

## Proyecto 1 - Documento de diseño

### Nivel 1

#### 1.1 Componentes

- Dada la necesidad de tener un sistema de verificación de identidad (log-in), el primer paso para interactuar con la aplicación es una base de datos que compare los parámetros de nombre de usuario y contraseña con los registrados. Esta primera componente será un controlador de acceso.
- La aplicación posee dos tipos de cuentas, los administradores y los usuarios. Los administradores necesitan subir información sobre la liga real y sus equipos. Estos deberán poder hacer modificaciones sobre las jornadas y los jugadores. Sin embargo, los administradores no deberán poder controlar equipos de fantasía para mantener el *fair-play* (juego limpio). Por esto, el componente será el controlador de liga.
- Toda la información de la liga real, sus equipos, jugadores y jornadas deben estar en una base de datos propia a la que los administradores pueden agregar los resultados manualmente. Además, esta debe poder brindar información al controlador de equipos de fantasía. El tercer componente será la base de datos de la liga.
- El segundo tipo de cuentas, los usuarios, son los dedicados a crear equipos de fantasía. Estos pueden modificarlos mediante la compra-venta de jugadores reales, la selección de un capitán, la modificación de la alineación. Además, cada equipo deberá conocer los puntos generados por cada uno de sus jugadores en cada una de las fechas. El siguiente componente será el controlador de equipos de fantasía.
- Toda la información de los equipos de fantasía debe ser guardada y administrada. Esto implica guardar un equipo por cada uno de los usuarios. Este equipo guarda la información de los puntos totales y según la jornada. El quinto componente será la base de datos de los equipos de fantasía.
- Debido al requerimiento funcional de los rankings se considera un sistema capaz de ordenar la información presentada de los distintos equipos de fantasía según un criterio. El sexto componente será el ranking. Nota: Este ranking no permite todos los criterios sino los especificados en los requerimientos funcionales.

## 1.2 Responsabilidades

Cada uno de los componentes anteriormente mencionados obtuvo responsabilidades necesarias para el funcionamiento de la aplicación.

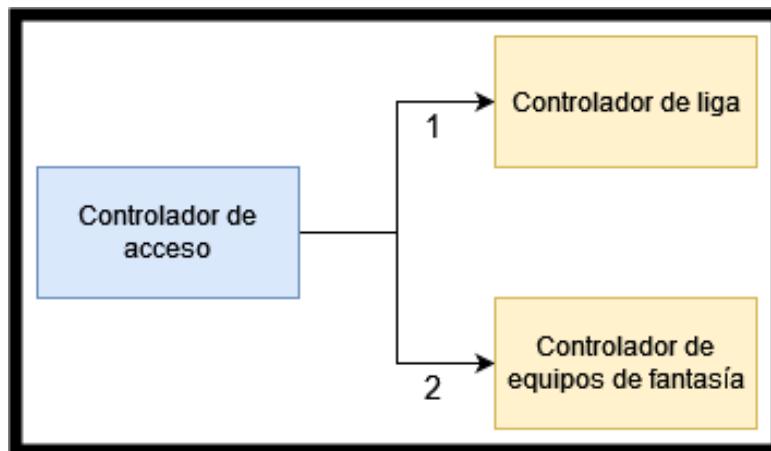
#	Responsabilidad	Componente
1	Iniciar sesión (usuario y contraseña)	Controlador de acceso
2	Guardar información de usuarios (usuario, contraseña, tipo de usuario)	
3	Acceder a menú (interfaz) según tipo de cuenta	
4	Cerrar sesión	
5	Crear liga y temporada (cargar datos de equipos, jugadores, jornadas y presupuesto)	Controlador de liga
6	Cargar resultados de cada jornada	
7	Enviar información actualizada a la base de datos de la liga (mostrar resumen de datos antes de enviar)	
8	Cancelar envío de datos	
9	Guardar información de la liga real (equipos, jugadores, jornadas y resultados)	Base de datos de la liga
10	Calcular los puntos obtenidos por cada jugador	
11	Enviar información de liga a controlador de equipos de fantasía	
12	Crear equipo de fantasía	Controlador de equipos de fantasía
13	Conocer presupuesto	
14	Comprar jugadores (límite de 15 jugadores por equipo)	
15	Vender jugadores (devuelve el 97% del valor al presupuesto)	
16	Modificar alineación para cada jornada (última modificación disponible: día anterior a media noche. En caso contrario, elige la de la jornada anterior)	
17	Seleccionar capitán del equipo de fantasía	
18	Consultar los puntos obtenidos por cada jugador en cada jornada	
19	Calcular la suma por jornada de los puntos	
20	Calcular los puntos totales de la liga del equipo	
21	Enviar actualización de equipo a la base de datos de los equipos de fantasía	
22	Eliminar equipo de fantasía	
23	Guardar información del equipo de fantasía (nombre usuario, equipo)	Base de datos de los equipos de fantasía
24	Calcular la suma por jornada de los puntos	
25	Calcular los puntos totales de la liga del equipo	
26	Enviar datos a otras componentes (controlador de equipos de fantasía y ranking)	Ranking
27	Ordenar información según criterio (puntos por equipo, por jugador o por jornada)	
28	Presentar información	

