

Pablo Martínez

Juan Diego Sarmiento

Jose Manuel Botello

Entrega #1 Proyecto #3

Gracias a que este proyecto, tiene muchas similitudes a los anteriores proyectos, se volverá a utilizar la misma estructura que se tuvo en el documento entregado en el proyecto, pero se agregarán las cosas que se consideren pertinentes, se tendrá muchas cosas similares, por lo cual serán señaladas con un asterisco (*) para indicar no hubo cambio respecto a esta parte de contenido.

Alcance:

- Al tomar en cuenta el alcance de este proyecto debemos notar algunas de las restricciones que vamos a tomar en cuenta para desarrollar la solución, la primera que se puede notar se basa en el aspecto de actualización de información, esto se basa en como se va a actualizar la información, de la liga mientras ocurren varios partidos. Para esto, decidimos tomar un enfoque donde obtenemos la información de cada partido línea por línea en un documento CSV. Este documento se va a actualizar una vez los partidos ocurran (añadiendo otra línea al CSV).(*)
- Otras restricciones que pudimos notar para el desarrollo de este proyecto se vieron reflejados en el documento de análisis, pero de igual forma serán mencionadas levemente en este documento. Previamente incluimos restricciones con el ingreso del administrador y el usuario, donde ambos ingresan desde la misma aplicación, esto se podrá lograr a través del nombre del usuario del administrador y en este caso se cambiará las opciones del usuario/admin. (*)
- Para solucionar muchas de las restricciones establecidas en el documento de análisis que se relacionaban a la nómina se va a lograr con un método que altera el valor del atributo (Presupuesto) del usuario basado en los métodos de comprar_jugador(), vender_jugador() y cambiar_presupuesto().
- Las transacciones similarmente se lograrán con los métodos en el usuario y la relación que tiene con la clase dream_team
- Se deberá tener en cuenta todas las estadísticas que se obtengan por los CSV, para ir armando información respecto al desempeño de los partidos, además se tendrá

funciones especiales para volver a cargar estas estadísticas que se irán consiguiendo con cada partido que se obtenga.

No-Objetivos:

- Algunos aspectos que queremos dejar explícitamente afuera de la solución pueden incluir el hecho de que se utilicen jugadores de diferentes ligas de futbol.
- A la vez también queremos evitar la selección de jugadores en posiciones que no les corresponden, un ejemplo de esto sería utilizar un delantero como arquero para asegurar más puntos en la liga de fantasía.
- También queremos dejar afuera del producto la opción de repetir más de dos veces al mismo jugador en la liga de fantasía.
- También se tiene como objetivo dejar afuera la opción de alterar la alineación de suplentes y jugadores mientras ocurre el partido, el cambio debe tener cierto tiempo de notificación.
- Se pueden tener dos equipos por jugador, cada uno deberá tener nombre y alineación diferente.

Decisiones de diseño:

Para desarrollar muchos de los diagramas de diseño que se pueden ver en futuras páginas, tomamos en cuenta el diagrama desarrollado en la fase de análisis de la primera entrega, puede que estos no tengan cambios mayores ya que en general se tendrá la misma funcionalidad de todos los requerimientos, pero se añadirán cosas necesarias para cumplir con las nuevas especificaciones, que se mencionarán mientras aparezcan. Se implementará de nuevo con diagramas de secuencia y de caso de uso que determinan la usabilidad y funciones disponibles, a la vez en el diagrama de diseño aplicamos diversas relaciones que nos permitirán tener la información nueva y necesaria para tener un registro detallado de todos los equipos y jugadores.

También se debe tener en cuenta el uso de ligas pasadas y las estadísticas que están tengan, ya que agregarán más información respecto al control, además se tendrán nuevas formas para manejar excepciones, ya sea que no se tenga equipo y se desea realizar acciones, o no se tenga un equipo completo y se desee también participar, además de posibles argumentos y expresiones erróneas entradas por el usuario.

