Documento de diseño

Proyecto 3 - Diseño y Programación Orientada a Objetos

Semestre 2022-20

Grupo de trabajo #5

Contenidos que se incluyen:

- Diagrama de clases completo (archivo externo)
- Diagrama de clases de alto nivel (archivo externo)
- Introducción al proyecto 3
- Cambios realizados respecto al proyecto 2
- Justificaciones de diseño
- Diagramas de secuencia para algunas funcionalidades
- Glosario

1. Introducción

Teniendo en cuenta el desarrollo realizado y entregado en el proyecto 2, se quieren realizar algunos cambios en las reglas del juego de futbol de fantasía para identificar los componentes del diseño y la implementación que deberían alterarse.

Los cambios se dan principalmente en las reglas de puntuación de los jugadores, de los equipos en cada fecha, la posibilidad de crear múltiples equipos de fantasía y la adición de funcionalidades para realizar reportes y gráficas.

A continuación, se presentarán los contenidos que conciernen al desarrollo del diseño de la solución que en grupo se ha planteado.

2. Cambios realizados respecto al proyecto 2

- En esta entrega habrán muchos más paneles y algunos cambios en el diseño de las ventanas con respecto al anterior

- Habrá nuevas clases dentro del modelo que nos permitirán modelar los objetos del mundo de forma más eficiente y de ser posible intentando seguir principios SOLID y patrones de diseño para facilitar y mejorar nuestro código, principalmente con la adición de tener más equipos algunos componentes que antes pertenecían al participante pasaran a ser del equipo como el presupuesto, el manejo de la plantilla de los equipos cambiará pues los jugadores ahora tendrán un atributo que define si son suplentes o titulares dada por una enumeración.

- Los puntos deben ser distribuidos de forma diferente y deben tenerse en cuenta nuevos parámetros para la asignación de puntos

- Así como un participante puede tener muchos equipos también debería poder eliminar alguno si desea y además debe poder ser capaz de administrar cada uno por separado lo que ha puesto nuevos desafíos a tener en cuenta a la hora de operar con la plataforma y asignar los puntos.

- El controlador ha pasado de ser una clase enorme a tener muchas clases que se encargar de conectar la interfaz con el modelo dependiendo de cual sea el usuario que lo necesite, es decir, habrá controladores solo para el administrador, otros para el participante y otro para el manejo del usuario en general

- El UML también debe cambiar puesto que hemos rediseñado muchas clases con sus métodos y atributos.

