



PROGRAMACIÓN – 1º DAM

PRUEBA 9 (30-MARZO-2022) curso 2021/2022

Nombre y Apellidos:

Puntuación:

Se tiene en la url: <https://github.com/luisdbb/federacion.git> un proyecto Java implementado en Eclipse que responde al diagrama de clases y a las especificaciones del final de este documento y que contiene la solución a todos los ejercicios propuestos en las pruebas anteriores hasta el momento. Este proyecto servirá de base para la resolución de los ejercicios de esta prueba.

La entrega consistirá en un **NUEVO PROYECTO** ampliado, al que se le habrá cambiado el nombre del proyecto de `FederacionSQL` por `FederacionXXXEVAL3` siendo XXX el nombre del estudiante. El proyecto se subirá a un **NUEVO REPOSITORIO** Github de cada estudiante de nombre `FederacionXXXEVAL3` y la url de éste se copiará en la tarea del aula virtual del módulo:

<https://aulasvirtuales.educastur.es/mod/assign/view.php?id=730660>

NOTA: Será requisito indispensable que se haya subido correctamente el proyecto de entrega de cada estudiante a su repositorio, sin errores de compilación, y enviar una invitación al profesor (luisdbb@educastur.org) para que disponga de permisos de Colaborador en dicho repositorio, de manera que el proyecto subido deberá estar sin errores de compilación y podrá ser clonado y ejecutado sin necesidad de ninguna configuración adicional.

0. (BASE 2ptos.) **Preparación del nuevo proyecto:** clonado desde github del repositorio del profesor, creación y nomenclatura del nuevo proyecto correcta, inclusión en este de todas las librerías necesarias y otros ficheros de recursos, funcionamiento correcto del nuevo repositorio. Creación de la BD relacional `bdfederacion` en el propio dispositivo (servidor `localhost`) y de las tablas siguientes:
 1. Tabla `personas` con la siguiente estructura:
 - `id INT(11) PRIMARY KEY`
 - `nombre VARCHAR(50) NOT NULL`
 - `telefono VARCHAR(15)`
 - `nifnie VARCHAR(10) NOT NULL UNIQUE`
 2. Tabla `lugares` con la siguiente estructura:
 - `id INT(11) PRIMARY KEY`
 - `nombre VARCHAR(50) NOT NULL`
 - `ubicacion VARCHAR(50) NOT NULL`
 - `airelibre BOOLEAN NOT NULL`
 3. Tabla `categorias` con la siguiente estructura:
 - `id (11) PRIMARY KEY`
 - `nombre VARCHAR(50) NOT NULL`
 - `abreviatura CHAR(1) NOT NULL UNIQUE`

- Insertar “a mano” en las tablas `lugares` y `categorias` los valores actuales de las enumeraciones `Lugar` y `Categoria` respectivamente.

