FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2: Kahoot

[7507] Algoritmos y Programación III
Curso 1 - Departamento de Computación
Primer cuatrimestre de 2020
Grupo 11

Alumno	Número de padrón	Mail
Cueto Quinto, Alan Ramiro	104319	acueto@fi.uba.ar
Graziosi, German	104623	ggraziosi@fi.uba.ar
Kasman, Lucía	97112	lkasman@fi.uba.ar
Ramos, Gisela	91553	garamos@fi.uba.ar

Índice

1. Supuestos

1. En el caso de los multiplicadores no dejamos que un jugador pueda usar dos multiplicadores en un mismo turno, al usar uno se desabilitaba el otro directamente.

2. En el caso de la exclusividad de puntaje hay un caso no contemplado por el enunciado, el cual es en una pregunta con respuesta parcial, que pasa si un jugador obtiene mas puntos que el otro pero sin haber elegido ninguno de los dos una de las opciones incorrectas. Nosotros decidimos en este caso directamente que el jugador que obtiene mas puntos sea el que obtiene el beneficio de la exclusividad, y el otro no recibe ningun punto.

3. En el caso donde el jugador esta respondiendo una pregunta, se acaba el tiempo y no envio su respuesta o en el caso de V o F no selecciono ninguna de las dos opciones, nosotros lo tomamos como que se envia una respuesta vacia, osea que no selecciono ninguna de las opciones.

4. En el caso de la pregunta Group Choice, nosotros decidimos que todas las opciones disponibles tienen que pertenecer a alguno de los dos grupos, osea que no pueden quedar opciones sin ser asignadas a alguno de los dos grupos.

2. Diagramas de clases

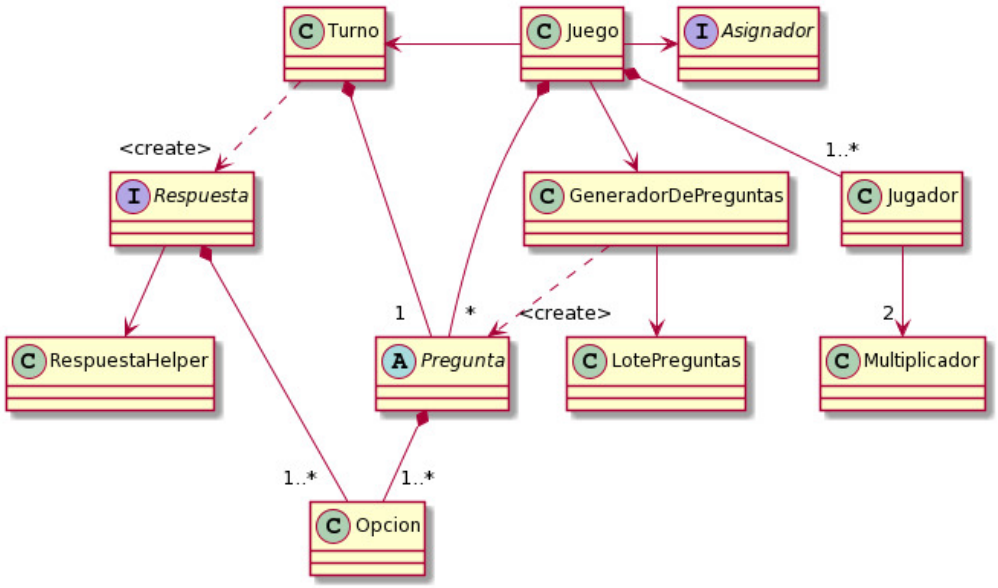

Figura 1: Diagrama general de Modelo.

