

LIGA DE FÚTBOL

Trabajo de equipo. 2º Cuatrimestre.

Introducción

En esta práctica del segundo cuatrimestre vamos a realizar el software de gestión de una liga de fútbol de barrio.

El objetivo es crear un programa que sitúe a la liga en un punto determinado (por ejemplo, la jornada 6) y que nos permita introducir nuevos datos con los que actualizar los ya introducidos de manera automática.

Programa Principal:

El programa principal ejecutará automáticamente (y de forma oculta para el usuario) los datos de un número de jornadas, de manera aleatoria para posteriormente darnos las siguientes opciones.

- Ver la clasificación actual.
- Ver el calendario (mostrando los resultados de los partidos ya jugados)
- Introducir nuevos resultados (puede hacerse de manera parcial, es decir, uno o más partidos de la jornada actual)

OPCIONAL: Opcionalmente también se podrán añadir las siguientes opciones.

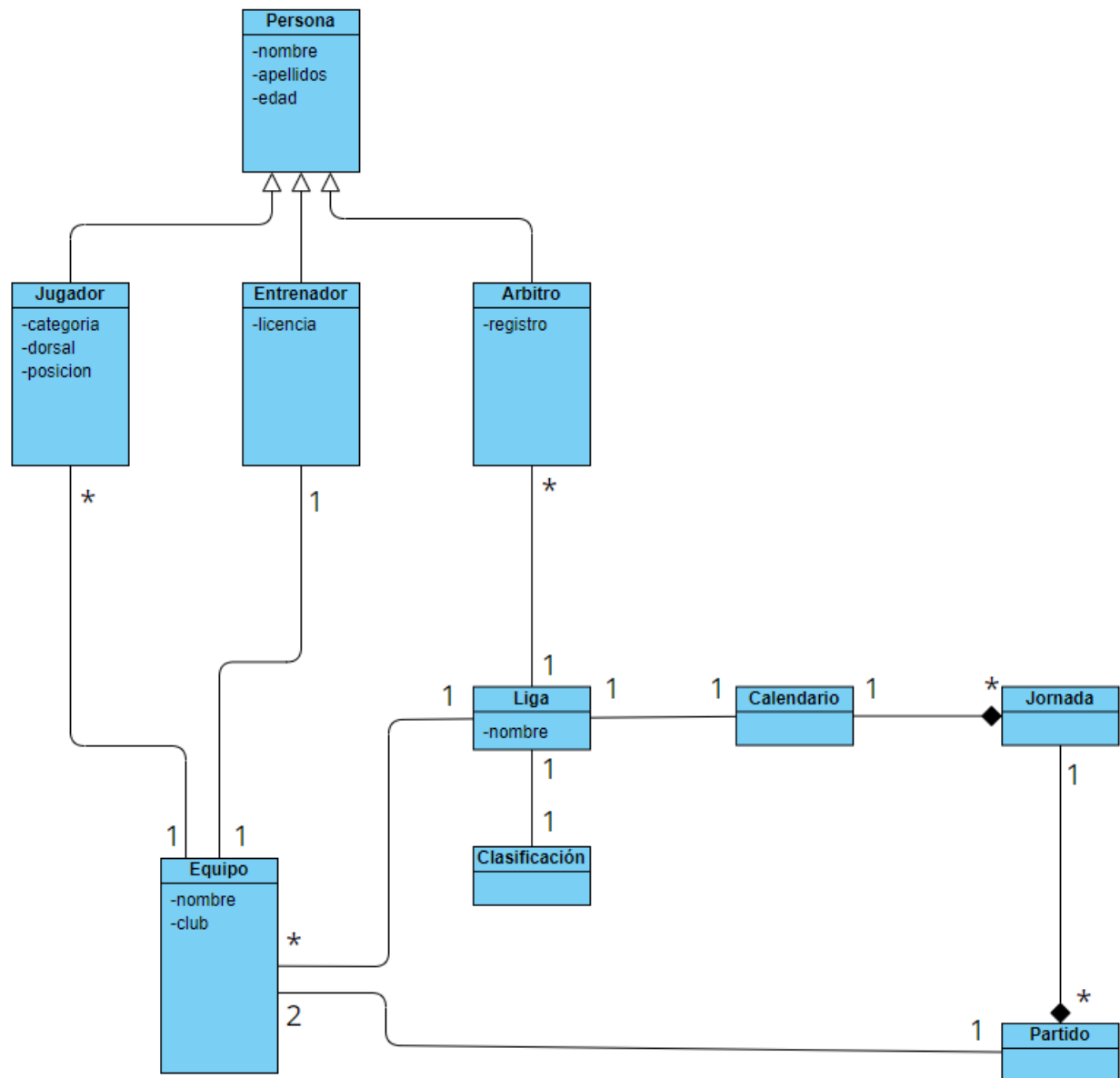
- Máximos Goleadores.
- Mejores Porteros (solo si se habilitan los esquemas de equipo, ver más abajo)
- Elegir número de jornadas a simular
- Resetear liga (con el número de equipos predeterminado)
- Introducir número de equipos desde teclado (para luego hacer la simulación)
- Proyecciones de campeones (en jornadas cercanas al final de la liga, ver quien tiene mejores posibilidades de ganar – esto implicará investigación aparte-)

