

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III
Curso 2
Primer cuatrimestre de 2020

Alumna/o	Padrón	Email
Besednjak, Mario	103287	mbesednjak@fi.uba.ar
Goyzueta, Alan	102988	agoyzueta@fi.uba.ar
Lopez Giles, Gisela	104842	glopezg@fi.uba.ar
Paredes, Luis	104851	lparedesr@fi.uba.ar
Villordo, Micaela	103828	mvillordo@fi.uba.ar

Índice

1. Supuestos	2
2. Diagramas de Clase	3
2.1. Modelo General del Juego	3
2.2. Modelo Pregunta	3
2.3. Modelo Multiplicador	5
2.4. Modelo Exclusividad	5
3. Diagramas de Secuencia	6
3.1. Un jugador responde	6
3.2. Fin de Ronda	7
3.3. Cargar Preguntas	7
4. Diagrama de Paquetes	8
5. Diagramas de Estado	8
6. Detalles de implementación	8
7. Excepciones	9

1. Supuestos

- Todas las preguntas tienen por lo menos una respuesta correcta.
- Multiple Choice: Puede tener como máximo todas las respuestas correctas.
- Multiple Choice Penalidad: Suma puntos por respuesta correcta seleccionada, siempre que no se haya seleccionado ninguna incorrecta.
- Multiple Choice Penalidad: En caso de no seleccionar ninguna opción el jugador sumará 0 puntos.
- Multiple Choice Parcial: se considera que un jugador ha contestado incorrectamente sólo cuando no ha elegido ninguna respuesta correcta.
- Multiple Choice Parcial: No se sumarán puntos a ningún jugador si ambos han elegido al menos una respuesta correcta.
- Group Choice: Deben tener al menos una opción de cada grupo.
- Group Choice: Todas las opciones deben pertenecer a un grupo.
- Fin de turno por tiempo: Se tomará lo que tiene marcado hasta el momento.
- Todas las preguntas tendrán el mismo tiempo máximo establecido.
- El tiempo máximo establecido es de 20 segundos para responder.
- No se puede iniciar el juego si los jugadores no completan los campos nombres y se lanzará un mensaje de error por cada vez que se intente la acción.

