

Documento de diseño

Integrantes:

Moisés Agudelo 202113485

Santiago Díaz 201912247

Juan Orozco 202112353

Catalogo:

- Contexto
- Objetivos
- Dominio
- Requerimientos Funcionales
- Requerimientos No funcionales
- Diagrama de Secuencias

Contexto:

Dentro del proyecto del curso vamos a construir un sistema para un juego de “Fútbol de Fantasía”. Existen muchas implementaciones de este tipo de sistemas, no sólo para fútbol sino también para deportes como baseball, fútbol americano y baloncesto. La idea principal de una liga de “Fútbol de Fantasía” es siempre la misma: al inicio de una temporada, cada uno de los participantes debe armar un equipo de fantasía seleccionando jugadores de una liga real y esos equipos ganarán puntos de acuerdo con el desempeño de los jugadores reales. Por ejemplo, si el arquero real de un equipo tiene un buen partido en el que no le anoten ningún gol y además detenga un penalti, todos los equipos de fantasía que tengan en su alineación a ese arquero ganarán puntos por su buen desempeño. El objetivo que persiguen los participantes del juego es lograr acumular la mayor cantidad de puntos al final de la temporada.

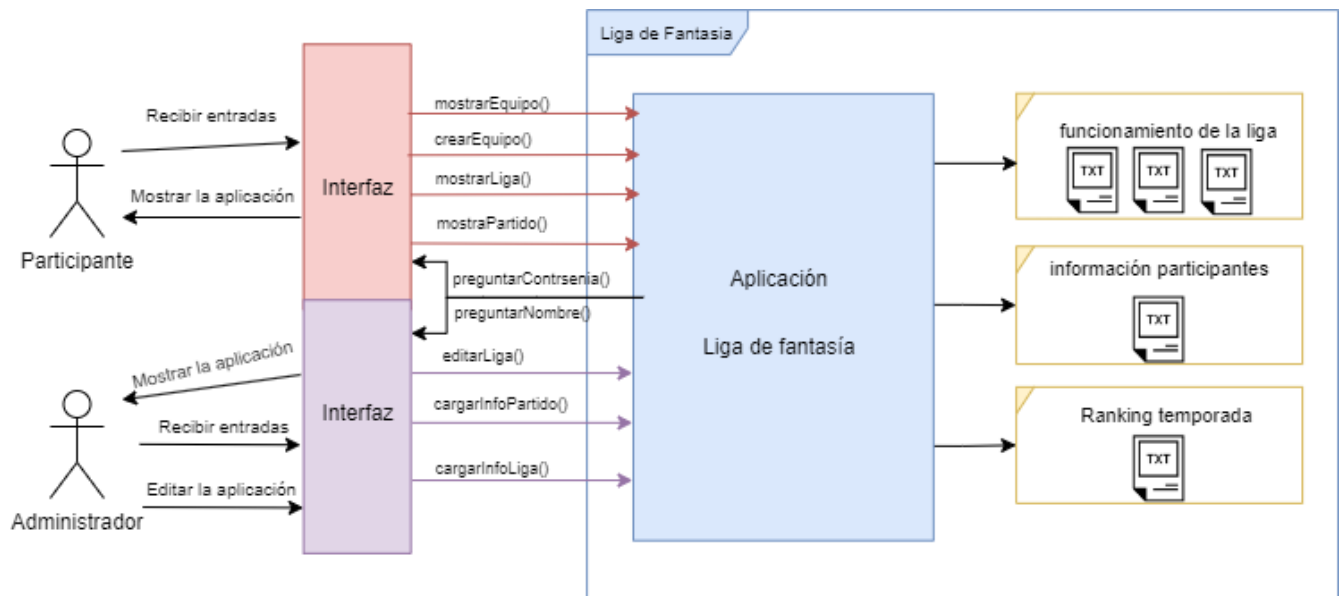


Figura del contexto

En relación con la interacción que el usuario tendrá con la aplicación como podemos ver en la figura dependerá del tipo de usuario que sea, en el caso del participante debe ingresar con su nombre y contraseña al ser una aplicación persistente toda la información ya estará guardada, pero en caso de ser la primera vez que el participante ingresa tendrá que entrar toda la información inicial para crear su equipo dentro de la aplicación y quedar registrado.

A su vez el administrador también deberá ingresar su nombre y contraseña, el administrador podrá editar y cargar información en todo momento, la interfaz debe solicitarle la carga de los datos de la liga y de los partidos, además de actualizar la información para la puntuación de los equipos y la clasificación del top-10 de la liga.