OPCIONAL: Como variantes a elegir sobre nuestro planteamiento inicial podemos añadir elementos algo más deterministas a nuestros equipos tales como:

- Limitar número de jugadores por posición
- Crear esquemas de equipo (4-4-2, 4-3-3, 3-4-3, 5-4-1, etc.) que posicionen a los jugadores en los equipos en base a su posición. Es decir, en un 4-4-2 el programa elegiría a 4 defensas, 4 centrocampistas y 2 delanteros, así como a un portero.
- Crear titulares (usando el esquema) y suplentes
- Generar bajas y lesiones aleatorias.
- Ver resultados, compromisos y otros datos individuales (jugadores, entrenadores, árbitros y equipos)
- Gestión de uniformes en casa y fuera en igualdad de colores.

Esquema base

En clase hemos diseñado algunas de las clases que necesitamos para crear la estructura sobre la que basaremos nuestro programa. La relación entre dichas clases se muestra en el siguiente diagrama.



Dichas clases son una guía para realizar el programa y están abiertas a modificaciones. También podéis añadir o retirar clases siempre que me deis una explicación adecuada de por qué lo habéis hecho.

El funcionamiento de las clases aquí descritas es como sigue:

Clase Liga:

La clase central con la que trabajaremos. La liga está relacionada con todas las demás clases de manera directa o indirecta.

Los elementos que forman la clase liga son (atributos):

- El nombre de la liga
- El Calendario de la liga. Solo hay un calendario para cada liga.
- La clasificación de la liga
- Los equipos que forman la liga
- Los árbitros que están adscritos a la liga.

Clase Calendario:

La clase calendario es la interfaz que comunica a la clase liga con las diferentes jornadas. Estudiad su viabilidad y necesidad. Si veis que no es necesaria, eliminadla (siempre justificando por qué).

En esta clase podremos ir ajustando las fechas de las diferentes jornadas, tendremos un listado de las diferentes jornadas y las operaciones necesarias para trabajar con ellas.

La utilidad inicial de clase es dejar el código más claro en la clase Liga que “hace uso” de dicha clase (mirad esto en teoría).

Clase Jornada:

La clase jornada se refiere a una sola jornada del Calendario de la liga. En esta jornada habrá una lista de partidos determinados en base al calendario original.

De tal modo en la clase jornada podremos:

- Crear Partidos
- Asignar árbitros a partidos (ver más abajo)
- Asignar horarios a partidos.
- Modificar datos de partidos.
- Etc.

OPCIONAL:

Considera la situación en que el número de árbitros pueda limitar los horarios de los partidos. Por ejemplo, si tus partidos son en sábados y domingos en horarios de mañana y tarde pero solo tienes 1 arbitro por cada 8 equipos, entonces tendrás que calcular (de manera automática), que cada árbitro tendrá que pitar dos partidos cada día, uno en cada horario.

