

Descripción de la aplicación

Se desea desarrollar una aplicación para gestionar una liga de baloncesto femenino (sirva de ejemplo LF Endesa (<http://www.ligafemenina.es/>)).

De cada **Jugadora** se guardará:

- Nombre
- Posición (Ala-pívot, Pívot, Base, Escolta, Alero)
- Dorsal
- Nacimiento (fecha, ciudad y provincia)
- Nacionalidad
- Altura

De cada **Equipo** se almacenará:

- Nombre del equipo
- Dirección
- Teléfono
- Web
- Email
- Jugadoras (colección con las jugadoras del Equipo)

De cada **Jornada** se guardará:

- Número de jornada
- fecha
- Partidos (colección con 8 partidos)
- Clasificación (colección de 16 Datos_equipo)

Cada **Datos_equipo** para una jornada contendrá:

- Nombre equipo
- Número de partidos jugados (PJ)
- Número de partidos ganados (PG)
- Número de partidos perdidos (PP)
- Puntos a favor (PF)
- Puntos en contra (PC)
- Puntos de la clasificación ($2*PG + PP$)

De cada **Partido** se guardará:

- Nombre equipo local
- Nombre equipo visitante
- Puntos equipo local
- Puntos equipo visitante
- Fecha del partido
- Hora del partido

En el paquete del modelo (o datos), además de las clases anteriores, se dispondrá de una clase llamada **LigaFem** (que será el Modelo o Datos general), que contendrá los siguientes atributos:

- Temporada (por ejemplo, la presente sería "2020/21")
- Una colección de objetos de tipo **Jornada (15 jornadas)**.
- Una colección de objetos de tipo **Equipo (16 equipos)**.

Funcionalidad de la aplicación

Todos los datos que se faciliten a la aplicación o que la aplicación genere, estarán ubicados en una carpeta del escritorio llamada "LigFemBal".

Arranque de la aplicación

Se comprobará que existen datos dentro de la subcarpeta "binarios". Si existen se cargarán y se trabajará con estos datos. La primera vez que se ejecute la aplicación, esta carpeta estará vacía.

Una vez iniciada la aplicación aparecerá un menú con submenús que recoja las siguientes funcionalidades.

- **Gestión de temporada.** (La carga de todos estos datos sólo debería hacerse una vez al principio de la temporada, y deberían ejecutarse de forma secuencial y en este orden, se debe controlar que se ha ejecutado una opción antes de permitir ejecutar la siguiente)
 - **Iniciar temporada.** Se introducirá por teclado el valor de la temporada sobre la que se van a introducir datos.
 - **Cargar jornadas.** Se cargarán desde un fichero los datos de todas las jornadas de esa temporada, con los nombres de los emparejamientos, fechas y horas de los partidos, según el modelo de fichero que se adjunta: "datosjornadas.txt". Es un fichero de tipo CSV, con el separador "+" para los atributos de la jornada, el

separador “#” para separar partidos de una jornada, y el separador “\$” para separar los datos correspondientes a cada partido. En este momento no se cargan resultados.

- **Cargar equipos.** Los datos se toman del fichero “datosequipos.txt”, que se adjunta. Es un fichero de tipo CSV, en el que los atributos están separados por el carácter “#”.
 - **Cargar jugadoras.** Se deben cargar en cada equipo sus jugadoras. Para ello, se adjunta una subcarpeta “jugadoras”, que contiene un fichero para cada uno de los equipos con sus correspondientes jugadoras. Estos ficheros son de tipo CSV, con atributos separados por el carácter tabulador “\t”. Los ficheros tienen formato correcto, aunque en alguno de los ficheros faltan valores. Es decir que si falta un valor habrá dos tabuladores juntos. La aplicación debe gestionar esto.
- **Gestión de jugadoras.**
 - **Modificar datos de una jugadora.** Se puede cambiar cualquiera de sus datos menos el Nombre. Los cambios efectuados en los datos en la jugadora deben reflejarse en su equipo.
 - **Eliminar una jugadora de un equipo.** Sólo se elimina del equipo.
 - **Añadir una jugadora a un equipo.** Se toman todos sus datos por teclado y se añade a su equipo.
- **Gestión de Jornada.**
 - **Leer los resultados de la jornada.** Los datos de la jornada son los puntos que ha obtenido cada equipo en su partido correspondiente. Se pide el número de jornada a cargar y los datos se proporcionan a través de un fichero cuyo nombre es el número de la jornada, que hay que colocar en la subcarpeta “resul_jornadas” (se adjuntan algunos). Con estos valores se actualizarán los puntos de los partidos de la jornada. Además, hay que calcular la clasificación de la jornada. Para ello se toman los datos de la jornada anterior, se incrementan sus valores y los puntos correspondientes, dando lugar a una colección de objetos de tipo Datos_equipo, que se añadirá a la jornada.
 - **Modificar fecha de la jornada.** Se pide el número de la jornada, y se cambia su fecha.
 - **Modificar fecha u hora de un partido.** Se pide el número de la jornada, y el nombre de uno de los equipos y se modifica la fecha o la hora del partido.
 - **Mostrar los resultados de la jornada.** Se pide el número de la jornada y se muestra un listado con cuatro columnas con los nombres de los dos equipos de cada partido y los puntos de cada uno de ellos para esa jornada.
 - **Mostrar la clasificación de una jornada.** Se pide el número de la jornada y se muestra un listado similar al siguiente:

Puesto	EQUIPO	PJ	PG	PP	PF	PC	PTS
1	PERFUMERIAS AVENIDA	13	13	0	1045	761	26
2	VALENCIA B.C.	13	12	1	969	746	25
3	SPAR GIRONA	12	11	1	903	703	23
4	MOVISTAR ESTUDIANTES	13	9	4	957	836	22
5	LOINTEK GERNIKA BIZKAIA	13	8	5	869	790	21
6	CIUDAD DE LA LAGUNA TENERIFE	13	8	5	911	925	21
7	IDK EUSKOTREN	13	7	6	860	848	20
8	CAMPUS PROMETE	13	5	8	836	883	18
9	DURÁN MAQUINARIA ENSINO	13	5	8	860	920	18
10	CADI LA SEU	13	5	8	784	860	18
11	KUTXABANK ARASKI	12	4	8	700	780	16
12	CASADEMONT ZARAGOZA	12	4	8	797	889	16
13	ALTER ENERSUN AL-QÁZERES EXTRE	12	4	8	781	914	16
14	SPAR GRAN CANARIA	13	3	10	911	1001	16
15	EMBUTIDOS PAJARIEL BEMBIBRE PD	13	2	11	807	944	15
16	QUESOS EL PASTOR	13	2	11	827	1017	15

- **Visualización de resultados (listados)**
 - **Jugadoras de un equipo.** Se pide el nombre del equipo y se muestra un listado de sus jugadoras, ordenado por posición y altura.
 - **Relación de equipos.** Un listado de equipos con sus atributos, sin las jugadoras, ordenado por los teléfonos.
 - **Relación de jugadoras.** Se pide la letra inicial del nombre y se muestra un listado de las jugadoras cuyo nombre empieza por esa letra, ordenadas por la fecha de nacimiento.
- **Almacenamiento de resultados (ficheros que se almacenan en la subcarpeta “fichsalida”)**
 - **Jugadoras de un equipo.** Se pide el nombre del equipo y se genera un fichero encolumnado con ancho de columnas fijo, con el nombre del equipo y extensión .enc
 - **Relación de equipos.** Se genera un fichero encolumnado con ancho de columnas fijas, llamado “equipos.enc”, con la relación de equipos, conteniendo nombre, teléfono, web y email.
 - **Clasificación de una jornada.** Se pide el número de la jornada y se genera un fichero en formato html, de nombre “fich_html_n.html” (siendo n el número de la jornada) conteniendo una tabla con la clasificación de la jornada similar a la que se adjunta para la jornada 13, situado en la subcarpeta “fichsalida”.
- **Salida de la aplicación.**
 - Se guardarán en formato binario todos los datos de la aplicación en la subcarpeta “binarios”, de forma que cuando se arranque de nuevo se pueda continuar en el estado en el que se encontraba antes de salir.

Indicaciones Adicionales

1. La aplicación, realizada individualmente, se entregará en forma de un proyecto de NetBeans (NetBeans 12.1 y Java 14) comprimido en formato **ZIP**. Recuérdese que el proyecto debe estar correctamente identificado con el nombre y DNI del autor. Esto se hace adaptando la licencia genérica aportada por NetBeans para que muestre los datos del autor.
2. El proyecto debe funcionar correctamente en todas las plataformas; no se admitirán proyectos “cableados” en que las rutas de archivos estén generadas para una sola máquina o plataforma en particular.
3. La aplicación debe respetar las normas de correcta nomenclatura en Java empleadas a lo largo del curso. En particular, no se utilizarán signos diacríticos en identificadores y la codificación del proyecto será UTF-8. Todos los archivos de texto hacen uso de la codificación UTF-8. Las clases tienen nombres que empiezan por mayúscula, los identificadores y métodos empiezan por minúscula, etc.
4. Todos los listados deberán presentarse en forma de columnas, con encabezados y ancho de columnas fijas, para que aparezcan perfectamente alineados.
5. Se valorará la creación de un `makefile` y el correspondiente `jar`.

Arquitectura de la aplicación

1. La aplicación debe hacer uso del patrón MVC empleado a lo largo del curso.
2. Todos los ficheros que maneja la aplicación estarán situados en una carpeta del escritorio llamada “LigFemBal”, según se especifica en diversos apartados.