

Diseño Proyecto Liga de Fantasía

Sebastián Contreras Salazar 202020903

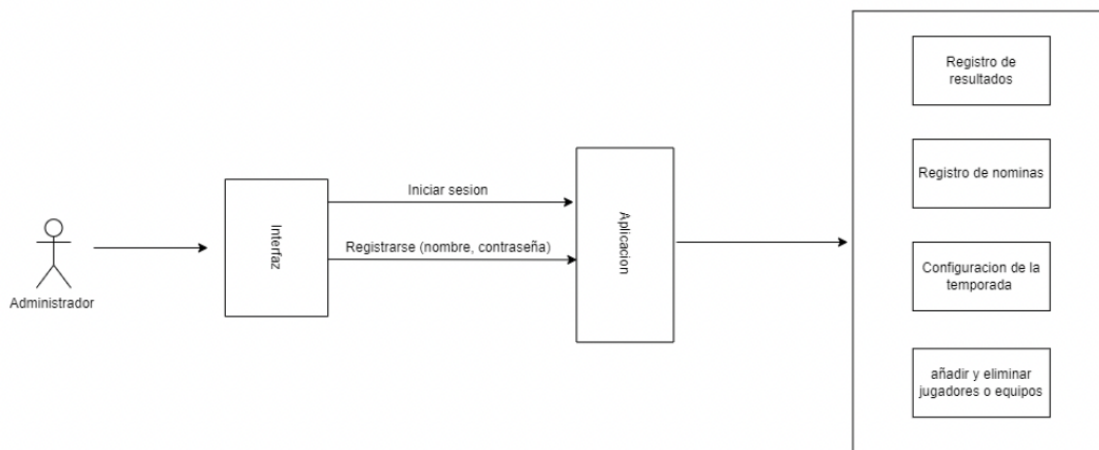
1. Contexto del Problema

Al iniciar el proceso de diseño es necesario analizar los componentes para la funcionalidad de alto nivel de la aplicación de liga de fantasía, esto debido a la necesidad de poder crear una interfaz visualmente atractiva en la cual se pueda navegar con facilidad para así poder sacar el mayor provecho frente a la funcionalidad de esta misma. Es así como en la siguiente imagen se ilustran las distintas interacciones que tanto un usuario como un administrador van a poder tener dentro de la aplicación:



La interacción del usuario con la interfaz corresponde a otorgar las entradas de: registrarse en la aplicación con nombre y contraseña y posteriormente al ingresar poder seleccionar/crear su equipo de fantasía y poder revisar datos de interés, tales como su posición en la liga, sus estadísticas, el desempeño de sus jugadores, como esta el mercado de compras, entre otros. La interfaz por su parte debe de mostrarle las interacciones que este hace dentro de la aplicación a el administrador .

El administrador le corresponde únicamente otorgar entradas correspondientes al inicio sesión con su nombre, también suministra información de los partidos existentes dentro de la liga más la información de cada temporada.



Igualmente, cabe resaltar, que al registrar resultados se incluye los datos de cada partido y puntaje de jugador, por lo que él mismo tiene que actualizar esa información e incluirla para facilitar a la aplicación de actualizar los puntajes y estadísticas de un equipo de fantasía.

Por otro lado, la interfaz debe poder solicitar a la aplicación que inicie sesión tanto como administrador como usuario y así registrarse en el juego. Además, debe de poder acceder a los datos proporcionados por la calculadora para así realizar las clasificaciones y puntajes de los jugadores de fantasía. Además podemos encontrar una tienda en donde el jugador va a hacer la compra y venta de jugadores para su equipo y también, en la misma tienda, observar las compras que se han hecho recientemente, mostrando de qué equipo a qué equipo se hizo la compra y cuánto se pagó.

2. Componentes candidatos y estereotipos

- a. Se usará el patrón MVC el cual se ve implementado la aplicación a través del bloque modelo, habrá diferentes clases que formarán parte del programa de fútbol de fantasía, y el controlador obtendrá información sobre las clases y comparará entre ellas para cumplir con los requisitos funcionales, cambiando En otras palabras , hacer toda la lógica de la aplicación, en términos de vista, hay una interfaz gráfica que permite al usuario desarrollar en nuestra aplicación
- b. Los usuarios del tipo "**Administrador**", deben ser reconocidos para mantener el control general sobre toda la aplicación, incluidas las cargas de archivos, como listados de calendario y cargas de listas para mantener constantemente actualizadas las ligas de fantasía y partes de las mismas.
- c. Otros tipos de usuarios que utilizan la aplicación y/o el sistema requieren una persona que actúe como un jugador real, este tipo de usuario debe poder crear uno o varios equipos de fantasías, cambiar el once inicial y el banquillo del equipo, y cambiar el Capitán, entre otras funciones. Esto corresponde a un candidato de "usuario" del que

el usuario es responsable, y el lado de fantasía almacena los cambios realizados por el usuario y guarda su información, permitiendo que no otro “usuario” no pueda usar el nombre de su equipo escogido y tampoco, pueda comprar un jugador que pertenece a un equipo de fantasía.

