**PRACTICA GRUPO**

**LIGA DE FUTBOL**



Carlos Soler Garcia

Roberto García Sánchez

Adrián López Espejo

Juan Antonio Enríquez Bustos

**Clase Principal**

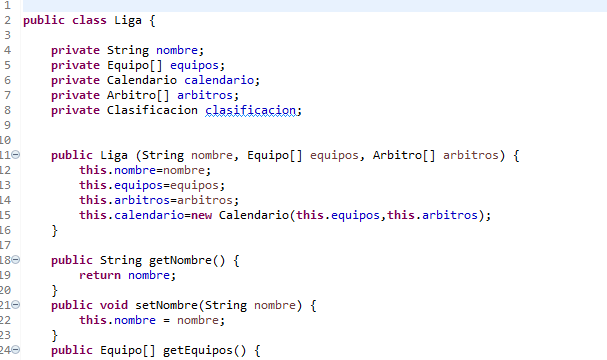
La clase principal es donde se generan todos los Jugadores, Equipos, Entrenadores y Árbitros que componen la Liga, además de generar precisamente dicha liga y hacer un llamamiento a la clase menú, que es el objeto con el que interactúa el usuario para pedir los datos que desee.

Los jugadores, equipos, entrenadores y árbitros se generan de manera aleatoria en sus métodos, los cuales devuelven una lista de Jugadores, una lista de Equipos, una lista de árbitros y un solo Entrenador que se le asigna a cada equipo que se le introduce al método.



**Clase Liga**

La clase Liga básicamente es donde se alojan todos los equipos, el Calendario (con sus jornadas y partidos), los árbitros y la clasificación. El constructor de Liga exige dos parámetros para poder generar el objeto, uno es la lista de Equipos y el otro la lista de Árbitros, los cuales se pasan como argumentos a los atributos de la liga antes de generar el Calendario, ya que este requiere de una lista de Equipos también. El atributo Nombre viene por defecto como “Liga Sin Nombre”, ya que la idea era añadir una opción en el menú para que el usuario pudiese cambiar el nombre de la liga como quisiera.



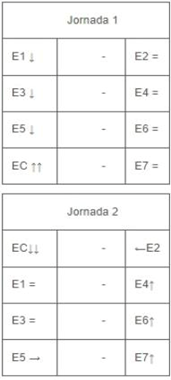
**Clase Calendario**

Esta clase es una de las que tiene más complejidad, ya que es la encargada de recibir los equipos desde liga y emparejarlos para asignar los partidos de cada jornada teniendo en cuenta una serie de condiciones, que debe de haber ida y vuelta, que no pueden repetirse partidos y que si los equipos son impares uno tiene que ser el que descansa.

El único atributo de esta clase es un array de Jornadas y el constructor requiere una lista de equipos. Este constructor crea la longitud del array de Jornadas teniendo en cuenta si son equipos pares o impares y llama al método emparejamiento que retorna dicho array de Jornadas.

Método emparejamiento:

El algoritmo de emparejamiento intenta que los equipos jueguen tanto en casa como de visitante de manera alternativa. El patrón que he seguido es el siguiente:



En la jornada 1 realmente no se mueven los equipos, se muestran tal cual vienen en la lista de Equipos que se generó al inicio, en las jornadas pares es cuando se lleva a cabo el movimiento del primer cuadrante en el que solo se mueven los equipos que jugaron de locales, y en las jornadas impares se lleva a cabo el movimiento del segundo cuadrante.

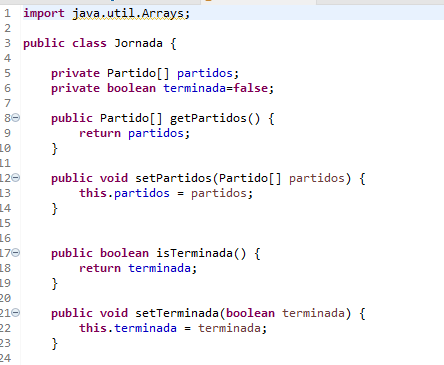
La simbología es la siguiente, una flecha indica que se mueve una posición, dos flechas que se mueve hasta el lado opuesto de la tabla, y un igual que no se cambió de posición. Nosotros por motivos de hacer el código más simple decidimos que el movimiento uno cambiase de posición en lugar de los locales los visitantes.

Solo hay una pega con este algoritmo y es que el último equipo de la lista siempre juega de visitante porque no rota hacia los equipos locales, se podría modificar, pero llevaría más tiempo de implementación.

El método de emparejamiento a diferencia de cómo podría pensar alguno que devolvería una matriz con los equipos.

**Clase Jornada**

El constructor de la clase Jornada pide como argumentos de entrada una lista de equipos que es la que viene dada del método emparejamiento del Calendario.