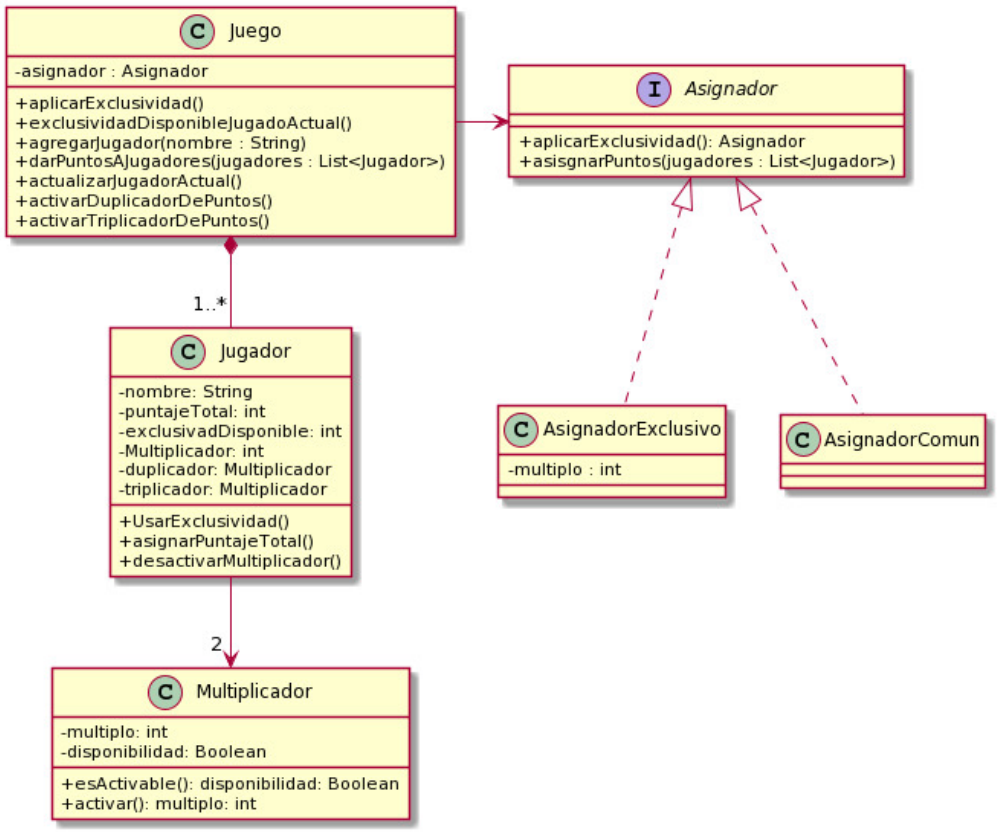

Figura 2: Diagrama de la relacion entre Juego y Jugador.

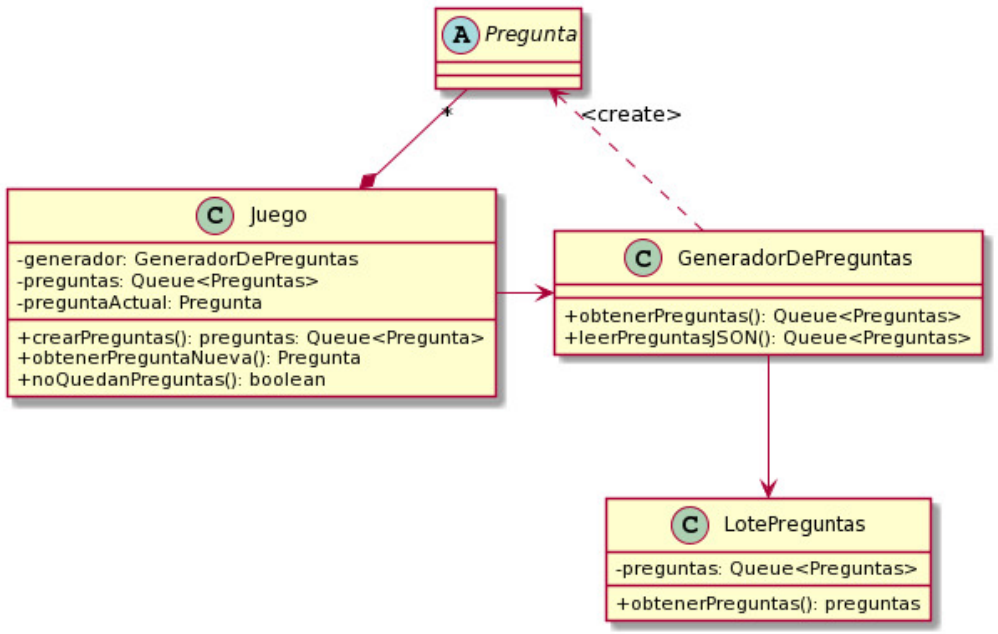

Figura 3: Diagrama de la relacion entre Juego y Preguntas con su generador.

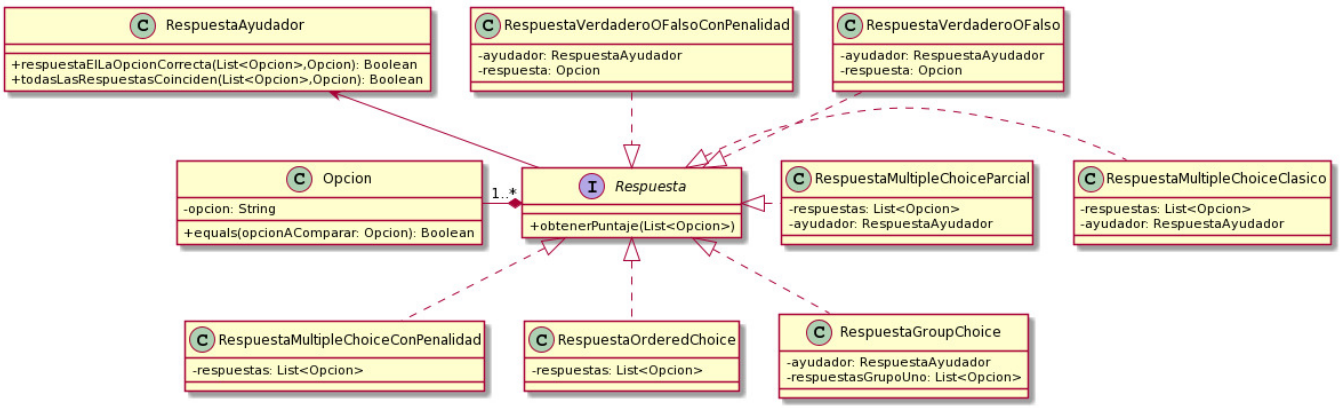

Figura 4: Diagrama de la interfaz Respuesta y las clases que la implementan.

