

Trabajo práctico 2

Algoritmos y Programación 3 75.03 Fiuba

Integrantes:

Jonathan Marcelo Huata

Alejandro Abraham Osco Cabrera

Samuel A. Camacho Ch.

Mateo J. Ausqui

Marco A. Manco Mantilla

Informe

El presente es un informe preliminar en el que se muestra una aproximación para la creación de AlgoHoot, un juego por turnos en el que los jugadores deberán responder preguntas.

Supuestos

El puntaje de un jugador no podrá ser negativo.

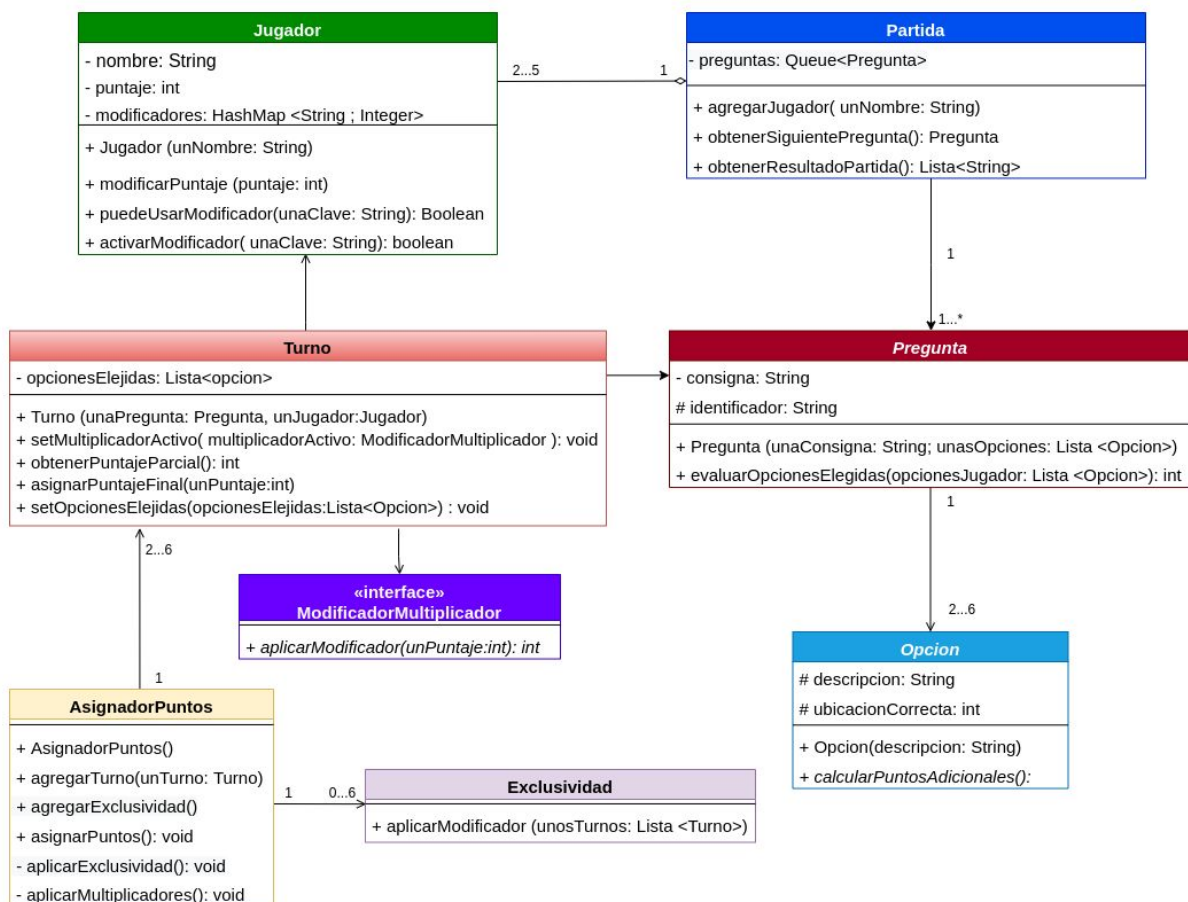
Siempre que no se reciba una respuesta de un jugador, sin importar el tipo de pregunta, el cambio al puntaje de dicho jugador será 0.

Para la exclusividad en preguntas de puntaje parcial se tomará como correcta la respuesta que más puntaje tenga.

