FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2 - Algohoot
Entrega Final

[7507/9502] Algoritmos y Programación III
Curso 1
Primer cuatrimestre de 2020

Grupo 4:

Kovnat, Leoni, Locatelli, Rosenblatt y Venglar.

Índice

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 implementar el juego de trivia Kahoot, denominado por nosotros como Algohoot, utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

2.1. Puntaje Negativo

Un jugador admite puntaje negativo si responde mal una pregunta con penalidad y tiene puntaje nulo.

2.2. Carga de Datos

Suponemos mejor no cargar los datos de los jugadores desde un botón en el menú, sino, previo a las rondas de preguntas. Además, si se ingresa algún jugador y se vuelve al menú inicial, se eliminarán todos los jugadores ingresados.

3. Diagramas de Clases

Dejamos a continuación el diagrama de clases que representa las relaciones establecidas hasta el momento:

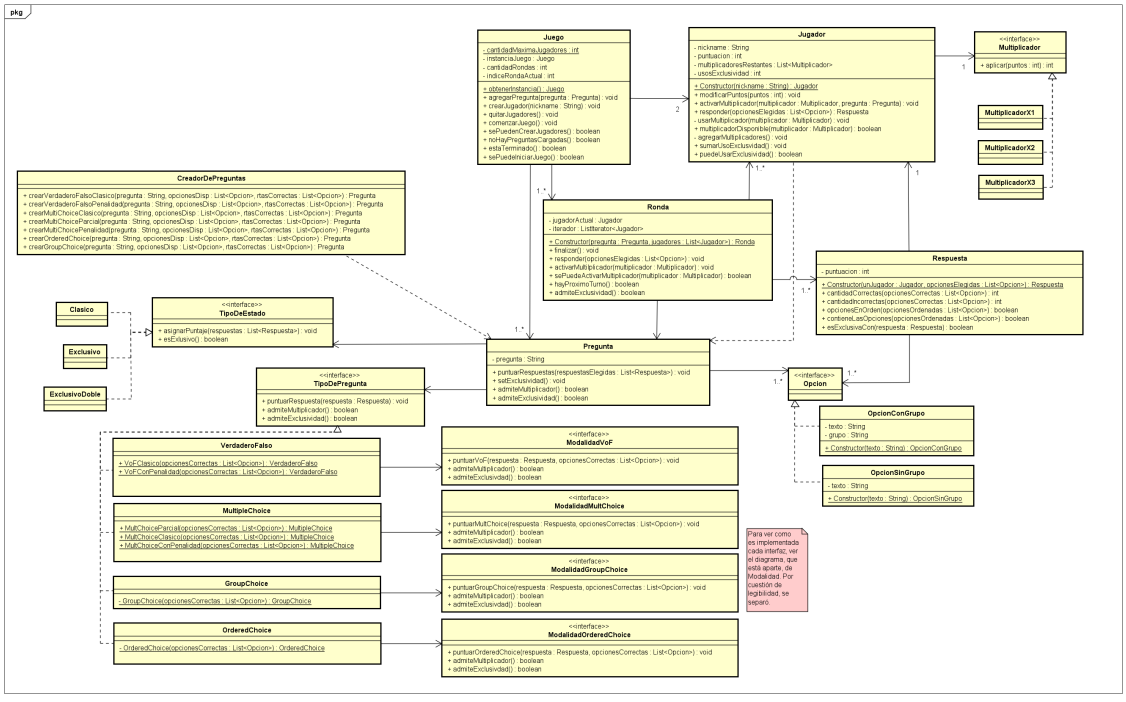

Figura 1: Diagrama de clases general.