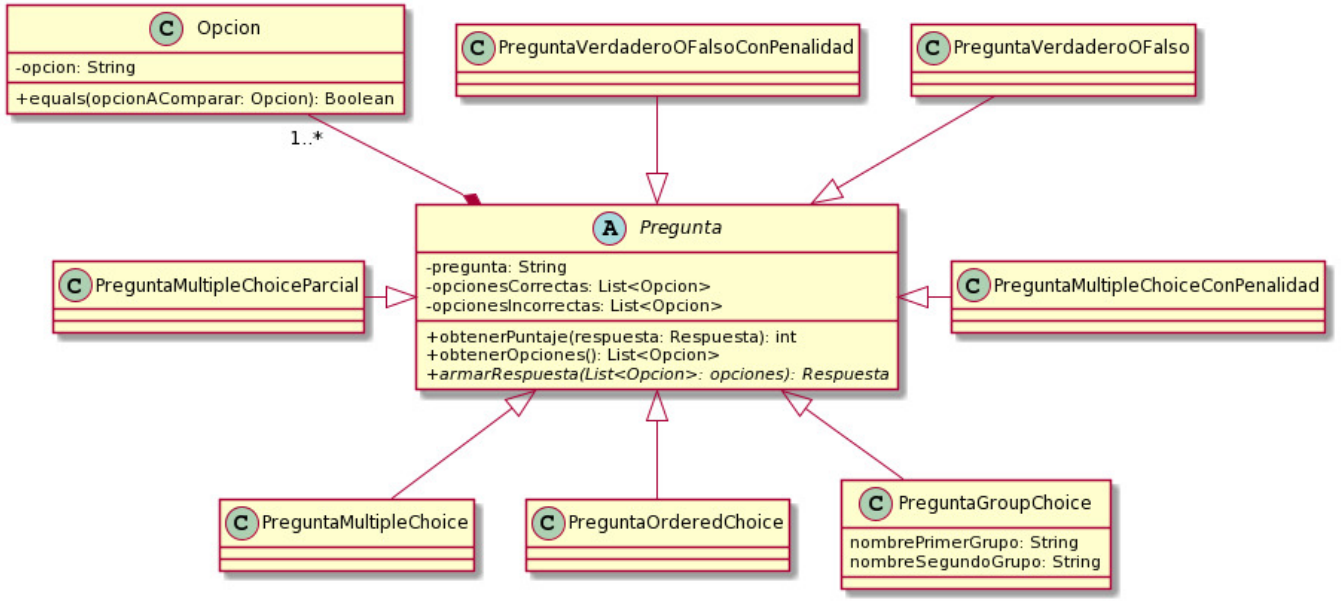

Figura 5: Diagrama de la clase abstracta Pregunta y todas sus hijas.

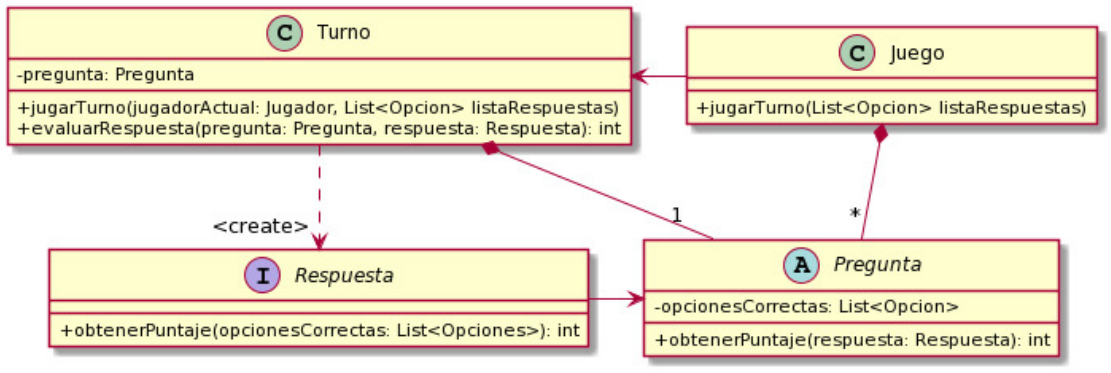

Figura 6: Diagrama de la relacion entre Turno y Juego.