Nota: Con respecto a las responsabilidades 23 y 24 se asignaron a la base de datos debido a que estas operaciones necesitan darse independiente de la actividad de un usuario. Estos suceden en el ciclo de uso de un administrador.

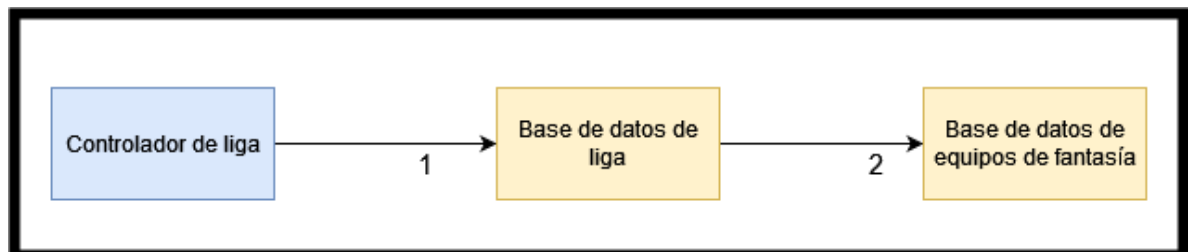
### 1.3 Colaboraciones

A pesar de que ya se definieron algunas en las responsabilidades, se busca aclarar la relación entre todas las componentes usadas.

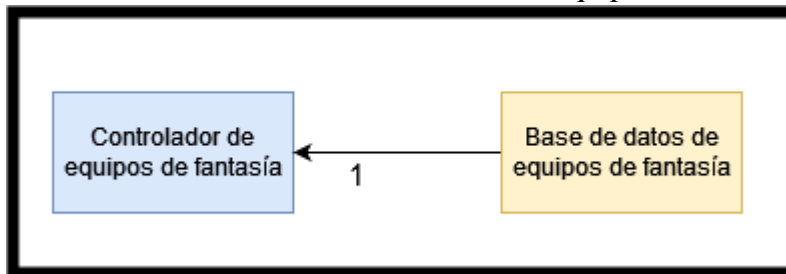
- Acceso de una cuenta: Después de hacer *log-in*, el controlador de acceso se comunicará con el controlador de liga o el controlador de equipos de fantasía según el tipo de cuenta con el que se acceda.
  1. Acceso a controlador de liga (usuario).
  2. Acceso a controlador de equipos de fantasía (administrador).



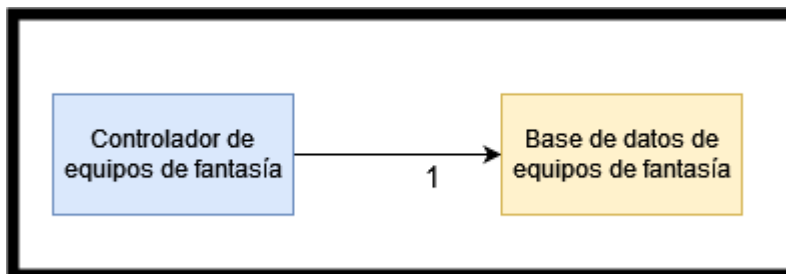
- Actualización datos liga y equipos de fantasía: Cuando un administrador sube los datos de la liga o los resultados de una jornada desde el controlador de liga, estos se envían a la base de datos de la liga.
  1. Guardar datos de liga o sobrescribir datos de jornada.
  2. Sobrescribir datos de jornada y puntos de los equipos de fantasía.



