

Trabajo Práctico 2 — Java

[7507] Algoritmos y Programación III

Curso 1

Segundo cuatrimestre de 2020

Alumno1:	Izquierdo, Stephanie
Número de padrón:	104196
Email:	sizquierdo@fi.uba.ar

Alumno2:	Segura, Agustina
Número de padrón:	104222
Email:	asegura@fi.uba.ar

Alumno3:	Correa, Valentina Laura
Número de padrón:	104415
Email:	vcorrea@fi.uba.ar

Alumno4:	Fontenla, Maria Sol
Número de padrón:	103870
Email:	msfontenla@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas	3
4.1. Diagramas de clase	3
4.2. Diagrama de estado	5
4.3. Diagramas de secuencia	5
4.4. Diagrama de paquetes	10
5. Detalles de implementación	10
6. Excepciones	11

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 un juego de preguntas y respuestas en Java.

2. Supuestos

- En las preguntas con penalidad, los puntajes se sumarán tanto si se ha elegido opciones incorrectas, con puntaje negativo, y opciones correctas, con puntajes positivos.
- Si seleccionada una o dos exclusividades el formato es parcial, entonces se asigna el puntaje con exclusividad al jugador que haya marcado más respuestas correctas (resultando en el que haya obtenido el mayor puntaje).
- Nunca se da el caso de que el jugador elija multiplicar en las preguntas que no son de penalidad porque no se lo permite la vista. Lo mismo para la exclusividad, que la vista solo permite mostrarla cuando las preguntas no son con penalidad.
- A medida que el jugador gasta su cupo de multiplicadores/ exclusividades, los botones para usarlos se deshabilitan, así el jugador queda inhabilitado para responder usando dichas opciones.
- Para las preguntas de formato clásico, se asignan diez puntos al jugador que haya elegido correctamente todas las opciones.
- Para las preguntas con parcial se asignan diez puntos por cada opción correcta elegida por el jugador, mientras ninguna de ellas sea incorrecta. En caso de alguna ser incorrecta, se anula el puntaje asignando cero puntos.
- Cada jugador posee únicamente un multiplicador por tres, uno por dos, y dos exclusividades.

3. Modelo de dominio

El diseño general consta de una clase Kashoot, la cual tiene una referencia a las entidades principales del juego, tales como las rondas y los jugadores. Además, una clase VistaKashoot se encarga de la visualización de las preguntas y opciones. Esta se relaciona directamente con Kashoot, la cual le proporciona esta información.

A medida que pasan los turnos, es decir, que juega cada jugador, se actualizan las instancias de VistaKashoot y de Kashoot, actualizando las preguntas con sus opciones, puntajes acumulados, puntajes asignados, entre otras cosas.

Cada pregunta posee cierta cantidad de opciones, entre las cuales se encuentran las opciones correctas e incorrectas. La pregunta ya posee una respuesta correcta, la cual se asocia a ella en el momento de su creación. Cuando un jugador selecciona las opciones que considere correctas para una pregunta, se crea una respuesta compuesta por dichas opciones y luego se compara con la respuesta correcta.

Las preguntas son creadas con su formato/modalidad correspondiente y determinan el puntaje obtenido por cada jugador luego de que estos responden.

Los puntajes de los jugadores se asignan al final de cada ronda, o sea, cuando ambos jugadores hayan respondido.

El juego consta de tres escenas principales. La primera, es una escena de bienvenida, donde se le pide a ambos jugadores que ingresen sus nombres y se encuentra el botón para comenzar a jugar. Una segunda escena, distribuye las entidades visibles al usuario, que se actualizan como ya se mencionó previamente. La tercera, es una escena final, donde se muestran los puntajes finales

obtenidos por los jugadores y se muestra el botón para salir. Todas estas escenas se muestran en una ventana principal.