3. Diagramas de secuencia

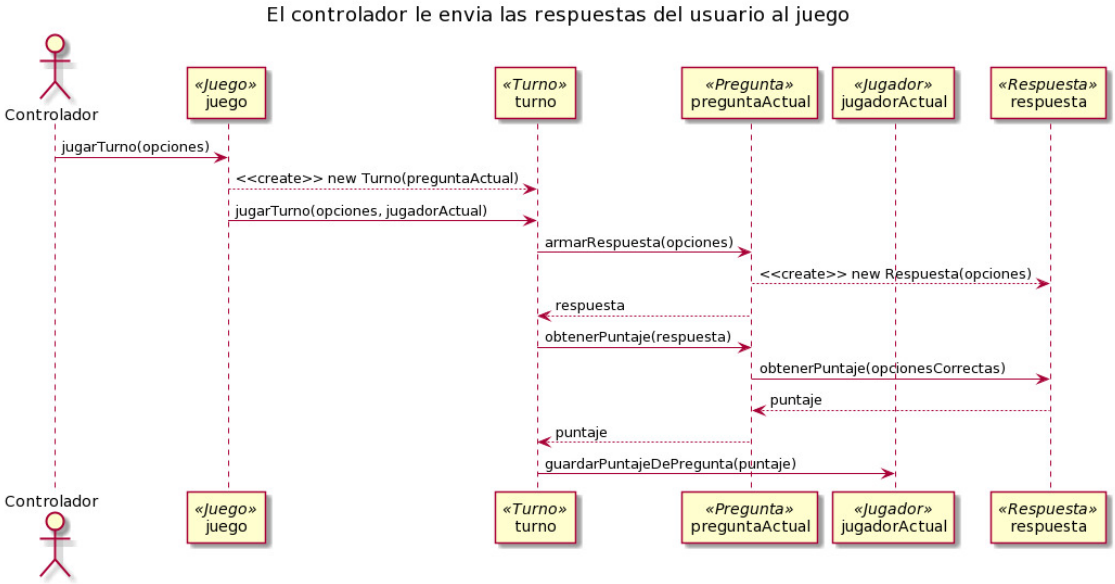

Figura 7

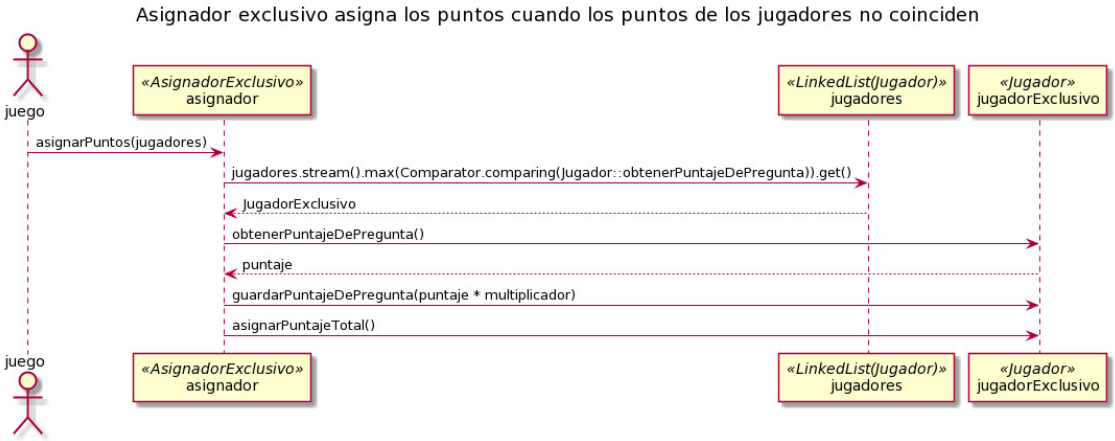

Figura 8

4. Diagrama de paquetes

PAQUETE MODELO Y SUS SUBPAQUETES

Figura 9: Vista general del paquete Modelo con sus subpaquetes.

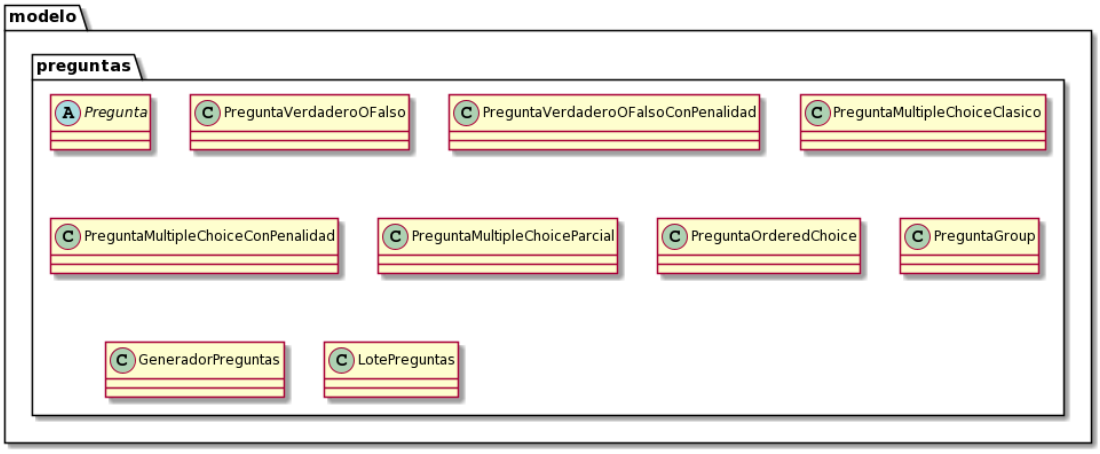

Figura 10: Vista completa del paquete “preguntas” dentro de modelo.