- d. Para el almacenamiento de datos se van a guardar en una carpeta llamada data, buscando tener diferentes variables que almacenan toda la información, compras, entre otros.
- e. Teniendo en cuenta las reglas de dominio que requieren que el usuario compre otro jugador al vender un jugador, además de asegurarse de que el presupuesto actual del usuario sea suficiente antes de comprar un jugador, es posible regular el precio en el mercado inmobiliario, por ejemplo, determinado por el precio del jugador administrador. Este conjunto de características corresponde al estereotipo de **"mercado"**.
- f. Para las compras de los jugadores de un mismo “usuario” para diferentes equipos que le pertenecen, las funciones de compras deben ser las mismas que para cualquier otro equipo, puesto que si quiere comprar un jugador para su equipo de fantasía, este necesita estar disponible y no pertenecer a ningún otro equipo respectivo.
- g. Para los reportes, estadísticas y gráficas presentadas, es necesario que se hagan a partir de los datos guardados en la carpeta de datos y de la información que sube el administrador, en el que los datos sean a partir de los jugadores que le pertenecen al equipo de fantasía.
- h. Para las comparaciones de otros equipos y jugadores, es necesario que se use la carpeta de datos y se compare con los mejores equipos y jugadores, para poder facilitar la comparación. Sin embargo, si se deseara, se podría comparar con un equipo en particular y específico que deseara el mismo usuario, en el que podría ingresar el nombre del equipo de fantasía que desea comparar con el suyo y a partir de eso, se pueden mostrar gráficas sobre los progresos, compras y desempeños respectivamente.

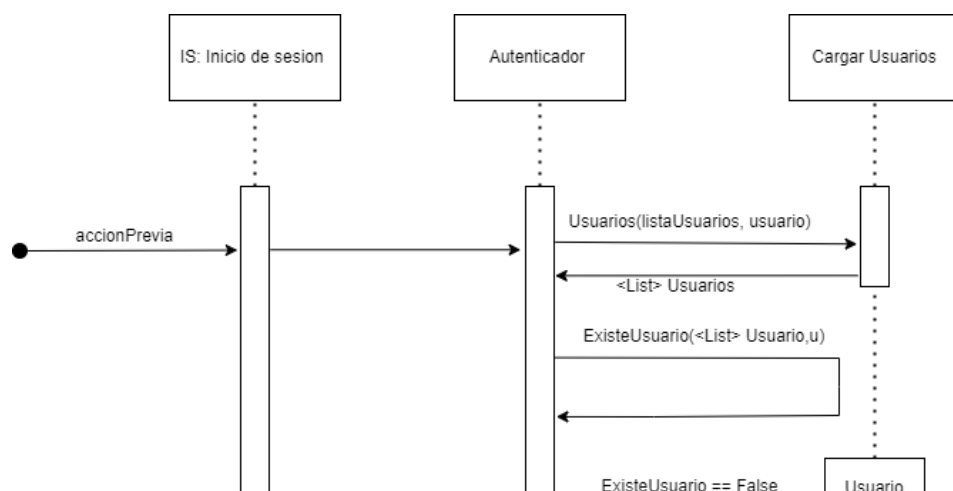
3. Responsabilidades

La siguiente tabla en detalle cada una de las responsabilidades y el componente asociado que debe asumir las responsabilidades