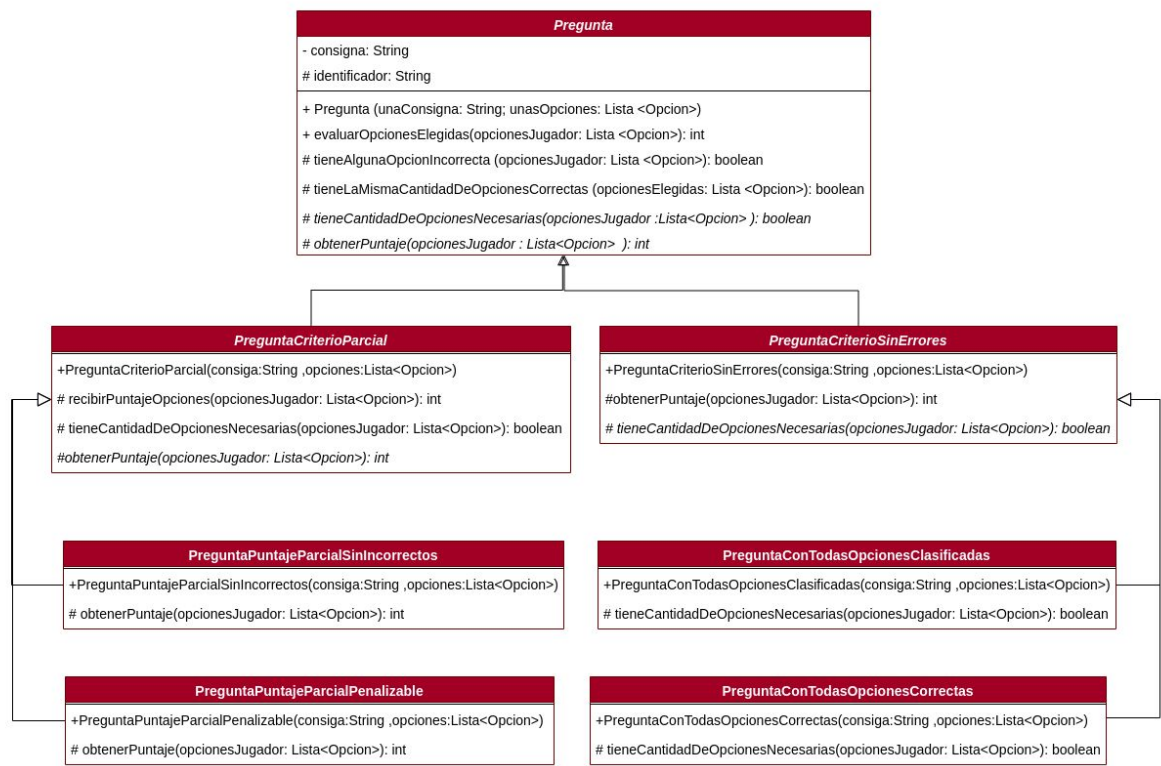
El archivo de pregunta siempre tendrá al menos un tipo de pregunta.

Diagramas de clases

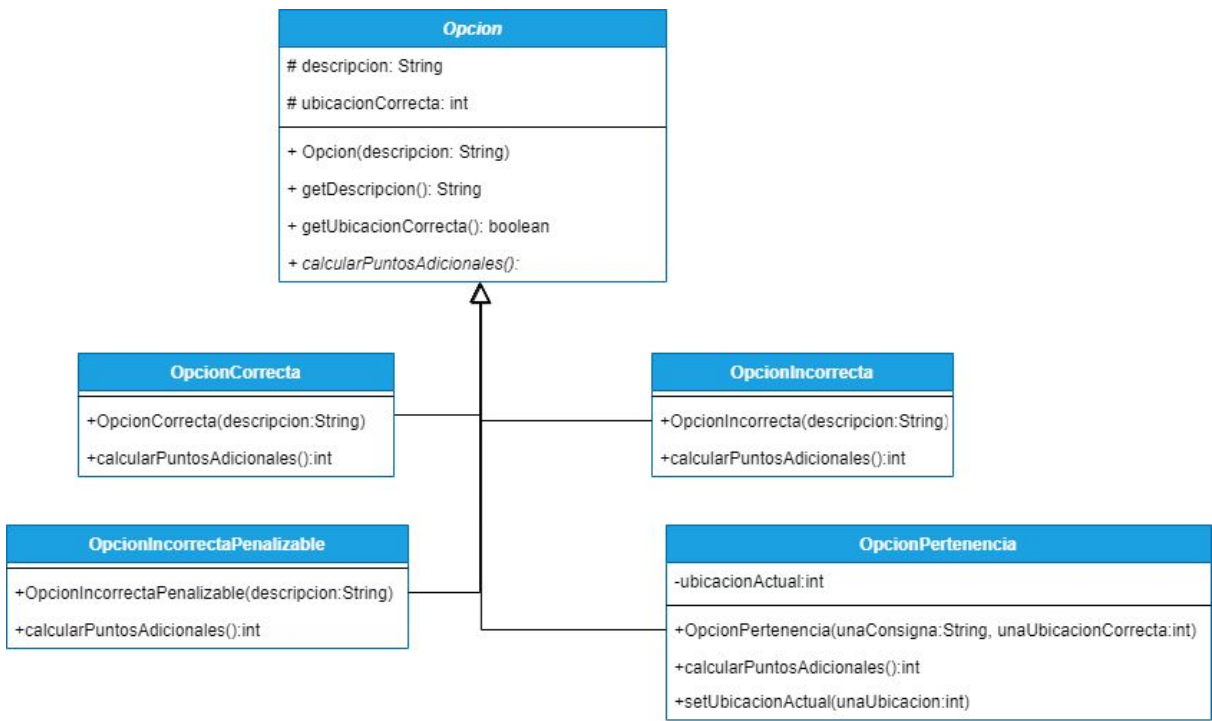
1º Diagrama, relación de clases de funcionamiento interno