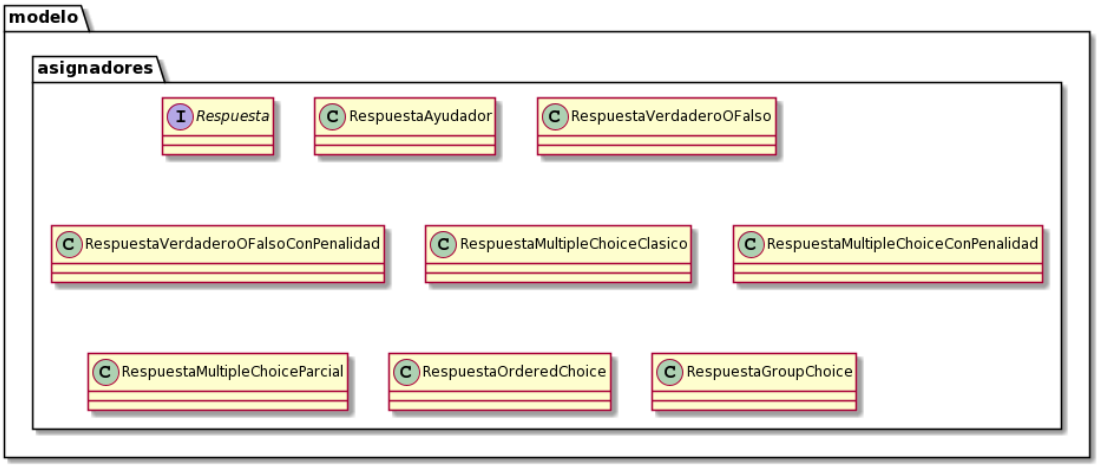

Figura 11: Vista completa del paquete “respuestas” dentro de modelo.

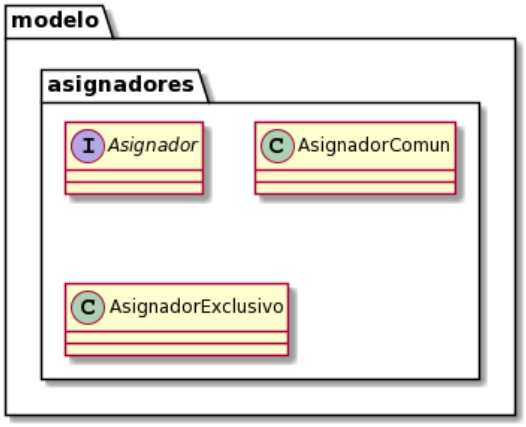

Figura 12: Vista completa del paquete “asignador” dentro de modelo.

PAQUETE VISTA Y SUS SUBPAQUETES

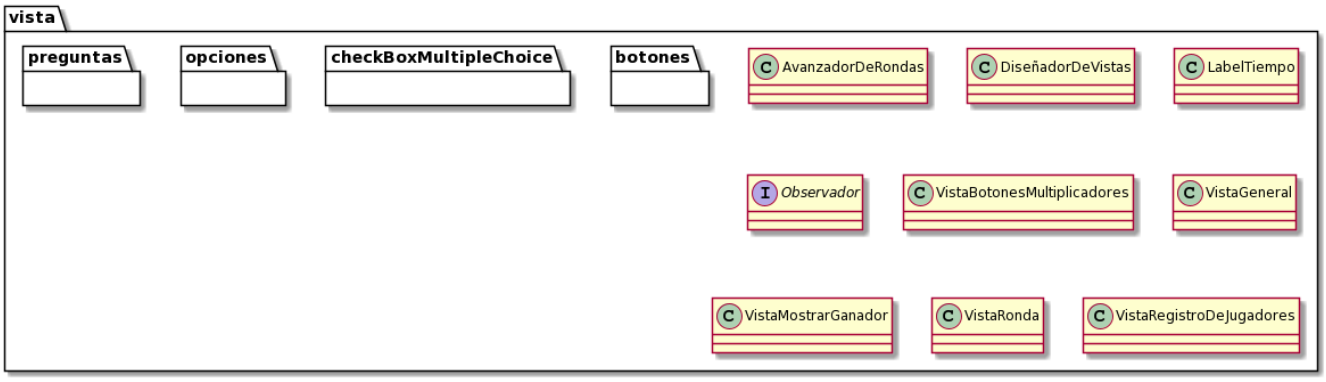

Figura 13: Vista general del paquete “vista” con sus subpaquetes.

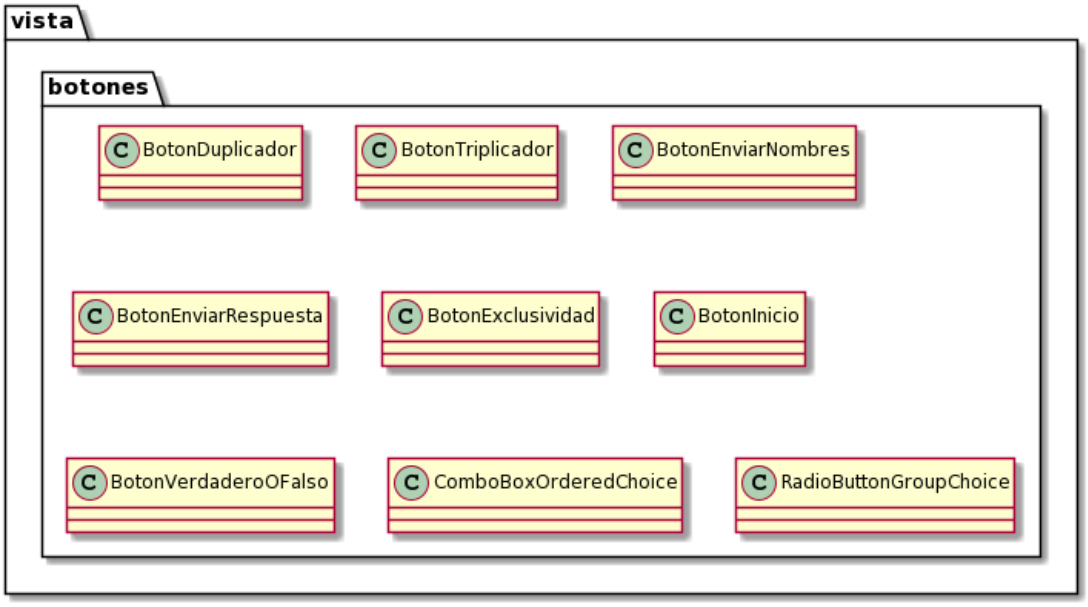

Figura 14: Vista completa del paquete “botones” dentro de vista.