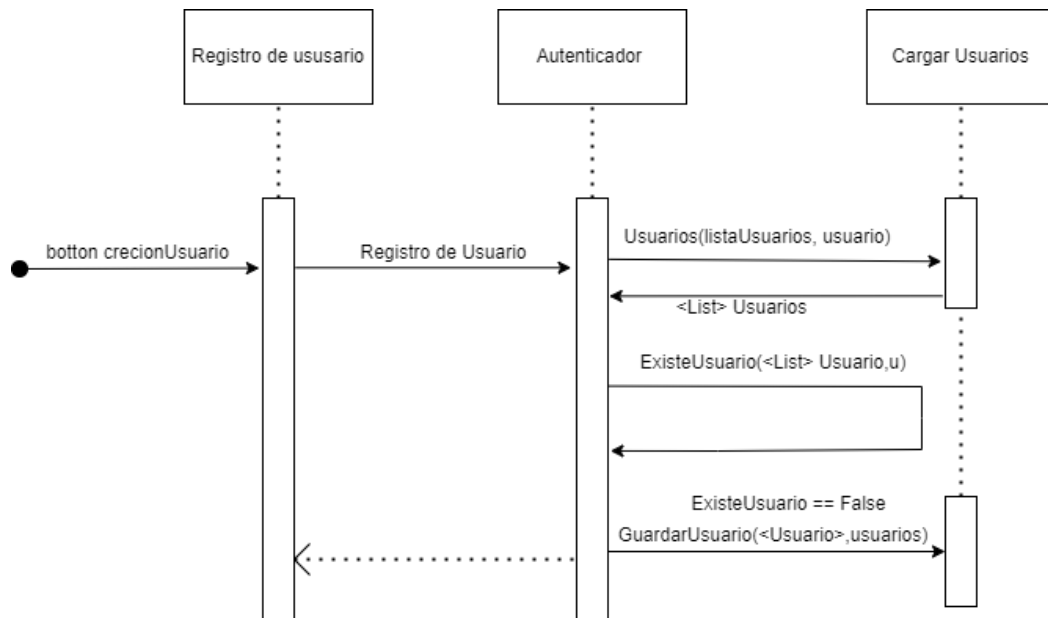
#	responsabilidad	componente al que pertenece
1	Crear equipo de fantasía	Usuario
2	Editar posiciones	Usuario
3	vender/comprar jugadores	Usuario
4	Cargar partido	administrador
5	Cargar temporada	administrador
6	Cargar resultados	administrador
7	añadir/ eliminar jugadores	administrador
8	ranking	calculadora
9	Eliminar equipo	Usuario
10	Comparar equipos	Usuario
11	Progreso/estadísticas por equipo	Usuario
12	Desempeño jugadores	Usuario
13	Historial partidos temporada	Usuario
14	Cargar estadísticas/progreso	Administrador
15	Mercado de compras y ventas	Usuario

4. Colaboraciones

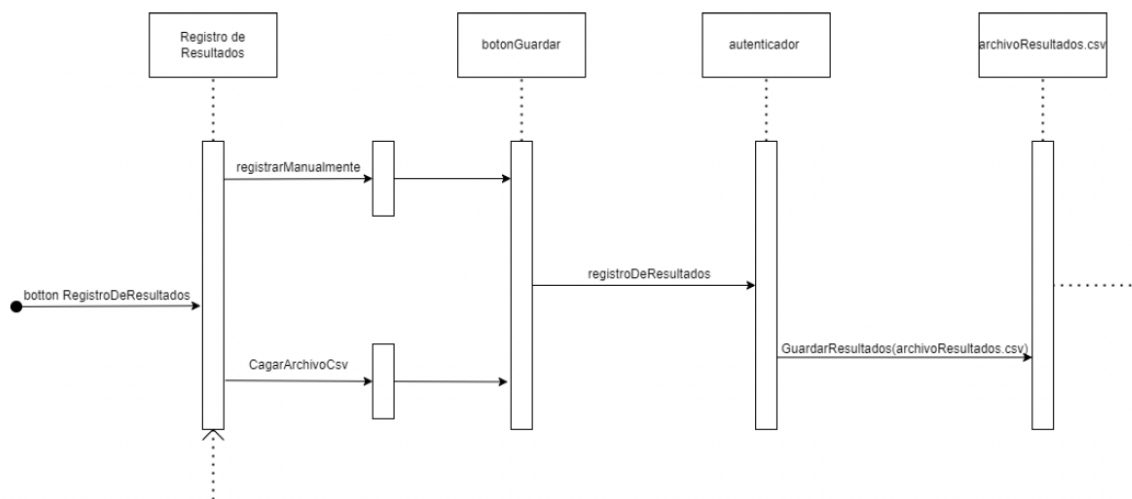
Inicio de sesión: Para registrar un usuario se accede desde la vista donde se ingresa la información del usuario (Nombre y contraseña), éste será añadido a un arraylist en donde se almacenarán todos los usuarios existentes en la aplicación. Posteriormente al iniciar sesión, un autenticador validará la información que está registrada y si esta no se encuentra retornara que el usuario o contraseña es invalido.



