

Trabajo Práctico 2 — Java

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

Integrantes	Padrones
Betz Rivera, Joaquin	104348
Fontela, Joaquin	103924
Gherzi, Agustin	104330
Podesta, Nicolas	104077

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	3
5. Diagramas de Paquetes	7
6. Diagramas de Estado	8
7. Detalles de implementación	8
8. Excepciones	9
9. Diagramas de secuencia	9

1. Introducción

El presente informe reúne la documentación de la solución del segundo trabajo práctico de la materia Algoritmos y Programación III. El mismo consiste en desarrollar un juego de preguntas y respuestas por turnos en java utilizando los conceptos del paradigma de la orientación a objetos vistos en el curso.

2. Supuestos

La exclusividad en el puntaje parcial sólo se aplica si un jugador obtiene puntos (aunque sea parcialmente) y el resto falla al menos una opción.

Si un jugador elige un tipo de bonificación no puede elegir otro a la vez.

En el caso de que los jugadores empaten luego de la última pregunta no habrá un desempate.

Los jugadores no pueden enviar opciones no ordenadas/agrupadas pero sí respuestas de selección vacías.

Los nombres de los jugadores deben ser distintos.

3. Modelo de dominio

El modelo realizado consta de un objeto que administra las acciones que se llevan a cabo durante el juego y posee las referencias necesarias para poder comunicarse con la mayoría de las entidades del dominio llamado GestorDeJuego.

También consta de un objeto Pregunta genérico compuesto por una Modalidad (que cambiará la forma de calcular el puntaje y puede ser ModalidadClasica, ModalidadPenalidad y ModalidadPuntajeParcial) y una Respuesta que se tomará como correcta y con la que se compararán otros objetos Respuesta provistos por los usuarios para saber cuántas opciones acertó y cuántas falló cada uno.

Las Modalidades dependen de objetos Bonificación que efectuarán las modificaciones correspondientes a los puntajes. Las Respuestas por otro lado pueden ser: RespuestaVerdaderoFalso, RespuestaMultipleChoice, RespuestaOrdererChoice o RespuestaGroupChoice. Todas contienen los atributos correspondientes para almacenar los datos usados.