2. Diagramas de Clase

2.1. Modelo General del Juego

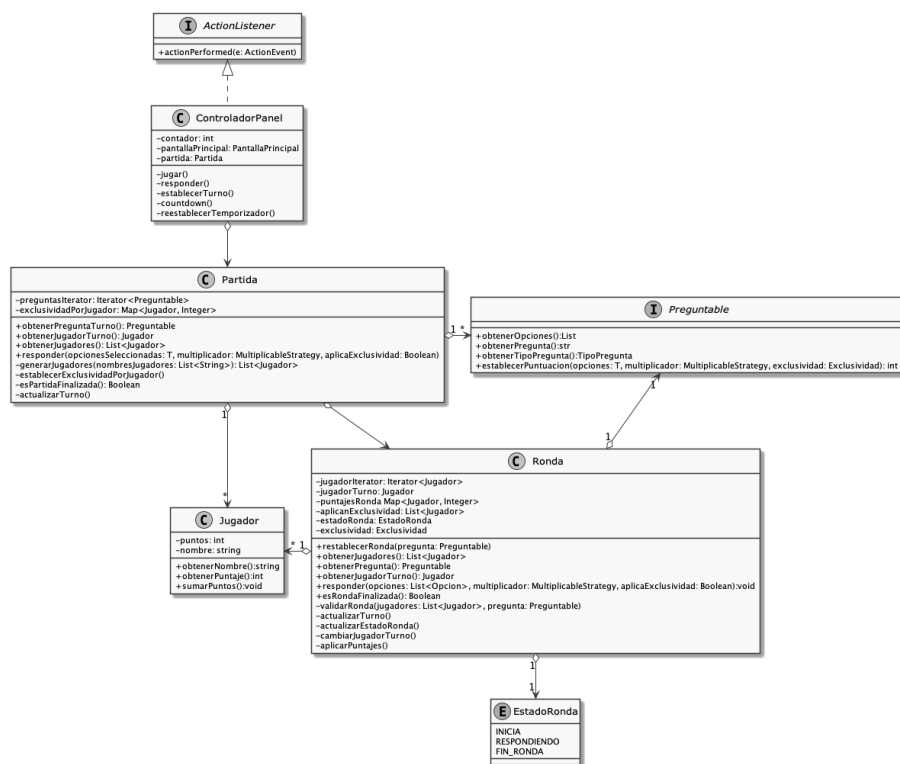


Figura 1: Modelo General del Juego.

2.2. Modelo Pregunta

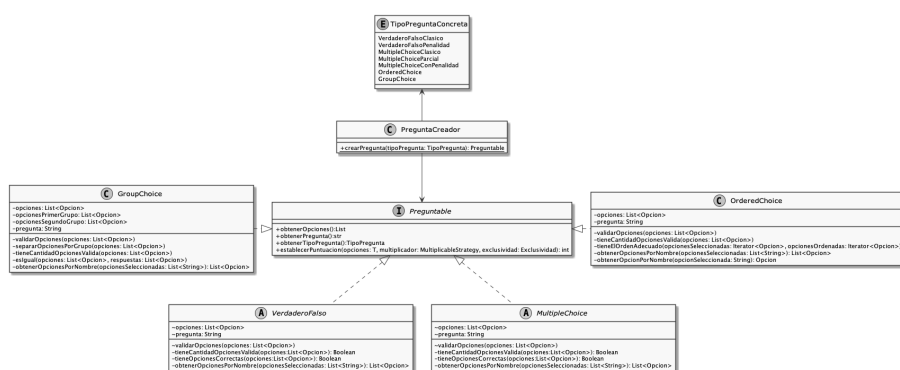


Figura 2: Modelo Pregunta.

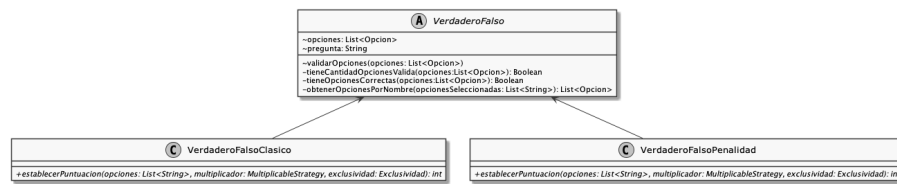


Figura 3: Modelo Verdadero Falso.

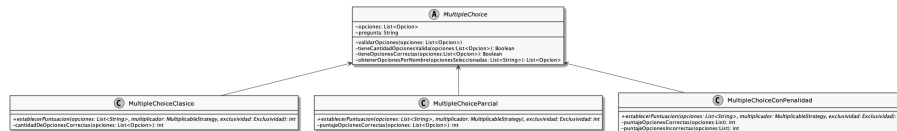


Figura 4: Modelo Multiple Choice.

