**@Deprecated (ya no sirve)**

**~~Nuevos Jugadores~~**

Lo que queremos hacer es que un jugador pueda proponer a un amigo suyo (o sea tiene que estar en la lista de amigos del jugador) para jugar un determinado partido.

El administrador recibe la propuesta, y le pregunta al partido si se puede inscribir. En este momento, el partido va a inscribirA(jugador, partido) y en caso de que haya algun error, el partido va a lanzar una excepcion que va a ser atrapada por el administrador.

Si el administrador atrapa una excepcion significa que no se pudo inscribir al jugador y debera justificar el motivo (que al fin y al cabo es el nombre de la excepcion atrapada).

Ejemplo

persona.proponerAmigo(juan, partido1);

admin.evaluar(juan, partido1) <- En este metodo se va a verificar si juan puede inscribirse o no, y en caso de que no, aca se van a manejar las excepciones

Pseudocodigo

**Persona**

>> proponerAmigo(Persona amigo)

admin.evaluar(this, amigo, partido)

>>esAmigoDe(amigo)

amigos.contains(amigo)

**Admin**

>> evaluar(Persona jugadorQueSugiere, Persona amigo, Partido partido)

if jugadorQueSugiere.esAmigoDe(amigo) then

try {

*partido*.inscribirA(amigo, partido);

}

catch( PartidoEstaLlenoException ){

mostrar\_mensaje(“El partido ya esta lleno”);

registrarDenegacion(Date dia);

}

catch( YaEstaInscriptoException ){

mostrar\_mensaje(“El jugador sugerido ya esta inscripto”);

registrarDenegacion(Date dia);

}

end

***Alternativa 2***

En vez de que la validacion de esAmigo(amigo) la realice el admin, que la realice la persona misma antes de enviar el evaluar. Entonces seria:

**Persona**

>> proponerAmigo(Persona amigo)

if ( this.esAmigoDe(amigo) ) {

admin.evaluar(amigo);

}

>>esAmigoDe(Persona p)

return amigos.contains(p);

En el caso de que el partido quede confirmado, todas las propuestas para ese partido deberian borrarse