Los siguientes diagramas muestran como las clases núcleo, es decir, las más impórtantes están relacionadas:

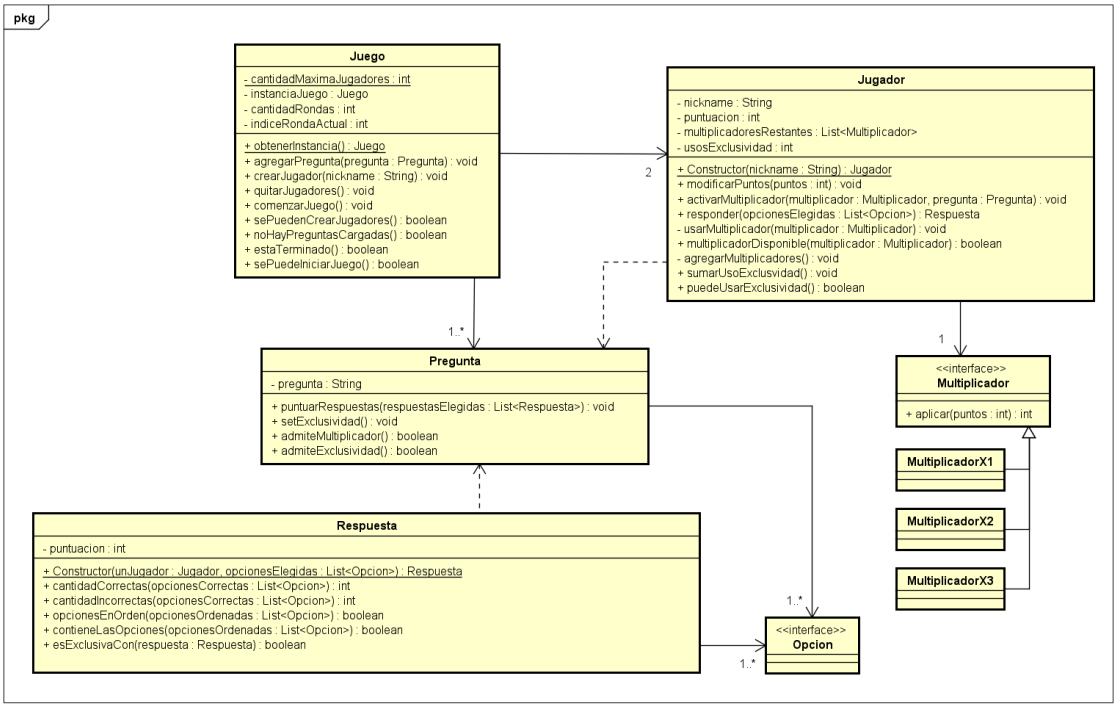

Figura 2: Principales relaciones entre Pregunta, Jugador, Respuesta, Opción, Multiplicador y Juego.