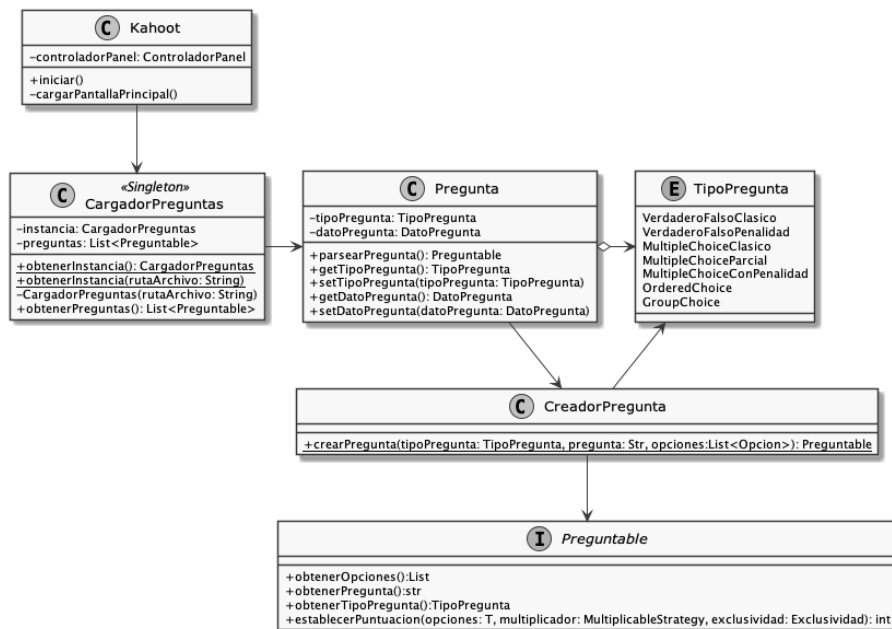


Figura 5: Modelo Cargador Pregunta.

2.3. Modelo Multiplicador

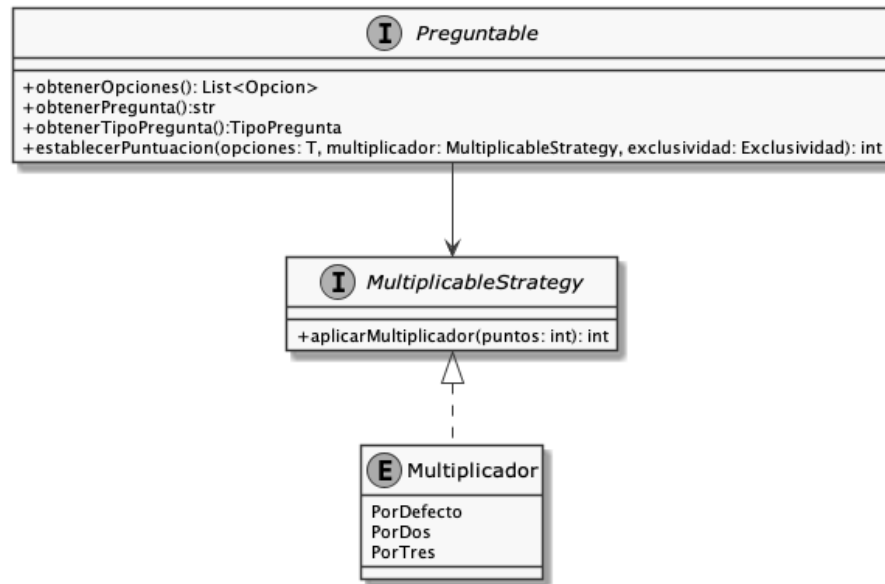


Figura 6: Diagrama Modelo Multiplicador.

2.4. Modelo Exclusividad

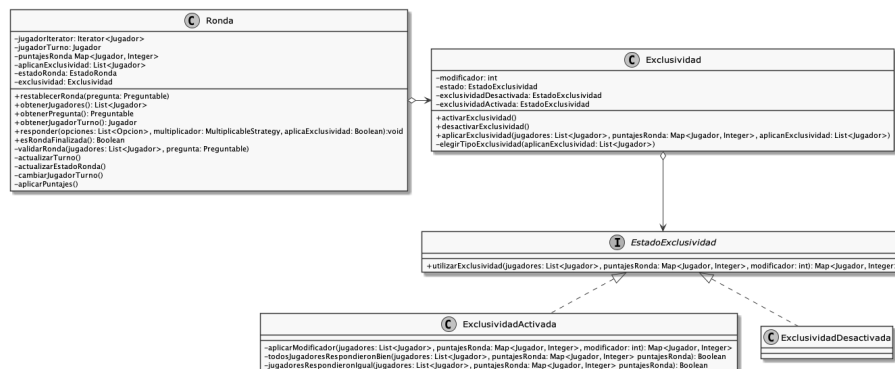


Figura 7: Diagrama Modelo Exclusividad.