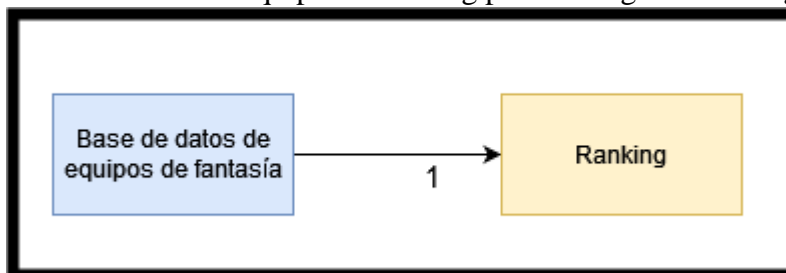
- Obtener datos de equipo de fantasía: Cuando se llega al controlador de equipo de fantasía el usuario puede pedir ver la información de su equipo.
  1. Envío de información histórica del equipo de fantasía del usuario



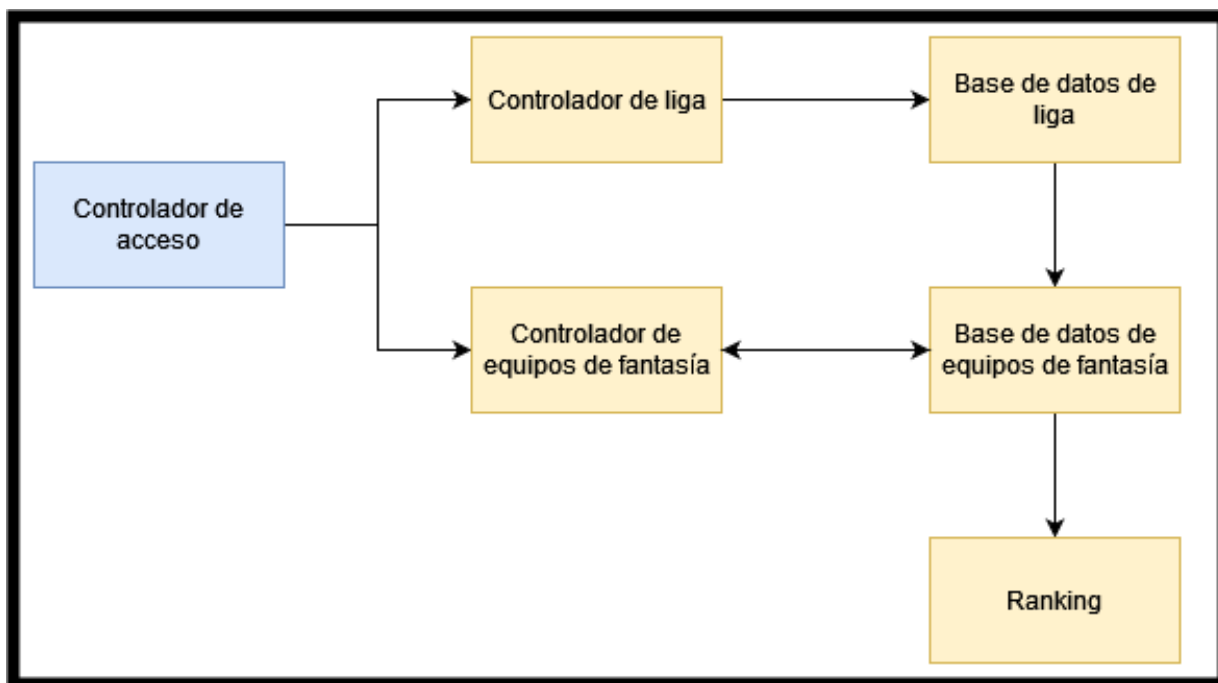
- Actualización plantilla o alineación: Al realizar alguna compra-venta de un jugador, al cambiar el capitán o modificar la alineación toda la información del controlador de equipos de fantasía se envía a la base de datos de los equipos de fantasía.
  1. Guardar modificaciones de controlador en la base de datos según la jornada cuando se realizó.



- Actualización ranking: La información obtenida de todos los equipos de fantasía es enviada a el componente Ranking.
  1. Enviar los equipos al ranking para ser organizados según el criterio dado.



Todo esto resulta en el siguiente diagrama de alto nivel, donde se observa cómo se organiza cada uno de los componentes según sus colaboraciones.