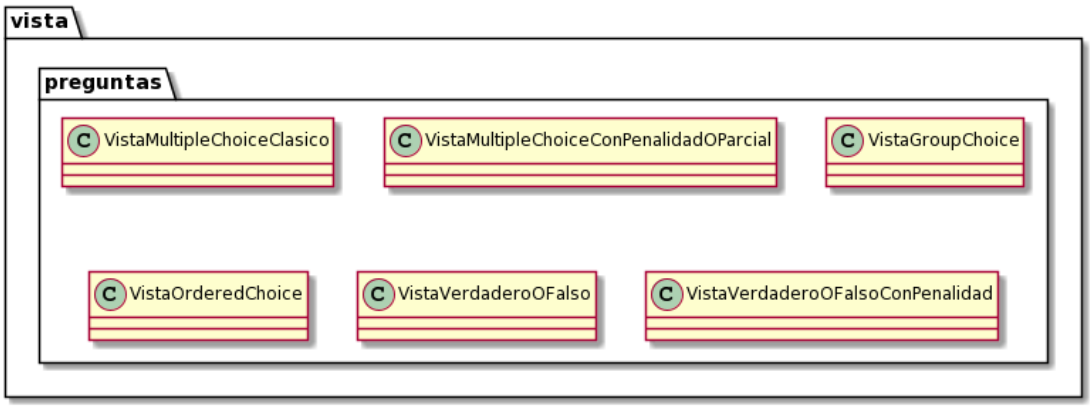

Figura 15: Vista completa del paquete “preguntas” dentro de vista.

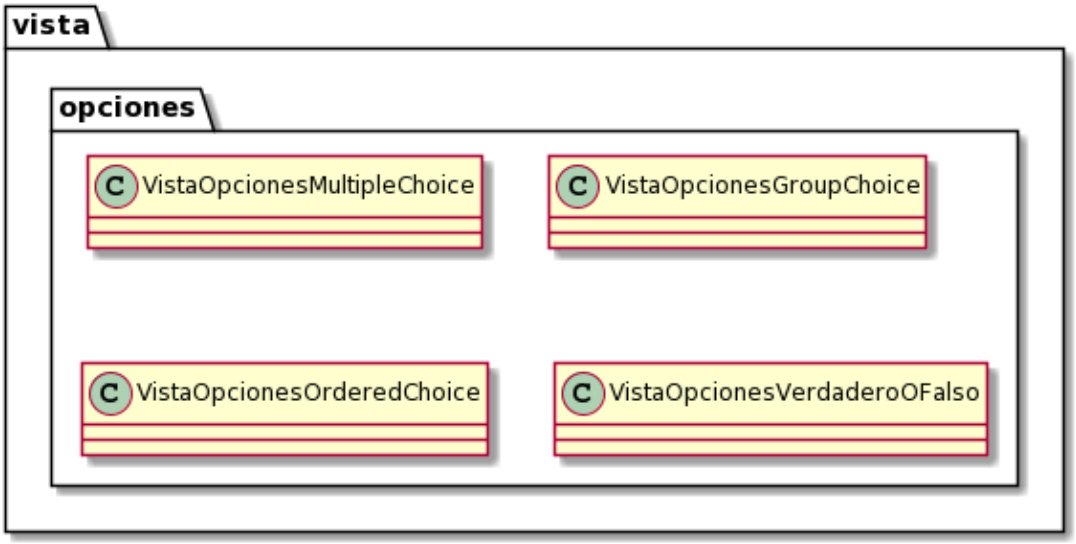

Figura 16: Vista completa del paquete “opciones” dentro de vista.

PAQUETE CONTROLADOR Y SUS SUBPAQUETES

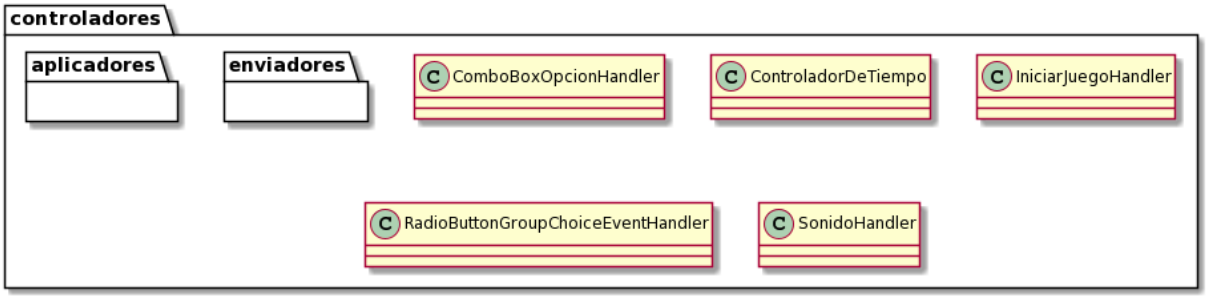

Figura 17: Vista general del paquete “controlador” con sus subpaquetes.