Objetivos:

El objetivo general de este proyecto es practicar varias etapas del desarrollo de una aplicación de software, desde el análisis hasta la construcción una aplicación funcional. A través del proyecto, los estudiantes pondrán en práctica todas las habilidades desarrolladas en el curso. Objetivos específicos del proyecto:

1. Identificar y abstraer entidades en un dominio particular, así como sus características y relaciones.
2. Construir diagramas de clase UML para expresar modelos de dominio y modelos de diseño.
3. Diseñar una aplicación basada en el paradigma orientado a objetos y expresar el diseño, justificando las decisiones importantes que hayan tomado.
4. Implementar un diseño utilizando el lenguaje de programación Java.

Tipos de roles:

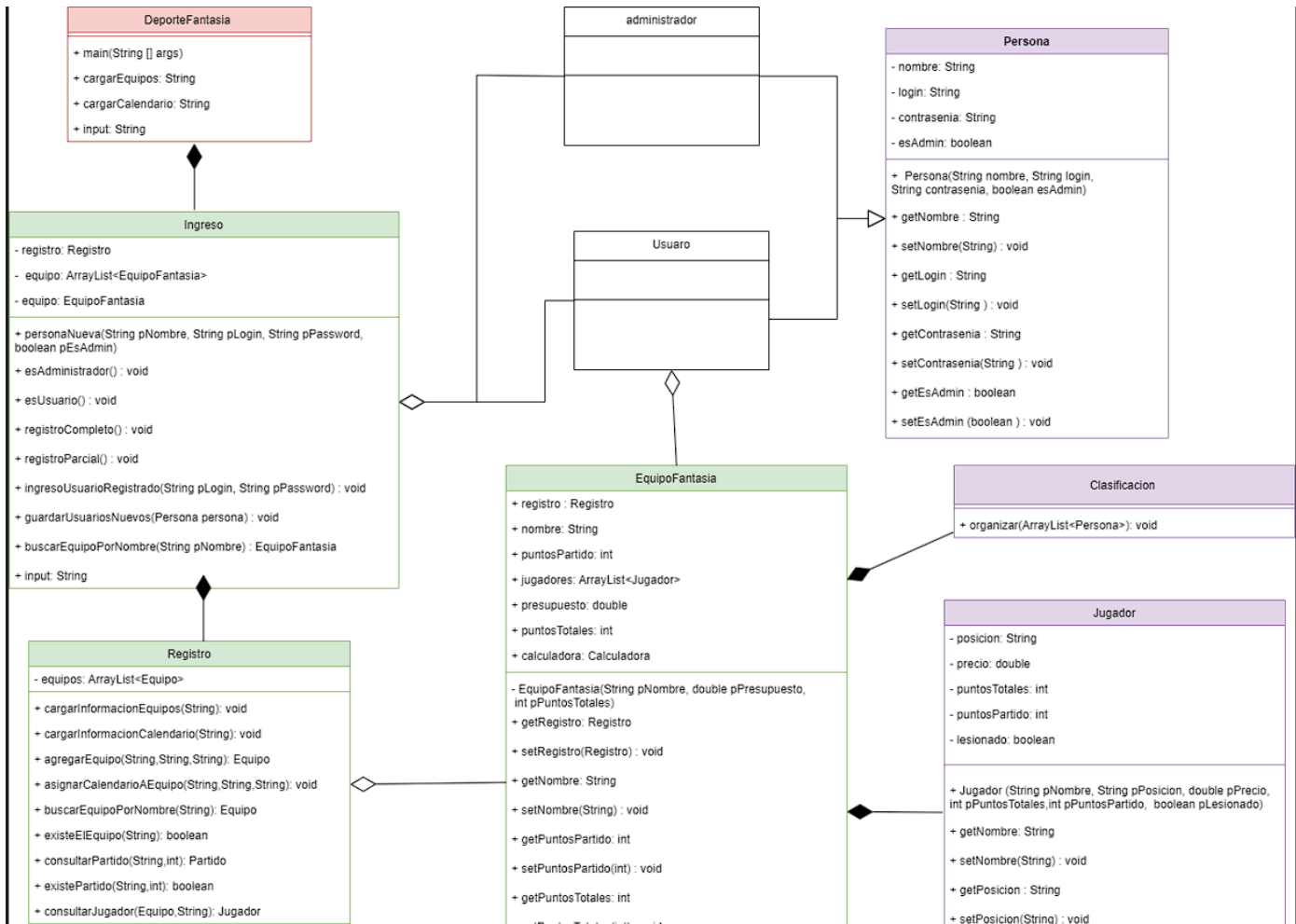
1. Information holder: mantiene y entrega información
2. Structurer: mantiene las relaciones entre objetos y provee información sobre esas relaciones
3. Service provider: hace algún trabajo para los demás (ofrece un servicio complejo)
4. Coordinator: su principal responsabilidad es delegar tareas a otros componentes
5. Controller: toma decisiones importantes y controla las acciones de otros componentes
6. Interfacer: transforma información y peticiones entre diferentes partes de un sistema