4. Diagramas de clase

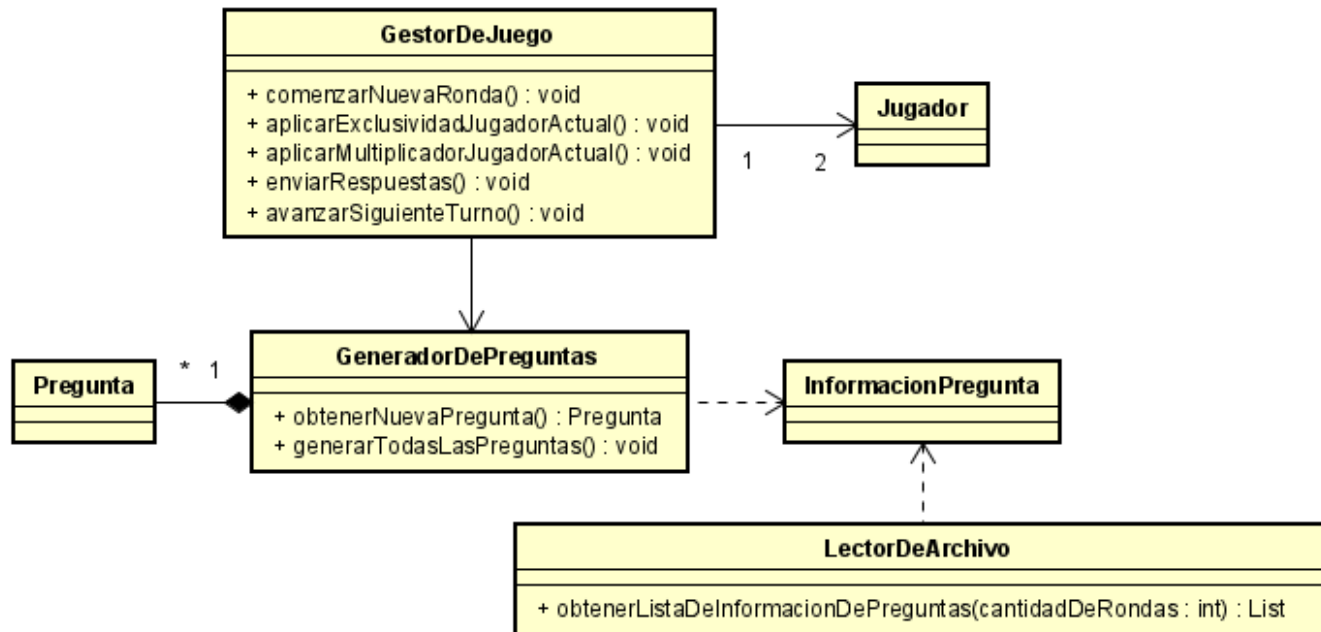


Figura 1: Relaciones de GestorDeJuego

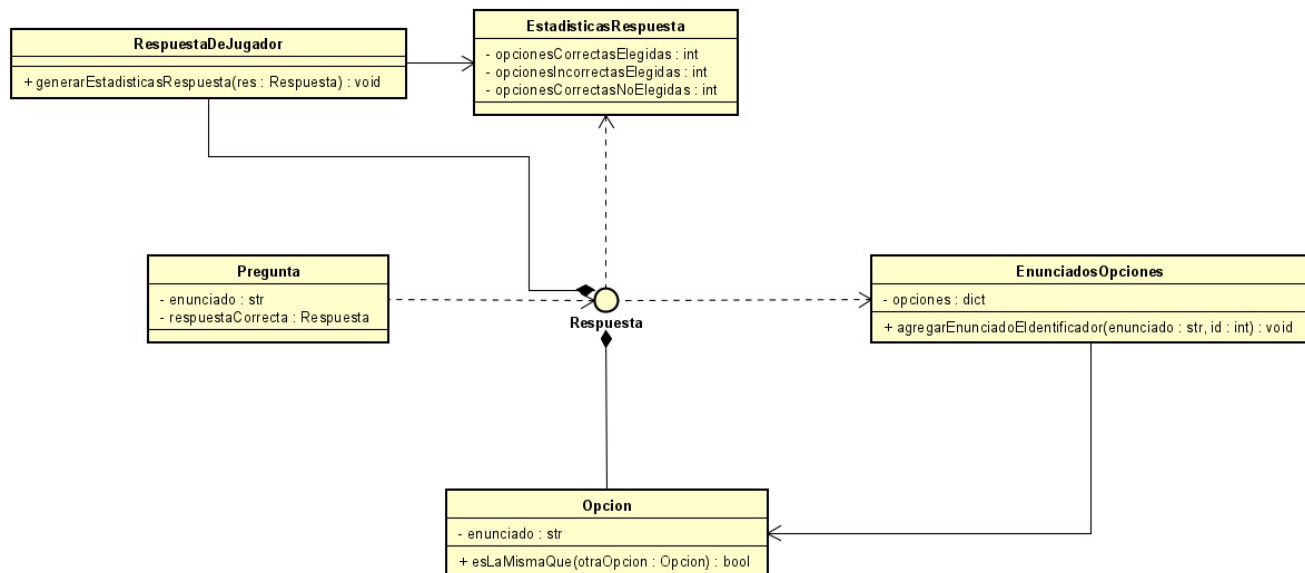


