

## Objetivo general del proyecto

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

Durante el desarrollo de este proyecto se buscará el desarrollo de las siguientes habilidades:

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.

## Instrucciones generales

A continuación, encontrará la definición inicial del proyecto, que será la base del trabajo de los 3 proyectos que se construirán durante todo el semestre.

El trabajo del proyecto #2 debe partir de la base del proyecto #1, y el trabajo del proyecto #2 será la base del proyecto #3. Si es necesario, podrán hacerse cambios en cada una de las etapas para mejorar o corregir decisiones que se hayan tomado en las etapas anteriores.

En el proyecto 1 no es necesario que haya una interfaz gráfica: la implementación que realicen puede tener una interacción con el usuario basada en consola.

El proyecto debe desarrollarse en **grupos que deben mantenerse durante todo el semestre**.

## Contexto del proyecto

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.

El sistema que vamos a construir lo usaremos para construir específicamente una liga de Fútbol de Fantasía y debe tener las siguientes características particulares, la mayoría de las cuales están inspiradas en plataformas existentes:

1. Los usuarios del sistema (los participantes) se identificarán con un nombre de usuario y una contraseña, y podrán tener un equipo de fantasía por temporada.
2. El sistema también tendrá un administrador con un nombre de usuario y una contraseña. El administrador podrá configurar todos los aspectos de una temporada, así como registrar la información de los partidos que vayan ocurriendo.
3. Una temporada (del campeonato real) tiene una cierta cantidad de fechas y en cada una se llevan a cabo los partidos. Usualmente (no siempre), todos los equipos que hacen parte de la liga (real) jugarán un partido en cada fecha. Antes del inicio de la temporada se conocerá la planeación del campeonato, fecha por fecha. En particular, se sabrá qué equipo jugará contra qué equipo, quién será el local, y para qué día y hora está programado el encuentro.
4. Al finalizar cada partido del campeonato, el administrador registrará el resultado y un registro detallado del desempeño de cada jugador (más abajo se encuentra la descripción de la información mínima necesaria).
5. Al inicio de cada temporada, el administrador registrará también las nóminas de los equipos reales: cada jugador tendrá un nombre, una posición (arquero, defensor, mediocampista, delantero) y un precio. Tanto las nóminas de los equipos reales como los precios de los jugadores se mantendrán fijas durante toda la temporada.
6. Un equipo de fantasía debe tener un nombre y en todo momento estará compuesto por 15 jugadores: 2 arqueros, 5 defensores, 5 mediocampistas y 3 delanteros (todos deben ser jugadores de la liga real).
7. Los usuarios del sistema podrán crear nuevos equipos de fantasía en cualquier momento de la temporada y se convertirán en los dueños del equipo. Equipos que hayan sido creados después del inicio de la temporada estarán en desventaja frente a los que hayan sido creados antes del inicio (empezarán con 0 puntos acumulados).
8. Cuando se cree un nuevo equipo de fantasía tendrá un presupuesto inicial, cuyo valor se definirá antes del inicio de la temporada. El dueño del equipo deberá usar ese presupuesto para “comprar” jugadores para su equipo, teniendo en cuenta que el valor total de sus jugadores no puede superar el presupuesto disponible y que el equipo debe tener siempre 15 jugadores con la distribución explicada antes.
9. Dos equipos de fantasía pueden incluir al mismo jugador real en su nómina.
10. Los equipos no están obligados a agotar su presupuesto.
11. Antes de cada fecha del campeonato, los dueños de los equipo de fantasía deberán configurar su alineación para la fecha: seleccionará a los 11 titulares (1 arquero, 4 defensores, 4 mediocampistas y 2 delanteros) y designará a uno de estos 11 como capitán del equipo. La configuración de un equipo se podrá realizar hasta la media noche anterior al inicio del primer partido de la fecha. La idea de tener jugadores suplentes es cubrir el riesgo de que algunos de los jugadores seleccionados para un equipo no jueguen ni siquiera un minuto en los partidos de la fecha.
12. Si un jugador no configura la alineación de su equipo para una fecha, se asume que usará la misma configuración de la fecha anterior.