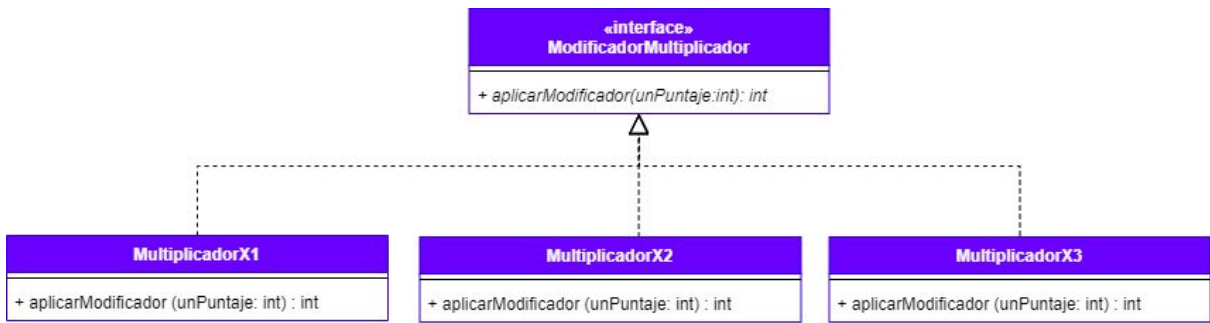
2º Diagrama de herencia de Pregunta:



3º Diagrama de Herencia de Opcion



4º Diagrama clases que implementan la interfaz ModificadorMultiplicador:



5º Diagramas de recuperación de datos de un archivo Json:

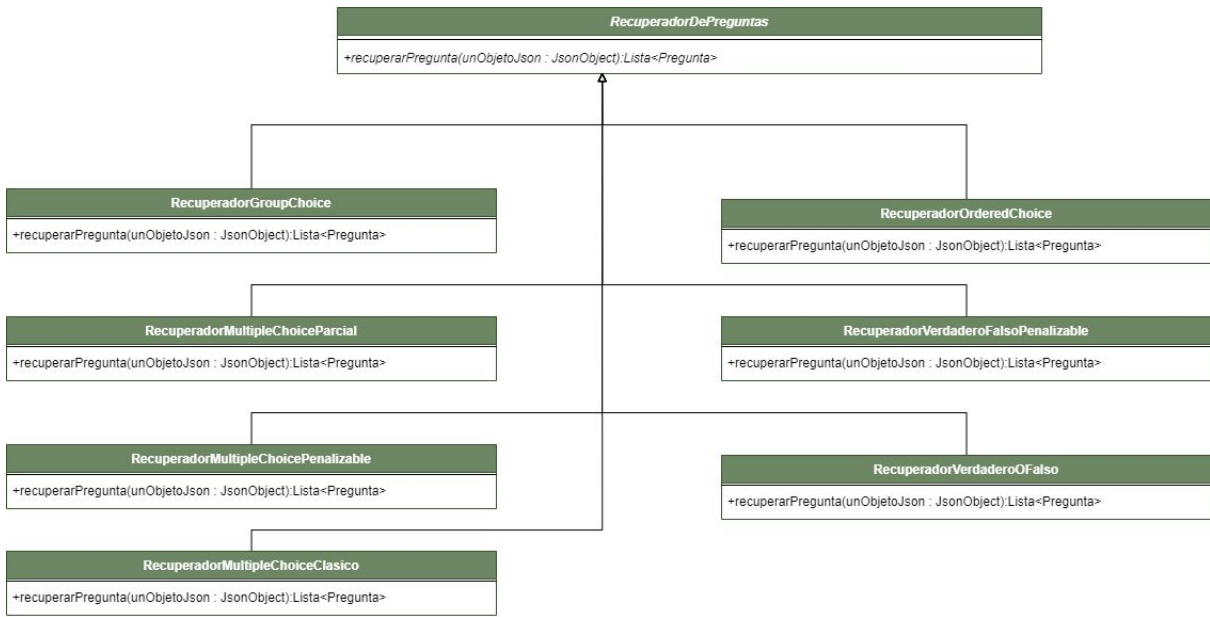
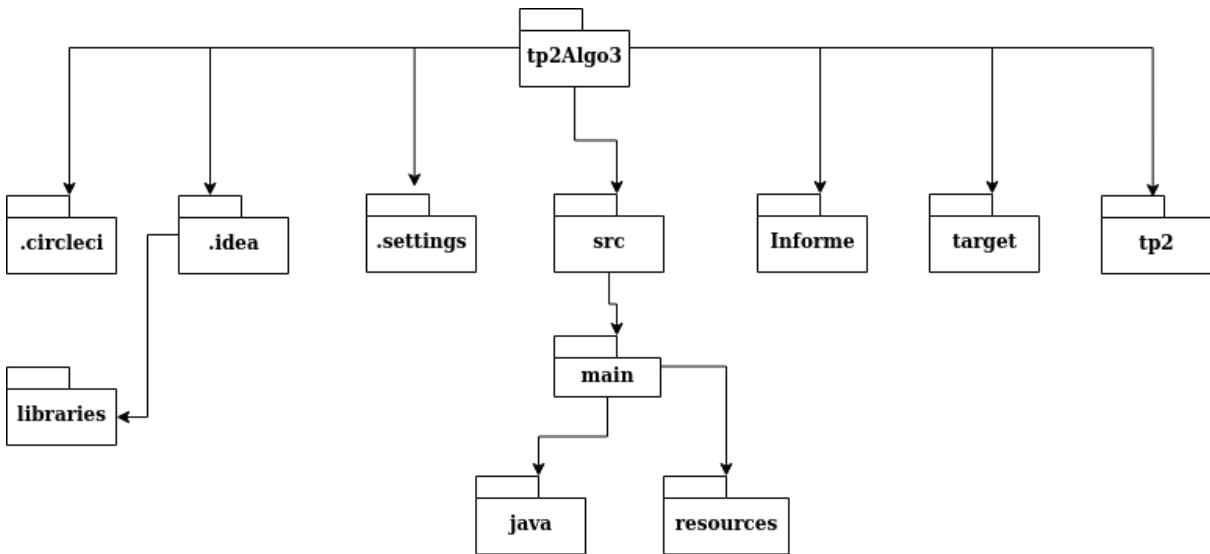
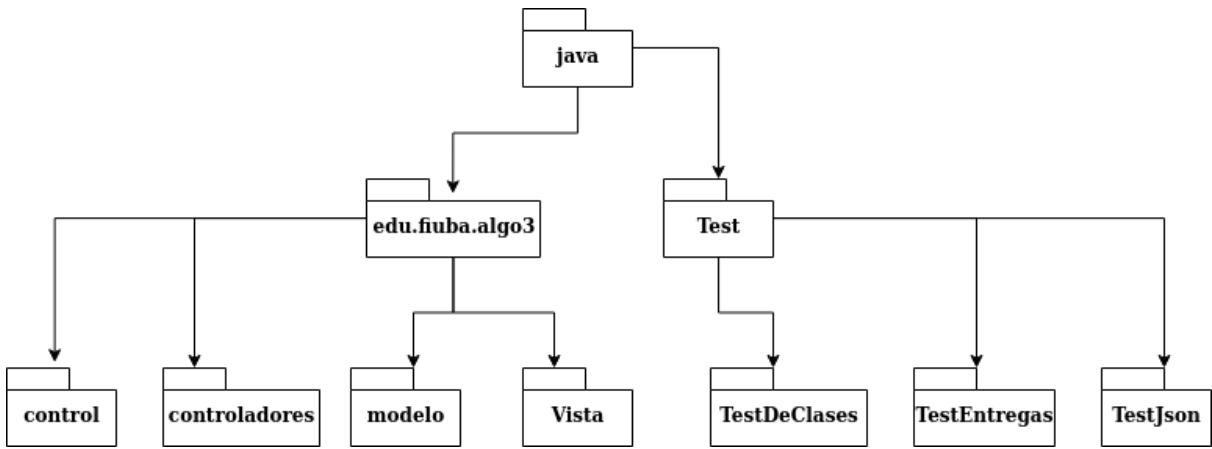