Figura 2: Relaciones de Respuesta

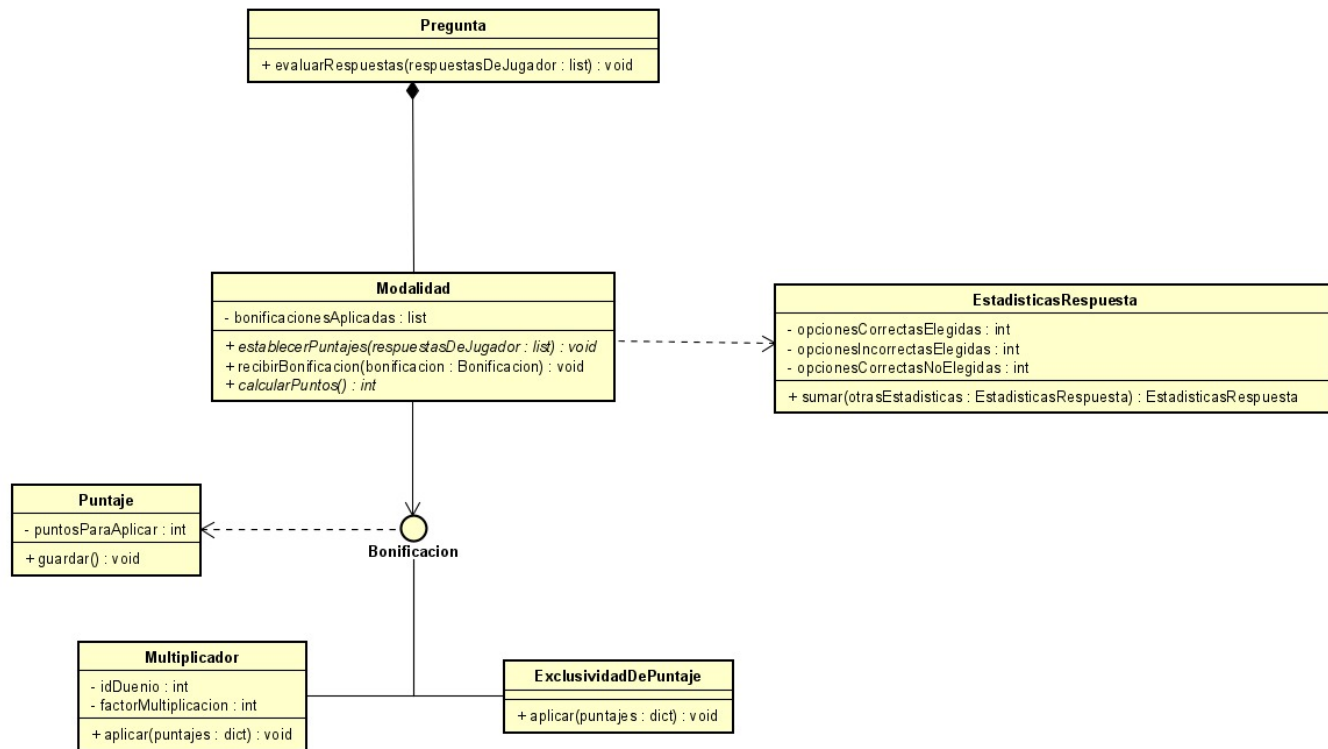


Figura 3: Diagrama de clases Pregunta.