4. Diagramas

4.1. Diagramas de clase

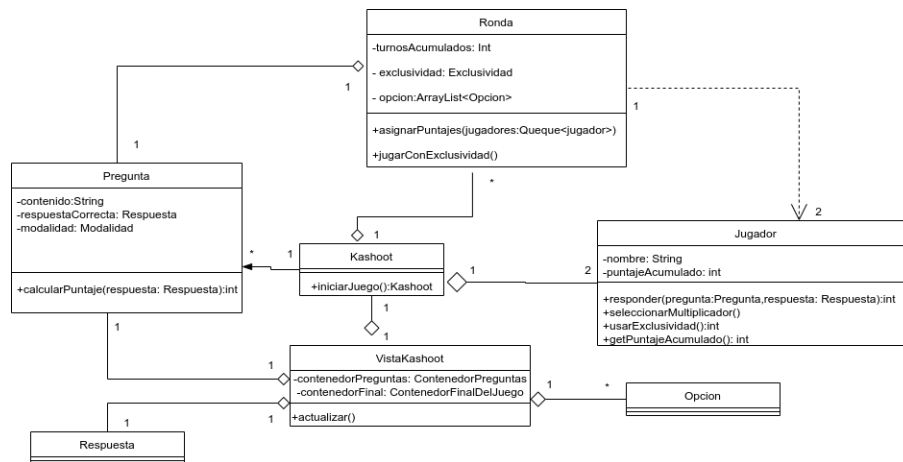


Figura 1: diagrama general de clases

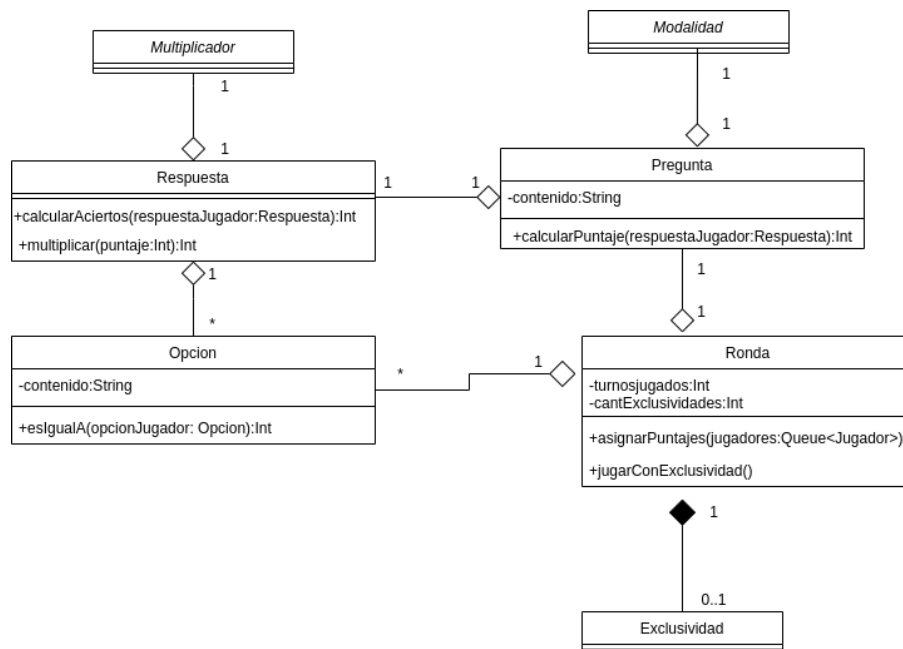


Figura 2: clases Pregunta, Ronda, Respuesta y Opcion

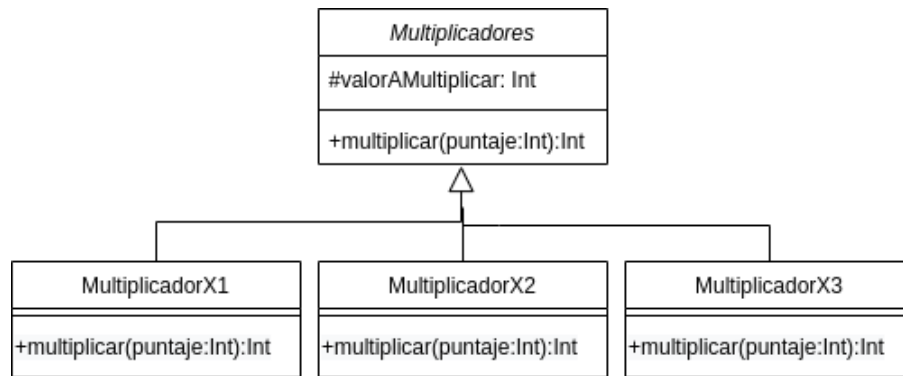


Figura 3: clase Multiplicador

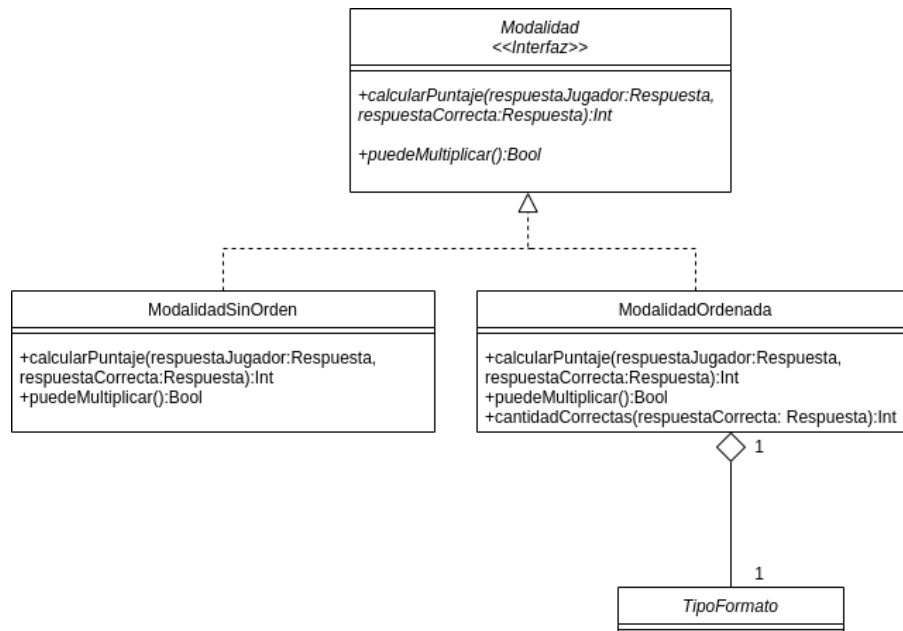


