

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III
Curso 1
Primer cuatrimestre de 2024

Grupo 05	
Alumno:	Número de padrón:
Alperovich, Denisse	110266
Danica More	109502
Cruceño, Juan Ignacio	110286
Apaza, Axel	105006

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	2
5. Detalles de implementación	3
5.1. ¿Herencia o delegación?	3
5.2. Principios de diseño	3
5.3. Patrón de diseño	3
6. Excepciones	4
7. Diagramas de secuencia	4

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 utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

En la especificación no estaban contempladas la modalidad específica del juego mas allá de ser de preguntas y respuestas por lo que decidimos que el juego tenga partidas y que cada una de estas contengan diez rondas, cada ronda corresponde a una pregunta. Además decidimos que el juego pueda ser jugado por N jugadores.

3. Modelo de dominio

Para la realización de la entrega 1 se utilizaron las siguientes clases: Modalidad (subclases: VFClasico, MultipleChoiceClasico, VFConPenalidad y MultipleChoiceConPenalidad), Jugador y Pregunta.

Decidimos que Modalidad tenga las subclases VF y MultipleChoice y Clásico y con penalidad ya que a la hora de asignar los puntajes se manejan de distinta manera pero la pregunta nunca cambia.

No vimos necesario implementar una clase Respuesta ya que vimos que de hacerlo, la misma quedaría anémica y se puede resolver de otra manera mejor.

4. Diagramas de clase

A continuación, se puede ver la interacción entre las clases a través de 2 diagramas de clase.

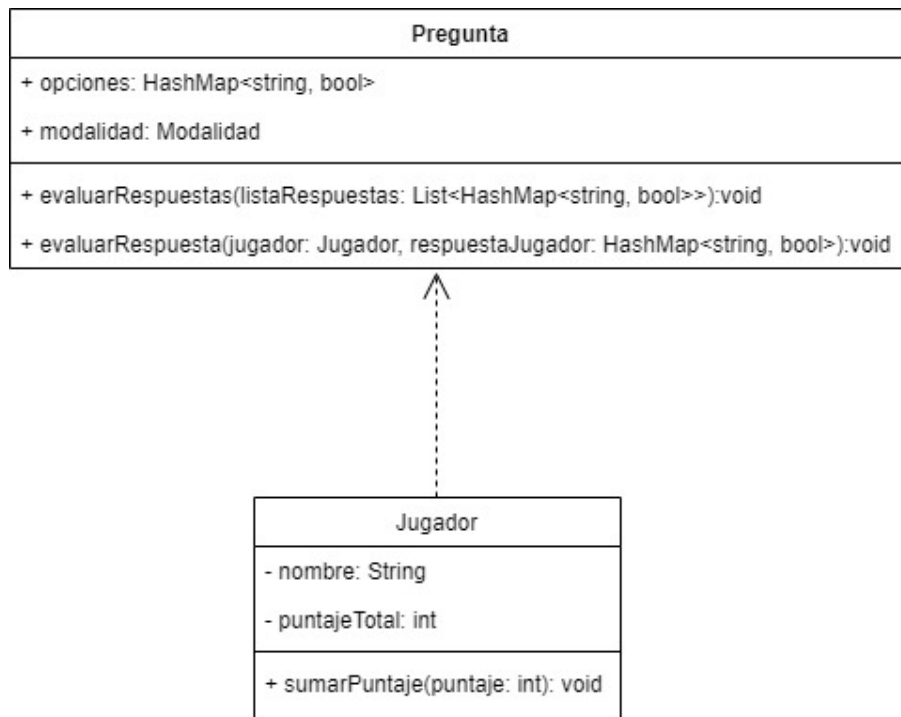


Figura 1: Diagrama 1 del Juego.

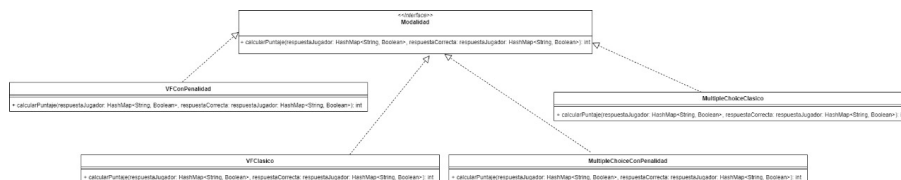


Figura 2: Diagrama 2 del Juego.

5. Detalles de implementación

5.1. ¿Herencia o delegación?