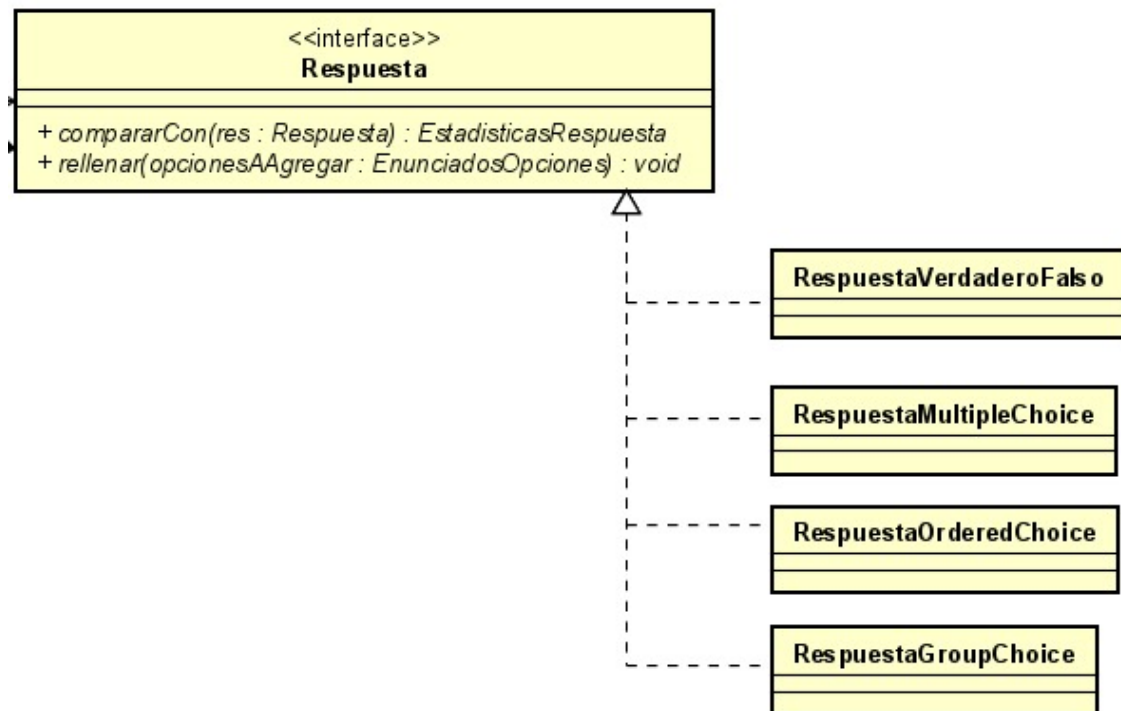


Figura 4: Diagrama de clases Respuesta.