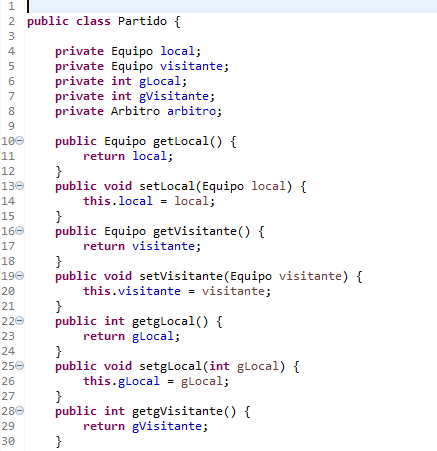
Como expliqué la lista de equipos emparejados no es una matriz sino un simple array ordenado como EquipoLocal, EquipoVisitante, EquipoLocal, EquipoVisitante, EquipoLocal, EquipoVisitante… etc. De esta manera solo hay que coger las posiciones del array de dos en dos y a la hora de crear el objeto partido el primer equipo se asignará como equipo local y el segundo como equipo visitante.

¿Y qué ocurre si los equipos son impares? Realmente no hay mucho cambio, el ejemplo sería el mismo, pero a la hora de asignar equipo local y equipo visitante al partido si uno de los dos es un espacio nulo del array simplemente ese partido no se genera y el equipo de esa pareja es el que descansa ya que no tiene enfrentamiento.

Como se muestra en la captura el método generarPartido() crea un array de partidos. Un for que recorre el array equiposEmparejados (que viene de la clase Calendario) de dos en dos, como se ha explicado antes, y con el condicional if comprobamos si equipo local o equipo visitante es null antes de generar el objeto partido y asignarle los valores.

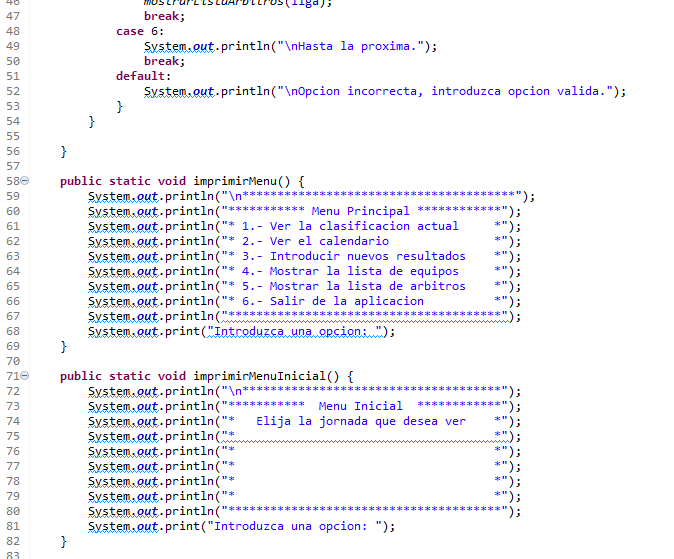
**Clase Partido**

La clase partido tiene los atributos Equipo Local, Equipo Visitante, Arbitro y los goles de cada equipo. Además tiene los métodos que pueden ser llamados de goles randoms para equipos locales y equipos visitantes.



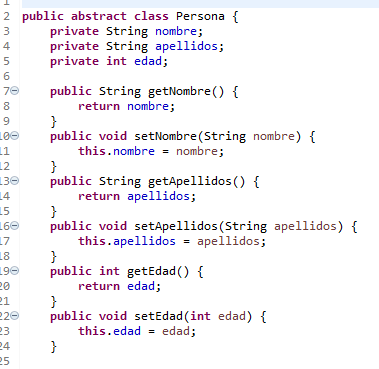
**Clase Menú**

La clase menú como constructor tiene de entrada el objeto liga. El menú imprime las opciones que el usuario ve con el método imprimirMenu(), le pide una opción al usuario desde el scanner y según lo que este escoja se llamará a otro método gracias a un Switch. Los métodos que llama el Switch son específicos de esta clase y se encuentran en la parte posterior del código.



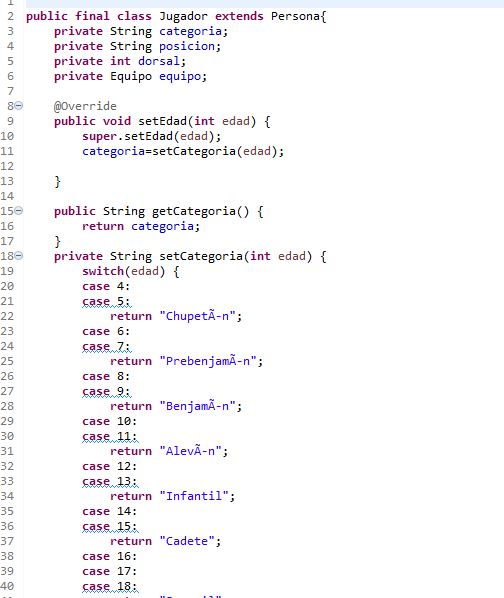
**Clase Persona**

Se trata de una superclase que guarda atributos que son comunes a otras subclases, jugador, entrenador y arbitro. Aquí encontramos atributos como la edad, nombre y apellidos. Es una clase bastante sencilla y sirve como base para las otras clases hijas.



