**CU Calificar Jugadores: justificación**

Enunciado:

*Una vez jugado el partido, cada jugador que participó puede calificar a cada uno de los jugadores con una nota numérica del 1 al 10 y agregar un texto a modo de crítica a cada jugador.*

Acá pasa lo mismo que en el caso de uso anterior: hay distintos tiempos para hacer las cosas. Como se ve, NO dice que la calificacion ocurre en el momento de finalizar la fecha, por lo tanto, obligar a los jugadores a que hagan las calificaciones en ese momento (o quedarnos esperando hasta que a todos se les cante hacerlas), seria un error. Pueden hacerla cuando ellos quieran, despues de la fecha del partido.

Entonces, tenemos:

1) En un primer momento, luego de jugar el partido, se llega a la fecha, y se lo cierra (en nuestro caso, al hacerse confirmarEquipos(); ). En este momento, construimos la lista confirmadosPresentes, que son aquellos que efectivamente jugaron el partido.

2) Luego, en ALGUN otro momento posterior a la fecha del partido, pueden ingresar al sistema los que jugaron ese partido, y ponerse a calificar a los que deseen (ninguno, uno o varios) y también cuando lo deseen.

Estos jugadores son aquellos en la lista de confirmadosPresentes del Partido, que pueden calificar a los demás; y también pueden ser calificados por los demás.

La forma en que haríamos las calificaciones sería mandando el mensaje…

Si suponemos que existe una UI que nos muestra las ventanitas para elegir una persona en un partido, y calificarla; la secuencia de la misma sería:

1) Se abre la UI, le muestra a la persona la lista de partidos que jugo <- ACA VEMOS LOS PARTIDOS (podemos pickear un objeto partido)

2) la persona hace clic en uno, y despues la UI muestra la lista de confirmadosPresentes de ese partido <- ACA VEMOS LAS PERSONAS CALIFICABLES (podemos pickear una persona)

3) la persona elige alguno, y hace clic en el boton “calificar” <- aca **ya tenemos** el objeto **persona**!

4) aparece otra ventanita, mostrando los datos minimos de la persona, y un textBox para poner la nota, y otro para el comentario. <- aca construimos el objeto unaCalificacion : Calificacion

5) Hace clic en “aceptar” -> queda la calificacion agregada a la coleccion de calificaciones de esa persona. <- aca la persona recibe el mensaje recibirCalificacion(unaCalificacion)

**Pseudocódigo de la secuencia:**

El CU calificar empieza cuando viene un interesado y dice:

unPartido.calificarA(juan, yo, 8, “Jugo bien”); //*yo* es el objeto Persona asociado al interesado

Luego la implementación sería algo asi:

Partido

>>calificarA(persCalificada, persCalificadora, nota, string)

. valido que la nota NO sea *null* y este entre 1 y 10(**int**)

. verifico que el partido ya se haya jugado (que haya un bool partidoJugado en true [que se pone en true cuando el admin hace confirmarEquipos()]) <-Sólo puedo validarlo en Partido

.**(\*)**

. verifico que la persCalificada haya jugado el partido (esten dentro de confirmadosPresentes) <- idem

. verifico que la persCalificadora haya jugado el partido <- idem, y en ese orden

. persCalificada.calificatePor(persCalificadora, *this*, nota, string);

Persona

>>calificatePor(persCalificadora, partidoCalificacion, nota, string)

. valido que en mi coleccion de calificaciones no exista ya una calificacion que tengan igual el par persCalificadora y partidoCalificacion que vienen por parametro (o sea, q no este repetida)

. Calificacion calificacion = new Calificacion(persCalificadora, partidoCalificacion, nota, string)

. misCalificaciones.add(calificacion)

**(\*) aca podriamos agregar verificaciones de que primero la personaCalificada este inscripta (y/o confirmada?) al partido, y despues lo mismo para personaCalificadora. Así el usuario podría distinguir mejor el error, si ocurre.**