Diseño de Sistemas 2014



K3001

Jueves TM

**Futbol 5 – 3ra Entrega**

**Grupo 4**

|  |  |
| --- | --- |
| Nombre y Apellido | Legajo |
| Julian Fuks | 147.291-4 |
| Axel Suvalski | 147.290-2 |
| Nicolas Orchow | 146.700-1 |
| Joel Melamed | 146.804-2 |
| Eric Lifszyc | 146.655-0 |



Diagrama de clases



Análisis de la solución

en la solución, tuvimos que adaptar algunas clases a los cambios propuestos para esta entrega. En la clase Jugador, agregamos tanto los métodos proponerAmigoAPartido como calificar.

En el primer caso, tuvimos que validar que el usuario solo proponga a un amigo y no a cualquier jugador, y en el segundo hubo que validar que tanto el usuario calificador como el calificado, hayan jugado ese partido, ya que sino, no correspondería que ese usuario califique o sea calificado.

Para ambas validaciones utilizamos excepciones que en algún momento pueden ser atrapadas o no, generando el error en el sistema.

Ejemplo:

**public** **void** calificar(Partido unPartido, Jugador unJugador, **int** puntaje, String critica) **throws** Exception{

**if**(!unPartido.participoElJugador(unJugador) || !unPartido.participoElJugador(**this**))

**throw** **new** Exception("Jugador no participó en el partido");

Para el caso de uso de calificar a un jugador, agregamos una clase calificación que es la encargada de guardar tanto el puntaje como la crítica realizada. Los objetos de esta clase se van a agregar en una lista que tiene el jugador con todas sus calificaciones.

El caso de uso de proponer un amigo era un poco más complejo ya que había que manejar el asincronismo en esta solicitud.

Para ello utilizamos a la clase partido donde implementamos una colección de mociones donde se guardan para que el administrador pueda consultarlas y también los metodos aceptar y rechazar para que pueda enviar la respuesta

>>Partido

**public** **void** agregarMocion (Mocion unaMocion)

{

\_mociones.add(unaMocion);

}

**public** **void** aceptar(Mocion unaMocion, Inscripcion unaInscripcion)

{

unaInscripcion.setJugador(unaMocion.getJugador());

**this**.agregarInscripcion(unaInscripcion);

}

**public** **void** rechazar(Mocion unaMocion, String motivo)

{

**this**.\_notifier.mocionRechazada(unaMocion, motivo);

}

En el caso de que el administrador acepte la solicitud, este dice con que modalidad hace la inscripción ese jugador y el mismo se inscribe automáticamente en el partido (cumpliendo todas las reglas anteriores sobre la prioridad)  
En caso de que la rechace, se genera una infracción (con el sistema usado para la iteración anterior) y deja registrado el motivo.