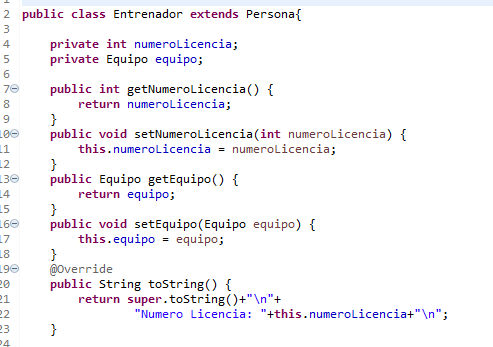
**Clase Jugador**

Es una clase que extiende persona y en ella se almacenan los atributos referentes al jugador, dorsal, posición, categoría en la que juega y equipo al que pertenece. Es una clase bastante importante ya que son los componentes esenciales de los equipos.



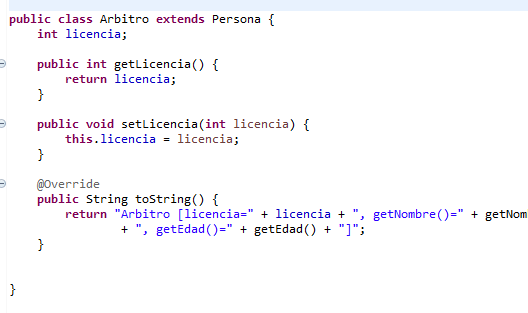
**Clase Entrenador**

Es una clase que viene de la super clase Persona, tiene solo 2 atributos que son el número de licencia y el equipo al que pertenecen, no es una clase muy importante.



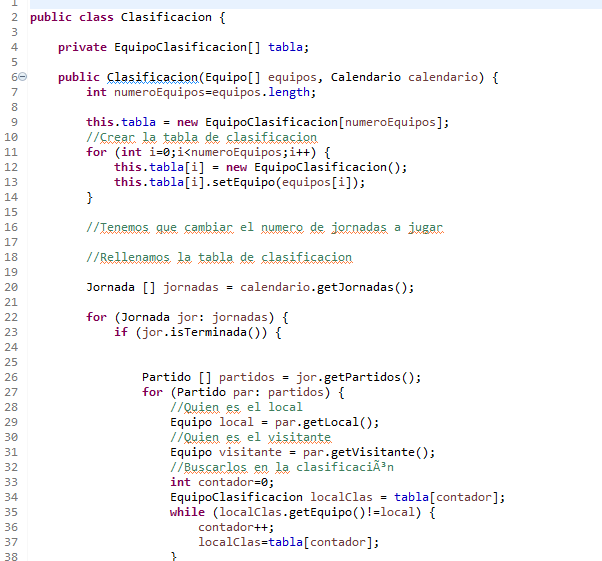
**Clase Árbitro**

Como las anteriores es una clase hija de la clase Persona, es una clase bastante importante en el programa ya que tiene que existir tantos árbitros como partidos se jueguen en la jornada, a cada uno se le asignará un partido que pitar cada jornada. Aunque solo tengo un atributo el papel que desempeña al pitar cada jornada un partido es importante y tiene que quedar constancia de ello.



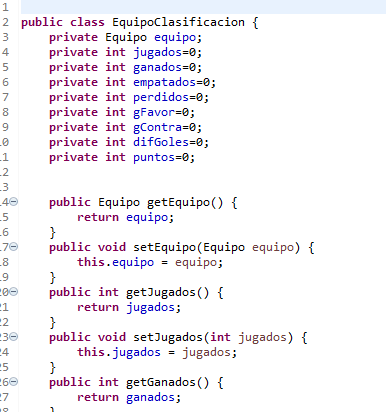
**Clase Clasificación**

En esta clase se revisan todas las jornadas jugadas y se hace un cómputo de los puntos y estadísticas de cada equipo, acto se guido se realiza una tabla donde se ordenan, mediante el algoritmo Bubble Sort, los equipos, estando en primera posición aquellos que hayan marcado más goles. Esta clase también actualiza los partidos y sus resultados en la clase EquipoClasificacion.



**Clase EquipoClasificacion**

Esta clase guarda los numerosos atributos referentes a la clasificación de los equipos, para cada equipo guarda los partidos jugados, su resultado, y los goles a favor y en contra. Mediante esta clase podemos luego calcular la posición que tendría el equipo respecto a los otros simplemente revisando sus atributos



Enlace donde se explica el detalladamente el sistema de emparejamientos.

<https://www.cihefe.es/cuadernosdefutbol/2013/03/la-confeccion-del-calendario-de-liga/>