

## **TRABAJO FINAL. ANTEPROYECTO.**

### **TÍTULO**

#### **Club Deportivo Modas Levante**

### **ANTECEDENTES**

El C.D. Modas Levante es un club situado en el barrio de Levante y que lleva ofreciendo la posibilidad de que los niños y niñas del barrio y sus alrededores puedan realizar actividad física y deportiva a través de la práctica del fútbol sala.

### **DEFINICIÓN DEL PROBLEMA**

Se trata de un club deportivo que desde su fundación en el año 1987 no ha llevado nunca un registro de las personas que han sido parte de él, tanto jugadores, entrenadores, masajistas....

### **JUSTIFICACIÓN**

Con el paso del tiempo, han sido muchos las instituciones, periodistas, padres, incluso los propios niños que pasaron por el club, que han preguntado por si existía algún tipo de registro para poder hacer uso de esa información en periódicos, revistas, trabajos... y la mayoría de las veces no han podido tener acceso a esa información, excepto recurrir a la memoria de algunos de los implicados. Por tanto, sería interesante tener un registro para evitar esta situación en un futuro.

### **OBJETIVO**

Llevar a cabo un registro desde el año actual en delante de los jugadores, entrenadores, y masajistas que han pasado por el club.

Almacenar en el caso en el caso de los jugadores: el nombre, apellidos, edad, fecha de llegada al club, categoría a la que pertenece, así como los goles marcados, partidos jugados... En el caso de los entrenadores y masajistas tanto sus años de experiencia como sus nivel de preparación.

Acceder de forma rápida y sencilla a cada uno de los datos recogidos.

## PROCEDIMIENTO

Se analizará cada uno de los contenidos estudiados en el Ciclo Formativo para ver cuál es la mejor forma de implementarlos en el proyecto que queremos llevar a cabo. Para ello, utilizaremos los siguientes recursos:

- ArrayList (estructuras de datos)

Se usará el ArrayList para almacenar de forma dinámica cada uno de los elementos, en este caso, personas o miembros del club de los cuales queremos guardar información. Este ArrayList será un ArrayList genérico de Miembros, por ejemplo.

- Enumeraciones

Para almacenar las categorías: PREBENJAMÍN, BENJAMÍN, ALEVÍN, INFANTIL, CADETE, JUVENIL Y SENIOR.

Para los niveles del entrenador: MONITOR, NIVEL 1, NIVEL 2, LICENCIADO E. F.

Para la titulación del masajista: FISIOTERAPEUTA, QUIROMASAJISTA, OSTEÓPATA

Las demarcaciones de los jugadores: PORTERO, CIERRE, ALA\_DERECHA, ALA\_IZQUIERDA, PIVOT.

- Herencia

Usaremos la herencia para crear una clase padre, en nuestro caso Miembros, que tendrá unos atributos y unos métodos comunes para todos, tanto si son jugadores, entrenadores, masajistas... y luego estarán las clases hijas que heredarán de la clase padre Miembros, pero tendrán sus propios métodos y atributos propios. Por ejemplo, todos los miembros tienen un nombre, un apellido, una fecha de alta en el club, un sueldo, una categoría... pero tienen papeles distintos dentro del club. Los jugadores tienen un dorsal, una demarcación, tienen una altura, una edad, goles marcados, partidos jugados.... El

entrenador tendrá una titulación que puede ser una enumeración con las siguientes constantes: MONITOR, NIVEL 1, NIVEL 2, LICENCIADO E. F.,, unos años de experiencia. El masajista tiene un sueldo distinto al de los entrenadores y jugadores, a parte de una titulación acorde al puesto que puede ser una enumeración: FISIOTERAPEUTA, QUIROMASAJISTA, OSTEÓPATA. Al igual que el entrenador también tiene unos años de experiencia en el puesto.

- Interfaces

Se usarán 2 interfaces. por ejemplo, la interfaz sueldos() y la interfaz comparable(). Sueldos cambiará el sueldo dependiendo de qué miembro del club sea, es decir, un jugador no cobra lo mismo que un entrenador ni que un masajista. El sueldo del jugador depende de la demarcación, el del entrenador de su nivel de estudios y sus años de experiencia, al igual que el del masajista. En la interfaz comparable() se usará para poder ordenar los miembros del club por nombre.

- Flujos de datos

Los flujos de datos nos servirán para poder guardar la información en ficheros mediante el uso de los buffers y métodos propios como BufferedInputStream, BufferedOutputStream...

- Ficheros

El uso de ficheros nos ayudará a guardar la información en un archivo y poder acceder a la información en cualquier momento cuando lo deseemos. Para ello, se hará uso de los métodos propios de los ficheros: writeObject, readObject, ObjectInputStream, ObjectOutputStream... Las opciones serán: Nuevo archivo, abrir archivo existente, guardar y guardar como...

- Excepciones

Controlaremos posibles fallos con las excepciones, como por ejemplo. Los niños y niñas de una edad pertenecen a una determinada categoría, por ejemplo la benjamín (entre 8 y 9 años), por tanto no pueden darse de alta en la categoría

alevín (entre 10 y 11 años). También si un niño se ha inscrito en un año, no poder volver a inscribirse porque ya existe en el registro, etc.

- Expresiones regulares

El uso de expresiones regulares nos será útil para mediante el uso de patrones controlar que un nombre (el nombre no puede tener número ni caracteres extraños), una edad (no puede contener letras y ha de ser un número entero), una fecha con un patrón válido (dd/mm/yyyy) sean válidos.

- Fechas

Se llevará un registro de fecha de alta para saber cuándo el niño ha sido inscrito en el club y calcular el tiempo que lleva en él, por ejemplo.

- GUI

Mediante la interfaz gráfica de java implementaremos el programa haciendo uso de los frames y diálogos disponibles en la GUI, y le añadiremos un comportamiento entre los miembros, por ejemplo buscar por categoría, por año de inscripción en el club...