Diagrama de paquetes

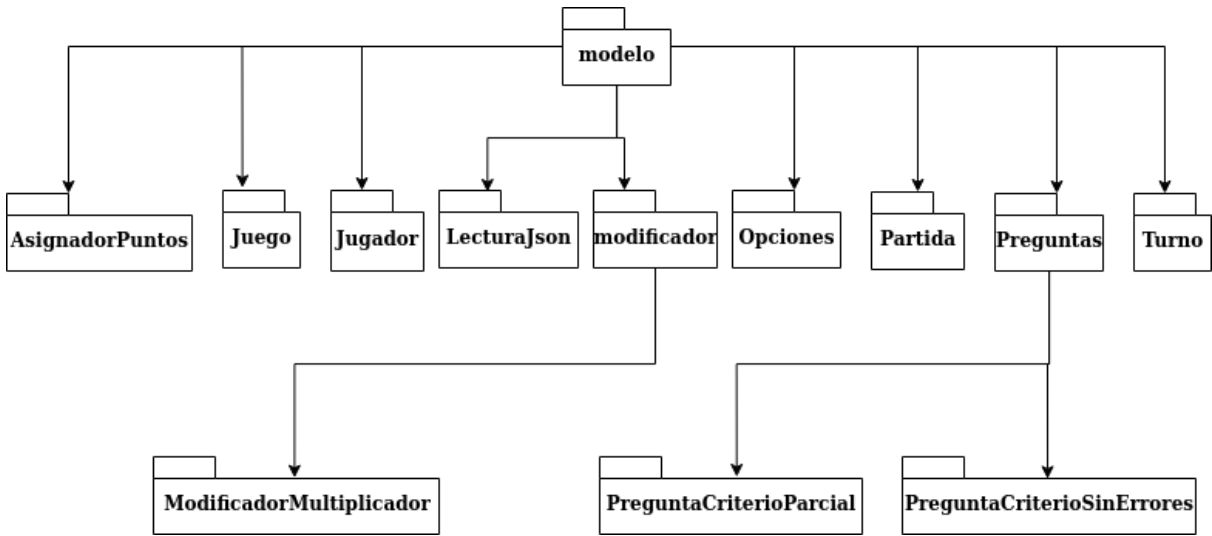
Paquetes Mayores



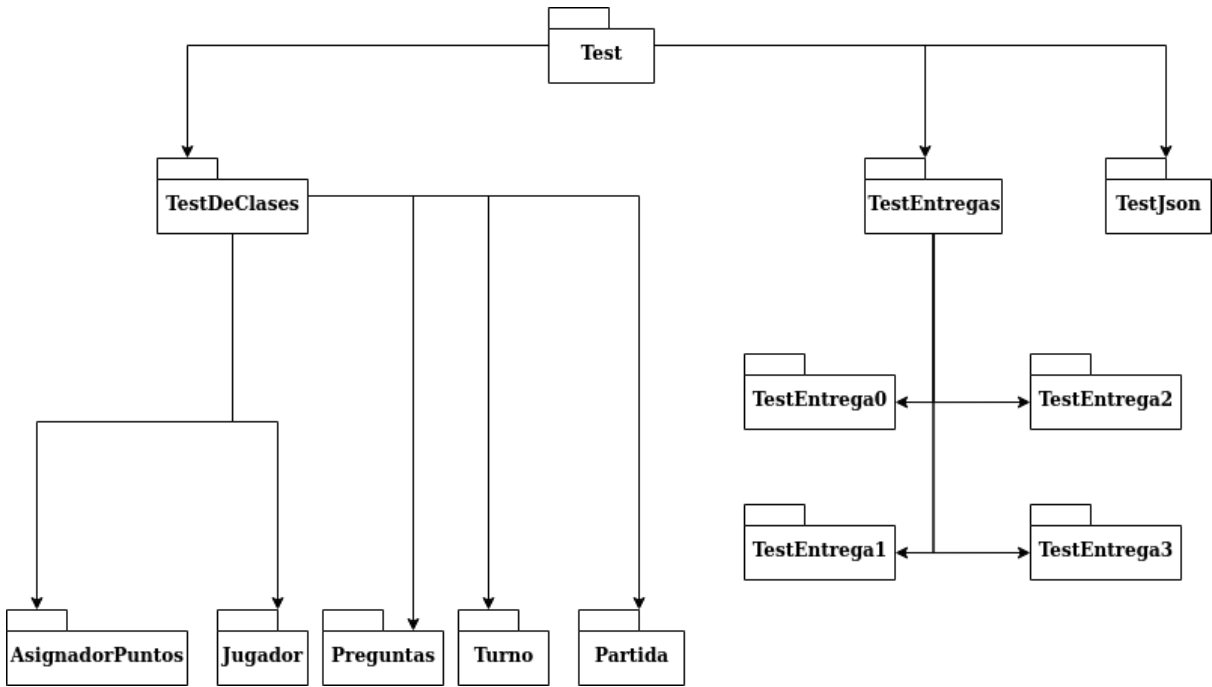
Paquetes Intermedios



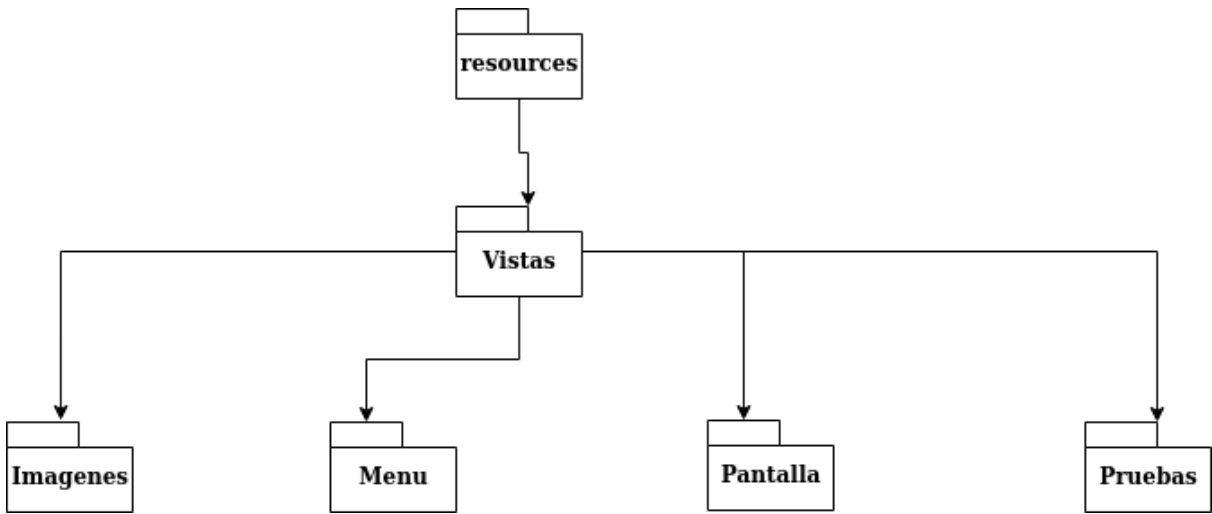