3. Diagramas de Secuencia

3.1. Un jugador responde

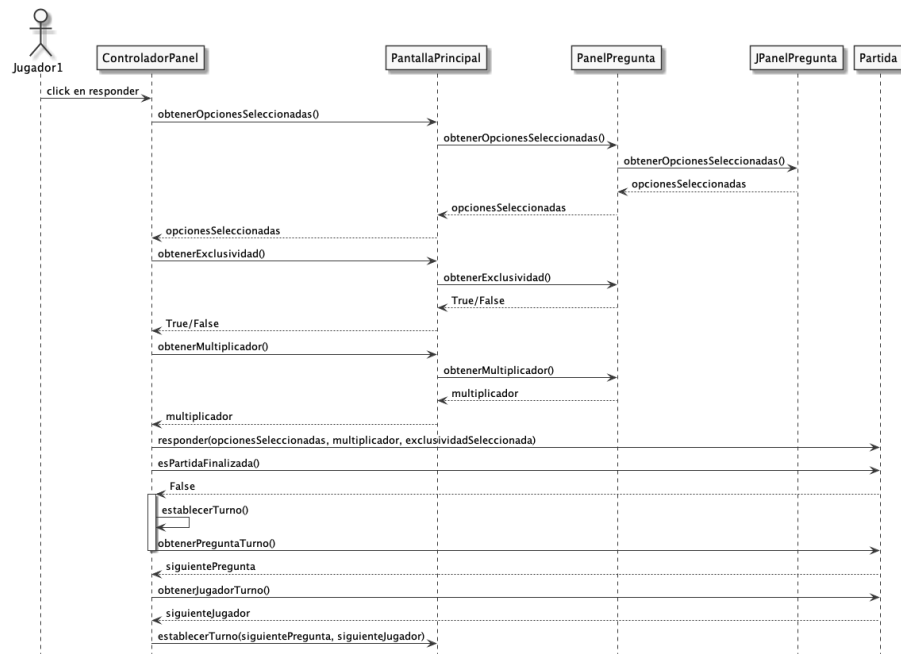


Figura 8: Un jugador responde UI.

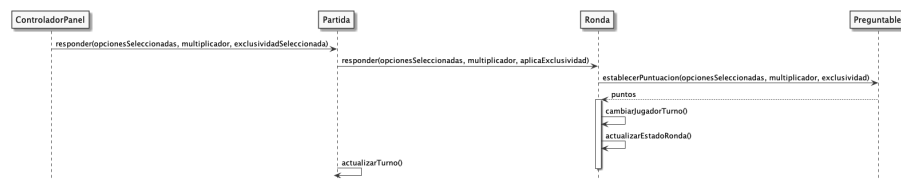


Figura 9: Un jugador responde sin multiplicador y sin exclusividad Backend.

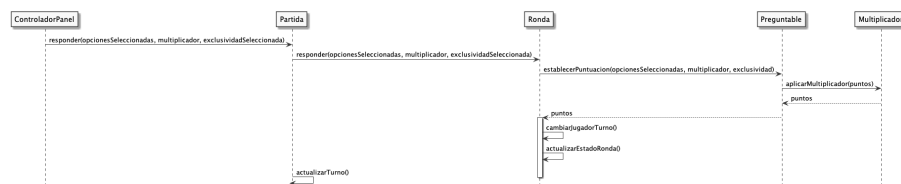


Figura 10: Un jugador responde una pregunta con penalidad con Multiplicador Backend.



Figura 11: Un jugador responde una pregunta sin penalidad con Exclusividad Backend.