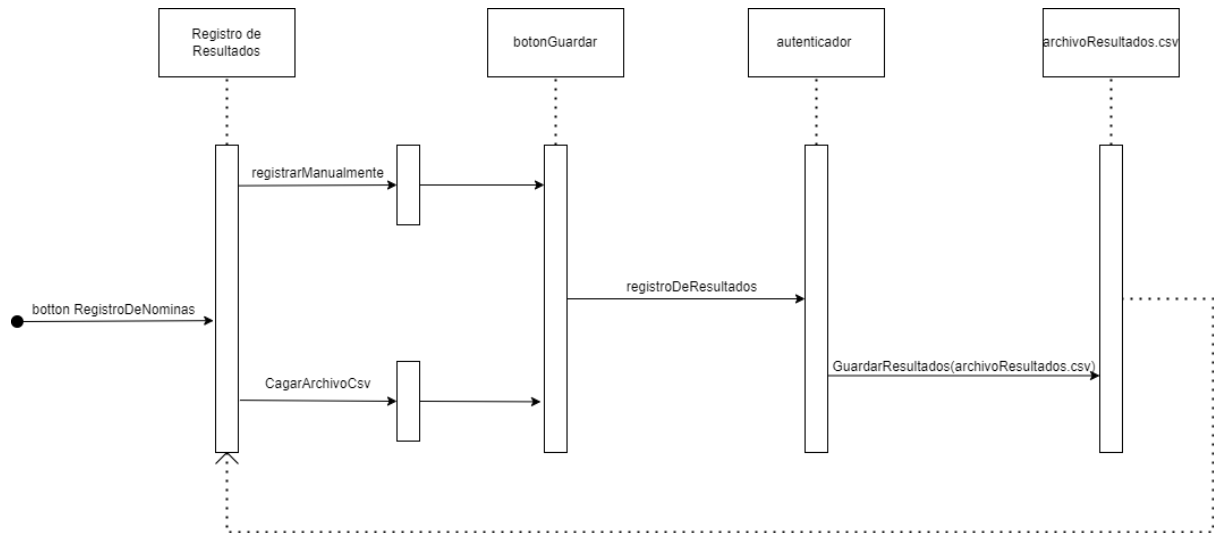
Creación de Usuario: Para la creación de un nuevo usuario se le solicitará al usuario que ingrese nombre y contraseña para este, posteriormente esta información será guardada en el arraylist correspondiente a los usuarios, si el usuario ya existe debe de retornar este usuario ya está registrado en el sistema y de lo contrario generará un mensaje de que el usuario fue creado exitosamente.



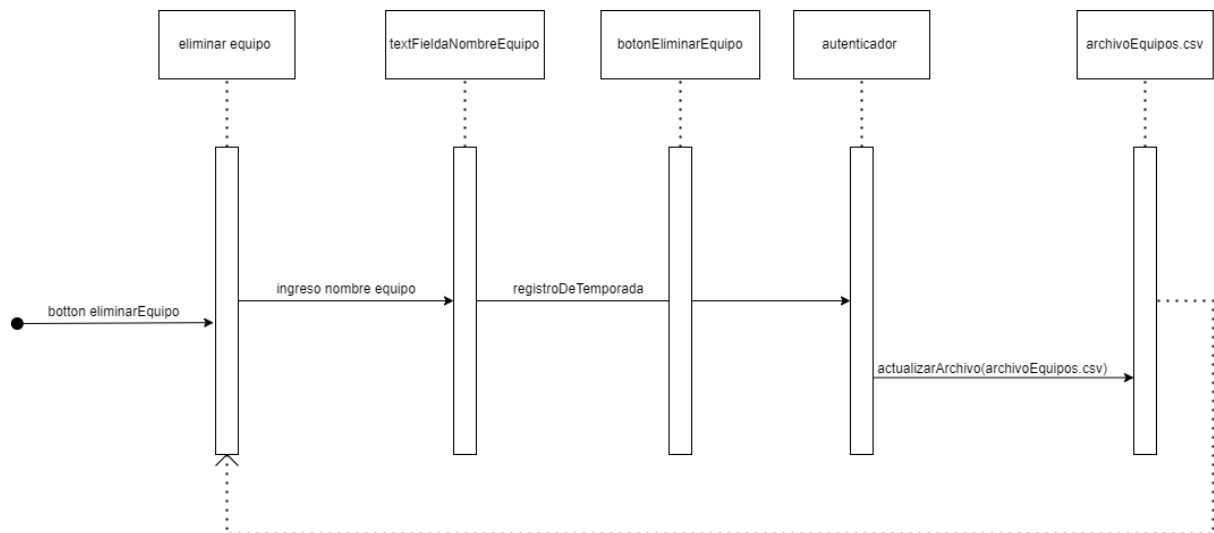
Registro de resultados: Para el registro de resultados se le solicitará al administrador que registre los datos ya sea manualmente para guardarlos en un archivo dentro de la aplicación o subiendolos por medio del botón de “cargar archivo csv”:



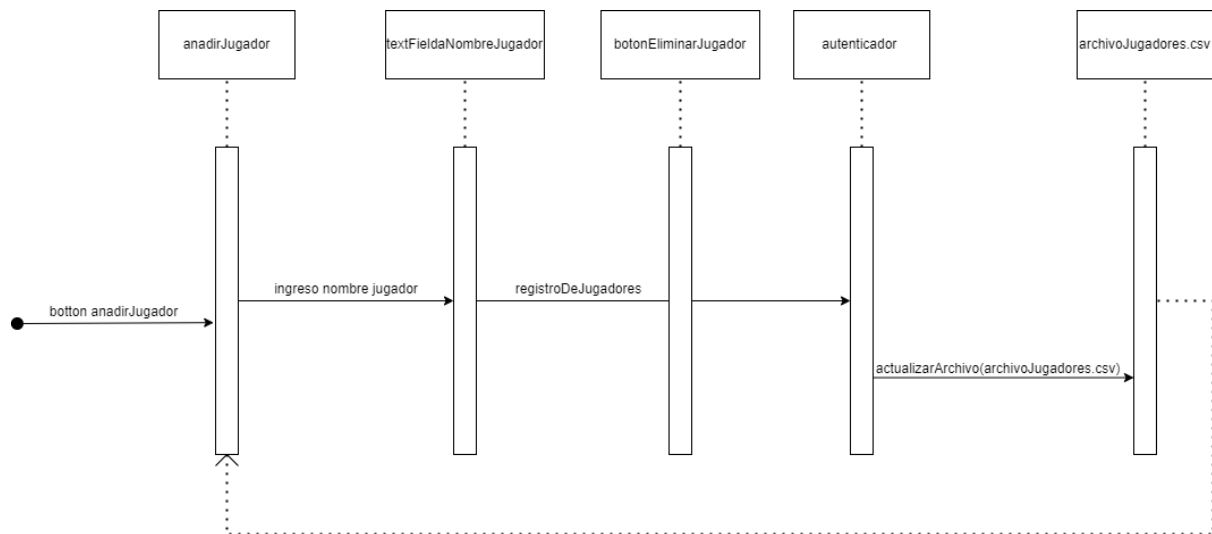
Registro de nóminas: Para el registro de resultados se le solicitará al administrador que registre los datos ya sea manualmente para guardarlos en un archivo dentro de la aplicación o subiendolos por medio del botón de “cargar archivo csv”:



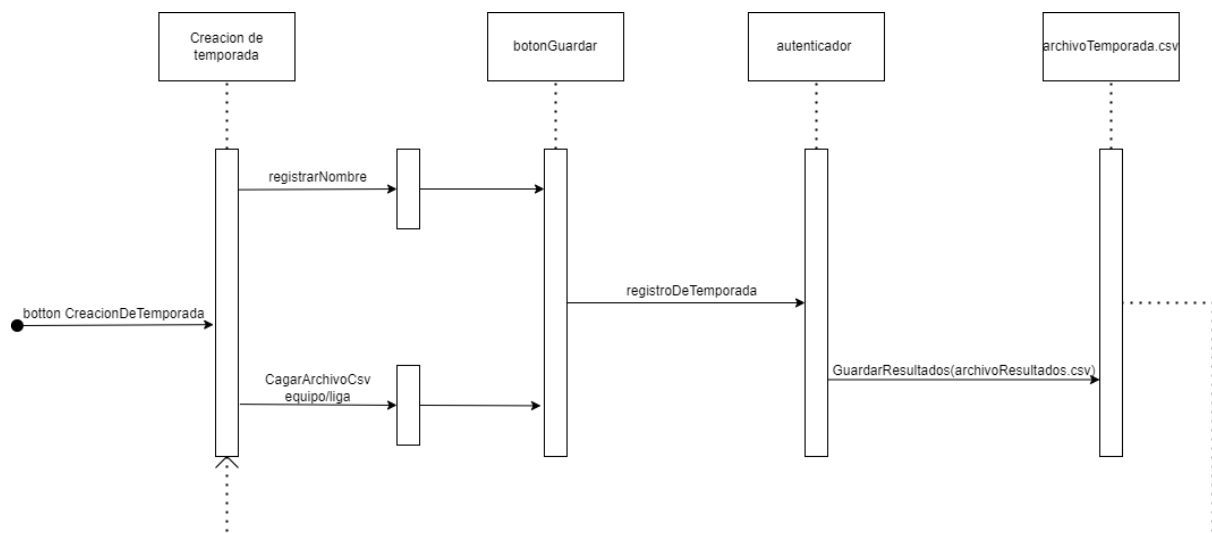
Configuración de aspectos(eliminar equipo): Por medio del botón “eliminar equipo” el administrador podrá acceder a los archivos correspondientes a la liga y al listado de jugadores para así poder eliminar estos de la temporada correspondiente:



Configuración de aspectos(eliminar un jugador): Por medio del botón “añadir jugador” el administrador podrá añadir jugadores al archivo de jugadores.csv por medio de la consola:



Crear temporada: Para la creación de una nueva temporada se le solicitarán al administrador que registre el nombre correspondiente a esta temporada además de archivos tipo csv tanto de los equipos pertenecientes como la información en general de la liga, estos archivos sera guardados en la carpeta de la temporada:



Crear nuevo equipo: Para crear un nuevo equipo nuevo de fantasía se espera que haya una carga de datos realizada ya por el administrador, puesto que se espera a que el Usuario ya tenga acceso a una información que ya existe y que puede cambiar. Ahora bien, el usuario que no es administrador y es un participante de la aplicación, se espera que pueda crear uno o varios nuevos equipos y ponerles el nombre que desee con la única restricción de que ese nombre no esté en uso. Luego de esto se lleva al usuario a añadir 15 jugadores respectivamente, de la posición que deseen, puesto que en base a videojuegos reales de fútbol, el usuario tiene permitido comprar todo tipo de jugador en un equipo, incluso si no incluyen todas las posiciones que hay, por lo que al llegar a cargar al nuevo equipo en la base de datos, se sigue el mismo proceso de configurar aspectos, que se mencionó previamente.

Vender y comprar jugadores: Para comprar y vender jugadores del equipo de fantasía, es necesario que ese equipo exista ya, por lo que se debe autenticar primero si realmente existe. Luego de ellos es necesario revisar si el jugador existe al igual que su equipo real, que no pertenezca a otro equipo de fantasía ya existente, y si se llega a este punto es posible comprarlo, el cual pertenecería al equipo de fantasía respectivo. Igualmente, si se desea comprar un jugador es necesario y obligatorio revisar el tamaño del equipo, puesto que debe tener 15 jugadores. Además, esto último también debe aplicarse para la venta de un jugador. Sin embargo, si se decide vender un jugador, es necesario comprar uno nuevo y ese jugador vendido regresa a pertenecer, en nuestra base de datos, a su equipo real y puede ser comprado por otro usuario.

Editar posiciones de equipo: En cuanto a editar las posiciones del equipo se espera que el usuario ya tenga los 15 jugadores respectivamente, y una vez los tenga este tendrá la opción de decidir qué posición va jugar cada jugador, en el que por ejemplo, un delantero puede jugar de arquero, y a pesar de que es ilógico, muchos videojuegos lo permiten, por lo que sí es posible, pero no es recomendable. Igualmente, el usuario simplemente tiene la opción de decidir en qué posición juegan y cuál es el 11 titular del mismo, para poder guardarlo en un HashMap que incluya el nombre del equipo y su alineación respectiva.

Rankings: Para los rankings se espera que por lo menos haya 2 equipos de fantasía, los cuales ya tienen su 15 jugadores establecidos, para que se puedan sumar la cantidad de puntos de cada jugador del equipo de fantasía y buscar sacar los 2 mejores equipos entre todos los equipos de fantasía. Ahora bien, cabe resaltar que es importante que haya una autenticación de que si hayan datos cargados y existan equipos de fantasía. Ahora bien, los rankings se calculan como una suma de puntos de todos los jugadores titulares que hacen parte del equipo de fantasía, por lo que es necesario que el administrador haya cargado previamente información de los jugadores y sus partidos.

Desempeño jugadores: Para los desempeños de jugadores, se espera que ya exista una carga de datos por parte del administrador de los jugadores y sus desempeños, con el fin de poder permitirle al usuario mirar cómo han ido ganando puntos los jugadores que le pertenecen a su equipo de fantasía a lo largo de las fechas. Ahora bien, es necesario que existan datos ya cargados y que el usuario únicamente puede ver el desempeño de los jugadores que pertenecen a ese equipo de fantasía en particular, que, cabe resaltar, debe ser de ese mismo usuario, por lo que es necesario hacer una autenticación de mostrar el desempeño de los jugadores que únicamente le pertenecen a ese equipo de fantasía.