Roles

- Deporte Fantasía: Structurer, al ser la clase que creara el menú se encarga de conectar con las demás clases y darles la información para que se ejecuten.
- Ingreso: information holder, porque esta clase se encargará de guardar los datos de los usuarios que ingresan a la aplicación
- Registro: Service provider , porque está la clase que registra toda la información para el funcionamiento general de la liga de fantasía.
- Calculadora: Service provider, porque esta clase se encarga de hacer todos los cálculos necesarios para dar la puntuación de los jugadores y equipos.
- Clasificación: information holder, porque es la clase donde se tomarán los puntos de todos los equipos fantasía y se clasificarán en base a los puntos obtenidos
- Equipo: information holder, debido a que guarda la información del equipo
- Calendario: information holder, debido a que guarda una lista de partidos.
- Partido: information holder, debido a que se almacena la información de los partidos.
- Jugador: information holder, debido a que se guardan los Jugadores que conforman al equipo.

Dominio:

El diagrama de diseño planteado para resolver el proyecto será el siguiente (el diagrama se encuentra en el repositorio de git para mejor resolución):





Como podemos observar el diagrama cuenta con varias clases:

- **Deporte Fantasía:** es la clase donde se creará el menú por consola, será la encargada de interactuar con el usuario y de hacer las búsquedas y llamados a métodos.
- **Ingreso:** esta clase se encargará de guardar los datos de las personas que ingresan a la aplicación, si la persona es nueva deberá realizar el registro a la aplicación; de lo contrario, la persona tendrá que ingresar sus datos de sesión para poder continuar. Si la persona es de tipo administrador, esta podrá acceder a clases del lado izquierdo del diagrama (las clases de registro), mientras que si es un usuario podrá acceder a las clases del juego propiamente (la clase fantasía).
- **Persona:** Es una agregación de ingreso, puede ser o administrador o persona, y guarda datos de cada persona que entra en la aplicación y se registra.
- **Registro:** La clase registro es la encargada de registrar los datos de la competencia real, equipos, calendario, partidos y jugadores serán incluidos en la base de datos del programa. Primero se realizará un registro de inicio de temporada con toda la información de la competencia, y cada vez que se jueguen partidos de la competición, se hará un registro de puntos de los jugadores.
- **Equipo:** Clase donde se guarda la información del equipo
- **Calendario:** Es una clase conformada por una lista de partidos.

- Partido: Clase donde se almacena la información de los partidos que jugara el equipo.
- Jugador: Jugadores que conforman al equipo.
- Fantasía: Esta clase es la que representa el juego en sí, aquí el usuario podrá crear su equipo de 15 jugadores y competir con el resto de las personas. Podrá realizar fichajes, vender jugadores y acumular puntos.
- Calculadora: Esta clase se encarga de calcular la cantidad de puntos que un equipo fantasía obtiene, esto teniendo en cuenta los puntos registrados por distintos jugadores en los partidos y los taxes que tenga el equipo.
- Clasificación: La clase donde se tomarán los puntos de todos los equipos fantasía y se clasificarán en base a los puntos obtenidos.

Funcionalidad:

Nuestro programa intenta modelar un equipo de fantasía de la liga premier de Inglaterra, para ello se crearon diferentes files txts con todos los datos necesarios para cargar la aplicación. Solo se tuvo en cuenta los datos completos de un equipo, ya que si no se hiciese así nos tomaría excesivo tiempo rellenar cada dato de cada equipo real (sin embargo, dentro de la aplicación se pueden modificar estos datos para que queden lo más detallados y actualizados posibles).

Para el óptimo funcionamiento de nuestro programa es necesario crear un usuario. Si la persona es de tipo administrador podrá cargar, modificar y consultar toda la información de la liga. Si la persona es de tipo usuario podrá crear o cargar su Equipo Fantasía, en el cual podrá fichar jugadores, cambiar jugadores y consultar sus puntos. Los datos de los usuarios son impresos en un documento tipo txt para guardar su información.

Requerimientos Funcionales:

1. La aplicación debe tener dos tipos de usuarios, administrador y participante.
2. Los usuarios deben poder identificarse con su nombre de usuario y contraseña.
3. El administrador debe poder configurar y registrar para la creación de la liga de fantasía .
4. Debe existir una clase que guarden las fechas de la liga
5. Debe existir una clase que guarde la información de todos los partido.
6. El administrador debe poder registrar el desempeño de los jugadores.
7. El administrador de poder cargar la información de cada partido
8. Debe existir una clase que guarde el equipo de cada participante
9. debe poder comprobarse que la alineación cumple con los parámetros. (15 jugadores, 2 arqueros, 5 defensores, 5 mediocampistas y 3 delanteros), además también debe cumplir con el presupuesto.
10. El participante de poder crear su equipo de fantasía con los parámetros que se definen al principio de la liga.
11. El participante debe poder vender y comprar jugadores con su presupuesto.
12. El participante debe poder definir una alineación para cada partido de la liga.
13. Debe existir una clase que guarde la información sobre la clasificación de la liga de fantasía.
14. El usuario debe poder ver la información de la clasificación junto con información importante sobre la temporada.