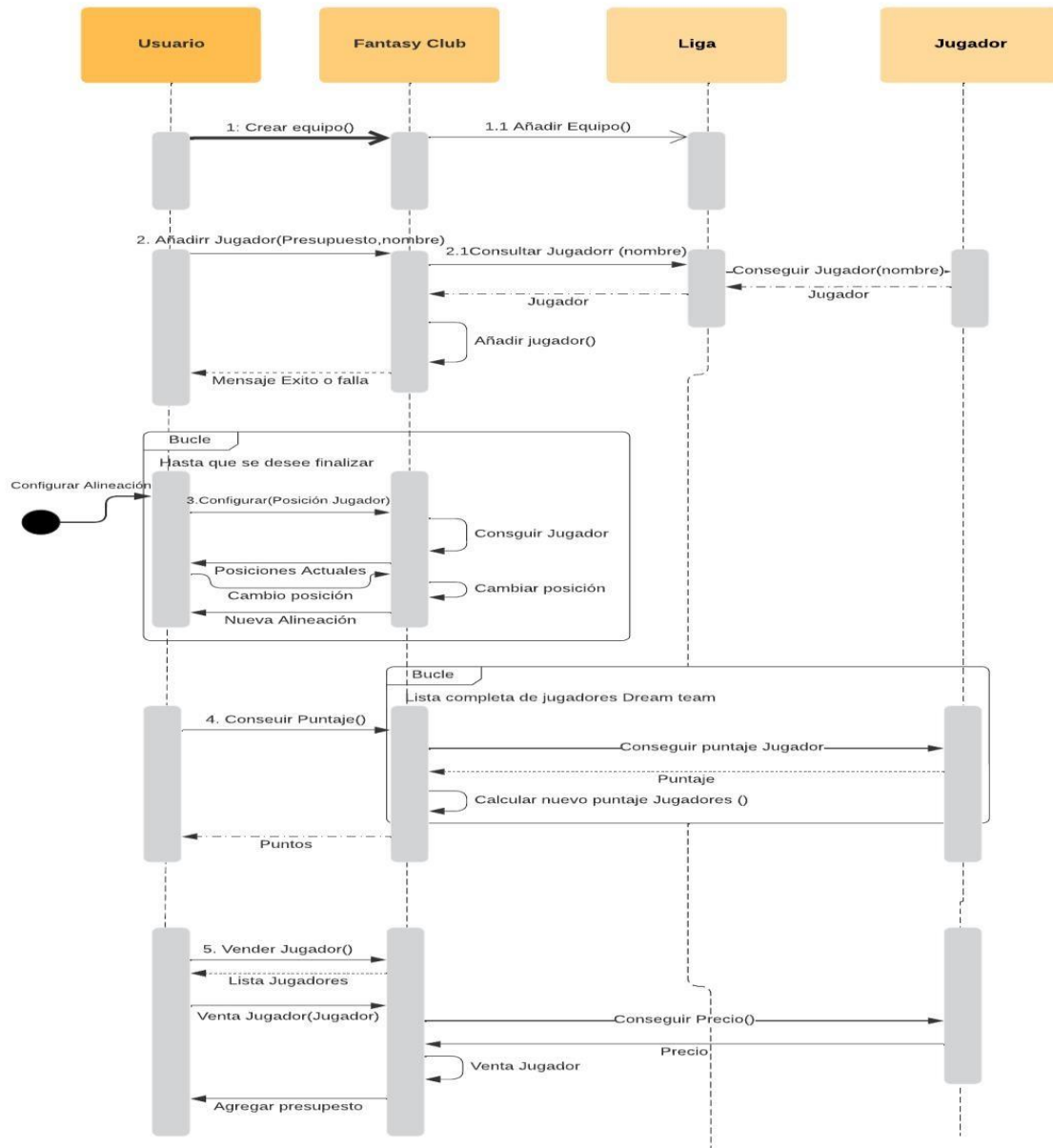
Casos de Uso:



Dentro del diagrama de casos de uso no se ve cambio alguno, ya que como tal la idea principal no se ve afectada, se tendrán los mismos usos por usuario. Estos usos en general no han cambiado, lo que cambiar será principalmente el alcance que se tenga en estos.

