Algoritmos y Programación II
Seguimiento 1 - Pruebas Unitarias Automáticas

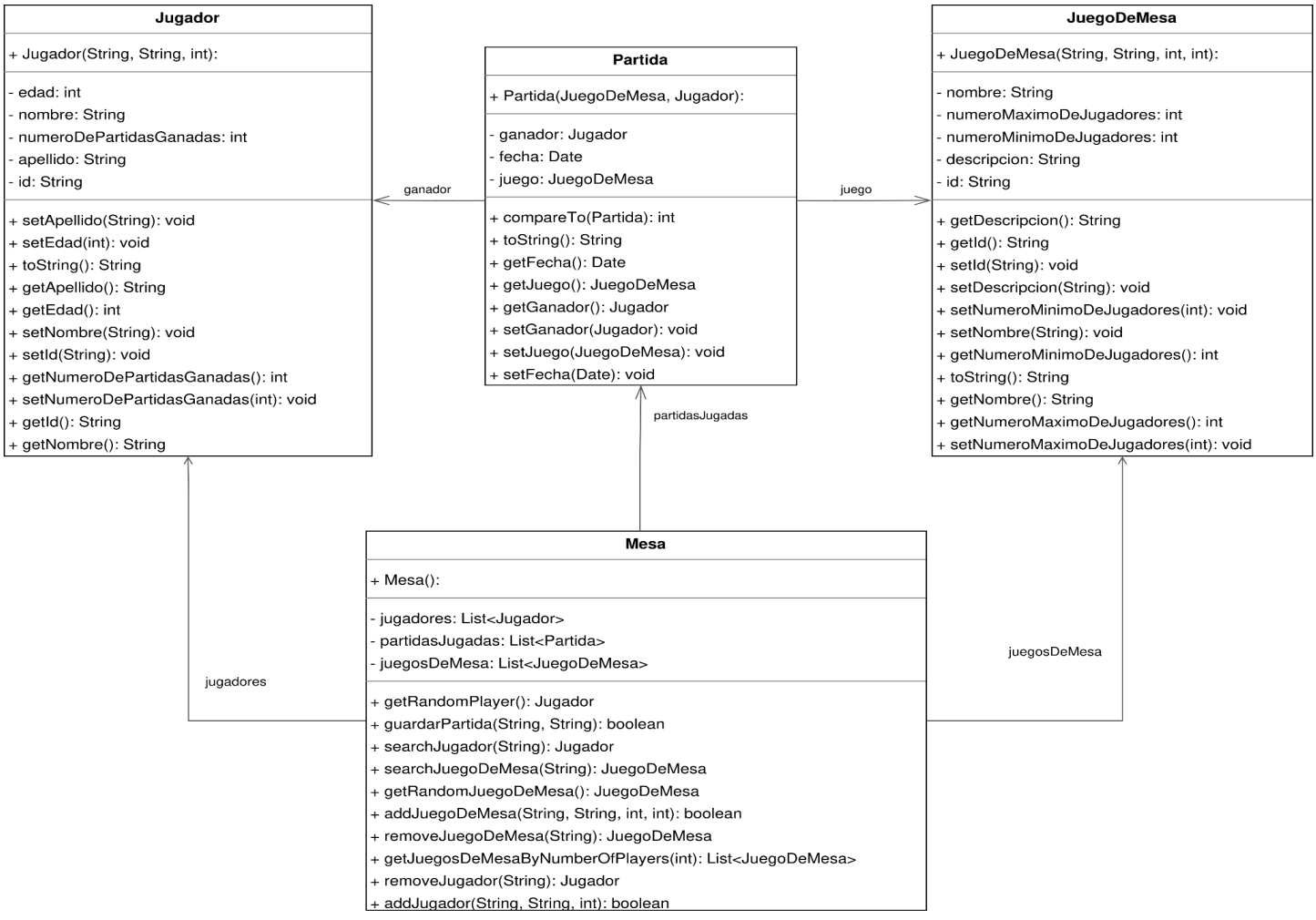
JUEGOS DE MESA

Un grupo de amigos ha decidido crear un club de juegos de mesa, últimamente han tenido problemas de convivencia por no decidir cuál es el juego que van a jugar en sus reuniones, para ello (ya que todos saben programar) han decidido crear un programa en Java que solucione estos problemas, y le han pedido a usted (porque ya no quieren hacerlo ellos) que desarrolle las pruebas unitarias del programa usando JUNIT, ellos ya tienen un diseño inicial que usted puede usar.