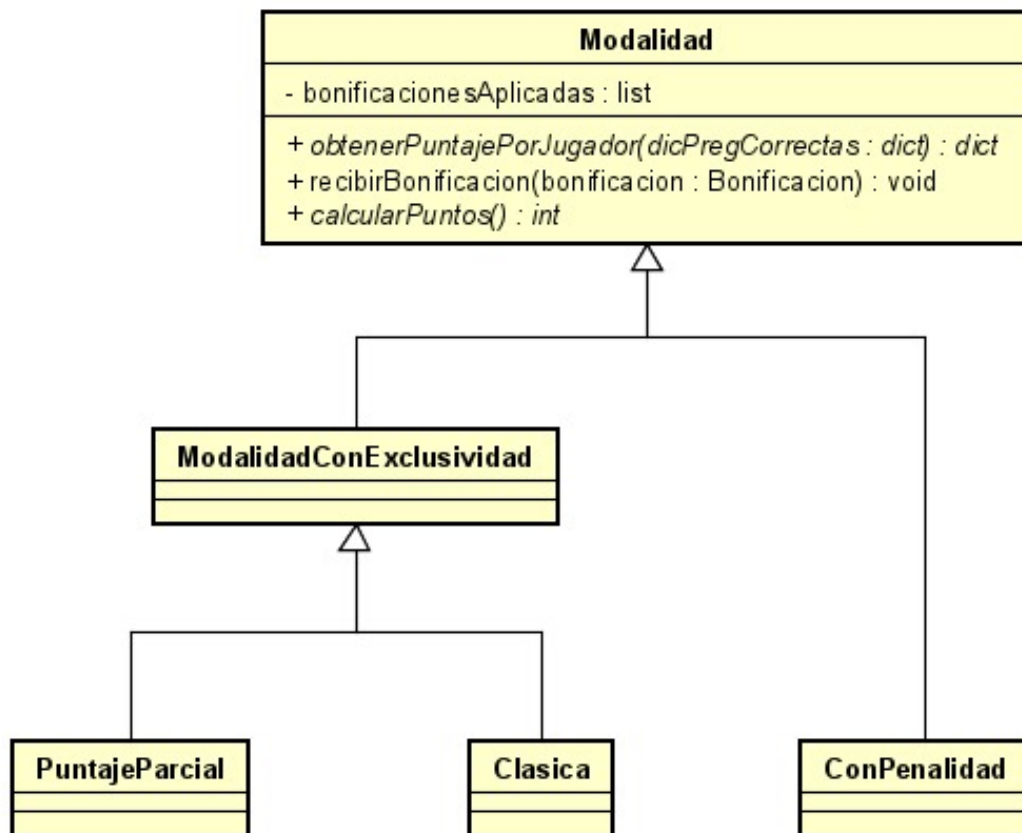


Figura 5: Diagrama de clases modalidad.

5. Diagramas de Paquetes

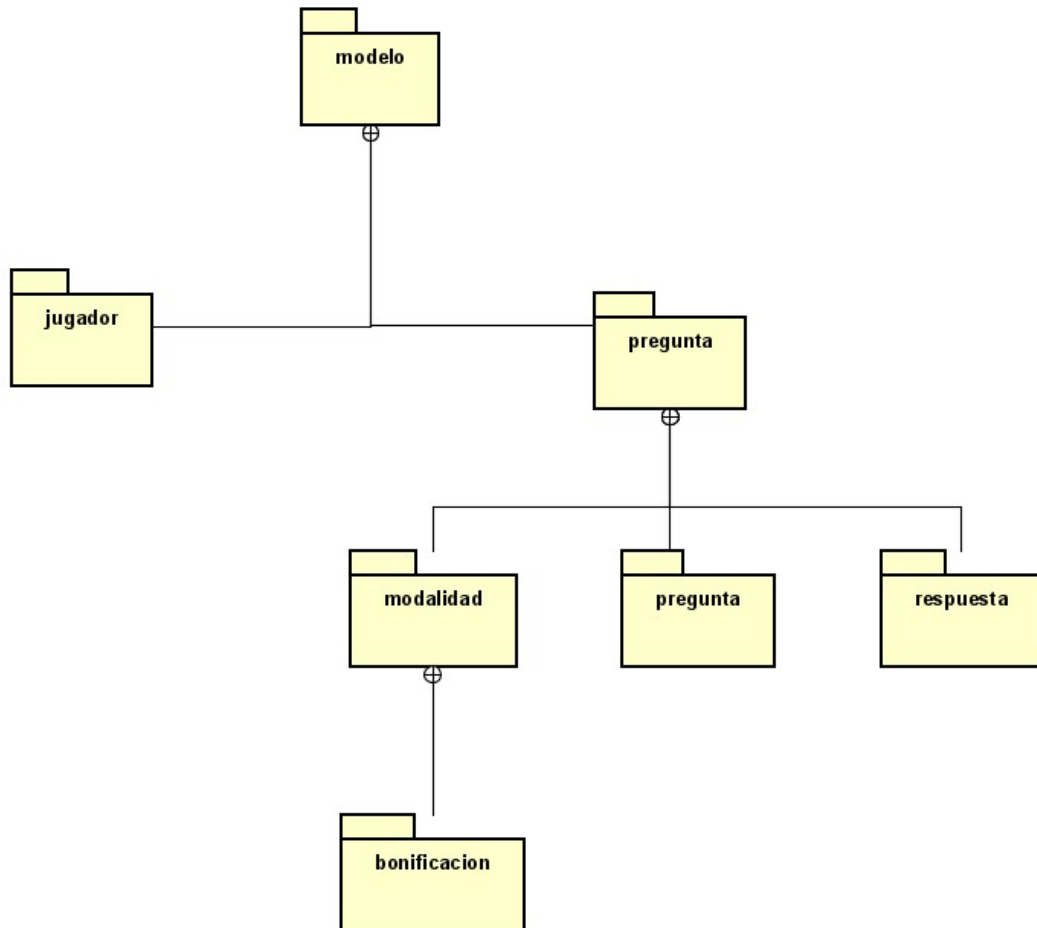


Figura 6: Diagrama de Paquetes del modelo.