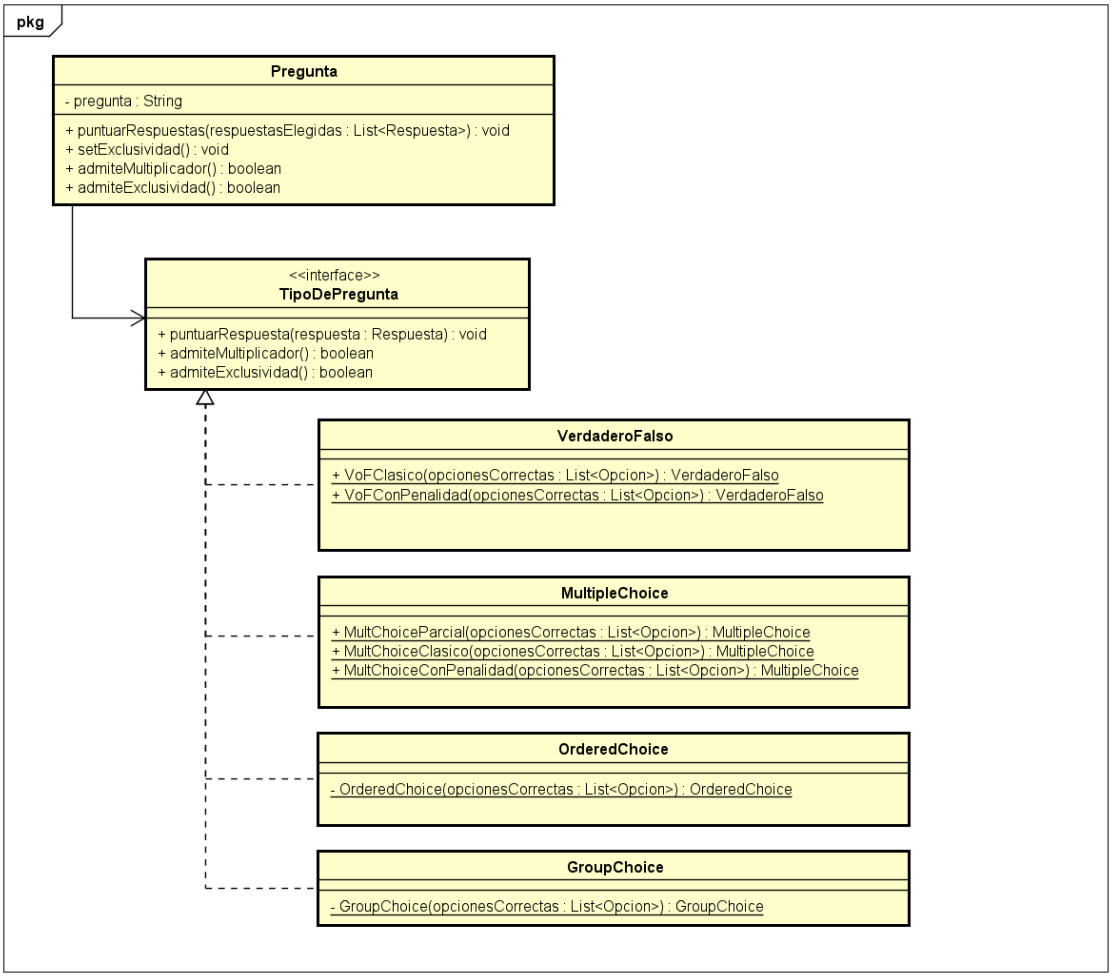

Figura 3: Cada instancia de Pregunta posee una instancia TipoDePregunta.

Cada pregunta conoce el tipo de estado actual y este puede ser clásico o exclusivo:

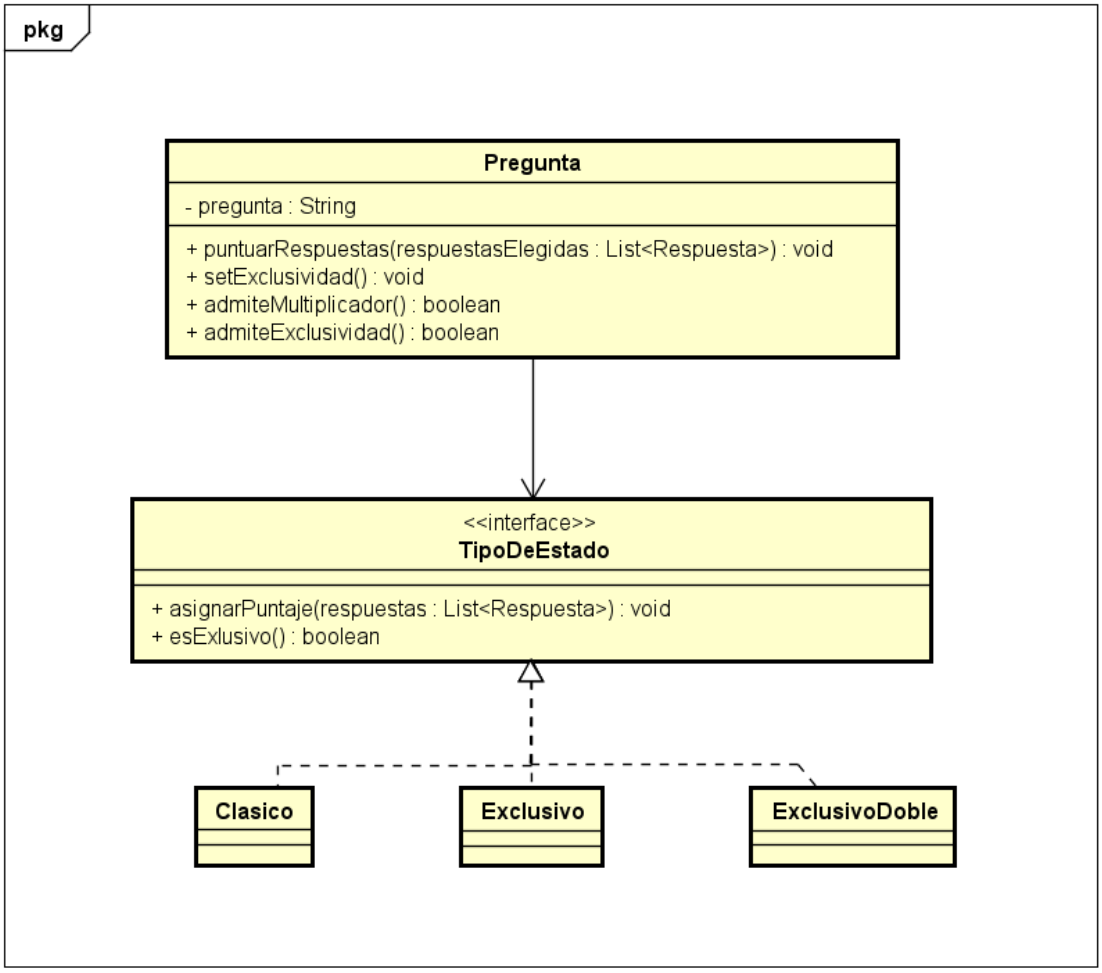

Figura 4: Relación entre el TipoDeEstado, las clases que la implementan y su relación con Pregunta.

