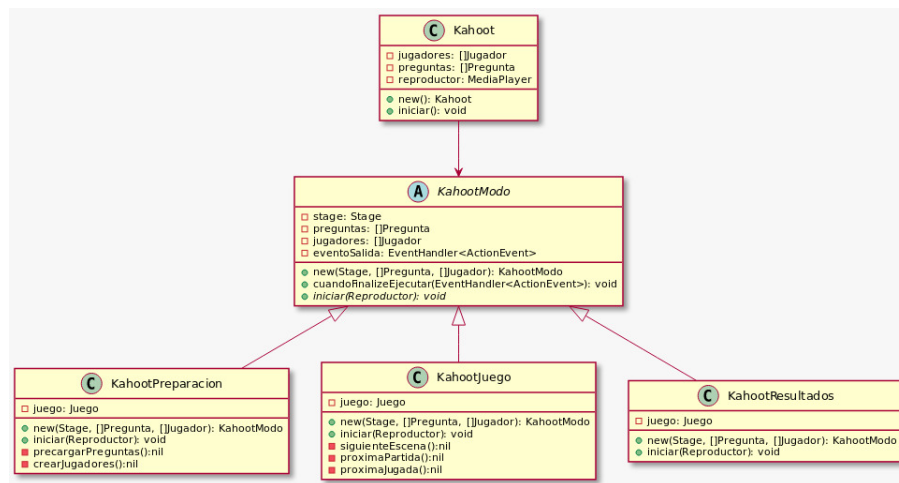


Figura 12: Kahoot

## 4.3. Flujo Juego

Clases utilizadas para poder realizar el flujo del juego.

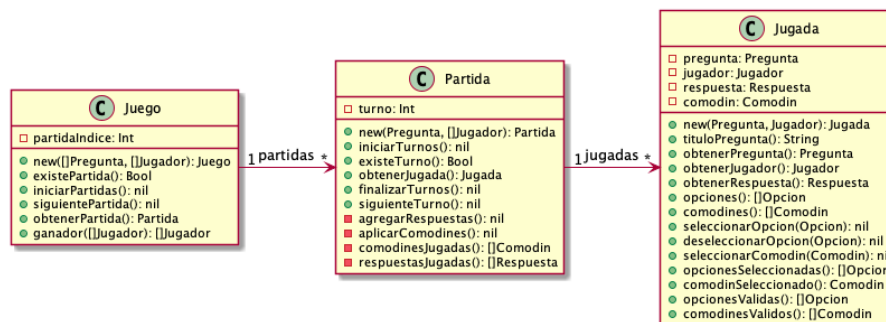


Figura 13: Juego

## 5. Detalles de implementacion

En el modelo se desarrolla un juego de preguntas y respuestas. La clase Juego es la encargada de guardar las partidas y cada Partida contiene la jugada correspondiente a cada jugador. La cantidad de partidas es igual a la cantidad de preguntas del juego.

La clase Jugador se encarga de guardar sus comodines y respuestas. Cada Respuesta contiene la pregunta a la cual pertenece, el jugador que responde, las opciones elegidas y el comodín que se aplica.

Las preguntas se clasifican en: VerdaderoFalso, MultipleChoice, GroupChoice y OrderedChoice. Cada pregunta tiene un estado (ConPenalidad o SinPenalidad) y un CalculadorDePuntaje. Este último puede ser del tipo Clásico, Ordenado, Parcial o ParcialEstricto. Cada uno, dependiendo de las opciones, calcula el puntaje correspondiente. Cada Pregunta tiene sus opciones del tipo Opción y cada opción tiene un Punto que puede ser Negativo, Positivo o Nulo.

### 5.1. Patrones utilizados

**State** Es utilizado por la clase Pregunta. Cada pregunta tiene un estado: ConPenalidad o SinPenalidad. Ambas clases se encargan de definir el puntaje de la pregunta, ya que dependiendo el estado cambia el punto de cada opción.

**Observer** Se utiliza en la interfaz gráfica para observar a la clases: Jugador y Pregunta.

### 5.2. Clases en las que se utiliza herencia

**Pregunta** En esta clase se aplica herencia ya que GroupChoice, MultipleChoice, OrderedChoice y VerdaderoFalso cumplen con la definición “es un”. VerdaderoFalso es una Pregunta y lo mismo para las demás clases. Otra razón es que entre las distintas clases se repetía la mayoría del código.

**EstadoPregunta** Las clases que heredan de la misma son: ConPenalidad y SinPenalidad. Ambas cumplen con la condición “es un” pues son los estados de la clase Pregunta. Además comparten la mayoría de sus métodos.

**Comodín** De esta clase heredan Exclusividad y Multiplicador. Ambas clases cumplen con la característica “es un”, es decir, Multiplicador “es un” Comodín y Exclusividad “es un” Comodín. Dado que la mayoría de sus métodos se repetían se decidió transformar a comodín como clase abstracta e implementar en sus herederas los métodos que diferían.