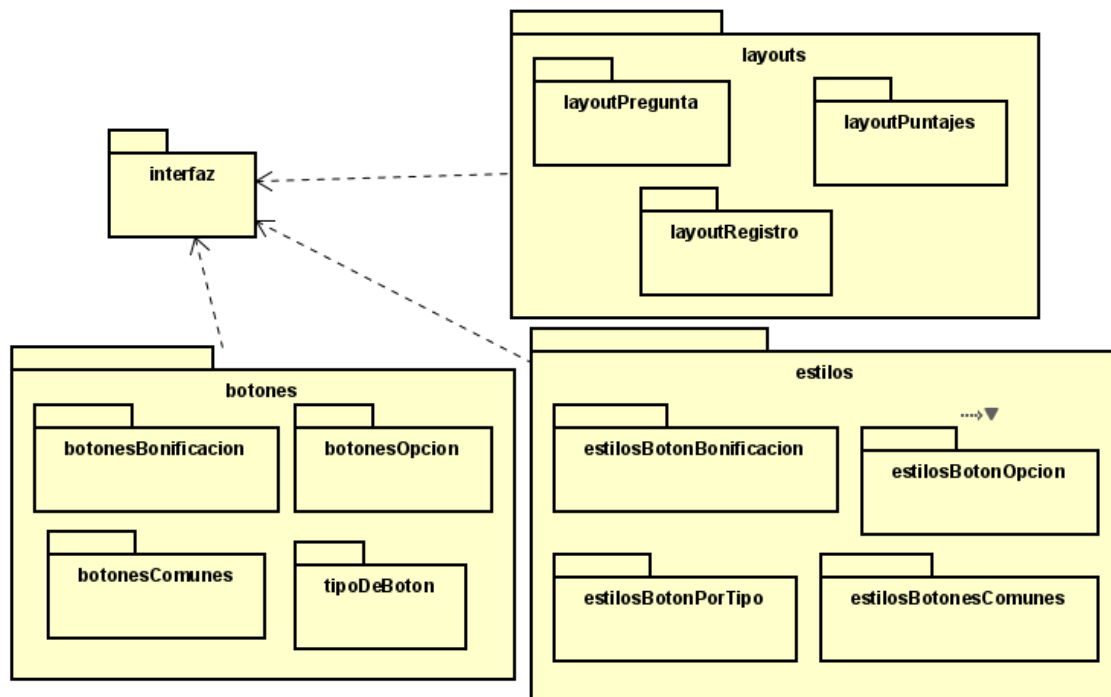


Figura 7: Diagrama de Paquetes de la interfaz.

6. Diagramas de Estado

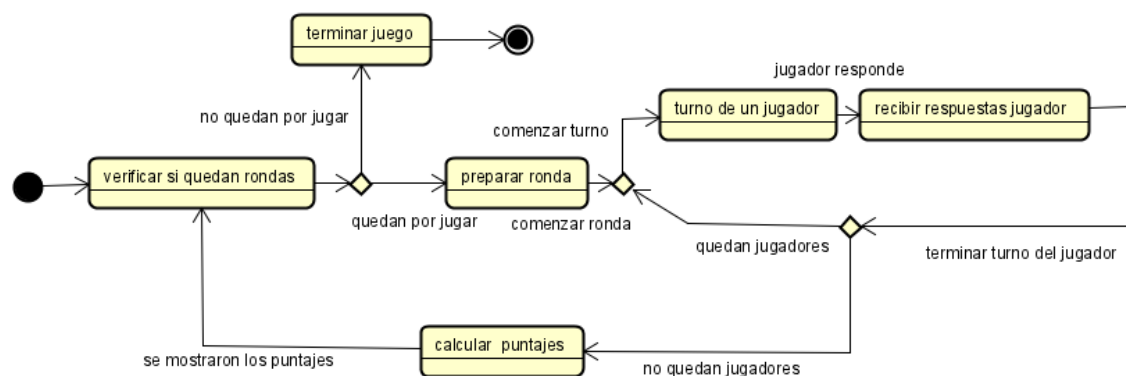


Figura 8: Transición durante el juego.