Para las clases decidimos usar delegación y no herencia ya que a nuestro parecer los métodos no eran estrictamente iguales y era conveniente delegar para tomar lo necesario y lo demás dejarlo a cada subclase. Esta decisión implica escribir cada implementación pero facilita aclarar la responsabilidad y marcar la diferencia de cada subclase con respecto a la otra.

5.2. Principios de diseño

En esta entrega usamos Single Responsibility Principle para que pregunta como clase no deba encargarse de muchas tareas y la razón de cambio sea una.

5.3. Patrón de diseño

En esta entrega utilizamos Strategy de patrón de diseño para separar el comportamiento de evaluación de la respuesta y lo pasamos al objeto modalidad.

6. Excepciones

En esta entrega no vimos necesaria la creación de excepciones.

7. Diagramas de secuencia

En el diagrama de secuencia se pide a una pregunta de verdadero/falso clásico que evalúe una respuesta de un jugador y le asigne puntaje.

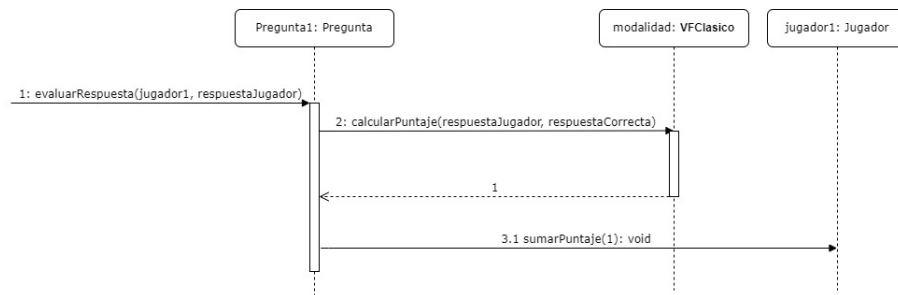


Figura 3: Diagrama de secuencia pregunta VF clásica.

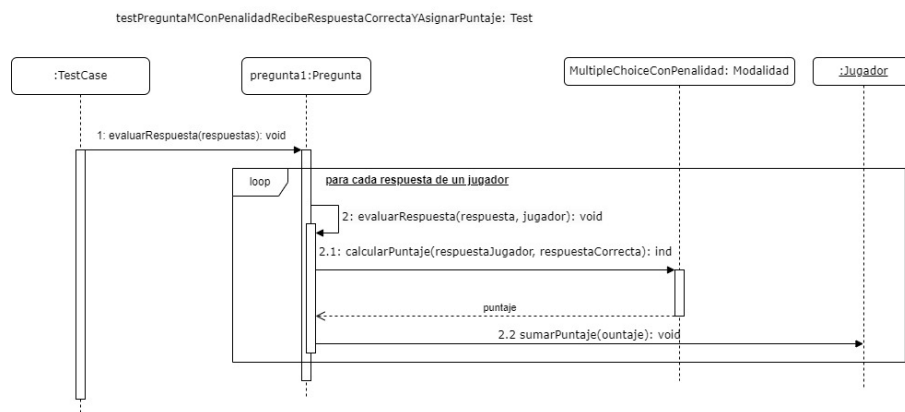


Figura 4: Diagrama de secuencia pregunta VF con penalidad.

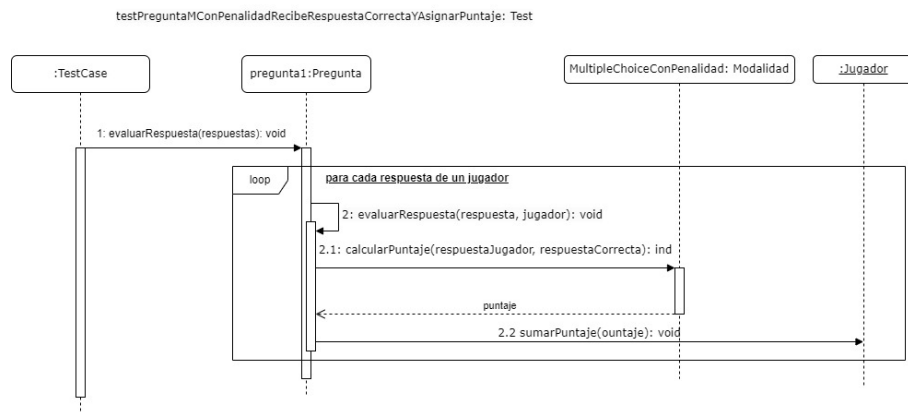


Figura 5: Diagrama de secuencia pregunta Multiple Choice con penalidad.