13. Antes de cada fecha, el dueño de un equipo puede vender y comprar jugadores. Al vender un jugador, el equipo recibirá una cantidad de dinero equivalente al 97% del valor del jugador (es decir que pagará un 3% de comisión por la transacción). Sin embargo, al comprar un jugador se pagará siempre por su valor completo. Esto quiere decir que un equipo irá perdiendo parte de su presupuesto a medida que compre y venda jugadores. Recuerde que un equipo no puede tener ni más ni menos de 15 jugadores.
14. A medida que se vayan registrando los resultados de los partidos y los registros de desempeño de los jugadores, se irán actualizando los puntajes de los equipos de fantasía. La cantidad de puntos se asignarán de acuerdo con las reglas que se describen más abajo.
15. Al finalizar la temporada, el ganador de la liga de fantasía será el equipo que haya acumulado más puntos.
16. La aplicación debe ser capaz de mostrar estadísticas interesantes sobre el campeonato en cualquier momento. Por ejemplo, el ranking de los equipos de fantasía, los jugadores con más y menos puntos acumulados, el mejor equipo de fantasía de cada fecha, etc.

### Desempeño de los jugadores

Al finalizar cada partido del campeonato, el administrador del sistema cargará un reporte del partido que incluirá el resultado y el desempeño de los jugadores que participaron en el partido. Este reporte debe tener al menos la siguiente información de cada jugador:

- Cantidad de minutos jugados, el minuto en que ingresaron y el minuto en el que fueron sustituidos si no terminaron el partido.
- Goles y penaltis anotados.
- Autogoles.
- Asistencias
- Goles recibidos (sólo aplica para los porteros y defensas).
- Penaltis detenidos (sólo aplica para los porteros)
- Penaltis errados.
- Tarjetas amarillas y rojas recibidas.

Esta información es la mínima requerida: ustedes pueden agregar información adicional si la consideran interesante.

### Reglas para la puntuación

Después de registrar el reporte de un partido, el sistema actualizará los puntos acumulados por los jugadores en la liga de fantasía y los puntos acumulados por los equipos de fantasía. Para actualizar los puntajes de los equipos, sólo se deberían tener en cuenta los jugadores que hayan sido designados como titulares para la fecha, a menos que el jugador no haya jugado ni siquiera un minuto en esa fecha. En esos casos, serán reemplazados automáticamente por el suplente correspondiente (el arquero por el arquero, el defensa por el defensa, etc.). Si llega a ocurrir que más de uno de los jugadores titulares que ocupan la misma posición se quedan sin jugar, sólo uno será sustituido por el suplente.

Los siguientes son los puntos que deben asignarse por jugador:

- 1 punto por jugar hasta 60 minutos.
- 2 puntos por jugar más de 60 minutos.
- 4 puntos por gol anotado por un delantero.

- 5 puntos por gol anotado por un mediocampista.
- 6 puntos por gol anotado por un arquero o un defensor.
- 3 puntos por cada asistencia para gol.
- 4 puntos para un arquero o un defensa cuyo equipo no haya recibido un gol.
- 5 puntos para un arquero que haya atajado un penalti.
- 5 puntos para el capitán del equipo, si su equipo real ganó el partido.
- -2 puntos por errar un penalti.
- -1 punto por una tarjeta amarilla.
- -3 puntos por una tarjeta roja.
- -2 puntos por un autogol.

Recuerde que estos puntos sólo pueden ser aplicados a jugadores que hayan jugado al menos un minuto en la fecha. Los puntos se aplicarán sólo a los jugadores que hayan sido designados como titulares por el dueño del equipo (o a los suplentes que hayan tenido que jugar porque el titular correspondiente no hubiera jugado).

## Aspectos técnicos y restricciones

Toda la información debe ser persistente (las listas de equipos y jugadores, la conformación de los equipos de fantasía, los resultados de los partidos, el desempeño de los jugadores, los puntos asignados a cada equipo, etc.). La información debe almacenarse en archivos de texto dentro de una carpeta que se comparta a través de GIT con todas las personas que estén participando en el juego. Vamos a suponer que sólo la aplicación va a escribir y leer de esa carpeta (ningún usuario malicioso va a modificar los archivos que ahí se encuentren sin utilizar la aplicación). La carpeta no puede ser la misma carpeta donde se encuentre el código fuente de la aplicación.

La sincronización usando GIT no tiene que hacerse desde la aplicación: los usuarios deben encargarse de sincronizar la carpeta usando algún cliente de GIT.

La persistencia no necesariamente debe hacerse en un solo archivo: diseñe con cuidado cuántos archivos habrá y cómo van a estar estructurados para disminuir la posibilidad de conflictos.