7. Detalles de implementación

En esta sección nos gustaría explicar el procedimiento por el cual se obtienen las puntuaciones correspondientes a las respuestas dadas por los jugadores para su posterior asignación. Explicaremos qué sucede con las respuestas luego de que los jugadores responden y éstas se almacenan para simplificar su comprensión.

Primeramente el GestorDeJuego llama al objeto Pregunta (cuya referencia se encuentra en una

variable del gestor para saber cuál es la pregunta actual) con su método evaluarRespuestas() al que se le pasa una lista de objetos RespuestaDeJugador que funciona para ligar la entidad respuesta al jugador que la respondió. Este método primero compara las Respuestas de cada Jugador con el objeto Respuesta que tiene almacenado en su atributo RespuestaCorrecta con el método "generarEstadisticasRespuesta()" (perteneciente a la clase Respuesta) para así obtener objetos EstadisticasRespuesta, que tendrán almacenados los datos de la cantidad de opciones que acertó y que falló el Jugador correspondiente.

Posteriormente Pregunta llama a establecerPuntajes() de su Modalidad pasándole la lista de RespuestaDeJugador que ahora tendrá almacenada las estadísticas; es así como dependiendo de la Modalidad (Clasica, Puntaje Parcial o Penalidad) se calcularán de forma distinta las puntuaciones. Además es en este método donde se aplicarán las Bonificaciones correspondientes y se guardarán directamente los puntos a los jugadores ya que respuestaDeJugador tiene una referencia a ellos.

8. Excepciones

Si el archivo .json no contiene al menos 15 preguntas validas para el juego, el programa lanza una excepción ya que tiene la opción de poder jugarse con un maximo de 15 preguntas.

Utilizamos una excepción para terminar el juego.

9. Diagramas de secuencia

En el siguiente diagrama de secuencia se mostrara la secuencia explicada en la seccion "detalles de implementacion".

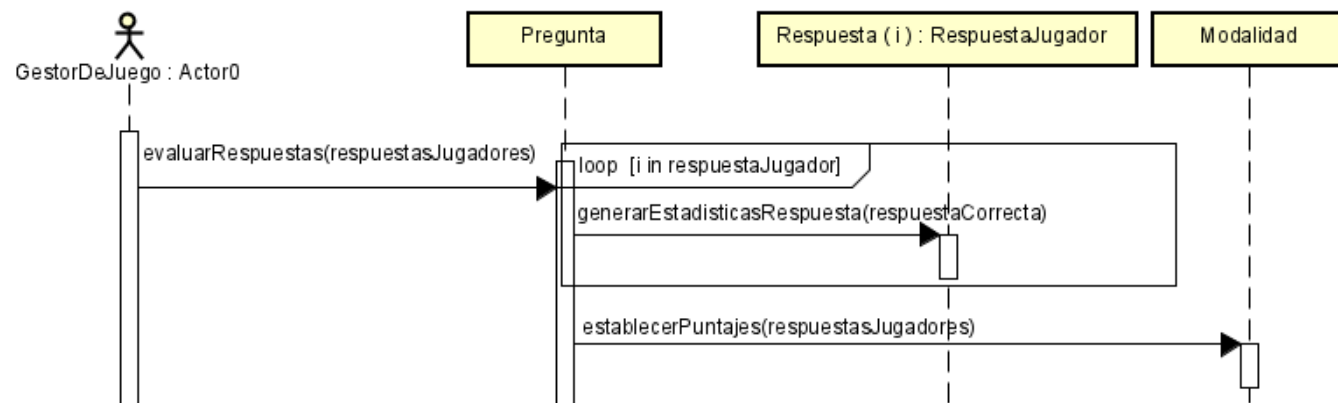


Figura 9: Diagrama de secuencia de la obtención y asignación de puntaje.