## 6. Excepciones

- AplicacionDeComodinInvalidaError;
- FactorComodinNegativoError;
- FactorComodinNuloError;
- JugadorInvalidoError;
- JugadorNoTieneAlComodinError;
- CantidadMaximaDeGruposError;
- CantidadMaximaDeOpcionesError;
- LectorFormatoDePreguntaError;
- LectorSintaxisError;

## 7. Diagramas de secuencia

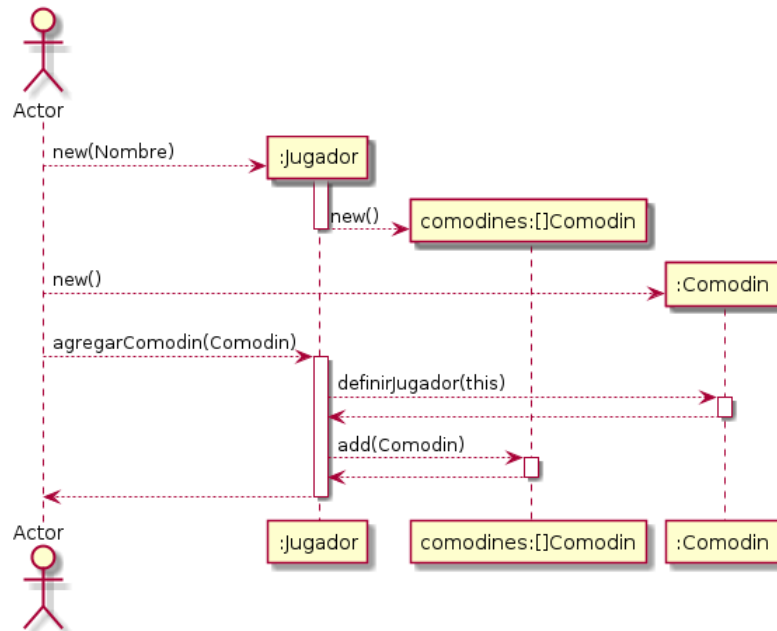