3.2. Fin de Ronda

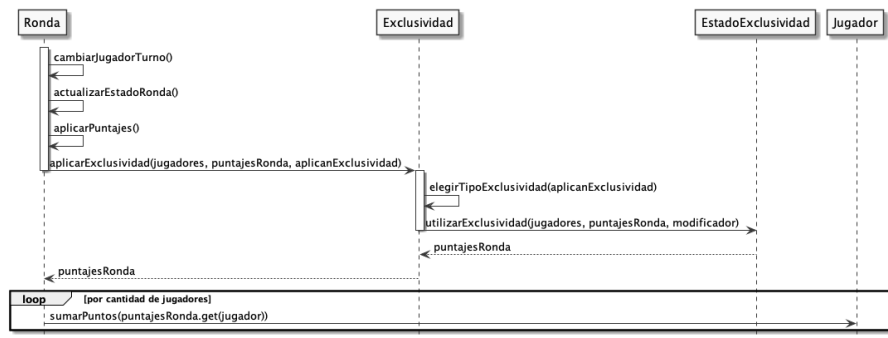


Figura 12: Fin de Ronda.

3.3. Cargar Preguntas

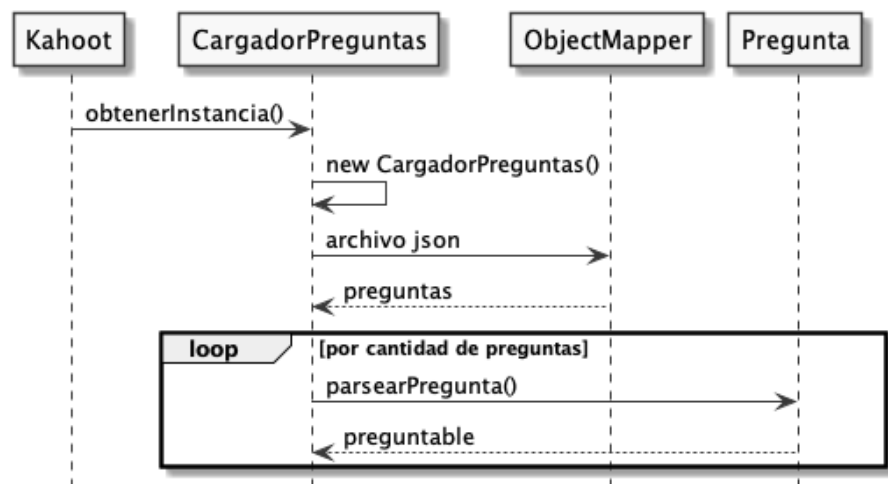


Figura 13: Cargar Preguntas.

4. Diagrama de Paquetes

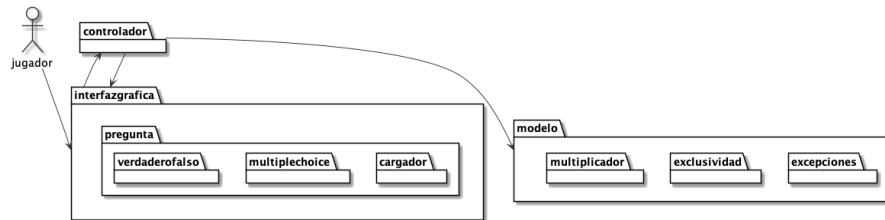


Figura 14: Diagrama Paquetes.