En el siguiente diagrama se ve, usando ModalidadClasica como ejemplo, como es el funcionamiento del ObtenerPuntajePorJugador de las modalidades.

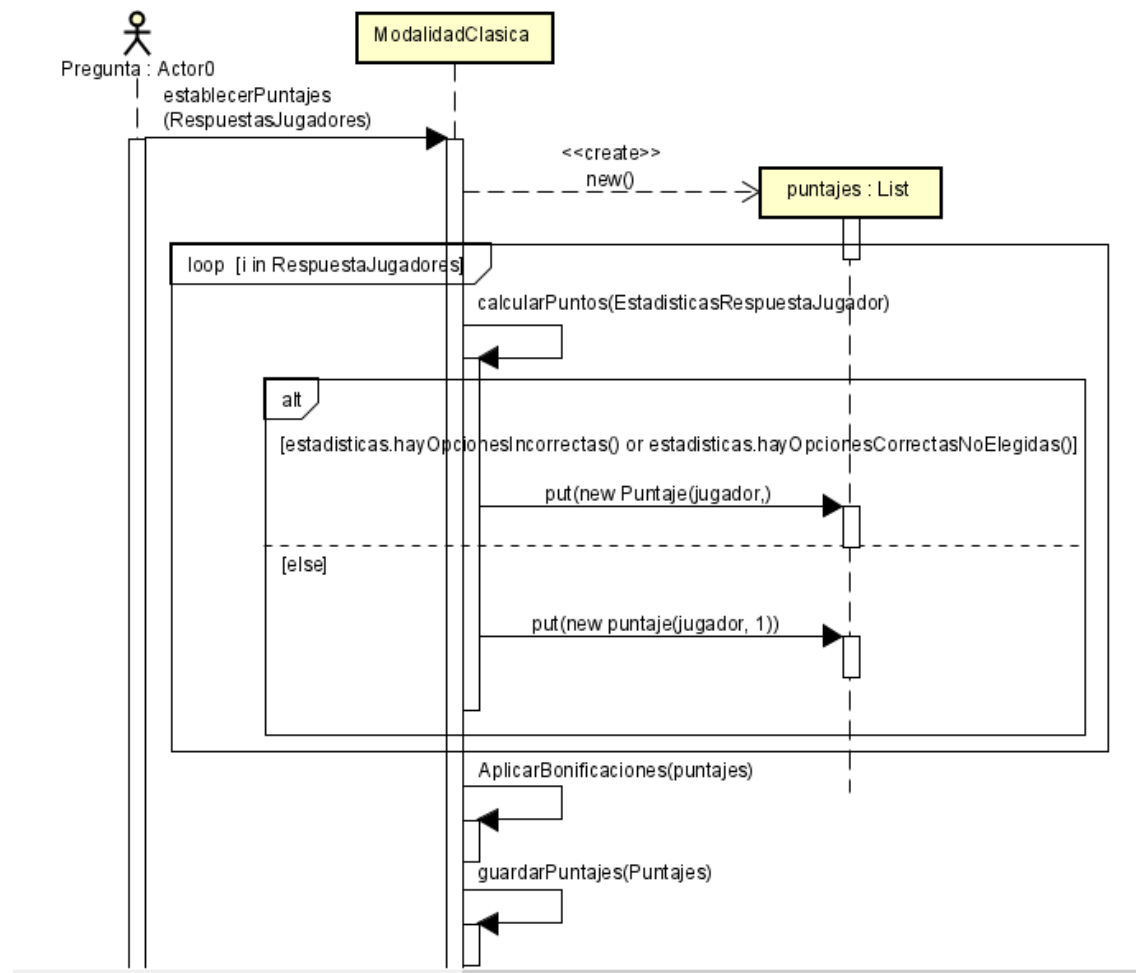


Figura 10: Diagrama de secuencia de la obtencion de los puntajes por la modalidad