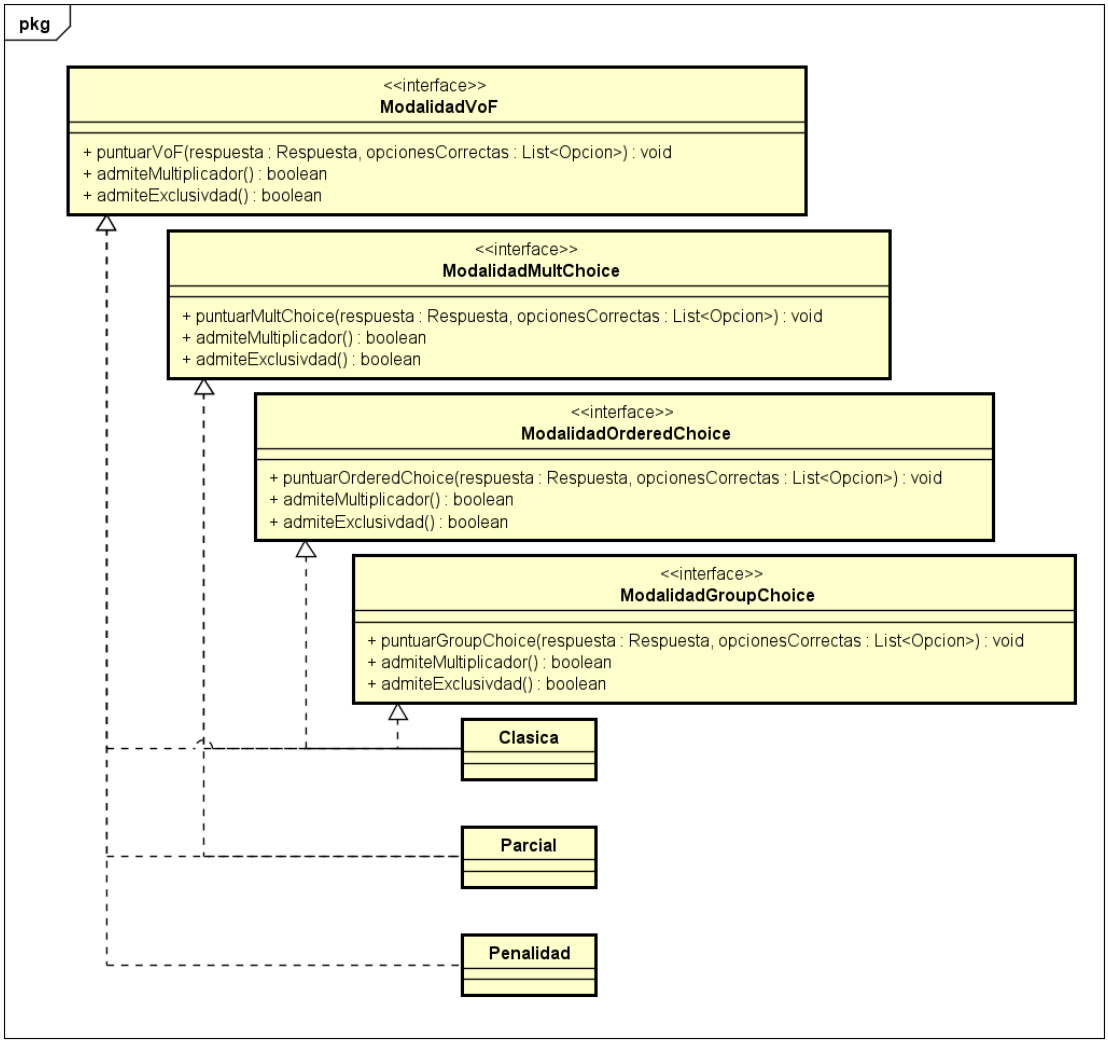

Figura 5: Como cada Pregunta puede tener una modalidad particular.

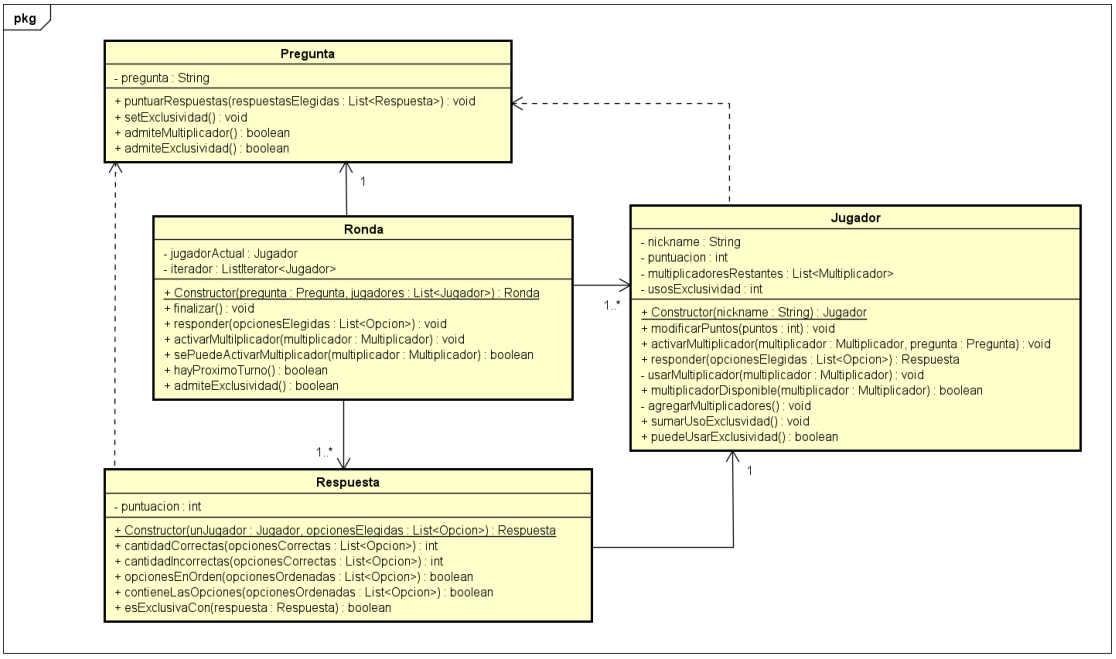

Figura 6: Relaciones con la Ronda y otras clases.