1. (Máx: 3ptos.) (apartado A:1pto) Implementar el método `data()` para las personas (clase `DatosPersona.java`) de forma que devuelva una cadena de caracteres con el siguiente orden:
`<idPersona> | <nombre> | <telefono> | <fechaNac(dd/MM/YYYY)> | <NIFNIE>`
 (apartado B: 1pto) Crear una nueva clase `ComparadorAlfabetico` que sirva para implementar la interfaz `Comparator` para las personas, de forma que se las pueda ordenar por orden alfabético creciente de su atributo `nombre`.
 (apartado C:1pto) Implementar luego una función estática en `DatosPersona.java` que recorra todo el array `PERSONAS` de la clase `Datos.java` y exporte hacia un fichero de caracteres de nombre `atletas_alfabetico.txt` los datos de todas las personas, cada una en una línea diferente, a través del método `data()` del apartado A y habiendo sido ordenadas previamente mediante el `ComparadorAlfabetico` del apartado B.
2. (Máx 3 ptos.) (apartado A: 1pto) Implementar la interfaz `Comparable` para las personas (clase `DatosPersona.java`), de forma que se ordenen según su fecha de nacimiento de menor a mayor edad y si 2 personas tienen el mismo valor de fecha, entonces se desempatará en función del campo `Documentacion` (NIF/NIE).
 (apartado B: 2ptos.) Incluir además en esa clase una función estática `insertarPersonas()` para obtener todos los datos del array `PERSONAS` de la clase `Datos.java` y ordenarlos según el comparador anterior (del apartado A de este ejercicio) sobre una lista. Finalmente recorrerla mediante un iterador e ir insertando en la BD relacional `bdfederacion` sobre la tabla `personas`.
3. (Máx 2ptos.) Implementar una función estática en un nuevo fichero `principal_XXX.java` que acceda a la BD relacional `bdfederacion` a la tabla `personas` y realice una consulta para seleccionar todos los datos de dicha tabla ordenados por el campo `nombre` ascendentemente. Con los valores de los campos para cada fila devuelta, componer un objeto de la clase `DatosPersona` y exportar sus datos hacia un fichero de caracteres de nombre `atletas_alfabetico_2.txt` a través del método `data()` de esa clase (apartado A del ejercicio 1), exportando los datos de una persona por cada línea en el fichero de salida.
 El resultado debería ser que el fichero generado a través del método del apartado C del ejercicio 1 contenga la misma información y en el mismo orden que el fichero generado en este ejercicio.