Paquetes de Modelo



Paquetes de Testing

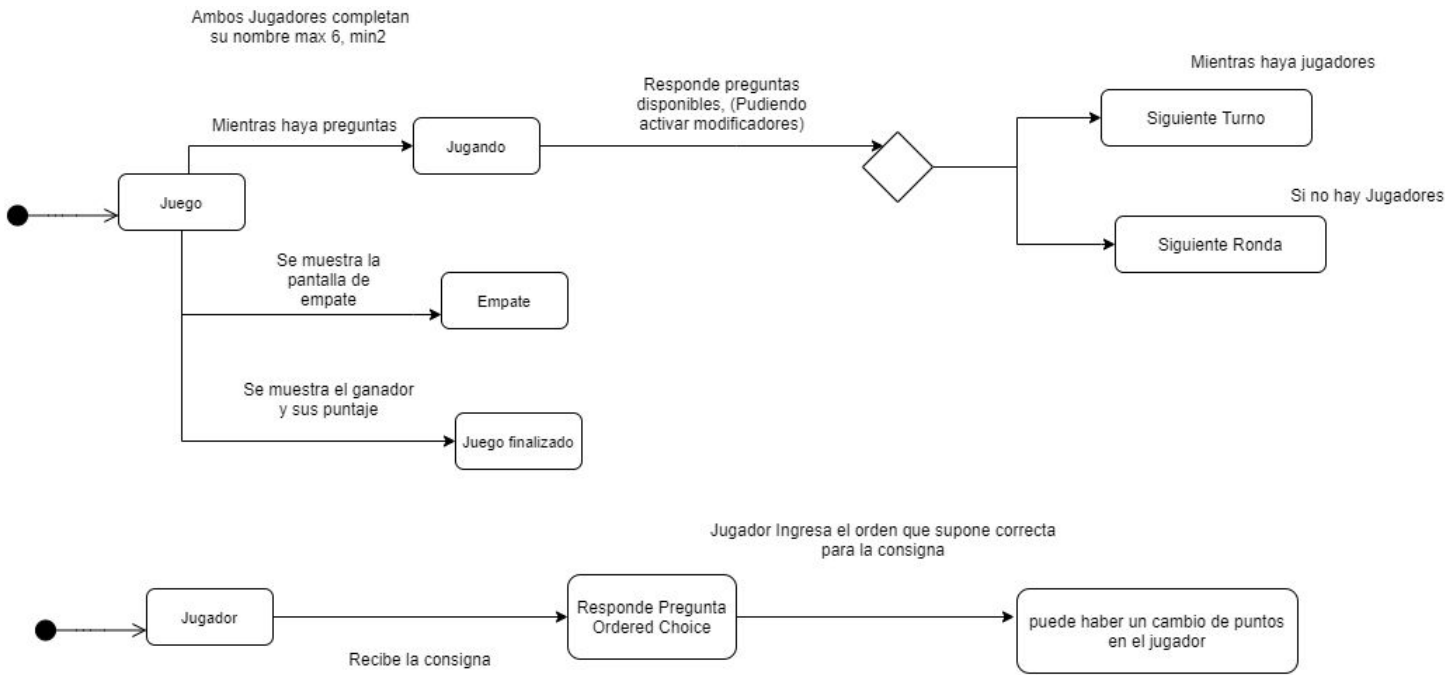


Paquetes de fxml



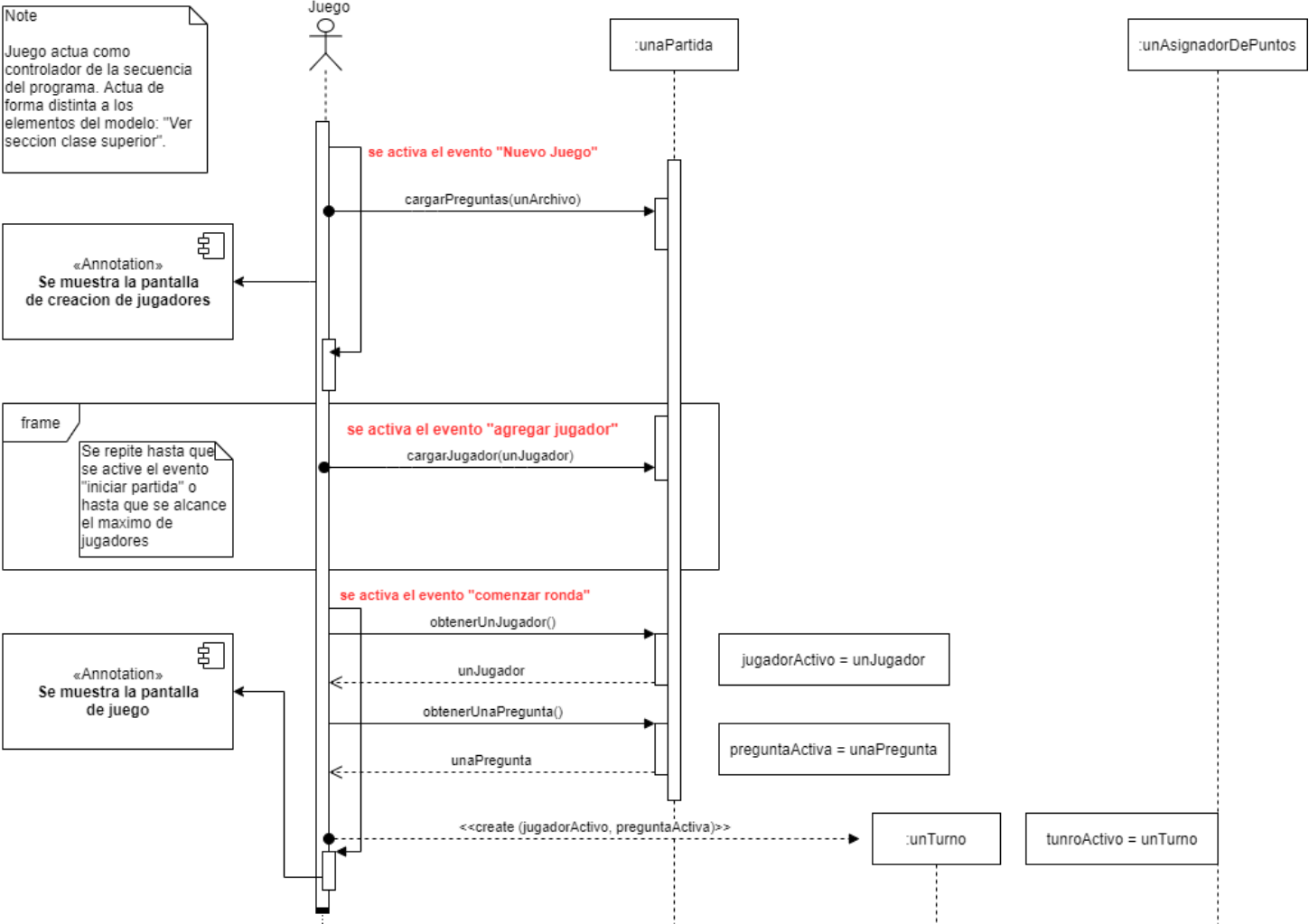
Diagramas de estado