4. Diagramas de Secuencia

Dejamos a continuación los diagramas de secuencia que muestran las acciones más importantes:

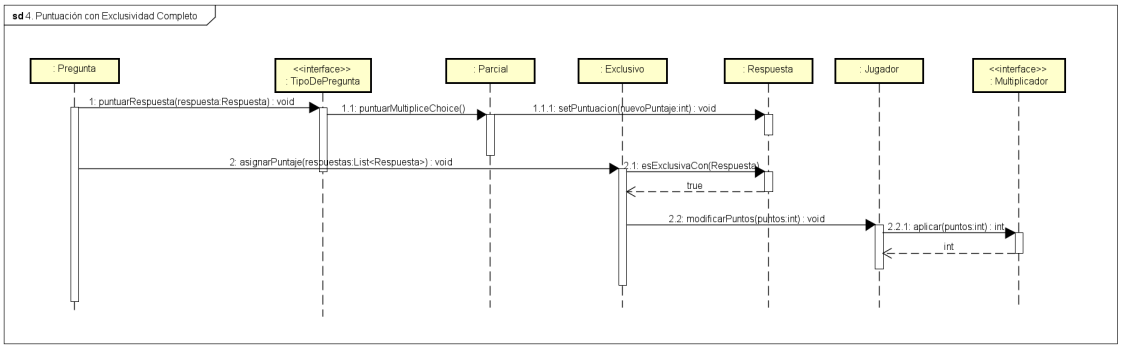

Figura 7: Puntuación de una instancia de pregunta.

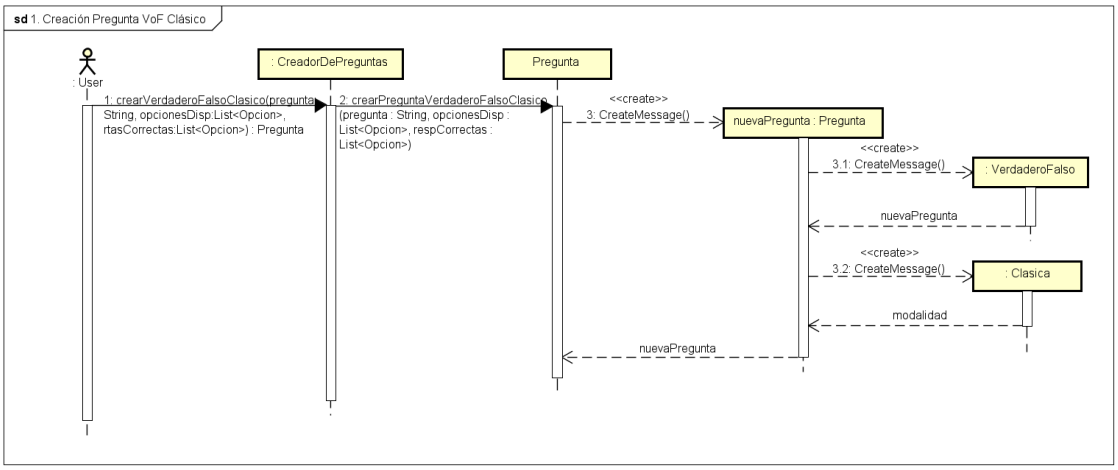

Figura 8: Creación de una pregunta.