Las funcionalidades generales que han pensado estos chicos para el programa se pueden resumir en la siguiente lista:

- Seleccionar un juego aleatorio, dada una cantidad de jugadores
- Agregar un nuevo juego al sistema
- Buscar conjunto de juegos con base a la cantidad de jugadores
- Eliminar un juego del sistema
- Buscar un juego con base al nombre
- Agregar un jugador al sistema
- Eliminar a un jugador del sistema
- Buscar a un jugador por nombre
- Aumentar la cantidad de partidas ganadas a un jugador
- Agregar una partida jugada al historial de partidas
- Crear una partida jugada
- Guardar los datos de una partida jugada

El diagrama de clases del proyecto es el siguiente:



Algoritmos y Programación II
Seguimiento 1 - Pruebas Unitarias Automáticas

Configuración de los Escenarios

Nombre	Clase	Escenario
SetUp1	MesaTest	Vacío
SetUp2	MesaTest	Mesa != null
SetUp3	MesaTest	Mesa != null JuegosDeMesa = [{ "nombre": "Catan", "max_jugadores": 6, "min_jugadores": 3, "descripcion": "Juego de azar y estrategia" }, { "nombre": "Exploding Kittens", "max_jugadores": 7, "min_jugadores": 2, "descripcion": "Juego de cartas" }, { "nombre": "Salem", "max_jugadores": 12, "min_jugadores": 4, "descripcion": "Juego de cartas y rol" },]

Objetivo de la Prueba: Verificar que se puede añadir un nuevo juego de mesa con sus atributos correctos, adicionando un juego más a la lista de juegos.				
Clase	**Método**	**Escenario**	**Valores de Entrada**	**Resultado esperado**
Mesa	addJuegoDeMesa	SetUp2	{ "nombre": "Carcassonne", "max_jugadores": 6, "min_jugadores": 2, "descripcion": "Juego de estrategia" },	***true*** Se agrega un nuevo juego de mesa al arreglo con nombre: "Carcassonne", número máximo de jugadores": 6, número mínimo de jugadores": 2, descripción: "Juego de estrategia"; el tamaño del arreglo aumenta en 1.
Objetivo de la Prueba: Verificar que se puede buscar un juego de mesa por su respectivo nombre.				
Mesa	searchJuegoDeMesa	SetUp3	“Exploding Kittens”	se retorna un juego de mesa cuyos atributos son: { "nombre": "Exploding Kittens", "max_jugadores": 7, "min_jugadores": 2, "descripcion": "Juego de cartas" },

Algoritmos y Programación II
Seguimiento 1 - Pruebas Unitarias Automáticas

Adicional a los casos de prueba anteriores, deberá diseñar e implementar el caso de prueba de **uno** de estos métodos (usted elige cúal):

1. searchJuegoDeMesa
2. getRamdonJuegoDeMesa
3. removeJuegoDeMesa
4. getJuegosDeMesaByNumberOfPlayers

Si lo considera necesario, puede crear uno o varios escenarios adicionales a los ya entregados.

Rúbrica

Codifica los escenarios de prueba. **El escenario cumple con todas las especificaciones del objeto dado.**	**Codifica el caso de prueba 1.** **Usa el escenario correcto, las entradas son las apropiadas y la salida es la esperada. La prueba pasa en verde.**	**Codifica el caso de prueba 2.** **Usa el escenario correcto, las entradas son las apropiadas y la salida es la esperada. La prueba pasa en verde.**	**Diseña correctamente el caso de prueba para alguno de los 4 métodos adicionales.** **Define correctamente la clase, el método a probar, la(s) entrada(s) son las apropiadas para la salida**	**Codifica el caso de prueba para alguno de los métodos adicionales:** **Usa el escenario correcto, las entradas son las apropiadas y la salida es la esperada. La prueba pasa en verde.**	**TOTAL**
10.00%	20.00%	20.00%	25.00%	25.00%	100.00%

Formato de escenarios y casos de uso

Configuración de los Escenarios

Nombre	Clase	Escenario
		-

Objetivo de la Prueba:				
Clase	**Método**	**Escenario**	**Valores de Entrada**	**Resultado esperado**