## Nivel 2

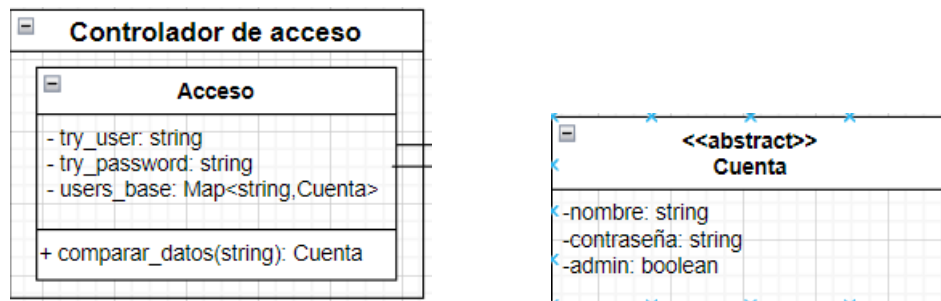
Continuando con el proceso de diseño por niveles, se pasa a descomponer cada componente presentado anteriormente.

### 2.1 Controlador de acceso

Para empezar, es importante mencionar las responsabilidades de este componente: iniciar sesión y guardar los datos en base al tipo de usuario (nombre de usuario, contraseña), acceder a la interfaz respectiva en base al tipo de cuenta (usuario, administrador) y cerrar sesión del usuario.

La clase Acceso se asegura de cumplir todas las responsabilidades de este. Para lograrlo, se tienen como atributos el input recibido de nombre de usuario y contraseña, así como el mapa que almacena el nombre de usuario como llave y la información de tipo Cuenta como valor. Así pues, el método que contiene esta clase toma el nombre de usuario que llega como parámetro y retorna la Cuenta relacionada con este.

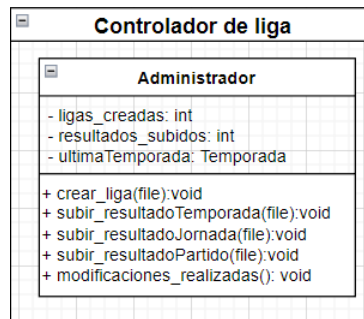
La clase abstracta Cuenta almacena la información del nombre de usuario y la contraseña ingresada, así como un atributo que toma el valor de True si la cuenta pertenece a un administrador y el valor de False si pertenece a un usuario.



### 2.2 Controlador de liga

La primera responsabilidad de este componente es lograr cargar los datos de equipos, jugadores, jornadas y presupuesto, creando así la liga y la temporada. Además, cumple la labor de cargar los resultados que se den en cada jornada, enviar información actualizada a la base de datos de la liga y ofrecer la opción de cancelar el envío de datos.

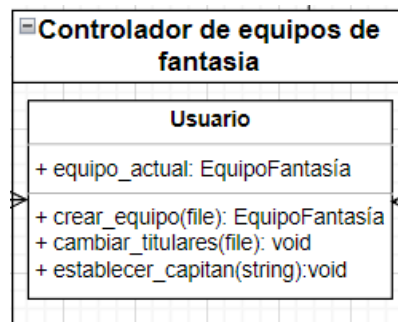
La clase Administrador, al ser la única que contiene la componente Controlador de liga, se encarga de satisfacer todas sus responsabilidades. Para lograrlo, se usan dos contadores que llevan cuenta de las ligas creadas y el número de resultados subidos por cada administrador. Además, se mantiene un registro del conjunto de resultados subidos por el administrador la temporada anterior. Así pues, se le brinda al administrador opciones para crear una liga, al igual que de subir los resultados de una temporada, una jornada y un partido. Finalmente, se ofrece la opción de revisar las modificaciones realizadas por este administrador.



### 2.3 Controlador de equipos de fantasía

El componente en cuestión se encarga de brindar al usuario las siguientes opciones: crear y eliminar un equipo de fantasía, conocer el presupuesto del equipo, comprar y vender jugadores, modificar la alineación para cada jornada, seleccionar al capitán del equipo, consultar los puntos obtenidos por cada jugador en cada jornada y, finalmente, calcular la suma de puntos por jornada y suma de puntos totales del equipo de fantasía.

La clase Usuario almacena como atributo toda la información del equipo de fantasía. Como métodos, logra crear un equipo de fantasía desde 0, así como de cambiar los titulares previos a cada jornada y seleccionar al capitán del equipo.



### 2.4 Base de datos liga

Este componente tiene la labor de guardar toda la información relacionada con la liga real (equipos, jugadores, jornadas y resultados), calcular los puntos obtenidos por cada jugador y de enviar toda esta información al Controlador de equipos de fantasía.