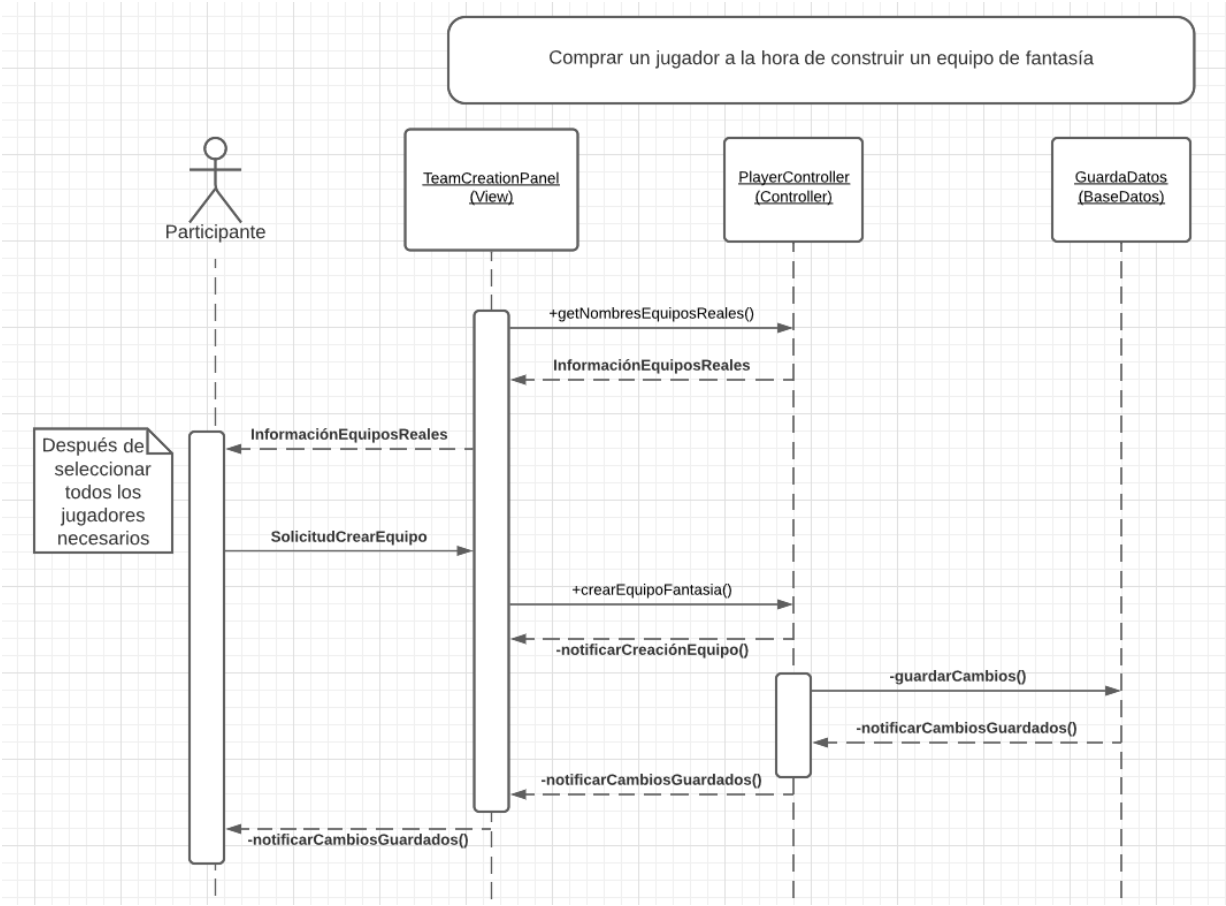
3. Justificaciones de diseño

Para este proyecto nuestro diseño ha sido rediseñado en cuanto a las estructuras que hemos venido usando, en nuestra segunda entrega tuvimos bastantes problemas a la hora de conectar la interfaz gráfica con nuestro controlador pues era una estructura que estaba diseñada para servir a través de la consola más no en una interfaz gráfica.

A medida que avanzaba el curso también pudimos darnos cuenta de varios errores que este diseño tenía pues nuestro patrón MVC no estaba bien implementado porque el controlador estaba sobrecargado. Para esta nueva entrega hemos decidido rediseñar todo y ahora habrá varios controladores que se encargaran de comunicar diferentes componentes de la interfaz gráfica con el modelo de la lógica.

Por otro lado, también hemos tenido que rediseñar muchas funciones y clases del modelo para poder facilitar la lectura y compresión de nuestra implementación (y la de quien lo lea). Además, han surgido nuevas ideas que podrían hacer más fácil el manejo de ciertas clases por lo que es posible que nuestro proyecto actual puede tardar aún un poco en terminarse y definirse por completo. Por esta razón también es probable que no tengamos un UML demasiado consistente en ciertos puntos pero que actualmente estamos tratando de construir sobre la base de una entrega anterior.

4. Diagramas de secuencia para algunas funcionalidades



5. Glosario de clases

Runner: Esta clase del paquete de vista se encarga de inicializar instancias de los objetos que comienzan con la lógica del programa. Es la clase que se debe ejecutar primero en toda la solución.

FirstFrame: Clase correspondiente a la creación de la primera ventana que aparece al comenzar el programa.

FirstPanel: Primer panel que se muestra en la primera ventana una vez se inicia el programa. Esta tiene información sobre las primeras opciones que al usuario le interesaría elegir. Las clases Frame que se encuentran en la aplicación son significativamente menores en cantidad a comparación de las clases Panel, pues la configuración de la ventana mostrada cambia considerablemente menos que el contenido dentro de ella.

TeamCreationPanel: Clase perteneciente al paquete de vista, se encarga de mostrar un panel diferente al “main” para el menú de un usuario. En este panel, el usuario encontrará listas desplegables con las opciones que tiene para elegir y armar su equipo si cumple con algunas condiciones como el presupuesto suficiente.

Administrador: Se va a encargar de subir los resultados de los partidos reales fecha por fecha. Solo hay un administrador registrado previamente en la plataforma.

Participante: Va a tener un solo equipo de fantasía y va a poder comprar y vender jugadores para armar su plantilla. Del mismo modo, va a poder elegir su alineación antes del inicio de cada fecha para que su equipo gane la mayor cantidad de puntos posibles.

TemporadaReal: Esta se relaciona con la clase fechas y con la clase de partidos reales. Por medio de esta podemos obtener la información de cuales fueron los partidos reales que se disputaron en una fecha.

Fecha: Esta clase se relaciona con PartidoReal y también con JornadaFantasia, por medio de esta es posible mantener el historial de todos los partidos y jornadas disputadas. A partir de esta podremos obtener los resultados de los partidos en una fecha específica. Del mismo modo, podremos obtener las estadísticas de un equipo de fantasía en una fecha determinada.

PartidoReal: Esta va a tener dos equipos reales, un número de fecha específico y un día y hora. Esta se relaciona con ResultadoPartidoReal.

ResultadoPartidoReal: A diferencia de PartidoReal, con esta clase podremos obtener el resultado de un partido. Del mismo modo, se van a obtener las estadísticas (desempeño) de todo el equipo y de sus jugadores uno por uno.

EquipoReal: Esta clase nos va a dar información de todos los jugadores reales que hacen parte de un equipo.

JugadorReal: Esta clase contiene la información básica de un jugador real y el historial de su desempeño en los partidos reales.

DesempeñoJugadorReal: Por medio de esta clase se van a obtener todas las estadísticas necesarias para asignar un puntaje a un jugador que esté en un equipo de fantasía.
Equipo Fantasía: Este va a ser el equipo manejado por el participante. Al inicio de la temporada el participante va a elegir a los 15 jugadores que harán parte de su equipo de fantasía. Esta clase se relaciona con la clase fecha a que una fecha puede tener varios equipos. También, se relaciona con la clase plantilla fantasía.

PlantillaFantasia: Esta clase contiene todos los jugadores de fantasía que hacen parte del equipo de fantasía de un participante.
Jugador Fantasía: Con esta clase podemos ver los puntos que ha obtenido un jugador específico en una jornada de fantasía. Del mismo modo, podemos ver su historial de puntos para observar su rendimiento a lo largo de la temporada.