Diagramas de Secuencia - Usuario:

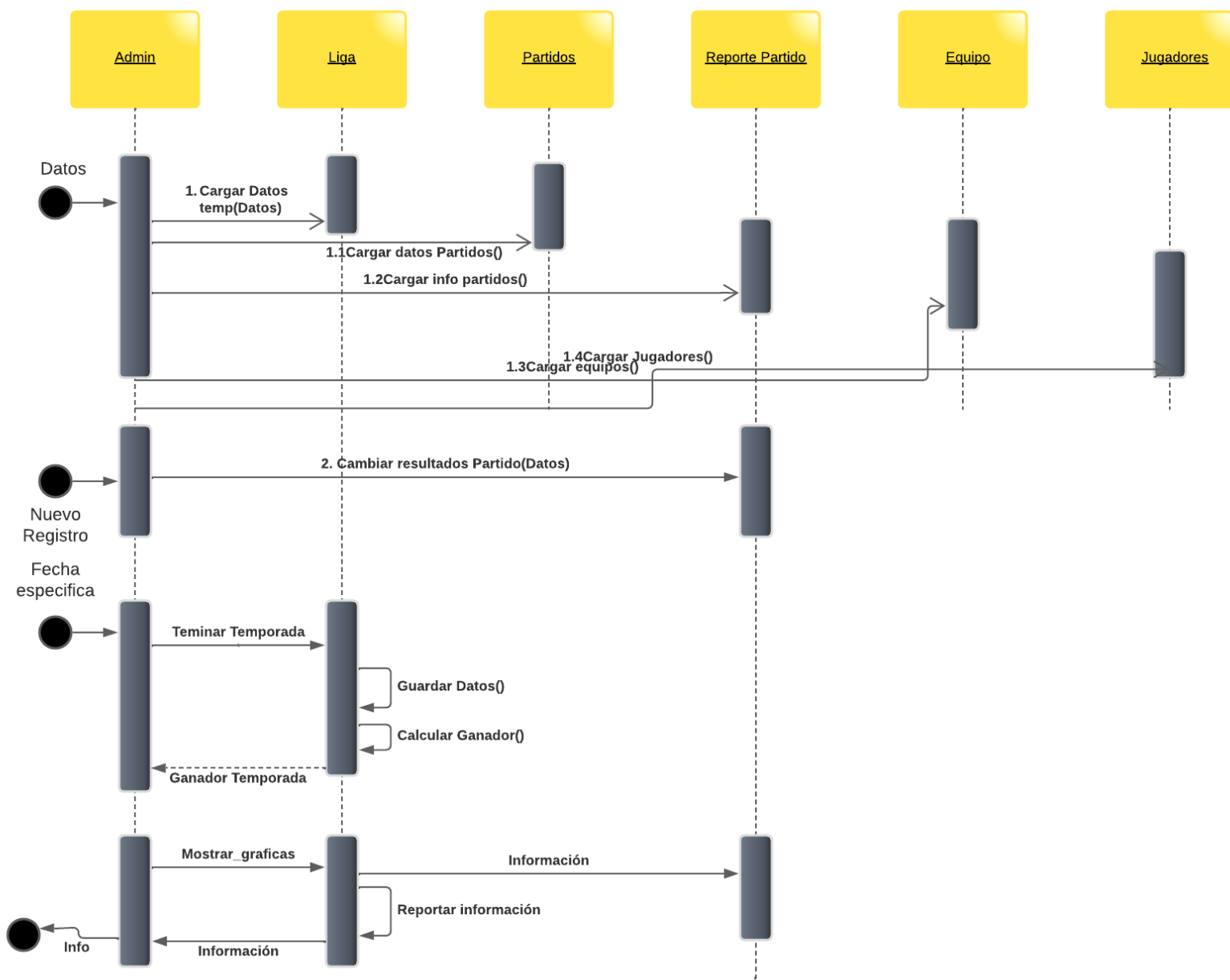
Diagrama de Secuencia, Caso de uso del Usuario