5. Diagramas de Estado

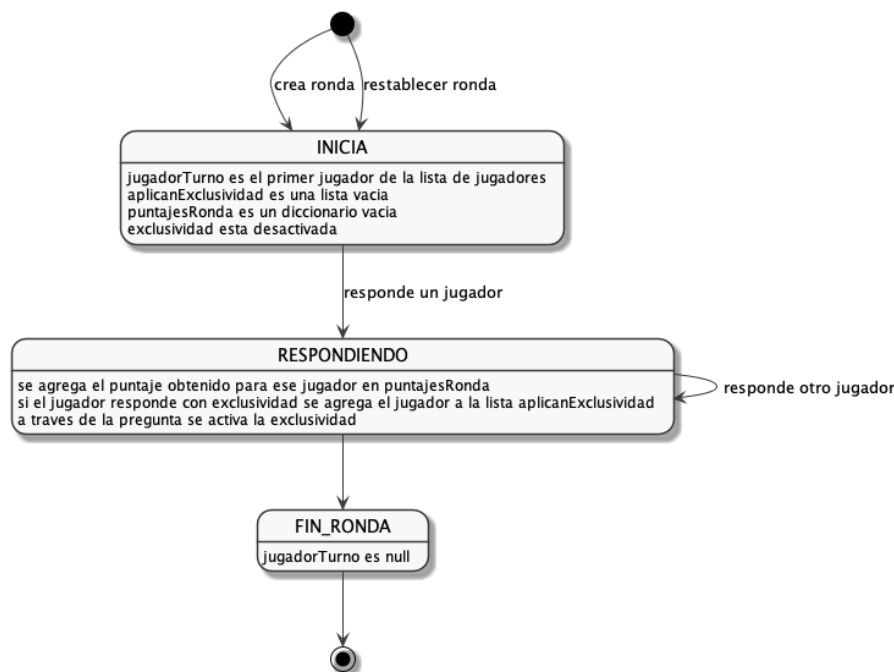


Figura 15: Diagrama Ronda.

6. Detalles de implementación

Para la implementación de este trabajo práctico utilizamos 3 patrones de diseño:

- Singleton
- Factory Method
- Strategy

Utilizamos el patrón Singleton en la carga de las preguntas desde el archivo, ya que nos permite tener una única instancia de las preguntas cargadas a la cual podemos acceder desde cualquier objeto de nuestra aplicación, permitiéndonos así evitar el traspaso de esta información de objeto en objeto.

Utilizamos el patron Factory Method para crear las preguntas. Decidimos utilizarlo ya que nos permite estandarizar la creacion de los objetos que implementan la interfaz Preguntable, abstrayendo a quien necesite crear estos objetos el como hacerlo.

Utilizamos el patron Strategy para implementar Multiplicador dado que identificamos a los diferentes multiplicadores como estrategias para modificar el puntaje. Al utilizar este patrón nos permitió aumentar la cohesión y minimizar el acoplamiento, ya que abstraemos al objeto cliente del conocimiento de cómo aplicar el multiplicador en cada caso.

Utilizamos Log4j para dejar registro en un log de aquellos fallos silenciosos que nos puedan servir, por ejemplo:

- Carga de preguntas con datos inválidos.
- No se puede cargar una imagen.
- No se puede cargar el audio.

Todas estas fallas no interrumpen la ejecución, pero se registran en el log.

En nuestro modelo Ronda es quien se encarga de llevar los puntajes parciales por pregunta y aplicarlos al finalizar cada ronda. Esto nos permite que tambien se encargue de llamar a aplicar exclusividad cuando todos los jugadores respondieron o haya pasado su turno. Si bien Ronda llama a exclusividad, queda exenta de la responsabilidad de aplicarla finalmente, ya que exclusividad es quien toma la desición.

7. Excepciones

ParametrosInvalidosExcepcion Excepcion que se lanza cuando ocurre uno de los siguientes casos:

- Cantidad de opciones no se corresponde con el tipo de pregunta.
- Si la lista de opciones no tiene una cantidad valida de opciones correctas.
- Si se desea iniciar una ronda con lista de jugadores vacia.
- Si se desea iniciar una ronda con pregunta en null.

TipoPreguntaNoImplementadaException Esta excepcion se lanza al querer crear un Preguntable de un tipo no implementado.

NombresInvalidosExcepcion Esta excepcion se lanza al querer iniciar la partida con los nombres vacios.