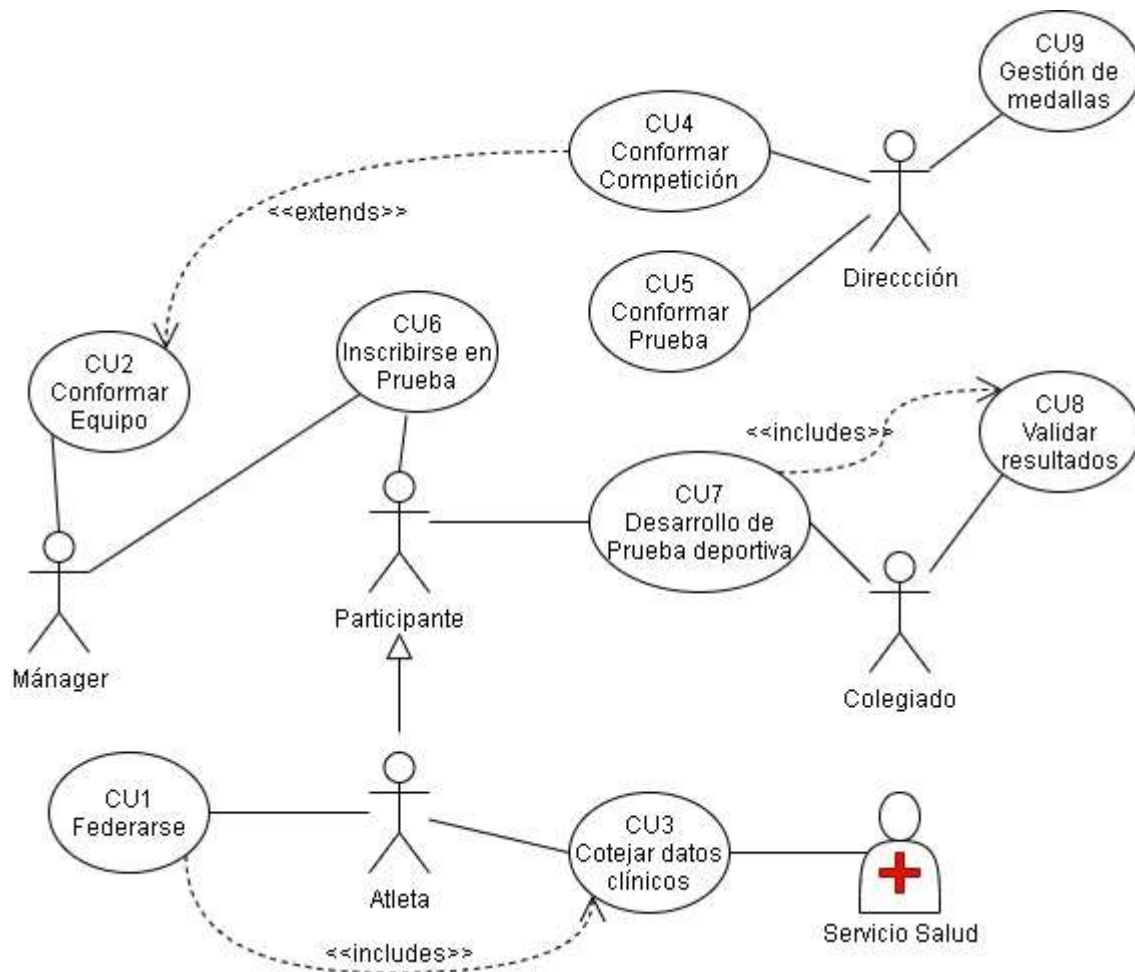


Ilustración 1 Diagrama de casos de uso

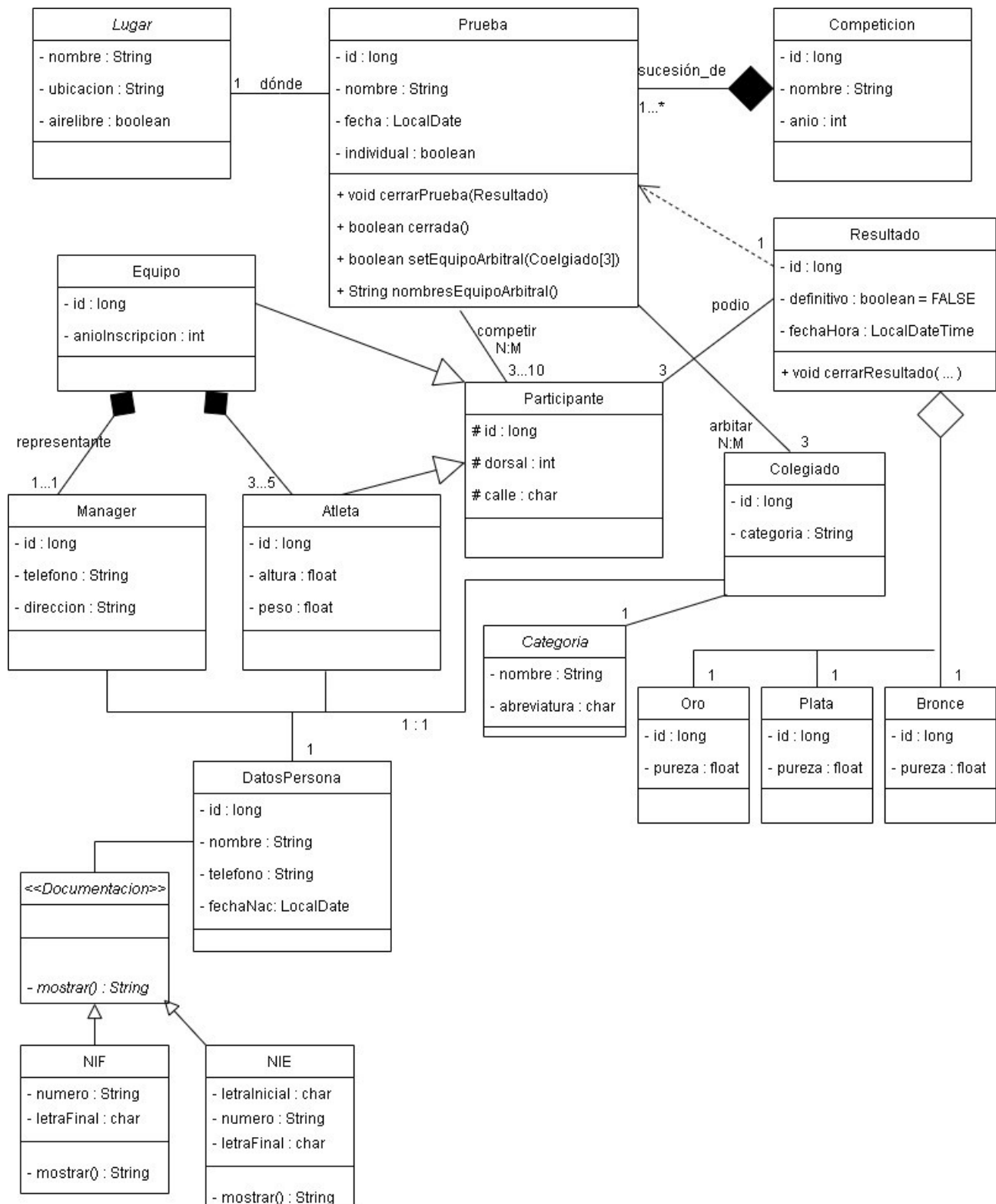


Ilustración 2 Diagrama de clases

Descripción del sistema a modelar:

“Gestión de competiciones de una Federación deportiva”

El sistema informático permite a la dirección de la Federación manejar los datos de las competiciones a lo largo de su historia. Para conformar cada **competición**, se *identifica* de forma única en el sistema, y se guarda su *nombre* y el *año* en que se celebra. Toda competición es una sucesión de diferentes **pruebas** (al menos una), que van *identificadas* y se registran sus *nombres*, *fechas* y *lugares* de celebración de cada prueba. También se marca si se trata de una prueba *individual* o colectiva. Los **lugares** donde se llevan a cabo las pruebas siempre son los mismos: de ellos se tiene su *nombre*, *ubicación* y un campo que indica si es al *aire libre*.

En cada prueba siempre compiten entre 3 y 10 **participantes**, cada cual con su propio *identificador*, nº de *dorsal* (que siempre es un valor entre 001 y 150) y la letra de la *calle* por la que correrá. Si la prueba es individual, entonces cada participante se corresponderá con un **atleta**. Si no lo es, entonces la prueba se denomina colectiva y los participantes serán **equipos** conformados por varios atletas (mínimo 3 y máximo 5). Todos los equipos tienen su propio *identificador* de equipo y se guarda el *año* en que se inscriben, dado que un atleta podría formar parte de varios equipos distintos siempre y cuando sea en años de competición diferentes. Además, por cada equipo se tiene un único **mánager** (que se *identifica* unívocamente