La clase Temporada guarda la información del nombre de esta, un mapa con sus jornadas y una lista con todos los equipos que la componen. La clase Temporada contiene también los procesos para añadir un equipo, añadir una jornada y añadir todos los

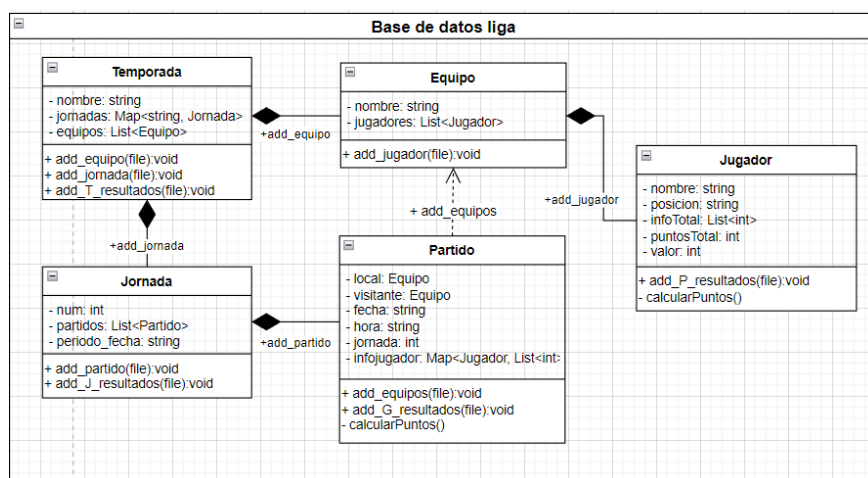
resultados de una temporada. Cabe resaltar que una temporada real se compone de jornadas y equipos.

La clase Jornada contiene información como: un entero que indica el número de la jornada real que se está disputando, una lista de partidos y un intervalo de fechas que indican entre qué días se disputan los partidos de esa jornada. Ofrece también la opción de añadir un partido y añadir los resultados de una jornada real. Cada jornada real está compuesta por partidos.

En la clase Partido se encuentra información sobre el equipo local, el equipo visitante, la fecha y hora en la que se va a disputar, el número de la jornada a la que pertenece y un mapa con el nombre de cada jugador como llave los puntos que sumó este durante le partido como valor. Esta clase también permite añadir equipos, añadir los resultados de un partido real y, finalmente, de calcular los puntos de los jugadores en base a la información recibida.

La clase Equipo solo contiene el nombre de cada equipo real y una lista con los jugadores que la componen. Así, se ofrece la opción de añadir a un jugador a dicha lista.

Finalmente, la clase Jugador almacena el nombre, posición, una lista con los puntos que ha obtenido en cada jornada, los puntos totales hasta ese momento y el valor económico de cada uno. Al igual que la clase Partido permite calcular los puntos del jugador y añadir los resultados de un partido.

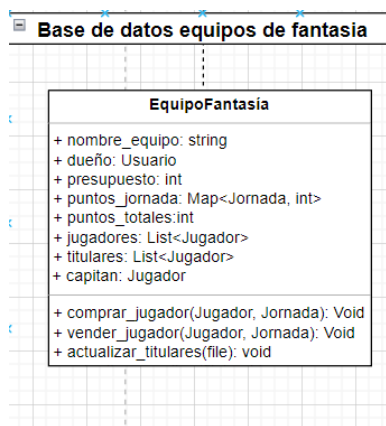


## 2.5 Base de datos equipo se fantasía

Esta componente guarda información del equipo de fantasía (nombre usuario, equipo), calcula los puntos por jornada y puntos totales de cada equipo. También debe ser capaz de enviar datos a otras componentes (controlador de equipos de fantasía y ranking).



En la clase EquipoFantasía encontramos almacenada información importante, entre la que se encuentra: el nombre del equipo de fantasía, el Usuario de su dueño, su presupuesto actual, un mapa con una Jornada como llave y el total de puntos sumados por el equipo como valor, los puntos totales de todo el equipo, una lista de todos sus jugadores, una lista con los jugadores titulares y el capitán para cada jornada. Además, presenta las opciones de vender y comprar jugadores al igual que de actualizar los titulares para cada jornada.



## 2.6 Ranking

Las responsabilidades de este componente son las siguientes: ordenar información según criterio (puntos por equipo, por jugador o por jornada) y presentar información solicitada.

La clase Ranking, única dentro de este componente, almacena una lista de equipos de fantasía, una lista de jugadores y la categoría por la que se quieren ordenar los rankings. Así pues, es capaz de, recibido las listas de jugadores/equipos y la categoría por la que se quiere ordenar, modificar las listas para que queden en el orden establecido.



## Diseño final