Mercado de compras: Para esto es necesario que no solo haya ocurrido una carga de información sobre los jugadores por parte del administrador, sino también es necesario que existan varios equipos de fantasía los cuales hayan comprado ya sus jugadores, puesto que el “mercado de compras” permite

es observar cuales han sido los últimos jugadores que se han comprado y por qué precios, el cual cualquier “usuario” debería tener acceso al mismo. Cabe resaltar, que para un mismo momento, lo que esté en esa opción y se muestre al usuario, que deberían ser las últimas 5 compras ocurridas exactamente junto con sus precios, debe ser lo mismo que pueda ver otro usuario en ese mismo momento.

Comparar con otros equipos: Para esto es necesario que exista una carga de datos de jugadores y sus desempeños, al igual que es necesario que existan equipos de fantasía. Ahora bien, esta opción permite comparar el equipo de fantasía de cada usuario con los 5 mejores equipos de fantasía, o los equipos de fantasía con más puntos, comparando los puntos acumulados y también los costos de cada plantilla. Para esta parte, siempre se van a mostrar los 5 mejores equipos de fantasía o los que tienen más puntos acumulados por el momento o por fechas, los cuales pueden cambiar por cada carga de datos del administrador en el que, un equipo puede entrar entre los 5 mejores o salir. Cabe resaltar, que es necesario que en las comparaciones haya una autenticación de que cada equipo exista y que efectivamente sean los que más puntos tienen acumulados.

Progreso/estadísticas por equipo: En cuanto al progreso/estadísticas por equipo, es necesario que exista de por sí una carga de datos y que un usuario tenga por lo menos un equipo de fantasía. Ahora bien, para sacar las estadísticas por equipo, es necesario que el administrador haga una carga de datos de las estadísticas de todos los jugadores por partido junto con sus puntajes, esto con el fin de que la aplicación pueda sacar, de todos los equipos, un top 5 de los jugadores que más puntos le han otorgado al equipo en todas las fechas acumuladas, puesto que con esto es posible obtener a los jugadores que tienen una mayor influencia en el equipo de fantasía. Sin embargo, es necesario hacer una autenticación, nuevamente, de los jugadores del equipo de fantasía, en el que para sacar este top 5 es necesario mirar si ese jugador efectivamente le pertenece al equipo de fantasía y que exista en la base de datos de la aplicación misma.