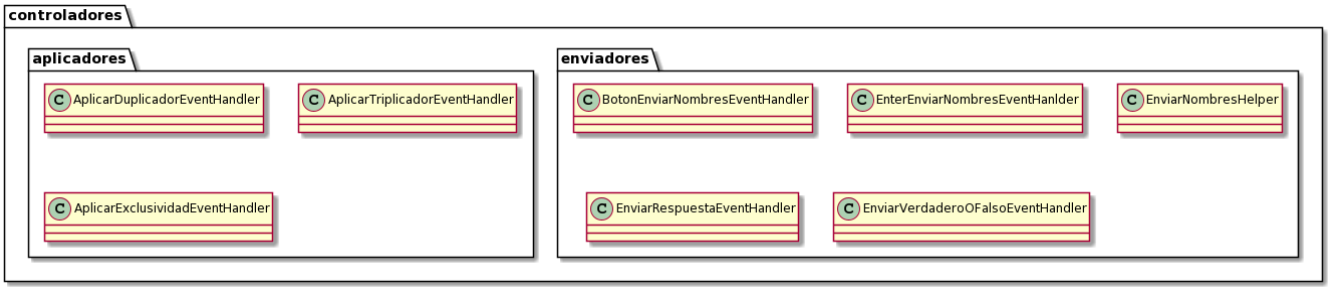

Figura 18: Vista completa del paquete “enviadores” y “aplicadores” dentro de controlador.

5. Diagramas de estado

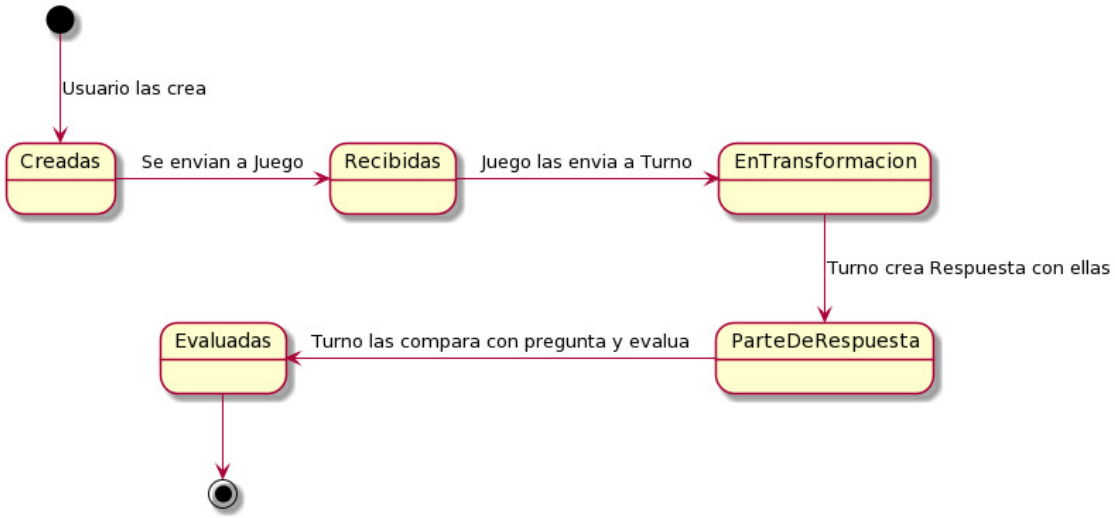

Figura 19: Estado de la clase Opcion cuando las crea el usuario para una Respuesta.