Se representa un cambio a través del tiempo de un objeto.

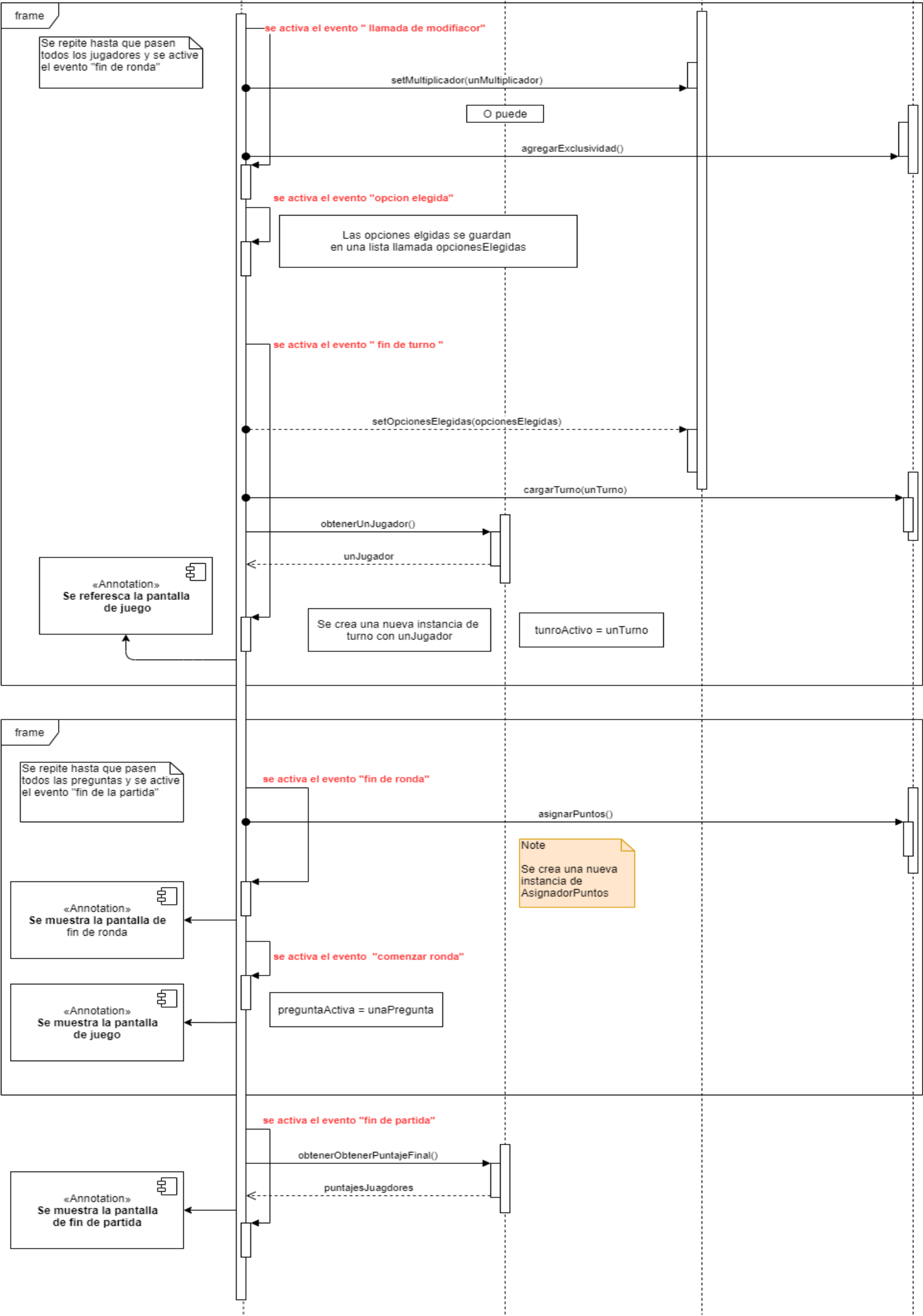


Diagramas de secuencia

Secuencia de una partida



continua...