Figura 4: clase Modalidad

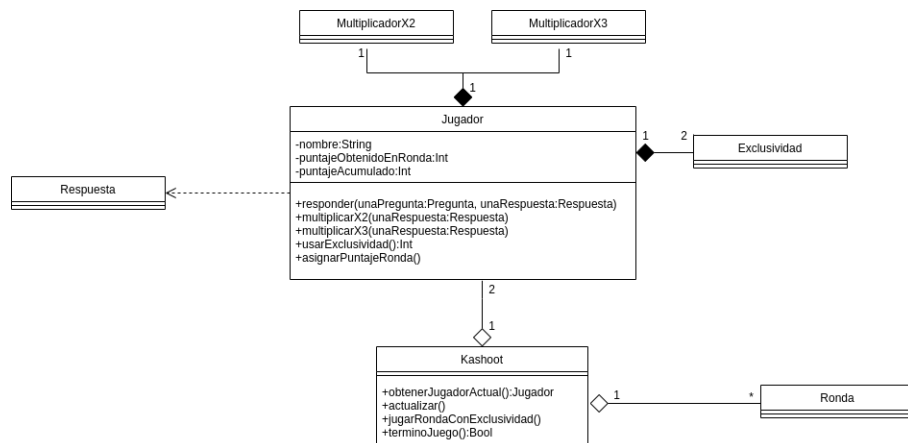


Figura 5: clases Kashoot y Jugador

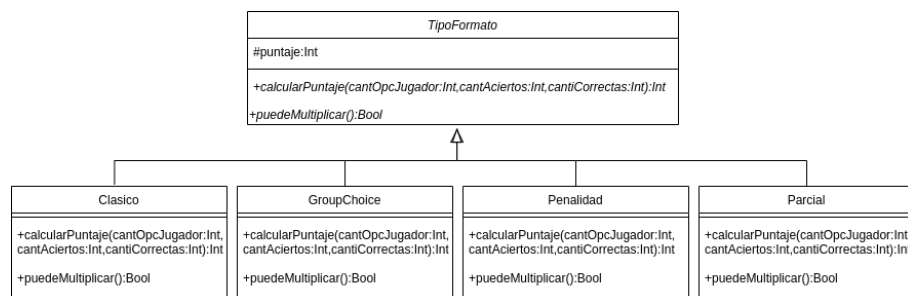


Figura 6: clase Formato

4.2. Diagrama de estado

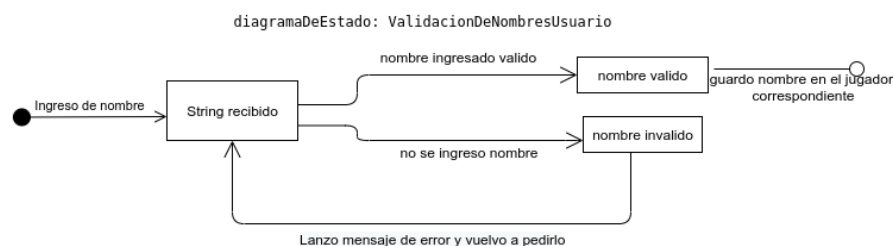


Figura 7: Validacion del nombre de un usuario.

4.3. Diagramas de secuencia

Todos los diagramas de secuencia arrancan con las instancias de sus clases correspondientes ya inicializadas

El jugador responde correctamente con multiplicidadX2 a una pregunta choice con penalida ya inicializada. La respuesta del jugador ya estaba inicializada.
 Responde correctamente