Figura 14: Creacion de jugadores asignando el comodin correspondiente

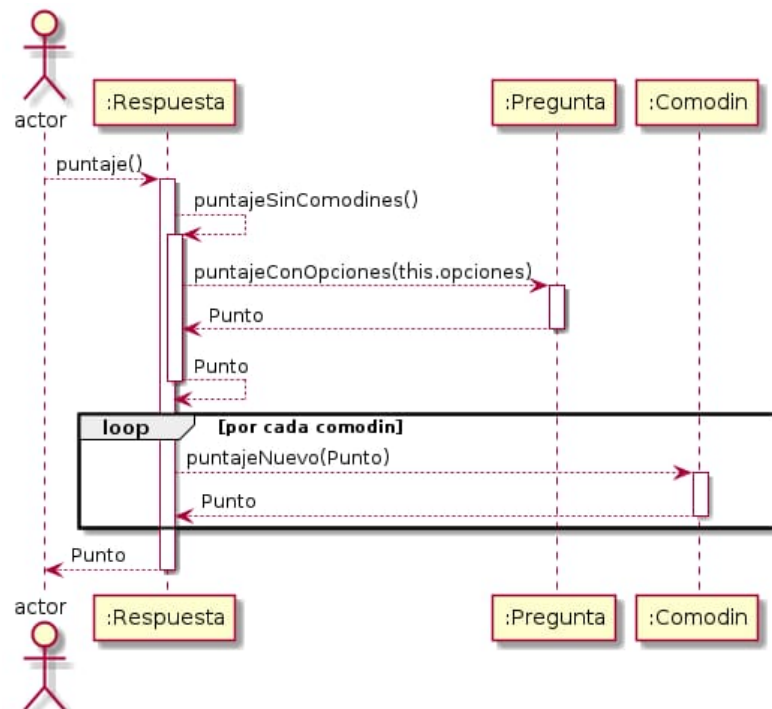


Figura 15: Obtención del puntaje de una respuesta

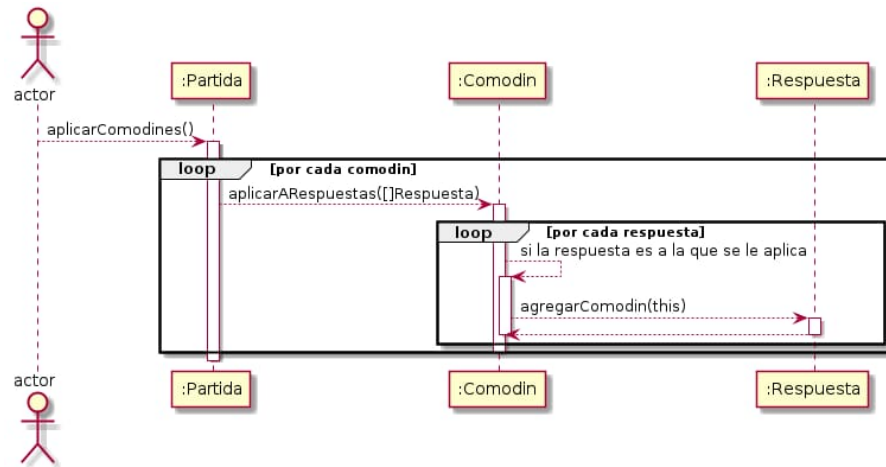


Figura 16: Aplicación de comodines

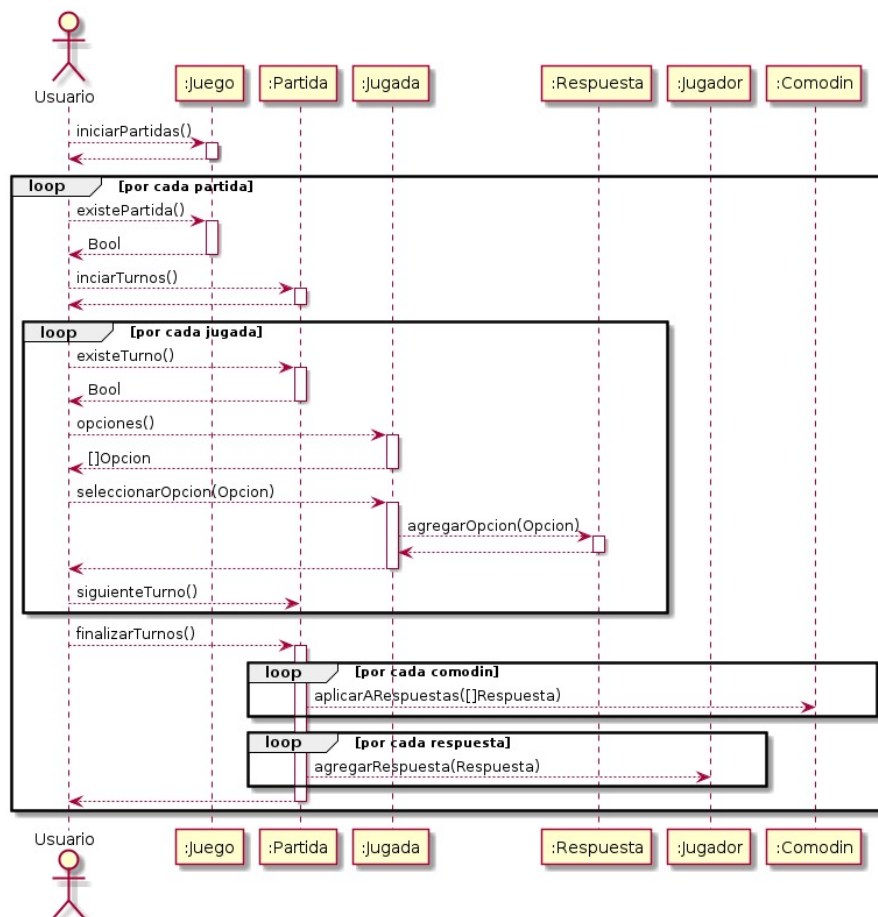


Figura 17: Flujo del juego

## 8. Diagramas de estado

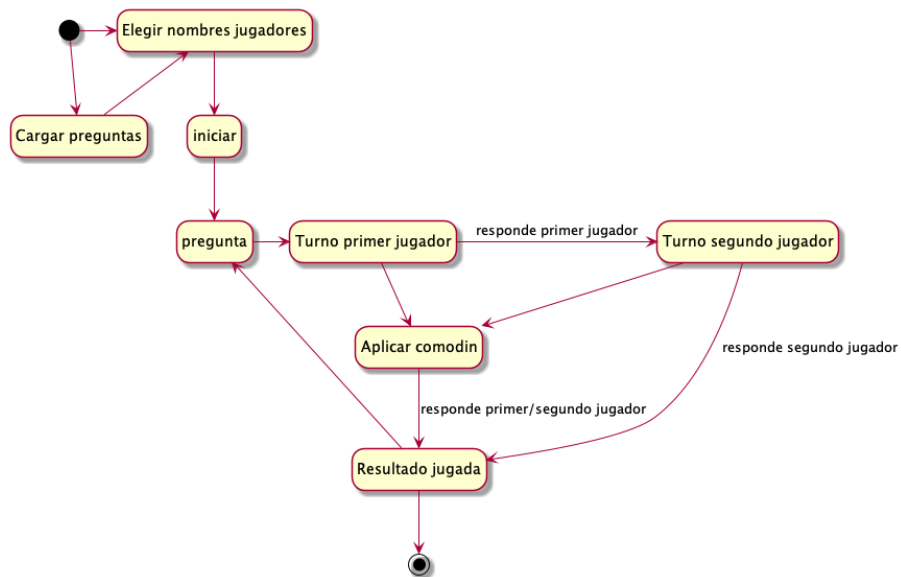


Figura 18: Diagrama de estados del juego