Detalles de implementación:

Al analizar los distintos tipos de preguntas, se observó que poseían comportamientos similares a la hora de evaluar una lista de opciones (respuesta de un jugador), además de compartir los mismos atributos. Esto nos motivó a crear abstracciones con el fin de no repetir código.

Durante las entregas semanales se fue refactorizando las abstracciones hasta llegar a lo mostrado en el diagrama 2.

PreguntaCriterioSinErrores:

Sus hijos tienen la particularidad de otorgar un punto o cero puntos.

Para que devuelva un punto, las opciones que se le pase a evaluar deben contener todas las opciones correctas de la pregunta, caso contrario devolverá cero puntos.

Las preguntas tipo Ordered Choice y Group Choice se pueden tratar como instancias de la clase PreguntaConTodasOpcionesClasificadas, clase concreta que hereda de la clase abstracta PreguntaCriterioSinErrores.

Las preguntas tipo Multiple Choice Clásico se pueden tratar como instancias de PreguntaConOpcionesCorrectas, clase concreta que también hereda de PreguntaCriterioSinErrores.

PreguntaCriterioParcial:

La lista de opciones a evaluar no necesariamente debe contener todas las opciones correctas para que la pregunta, tras evaluarla, devuelva puntos.

Por ejemplo, si la pregunta tiene 4 opciones correctas y se le pasa a evaluar una lista con solo 2 de las 4 opciones correctas, la pregunta retornaría 2 puntos.

PreguntaPuntajeParcialSinIncorrectos:

Si la lista de opciones a evaluar contiene alguna opción incorrecta, devolverá cero puntos, caso contrario, devolverá tantos puntos como aciertos haya en la lista de opciones evaluada.

Las preguntas tipo “Verdadero falso clásico” y “Múltiple choice de puntaje parcial” se pueden tratar como instancias de la clase PreguntaPuntajeParcialSinIncorrectos, clase concreta que hereda de la clase abstracta PreguntaCriterioParcial.

Obs: se puede tratar a una pregunta “Verdadero falso clásico” como un caso particular de una pregunta “Multiple choice de puntaje parcial”.

PreguntaPuntajeParcialPenalizable:

Al evaluar la lista de opciones (respuesta del jugador), devolverá un punto por cada opción correcta y restará un punto por cada opción incorrecta, retornando el resultado final.

Obs: el resultado final puede ser negativo.

Las preguntas tipo “Verdadero falso penalizable” y “Múltiple choice de puntaje parcial penalizable” se pueden tratar como instancias de PreguntaPuntajeParcialPenalizable, clase concreta que también hereda de PreguntaCriterioParcial.

Entrega Final:

Trabajando Con Vistas:

A Través de archivos FXML cargados con la datos de la interfaz, comenzó la tarea de asociación entre los elementos del modelo. Si bien no se siguió una metodología concreta, nos apegamos a seguir la secuencia del juego y asociar sus comportamientos con los elementos de nuestras interfaces.

Una vez se agiliza la comunicación entre el controlador y el modelo, se fue implementando la navegación entre escenas y se establecieron los comportamientos necesarios para que la parte gráfica refleje el estado del juego.

Cómo funciona:

Las pantallas de juego son muy similares entre sí, por lo que, por convención, decidimos unificar sus controladores en uno solo que maneje sus funciones en común. Claro está que sus diferencias deben ser interpretadas de forma diferente, por lo que tuvimos que hacer reconocibles cada uno de sus comportamientos propios.

Una parte interesante que surgió durante el diseño fue la implementación de las preguntas de ordenamiento y las de agrupamiento. Estas, a diferencia de sus contrapartes, precisan múltiples elecciones para cada una de sus opciones. Decidimos utilizar la herramienta “ChoiceBox” para realizar estas preguntas particulares. Pero el comportamiento de las “ChoiceBox” son diferentes a las de un botón tradicional, estas almacenan la opción elegida sin la necesidad de la intervención de un evento definido en el controlador.

La solución más directa que encontramos fue usar un “Recolector de ChoiceBox” que se encargue de recolectar (redundancia detectada) las opciones seteadas en cada una de las choicebox mostradas y prepararlas para su posterior evaluación.

Excepciones

Una de las metas del diseño del programa fue eliminar la necesidad de excepciones. Esto fue logrado satisfactoriamente, dado que todo lo que introduce el usuario es procesado por el programa sin que este requiera de un comportamiento extraordinario para hacerlo.

Agradecimientos:

- *A Edson Justo nuestro corrector de parte de todo el equipo dinamita.*
- *Gracias por tu infinita paciencia y buena onda, lo apreciamos mucho.*