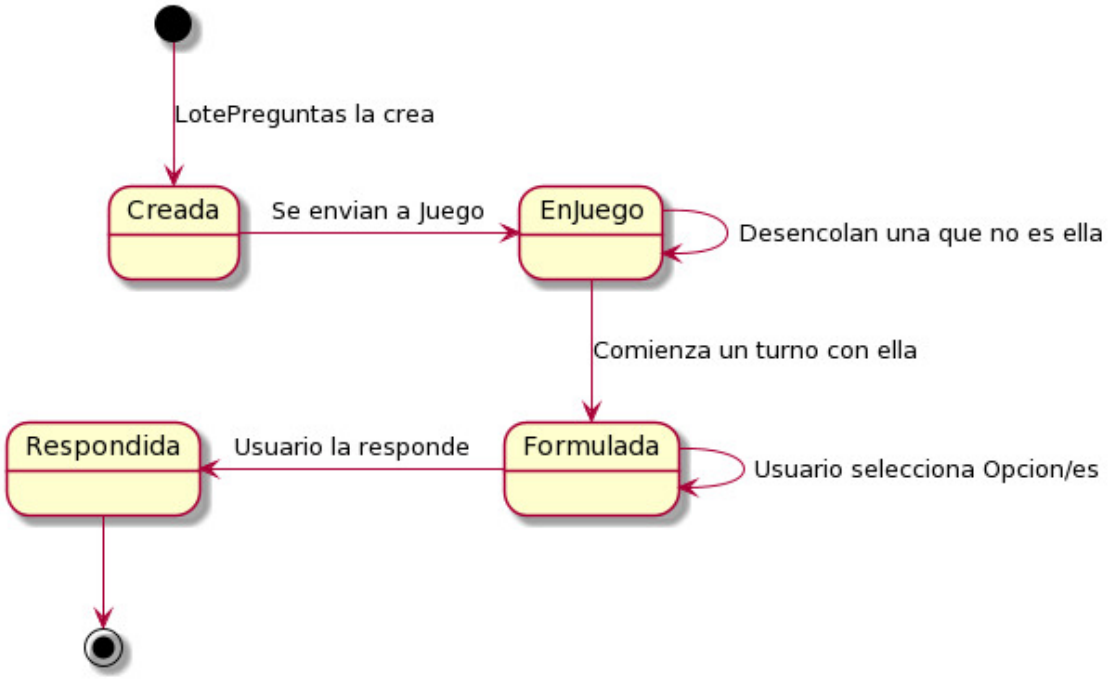

Figura 20: Estado de Pregunta desde que es creada.

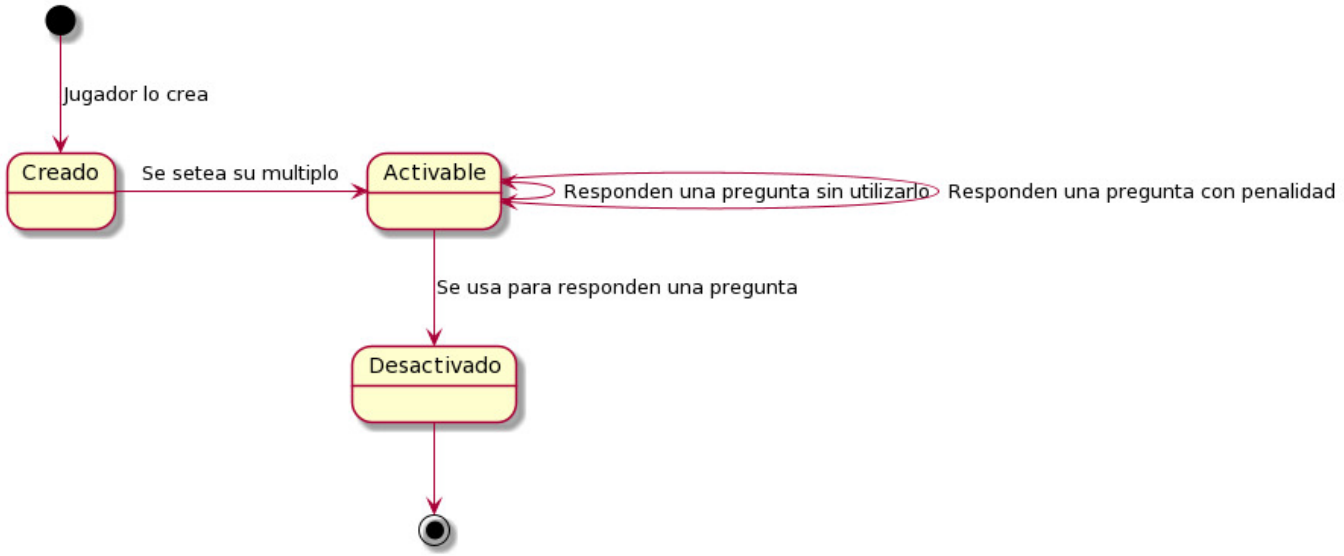

Figura 21: Estado de multiplicador desde que es asignado a un Jugador.

6. Detalles de implementación

1. En la implementacion de la exclusividad pedida en la consigna, nosotros usamos el patron "Strategy" y creamos una interfaz Asignador, la cual en tiempo de ejecucion decidimos si en el turno se usa el "asignador exclusivo" o el "asignador comun" en caso de que alguno de los jugadores haya seleccionado el boton de "Aplicar Exclusividad" o no.

2. Tambien usamos el patron "Strategy" en el caso de la vista para las preguntas multiple choice, en las check boxes donde se decide si una opcion esta seleccionado o no, decidimos que como eso se decide en tiempo de ejecucion tenga un estado "agregado" o "no agregado" los cuales reaccionan de manera distinta si se los selecciona, en caso de si se selecciona una con estado "agregado" pasa a "no agregado" y sale esa opcion de la respuesta y en caso de que el estado sea "no agregado" y se lo selecciona pasa todo lo contrario, la opcion pasa a ser parte de la respuesta y su estado pasa a "agregado".

3. Para las respuestas decidimos crear una Interfaz general de Respuesta, la cual tiene un metodo obtenerPuntaje(), el cual todas los distintos tipos de respuesta lo implementan de diferente manera comparando las opciones seleccionadas con las opciones de la Pregunta a responder.

4. En el caso de las preguntas, creamos una clase abstracta Pregunta, los distintos tipos de preguntas heredaban de esta, ya que todos manejaban los mismos atributos y todas las clases implementaban solamente un metodo, el cual era armarRespuesta(), el cual era implementado de diferente manera en todas sus clases hijas(En el caso de PreguntaOrderedChoice, arma una RespuestaOrderedChoice y asi el resto).