A su vez, este mismo no se afectado en su mayoría cambiaría en cada paso que se seleccione un equipo para el cual actualizar sus datos, ya que desde la perspectiva del usuario este no ve un mayor cambio, y/o debe realizar acciones extras respecto a las funciones principales. Lo nuevo se visualizará en la parte de la vista y será cargado por la clase Administrador y manejado por la clase de Liga.

Diagramas de Secuencia - Administrador:

DIAGRAMA SECUENCIA CASO DE USO ADMINISTRADOR



A diferencia de los anteriores este tiene más cambios, ya que el Administrador deberá actualizar los nuevos cambios y tener más capacidades para añadir los nuevos datos que se planean añadir reportes, pero en general tampoco será muy diferente.

Diagramas de Responsabilidad:

Roles:

Roles
R1: Admin
R2: Cliente
R3: Equipo
R4: Jugador
R5: Dream Team
R6: Partidos
R7: Reporte Partido
R8: Liga
R9: Temporada Pasada
R10: Penalties

Responsabilidades:

Responsabilidades
R1: Agregar Datos y Estadística para Gráficas [R1,R8]
r1: Registro de Usuario: [R1, R2]
r2: Registro de Administrador: [R1]
r3: Fecha Partidos: [R1, R3, R6, R7, R8]
r4: Crear Nuevo Fantasy Club: [R2, R5, R8]
r5: Agregar Jugador A Fantasy Club: [R2, R4, R8, R5]
r6: Escoger Capitán: [R2, R4, R5]
r7: Configurar Alineación: [R2, R4, R5]
r8: Default Alineación: [R3, R5, R8, R9]
r9: Puntaje Acumulado: [R2, R5, R8]
r10: Terminar la temporada: [R1, R5, R6, R8, R9]
r11: Venta de Jugador: [R2, R4, R5]
r12: Estadísticas [R1, R6, R7, R8, R9, R10]
r13: Ajustar Resultados [R1, R4, R5, R7]
r14: Set up Administrador [R1, R4]
r15: Contador puntos de la liga [R4, R5, R6, R7, R8]
r16: Ganador de la temporada [R1, R2, R5]
r11: Historial [R8, R9]

Colaboraciones:

Colaboraciones
C1: [R1, R2]
C2: [R1, R6]
C3: [R1, R7]
C4: [R1, R8]
C5: [R1, R9]
C6: [R2, R3]
C7: [R2, R4]
C8: [R2, R5]
C9: [R2, R9]
C10: [R3, R4]
C11: [R3, R6]
C12: [R3, R7]
C13: [R3, R8]
C14: [R4, R5]
C15: [R4, R6]
C:16: [R4, R7]
C17: [R4, R9]
C18: [R4, R10]
C19: [R5, R6]
C20: [R5, R7]
C21: [R5, R9]
C22: [R6, R7]
C23: [R6, R8]
C24: [R7, R8]

Diagramas UML:

El nuevo diagrama UML es más extenso y se tiene además las clases que se usarán dentro del modelo vista controlador, además de tener los métodos necesarios para implementar los nuevos requerimientos, pero en su generalidad no se verá afectado, ya que el objetivo es el mismo. Se ve estructuralmente como las clases tienen ciertas relaciones, ya sea de dependencia, cabe la pena recalcar que la App y controlador conectan la vista y el modelo, con la persistencia y respectivamente, ya que ambos son sistemas separados, se tiene una gran cantidad de clases y métodos independientes para evitar las dependencias y mejorar, pero aún se tiene un gran acoplamiento entre las partes respectivas, es decir entre el mismo modelo y vista.