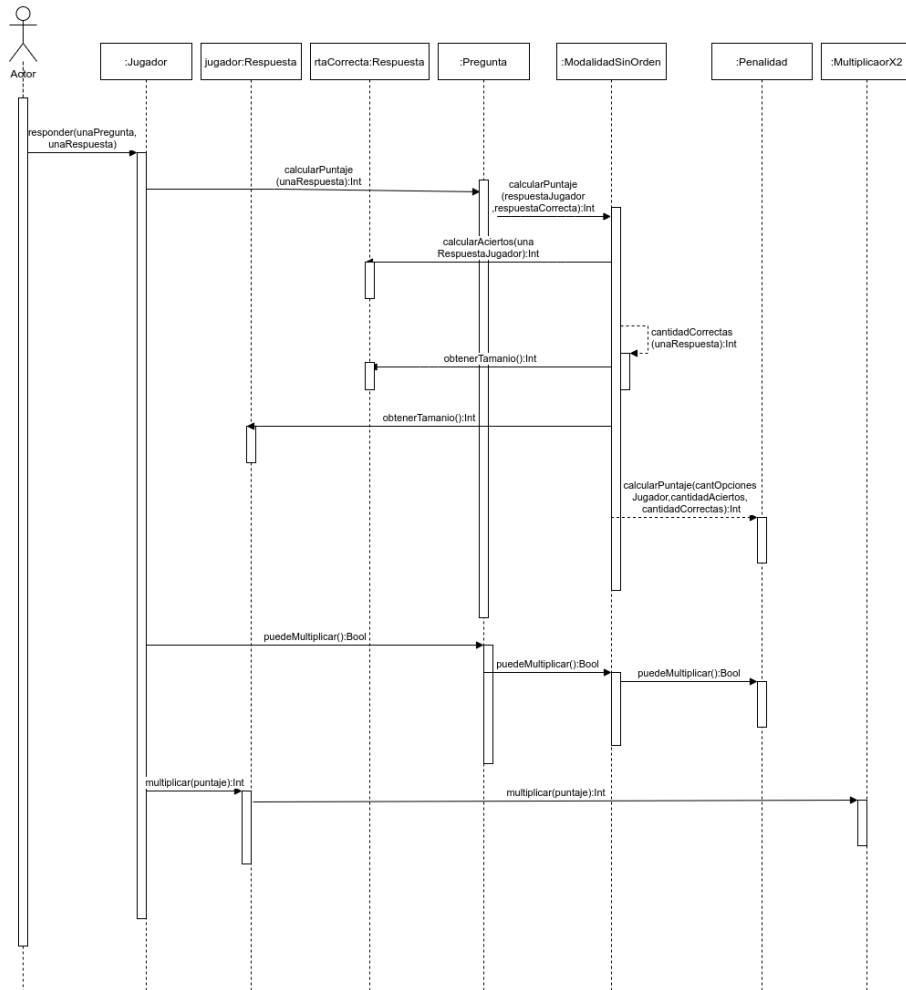


Figura 8: Flujo utilizando el formato penalidad

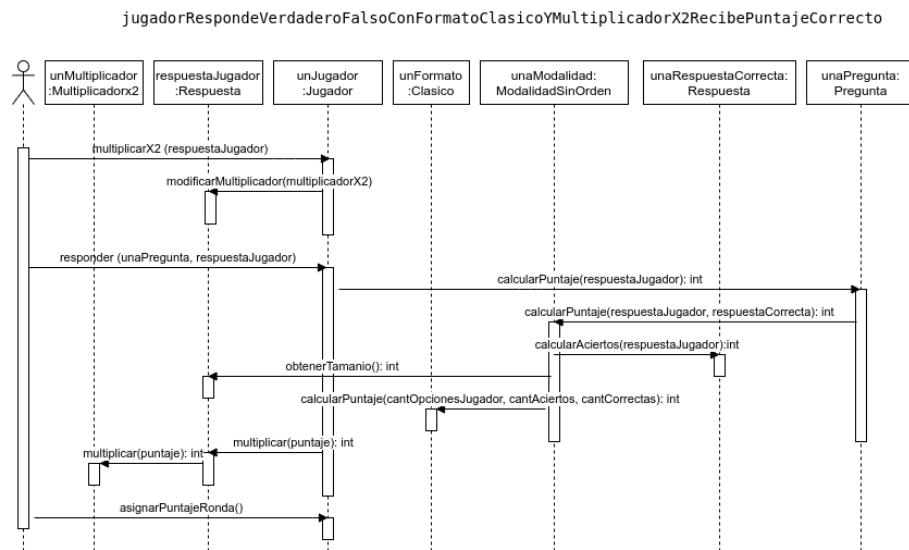


Figura 9: Flujo de una respuesta utilizando multiplicadorX2

Dos jugadores respondes a una pregunta choice con penalidad y multiplicidadX2. Estos ya se encuentran inicializad
Se les asigna el puntaje Correspondiente