Clase Clasificación:

En esta clase lo que haremos será crear una clasificación con un criterio (que os dejo que propongáis) que nos indique (basándonos en los diferentes partidos de cada equipo) el ranking de los diferentes equipos en la liga.

Estudia los atributos necesarios, la manera de calcularlos y los métodos útiles para la realización de esta clase.

Clases Persona, Jugador, Entrenador, Árbitro y Equipo.

Estas clases, desarrolladas en clase, pueden modificarse en base a la necesidad de los equipos de programación.

Algunas ideas posibles sería añadir algún tipo de elemento que permita directamente consultar las estadísticas, calendario y otros datos de un jugador, entrenador, árbitro o equipo. Esto podemos conseguirlo creando relaciones entre las clases de manera inteligente.

Consejos para el desarrollo:

Este es vuestro primer proyecto complejo a nivel organizacional. Necesitaréis trabajar como un equipo, distribuyendo las tareas y manteniendo altísimos vuestros niveles de comunicación. Necesitareis reuniones periódicas y objetivos a corto plazo.

Consejo 1. Aproximación Agile:

Tratad de mantener vivo el trabajo y funcionar en base a sprints con objetivos pequeños y claros. Planificar un road map con objetivos opcionales, cread tareas y usad las herramientas necesarias para comunicaros de manera efectiva.

Consejo 2. Prototipad:

Cread una versión pequeña funcional lo mas antes posible, aunque la funcionalidad sea limitada. Tal y como hemos hecho en clase, id funcionando de manera progresiva. No empecéis por el menú final, sino que tratad de que las cosas vayan funcionando.

Consejo 3. Distribuid:

Si en vuestro equipo uno programa y 3 miran, no vais a llegar muy lejos. Necesitáis distribuir el trabajo por áreas. Intentad que alguien se encargue del calendario, otro de los equipos, otro de los partidos, otro de la clasificación, etc. Todo esto partiendo de un diseño sencillo con las clases más básicas.

Consejo 4. Decidid:

No hay malas o buenas decisiones. Lo malo en realidad es no decidir. Si tomáis una decisión, justificadla y llevadla a cabo. Si no funciona, no abandonéis, corregid y continuad.

Consejo 5. Documentad:

Llevad un cuaderno de bitácora de lo que hacéis cada día que estéis en el proyecto. Incluid lo siguiente en las notas de campo.

- Qué pretendía hacer.
- Qué he hecho.
- Qué dificultades he tenido.
- Qué he hecho para solventarlas
- Qué conclusión saco.

Cada día pueden ser 4-5 frases. No escribáis la biblia. Adjuntad todos vuestros cuadernos de trabajo en la documentación del trabajo.

Equipos, Evaluación y Entrega:

Los equipos podrán estar formados por un mínimo de 3 personas y un máximo de 5. Si hay alguien que no tenga equipo, que hable con el profesor y se le asignará uno. Debéis anunciarme los equipos mediante un mensaje por Teams.

La entrega del trabajo consistirá en un archivo ZIP (comprobad que funciona) en el que se incluirá:

- Archivos Java (.java, nada de jars ni ejecutables)
- Breve Manual de Instrucciones (no quiero tener que adivinar cómo funciona) en PDF. Este manual podéis hacerlo en modo escrito o mediante un video (subid el video a una plataforma y compartid el enlace)
- Los cuadernos de bitácora de todos los miembros en un solo archivo PDF
- Otros documentos que consideréis necesarios (resultados, ejemplos, etc.)

La Evaluación del trabajo tendrá en cuenta:

- Corrección de los requisitos
- Presentación
- Facilidad de uso
- Toma de decisiones
- Cuadernos de Bitácora
- Compleción de requisitos opcionales e ideas originales.

Esta es una de las prácticas más importantes del año porque el contenido os va a elevar a un nivel mucho más alto como desarrolladores de software. Hay un antes y un después de esta práctica.

Buena Suerte.