5. Diagrama de Paquetes

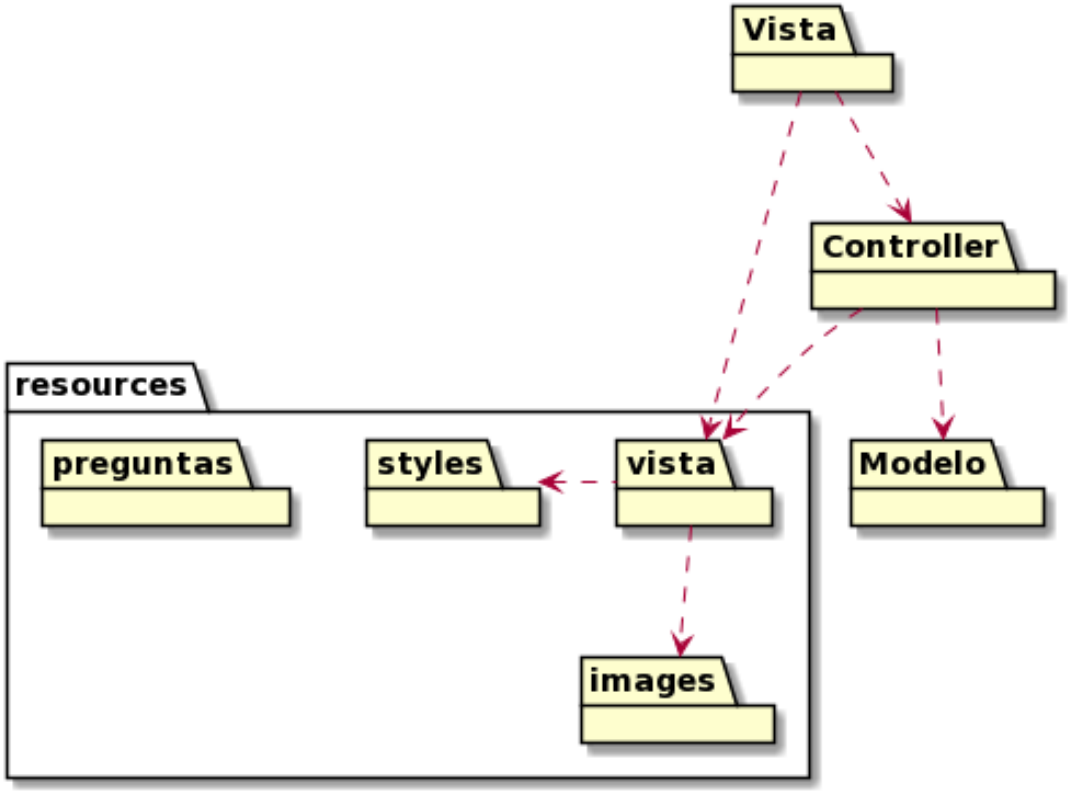

Figura 9: Diagrama de estados del juego.