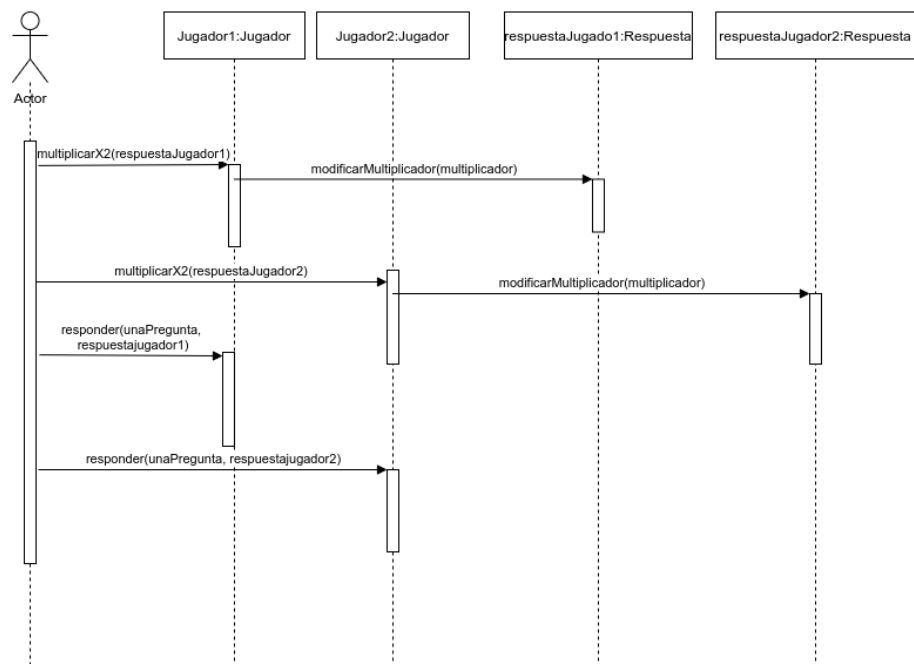


Figura 10: Flujo de dos jugadores utilizando formato clasico y multiplicador

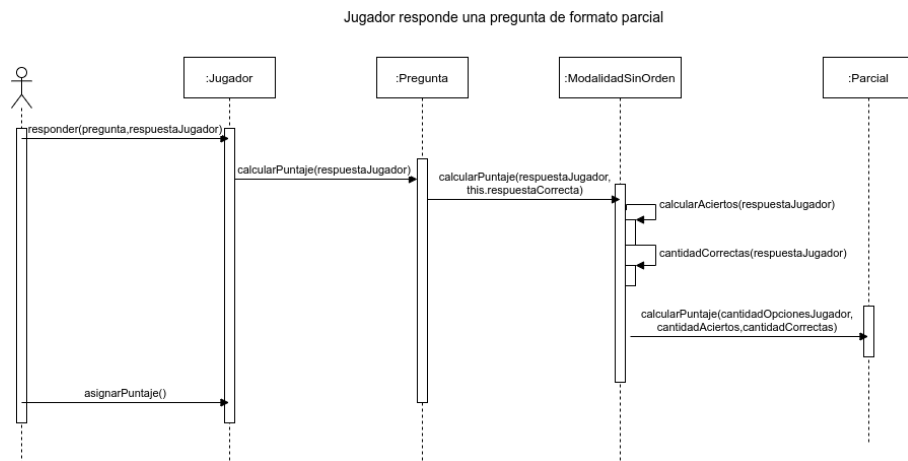


Figura 11: Flujo de jugador utilizando formato parcial

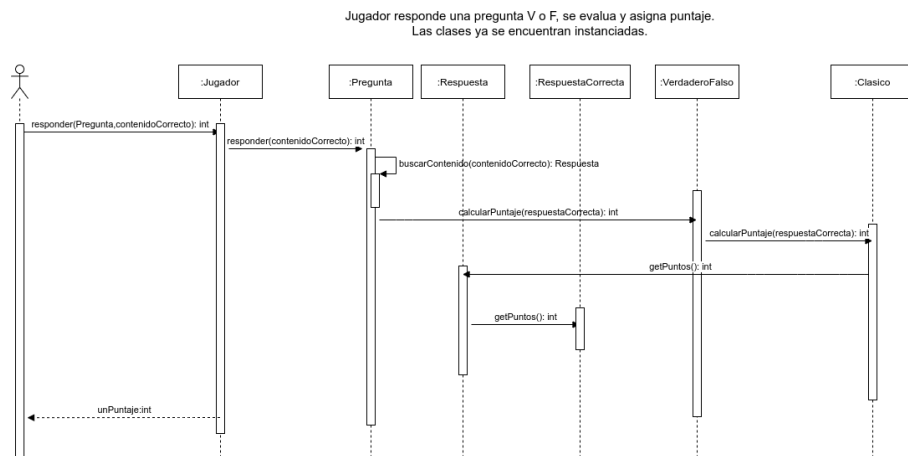


Figura 12: Flujo de jugador utilizando formato clasico

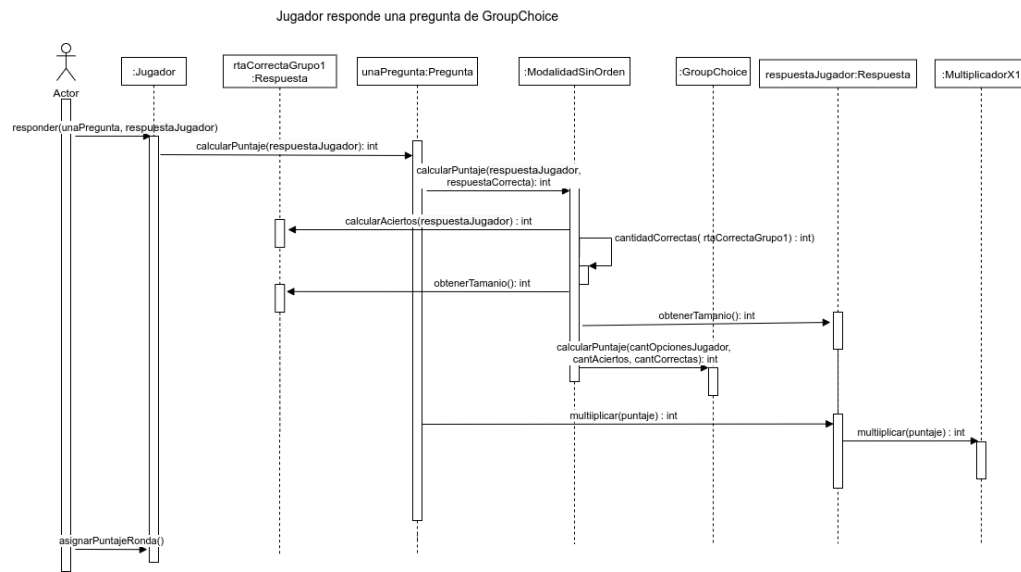


Figura 13: Flujo de jugador utilizando formato groupChoice

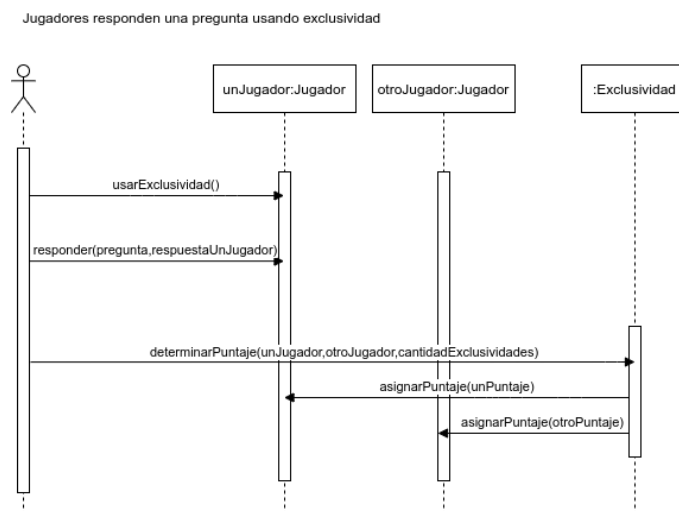


Figura 14: Flujo de jugador utilizando Exclusividad

4.4. Diagrama de paquetes

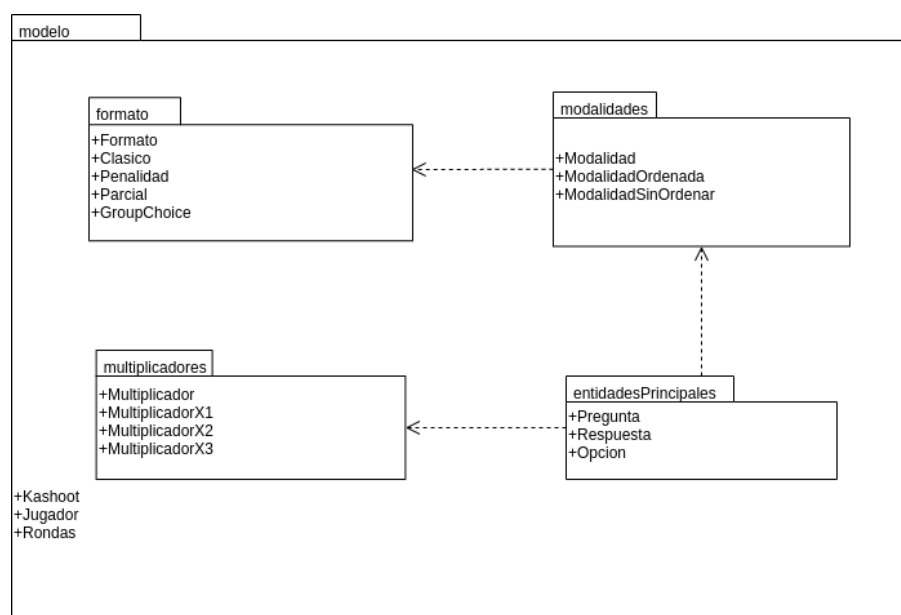


Figura 15:

5. Detalles de implementación

Responder con exclusividad: para la implementación de esta opción, la cual asigna puntaje a los jugadores en función de cómo fueron sus respuestas, se utilizan principalmente los puntajes obtenidos en la ronda jugada. Una vez que ambos jugadores responden, si al menos una exclusividad fue elegida, se le delega a esta última que actualice los puntajes obtenidos en la ronda, siguiendo el criterio de asignación de puntajes por exclusividad.