Tanto el administrador como los usuarios “normales” del sistema deben usar la misma aplicación: dependiendo del tipo de usuario, las opciones que se muestren deben ser diferentes.

Se sugiere que para configurar una nueva temporada el administrador pueda cargar un archivo JSON o CSV con la información de los jugadores y equipos. Además, para reportar el resultado de los partidos y el desempeño de los jugadores el administrador también debería cargar un archivo por partido <sup>1</sup>.

Para fines de auditoría, debe ser posible reconstruir todo lo sucedido durante el campeonato (qué equipos de fantasía había, qué alineaciones tenían, etc. ).

Vamos a suponer que todos los usuarios del sistema, así como los partidos, utilizan la misma zona horaria. Es decir que las fechas y horas no necesariamente tienen que tener en cuenta la zona horaria (ej. GMT -5).

La aplicación debe estar hecha en Java y la interfaz debe estar basada en consola.

No es necesario que se soporten funcionalidades que no hayan sido mencionadas, pero se pueden incluir funcionalidades adicionales si facilitan el trabajo.

---

<sup>1</sup> La aplicación también podría tener opciones para registrar la información de los jugadores uno por uno, pero seguramente eso será mucho más difícil de probar que si se pueden cargar archivos enteros con la información.

## Entrega 1: Análisis del proyecto

La primera etapa del proyecto consiste en realizar el análisis del sistema que debe construir.

### Actividades

1. Construya un modelo de dominio (diagrama de clases) a partir de la información del caso:
  - a. Identifique las entidades que aparecen dentro del caso.
  - b. Identifique las características (atributos) de esas entidades.
  - c. Establezca las relaciones entre las entidades, incluyendo asociaciones y relaciones de herencia.
2. Construya un documento de análisis para su proyecto. El documento debe incluir:
  - a. El modelo de dominio que construyó en el paso anterior.
  - b. Una descripción de los requerimientos funcionales que su aplicación soportará. Le recomendamos utilizar historias de usuario.
  - c. Una descripción con las restricciones del proyecto.

### Entrega

1. El proyecto debe entregarse en una carpeta dentro del repositorio GIT del grupo con el nombre **“Proyecto 1”**. Dentro de esta carpeta debe existir una carpeta con el nombre **“Entrega 1”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como imágenes que se puedan leer con facilidad.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **“Proyecto 1 - Entrega 1”**.

## Entrega 2: Diseño

Teniendo en cuenta el análisis realizado en la primera entrega del proyecto, para la segunda entrega debe realizar el diseño detallado del sistema.

### Actividades

1. Realice el diseño y construya un documento de diseño donde presente el diseño, haciendo especial énfasis en justificar las decisiones clave que haya tomado. El documento debe incluir, como mínimo, los siguientes elementos:
  - a. Un diagrama de clases de diseño que incluya todas las clases, incluyendo sus las relaciones, atributos y métodos.
  - b. Un diagrama de clases de alto nivel, que incluya todas las clases y sus relaciones, pero no todos los métodos ni atributos. Este diagrama facilitará entender las relaciones entre clases.
  - c. Diagramas de secuencia para las funcionalidades que usted considere críticas:Estos elementos NO son los únicos que debe incluir su documento: piense en qué otros diagramas, descripciones y justificaciones pueden ser necesarias para poder entender totalmente el diseño. Si su documento incluye únicamente los tres elementos mencionados antes, con seguridad será considerado insuficiente.

## Entrega

---

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre **“Entrega 2”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como el documento de diseño.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **“Proyecto 1 - Entrega 2”**.

## Entrega 3: Diseño e Implementación

Teniendo en cuenta el análisis y el diseño, para la tercera entrega debe realizar construir la implementación del sistema.

## Actividades

---

1. Si descubrió problemas con su diseño, actualícelo.
2. Implemente el sistema que diseñó. Tenga en cuenta que los detalles de la implementación deben ser coherentes tanto con el modelo de clases, como con los diagramas de secuencia.
3. Asegúrese de que la documentación del diseño sea consistente con la implementación.

## Entrega

---

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre **“Entrega 3”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto el documento de diseño actualizado como el proyecto Eclipse e instrucciones para su ejecución. Incluya también archivos de prueba para poder correr las aplicaciones y tener datos con los que se pueda probar con facilidad.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **“Proyecto 1 - Entrega 3”**.