6. Diagramas de Estados

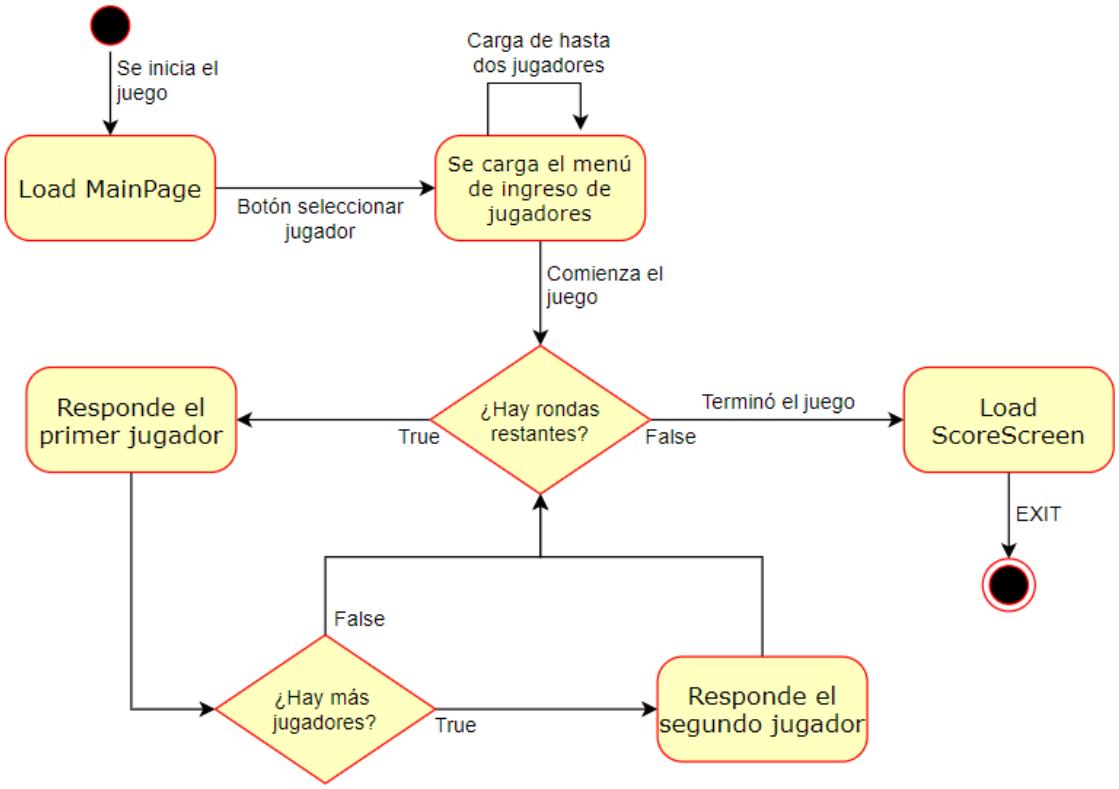

Figura 10: Diagrama de estados del juego.

7. Detalles de implementación

7.1. Como se contesta una pregunta

Una vez que el Jugador selecciona las opciones que desea, se le pide al Jugador que cree una instancia de Respuesta que contenga dichas opciones junto a una referencia al Jugador quien haya instanciado esa respuesta. Luego, esa respuesta es enviada por parametro a la instancia de Pregunta a traves del método puntuarRespuesta.

Luego, la Pregunta hace lo siguiente:

1. Delega la puntuacion de la respuesta a TipoDePregunta, que a su vez lo delega a la Modalidad(Clasico, Parcial, Penalidad).

2. Modalidad le envia a Respuesta la lista de las opciones correctas y esta devuelve la cantidad de opciones correctas e incorrectas. Modalidad se encarga de darle un puntaje a Respuesta dadas dichas cantidades. El método con el cual Modalidad puntua a la respuesta es exclusivo de cada tipo de Modalidad.

3. Una vez que se tiene la o las respuestas puntuadas, Pregunta se las envía a su Estado(Clasica, Exclusiva o Exclusiva Doble) la cual se encarga de puntuar al Jugador dado los puntajes de las respuestas de los jugadores.

El hecho de que cada objeto Pregunta y TipoDePregunta tenga objetos asociados que definan el comportamiento del mismo (Pregunta tiene Estado y TipoDePregunta, mientras que TipoDePregunta tiene Modalidad), corresponde al patrón de diseño Strategy.

7.2. Manejo de Turnos

El modelo cuenta con una clase Ronda, la cual se encarga del manejo de turnos para una pregunta en particular. Cuando se inicia el juego se instancia una Ronda por cada pregunta que se haya añadido al juego. La Ronda se encarga de mandar a puntuar las respuestas, cambiar de turno y de informar si ya no hay jugadores restantes para jugar. Una vez que se indica que no hay jugadores restantes, se mandan a puntuar las respuestas y se cambia de Ronda.

7.3. Conexión entre el Controlador y el Modelo

El controlador se encarga de conectar la Vista con el Modelo. Es decir, si se tiene una Vista que muestra los botones que representan las opciones, el controlador se encargara de recolectar las opciones seleccionadas mediante los clicks del usuario con las cuales se crea la instancia de Respuesta del jugador actual y, una vez que hayan respondido todos los jugadores, mandar a puntuar dichas respuestas. Tambien se encarga de modificar la vista teniendo en cuenta las limitaciones del modelo. Por ejemplo, si un jugador no tiene mas multiplicadores disponibles, el controlador se encarga de modificar la vista para que el boton que activa los multiplicadores este desactivado.