Diseño Final

- ```

classDiagram
 class PantallaNominasCargaLiga {
 + nombre liga :JLabel
 + nombre liga :JTextField
 + cargar archivo csv :JButton
 + guardar :JButton
 + tablaDatos :JTable
 + regresar menu registro de nominas :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosNominas()
 }
 class PantallaRegistrosNominas {
 + Cargar equipo :JButton
 + Cargar Liga :JButton
 + regresar menu registros :JButton
 + actionPerformer(ActionEvent) void
 + PantallaNominasCargaLiga()
 + PantallaNominasCargaEquipo()
 }
 class PantallaNominasCargaEquipo {
 + nombre equipo :JLabel
 + nombre equipo :JTextField
 + cargar archivo csv :JButton
 + guardar :JButton
 + tablaDatos :JTable
 + regresar menu registro de nominas :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosNominas()
 }
 class PantallaRegistrosAdministrador {
 + Registrar resultados :JButton
 + Registrar Nominas :JButton
 + Menu registros :JLabel
 + configuracion :JButton
 + regresar menu administrador :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosDeResultados()
 + PantallaRegistrosNominas()
 }
 class PantallaRegistrosDeResultados {
 + Registro de resultados :JLabel
 + fecha partido :JLabel
 + equipo local :JLabel
 + equipo visitante :JLabel
 + marcador local :JTextField
 + marcador visitante :JTextField
 + fecha partido :JTextField
 + equipo local :JTextField
 + equipo visitante :JTextField
 + marcador local :JTextField
 + marcador visitante :JTextField
 + cargar archivo csv :JButton
 + guardar :JButton
 + tablaDatos :JTable
 + registro de desempenos :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosDeDesempenos()
 }
 class PantallaInicioAdministrador {
 + Menu administrador :JLabel
 + registros :JButton
 + descripcion registros :JLabel
 + configuracion :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosAdministrador()
 + PantallaConfiguracionAdministrador()
 }
 class PantallaConfiguracionAdministrador {
 + creacion temporada :JButton
 + eliminar jugador :JButton
 + eliminar equipo :JButton
 + regresar menu administrador :JButton
 + actionPerformer(ActionEvent) void
 + PantallaCrearTemporada()
 + PantallaEliminarEquipo()
 + PantallaEliminarJugador()
 }
 class PantallaCrearTemporada {
 + nombre liga :JLabel
 + nombre liga :JTextField
 + nombre equipo :JTextField
 + nombre equipo :JLabel
 + cargar archivo csv liga :JButton
 + cargar archivo csv equipo :JButton
 + guardar Liga :JButton
 + tablaDatos :JTable
 + guardarEquipo :JButton
 + regresar menu configuracion :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosNominas()
 }
 class PantallaEliminarEquipo {
 + nombre equipo :JLabel
 + nombre equipo :JTextField
 + eliminar equipo :JButton
 + regresar menu configuracion :JButton
 + tablaDatos :JTable
 + actionPerformer(ActionEvent) void
 + PantallaConfiguracionAdministrador()
 }
 class PantallaEliminarJugador {
 + nombre Jugador :JLabel
 + nombre Jugador :JTextField
 + eliminar jugador :JButton
 + regresar menu configuracion :JButton
 + tablaDatos :JTable
 + actionPerformer(ActionEvent) void
 + PantallaConfiguracionAdministrador()
 }
 class PantallaCrearUsuario {
 + Usuario :textField
 + contraseña :passwordField
 + crear usuario :JButton
 + registrar :JButton
 + IngresarNombre()
 + actionPerformer(ActionEvent) void
 }
 class PantallaLogin {
 + Usuario :textField
 + contraseña :passwordField
 + crear usuario :JButton
 + iniciar session :JButton
 + PantallaLogin(String usuario, contraseña)
 + actionPerformer(ActionEvent) void
 }
 class PantallaRegistroDeDesempenos {
 + nombre jugador :JLabel
 + equipo :JLabel
 + minutos jugados :JLabel
 + goles :JLabel
 + autogoles :JLabel
 + asistencias :JLabel
 + position :JLabel
 + goles recibidos :JLabel
 + penales detectados :JLabel
 + penales errados :JLabel
 + tarjetas amarillas :JLabel
 + tarjetas rojas :JLabel
 + puntaje :JLabel
 + nombre jugador :JTextField
 + equipo :JTextField
 + minutos jugados :JTextField
 + goles :JTextField
 + autogoles :JTextField
 + asistencias :JTextField
 + position :JTextField
 + penales detectados :JTextField
 + penales errados :JTextField
 + tarjetas amarillas :JTextField
 + tarjetas rojas :JTextField
 + puntaje :JTextField
 + cargar archivo csv :JButton
 + guardar :JButton
 + tablaDatos :JTable
 + regresar menu registros :JButton
 + actionPerformer(ActionEvent) void
 + PantallaRegistrosDeResultados()
 }
 PantallaNominasCargaLiga --> PantallaRegistrosNominas
 PantallaRegistrosNominas --> PantallaNominasCargaEquipo
 PantallaRegistrosNominas --> PantallaRegistrosAdministrador
 PantallaRegistrosAdministrador --> PantallaRegistrosDeResultados
 PantallaRegistrosAdministrador --> PantallaInicioAdministrador
 PantallaInicioAdministrador --> PantallaConfiguracionAdministrador
 PantallaConfiguracionAdministrador --> PantallaCrearTemporada
 PantallaConfiguracionAdministrador --> PantallaEliminarEquipo
 PantallaConfiguracionAdministrador --> PantallaEliminarJugador
 PantallaCrearTemporada --> PantallaRegistrosNominas
 PantallaEliminarEquipo --> PantallaConfiguracionAdministrador
 PantallaEliminarJugador --> PantallaConfiguracionAdministrador
 PantallaCrearUsuario --> PantallaLogin
 PantallaLogin --> PantallaRegistrosNominas
 PantallaRegistrosDeDesempenos --> PantallaRegistrosDeResultados

```

- Que un jugador de un equipo de fantasía aparezca como uno de los más influyentes, cuando en realidad no debería serlo
- Que al abrir el mercado de compras estos mismos puedan cambiar las alineaciones de otros equipos de fantasía o no cambien
- Que las gráficas en las estadísticas no carguen o salgan errores
- Al editar posiciones del equipo pueden existir errores como poner a un delantero como arquero.

Estos son algunos errores que se deben tener en cuenta en la implementación los cuales el usuario, al usar la interfaz, no debería presenciarlos.