y se guarda tanto su *teléfono* como su *email*). Un mánager representa al mismo equipo siempre, pero no a los atletas individualmente. Los mánagers mandan la información de su equipo a la federación al principio de la temporada, cuando se conforman las competiciones. Para ello los propios atletas deben estar federados previamente. De los atletas interesa guardar su propio *identificador* de atleta y sus datos físicos: *altura*, expresada en metros, y *peso*, expresado en kilogramos (ambas unidades con 1 decimal). La información sobre los datos clínicos que cada propio atleta aporta al sistema al federarse se coteja con los del servicio de salud.

Por otro lado, la inscripción de un participante en una prueba individual la realiza el propio atleta, pero si es una prueba por equipos la realizará únicamente el mánager en representación de todos los miembros de ese equipo.

Por último, el desarrollo de las pruebas son controladas siempre por 3 **colegiados** distintos, cada uno de ellos tiene su *identificador* y un campo para marcar la *categoría*. Son los encargados de tomar las mediciones y de validar el **resultado** de cada prueba, marcándolo como *definitivo* en una *fecha* y *hora* determinadas. Todos los resultados van *identificados* y cada uno de ellos se compone de un **oro** (primer puesto en la prueba), una **plata** (2º puesto) y un **bronce** (3º puesto). De estos metales se lleva la cuenta de su *pureza* (valor decimal en %), además de su *identificador* propio. Se desea guardar cuál es la mayor pureza alcanzada de cada tipo de metal en todos los tiempos. Toda la gestión de las medallas recae también en la dirección de la federación deportiva.