Requerimientos no Funcionales:

Usabilidad:

- La aplicación debe proporcionar mensajes de error.
- La aplicación debe mostrar ayuda visual para que el usuario sepa que se le pide en cada interacción.
- La interacción por consola debe estar bien estructurada para que el usuario sepa en todo momento que acciones puede realizar y cual está realizando en cada momento.

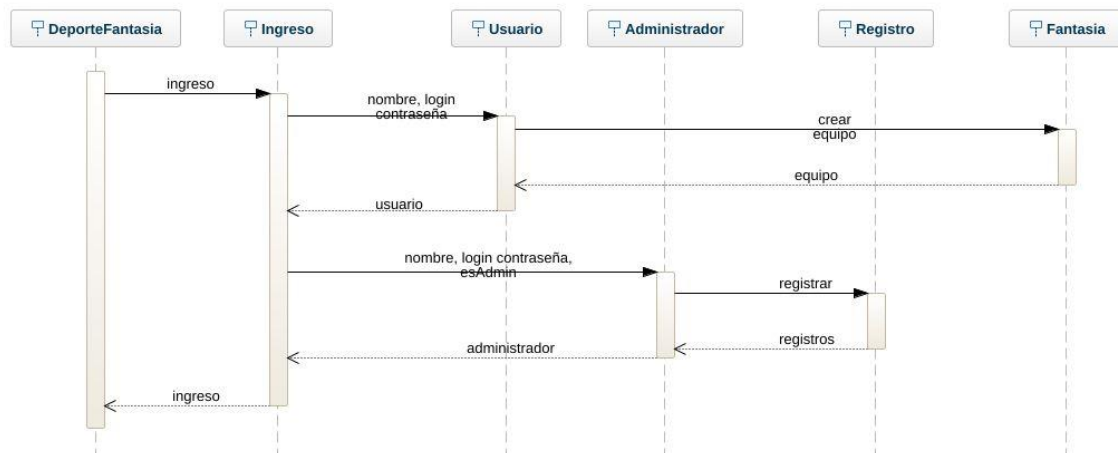
Escalabilidad:

- La aplicación debe estar preparada para contener cada vez más datos.
- La aplicación debe tener estructura de datos que sean estructuradas para guardar grandes cantidades de datos.
- La aplicación debe tener un tiempo de respuesta aceptable a medida que aumenta la carga de dato que contendrá la misma

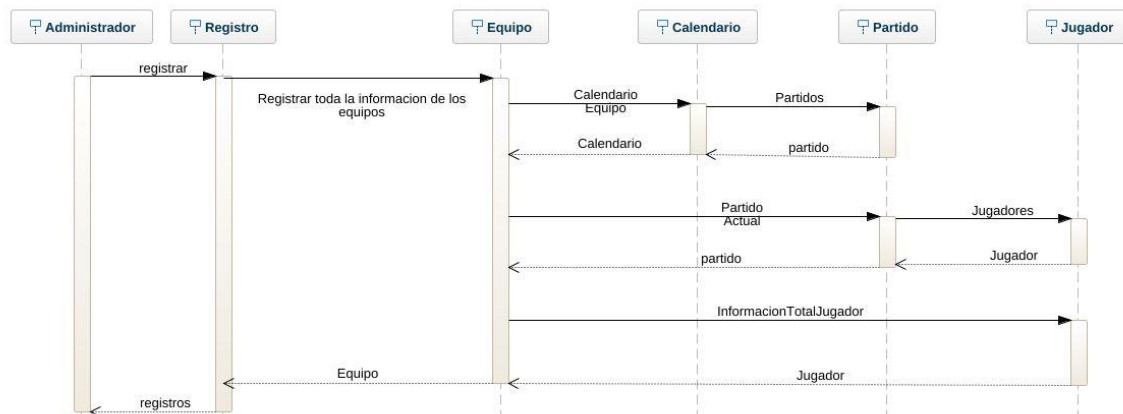
Diagramas de Secuencia:

Nota: todos los diagramas se encuentran en la carpeta image del repositorio.

El primer diagrama nos muestra el proceso de ingreso a la aplicación, como vemos la persona puede registrarse como usuario o administrador, se guarda su información, y dependiendo de su rol, este accede a una función o a otra.



En el segundo diagrama podemos ver las acciones que puede realizar un administrador, creando todo el sistema de registro y guardado de informacion.



Y por ultimo en el ultimo diagrama vemos las acciones que puede hacer un usuario al usar la aplicación, teniendo su equipo, el sistema de puntos y clasificacion respectiva.