Responder con multiplicador: para la implementación de esta opción, se optó por definir una clase abstracta de la cual hereden los distintos tipos de multiplicadores, incluyendo uno por defecto que multiplica por uno. Cada vez que el jugador responde, el puntaje obtenido se multiplica independientemente de si eligió multiplicador o no. Si ningún multiplicador fue seleccionado por el jugador, se responde por el ya mencionado multiplicador por uno. Esto se trata de un patrón strategy. Se utiliza debido a que el multiplicador es un comportamiento que se determina por el jugador en tiempo de ejecución. El comportamiento se modifica en la respuesta del jugador.

Modalidades: para los distintos comportamientos de las preguntas se definieron las modalidades ordenadas, relacionada con la pregunta de tipo ordered choice, y la modalidad sin orden, para las demás. La diferencia radica en la forma en la cual se verifica si la respuesta es correcta. En la ordenada se debe verificar que el orden de las opciones dentro de la respuesta del jugador sea igual que el orden dentro de la respuesta correcta asociada a la pregunta, mientras que en las preguntas que no son ordenadas esto no es necesario.

Kashoot: esta clase es la encargada de contener todas las entidades necesarias para jugar. Es la que contiene el modelo. Esta se actualiza cada vez que un jugador responde, modificando la referencia al jugador actual y aumentando los turnos jugados en cada ronda, a menos que la ronda termine y en tal caso se elimina de la cola de rondas.

6. Excepciones

No se consideró necesario el uso de excepciones debido a que la vista, la cual posee todas las entidades con las cuales el jugador interactúa con la interfaz, prevé todo tipo de caso borde que pudiera romper con el flujo del programa. Con esto se quiere decir que ante la posibilidad de un caso en el cual se deba lanzar una excepción, la vista lo maneja de manera estructurada, controlando el flujo del programa de manera consistente.

Un ejemplo de caso borde que controla la vista es el de los nombres ingresados por los jugadores. Cuando los jugadores deben ingresarlos. En caso de que uno o ambos casilleros quedan vacíos y quieran comenzar a jugar, se tomó la decisión de que no se les permita comenzar el juego hasta que no se ingresen nombres válidos. Esto nuevamente es controlado por la vista.