



PROGRAMACIÓN – 1º DAW

PLAN de recuperación para alumnos pendientes curso 2021-2022

En este documento se detalla el proceso de evaluación del módulo Programación del CFGS en Desarrollo de Aplicaciones Web para la evaluación ordinaria de Marzo de los estudiantes con este módulo pendiente en el curso 2021-2022.

En este plan de recuperación, la superación de 2 hitos corresponde con el 50% de la nota final de la evaluación, siendo el 50% restante a una prueba teórico-práctica que tendrá lugar en Marzo de 2022 y de la que serán avisados convenientemente los estudiantes a través de la plataforma virtual del módulo en la dirección <https://aulasvirtuales.educastur.es/course/view.php?id=12345>

A través de esa plataforma se pondrá a disposición de los estudiantes distintos materiales de apoyo, así como otros ejemplos que puedan servir de ayuda para realizar la implementación solicitada. Para comunicaciones con el profesor se deberá usar o bien el sistema de mensajería del aula virtual o bien a través del envío de un email a la dirección luisdbb@educastur.org (sólo se podrán utilizar estas vías, ligadas a la cuenta personal oficial de los estudiantes)

En el plan se planifican 2 hitos, ambos de carácter obligatorio, y la prueba teórico-práctica:

- hito1: 9-ENE-2022
- **hito2: 27-FEB-2022**
- prueba teórico-práctica: (fecha aún por determinar → MARZO 2022)

, para cada uno de ellos habrá que realizar una entrega sobre unas especificaciones concretas.

Tras cada fecha de cada hito se efectuará una corrección de la entrega y se pondrá una nota al estudiante. Para superar el módulo, deberá obtenerse una nota igual o superior a 5 puntos, resultantes de la ponderación anterior (50% de la nota a partir de los hitos y el 50% de la nota a partir de la prueba teórico-práctica)

HITO 2: Federación deportiva (fecha máxima: 27-FEB-2022)

Para este hito se deberán implementar todos los ejercicios de forma obligatoria en un mismo proyecto Eclipse en Java que será el que sea incluido en la entrega, en la url del aula virtual del módulo. Ese proyecto deberá estar correctamente subido en un solo archivo comprimido, de forma que pueda ser clonado directamente, sin errores de compilación y sin necesidad de ninguna configuración adicional.

El proyecto final deberá incluir las implementaciones para todos los ejercicios propuestos, que se puntuarán de la siguiente forma:

Ej1	Ej2	Ej3	Ej4	Ej5	TOTAL
2,5	1,5	1,5	1,5	3	10

“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.

La siguiente figura muestra el diagrama de clases del mismo:

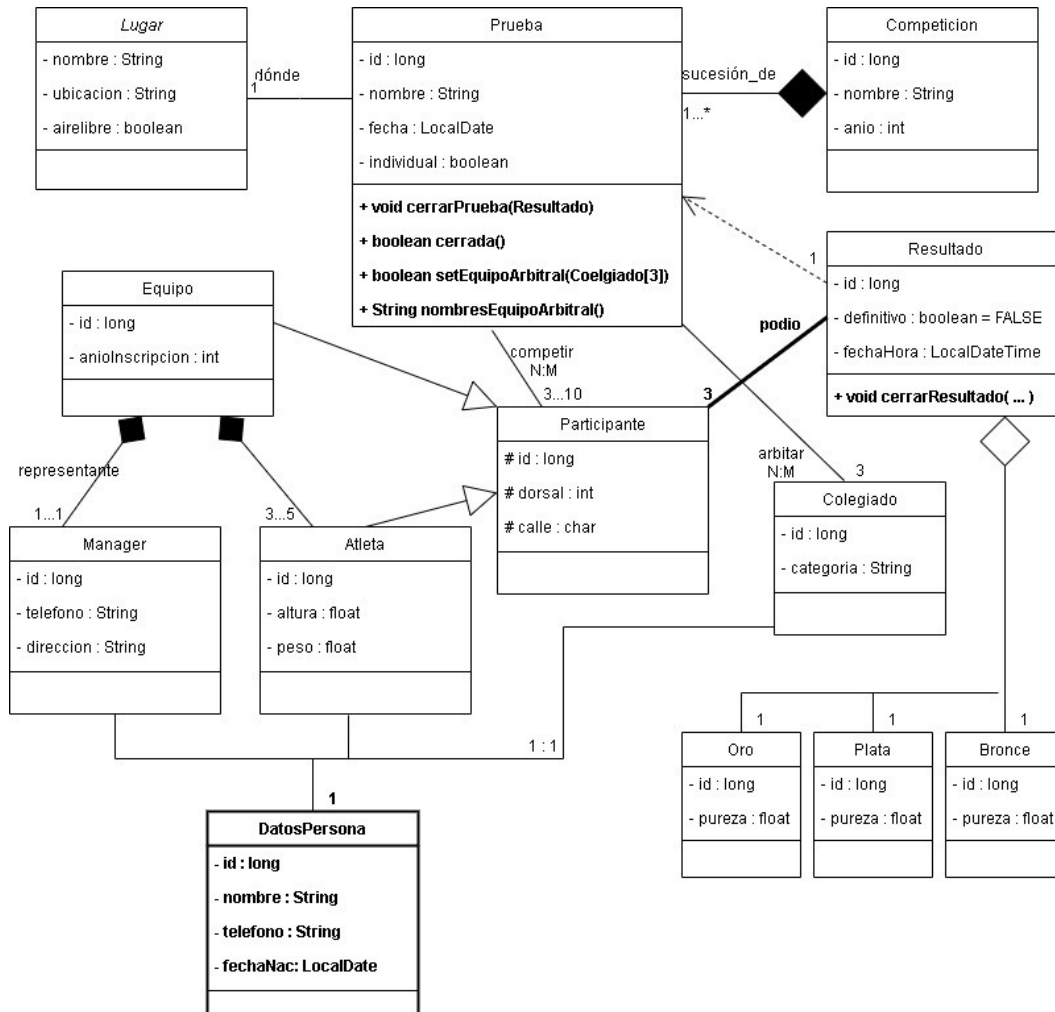


Ilustración 1 Diagrama de Clases

Para la aplicación se tienen 4 perfiles de usuario:

- **Dirección** → Se corresponde con alguien del equipo directivo de la Federación, con permisos para conformar los datos de las competiciones en cada temporada, incluyendo las pruebas de las que consta cada competición y los lugares donde se desarrollan. También es la figura encargada de la gestión de las medallas.
- **Mánager** → En representación de un equipo de atletas, esta figura los conforma al principio de la temporada y también realiza las inscripciones en las pruebas colectivas (por equipos).
- **Colegiado** → Son los jueces/árbitros de las pruebas, quienes controlan el desarrollo de las pruebas, toman los resultados de las mismas y los validan.
- **Atleta** → Corresponde con cada atleta individual, que se inscribe y compite en alguna o varias pruebas de una competición. Debe estar federado para ello y puede formar parte de un equipo.

A continuación se muestra el diagrama de casos de uso y una explicación de cada uno:

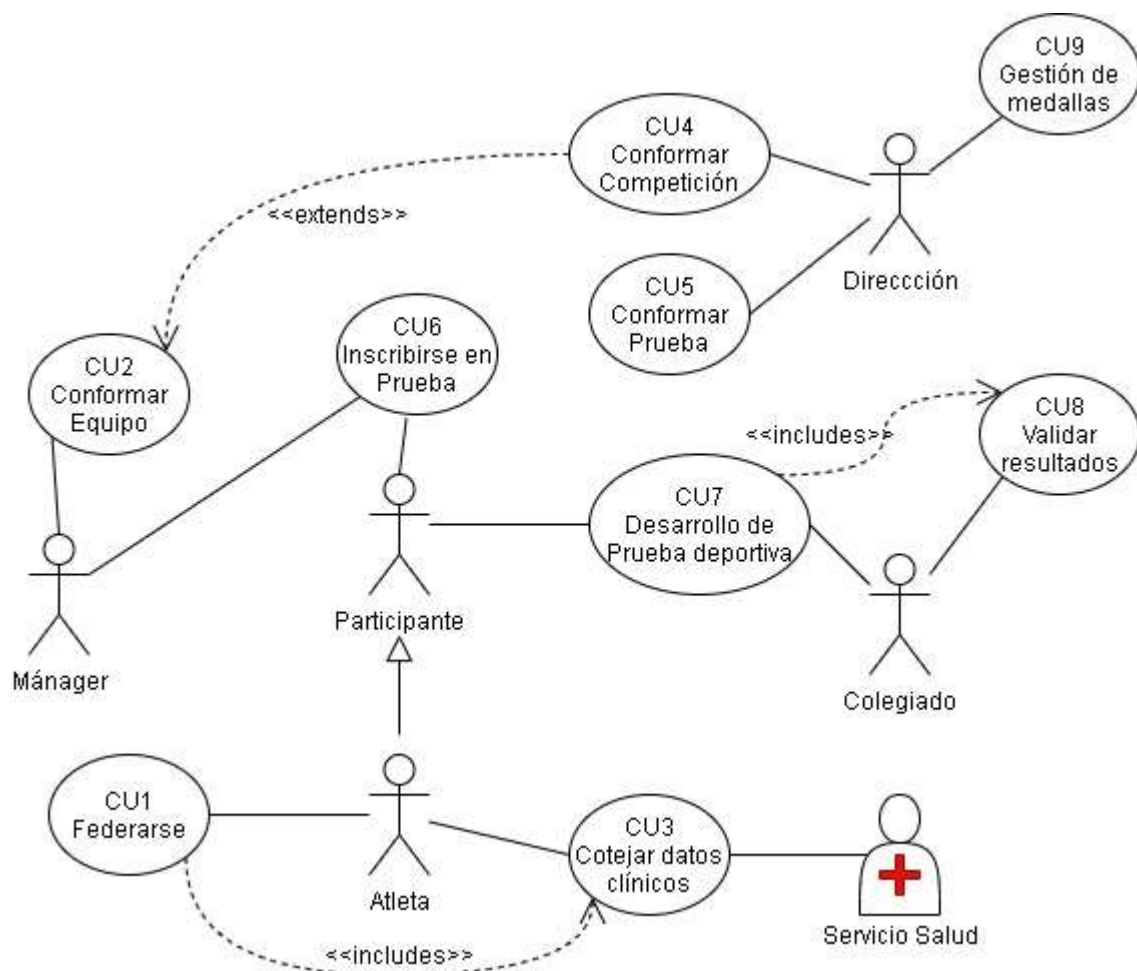


Ilustración 2 Diagrama de Casos de Uso

CU1. Federarse

Un atleta debe estar federado para poder competir en cualquier prueba de una competición de la federación. Aporta sus datos físicos (altura y peso) y se cotejan sus datos contra el servicio de salud (CU3).

CU2. Conformer Equipo

El mánager del equipo es quien conforma un equipo de entre 3 y 5 atletas diferentes. Lo hace al principio de la temporada cuando se conforma la competición del año y sólo está vigente esa temporada.

CU3. Cotejar Datos Clínicos

El Servicio de Salud es una entidad externa a la Federación contra la cual se cotejan los datos clínicos que los atletas incluyen al federarse, es decir, cuando un atleta se federa (CU1) siempre se cotejan sus datos clínicos contra los que están disponibles a través del servicio de salud.

CU4. Conformer Competición

La dirección de la federación es la encargada de conformar la/s competición/es de cada temporada al principio de ésta, de forma que se tienen todos los datos de todas las temporadas a lo largo de la historia de la federación.

CU5. Conformer Prueba

Es la propia dirección de la Federación la que conforma cada prueba de cada competición, indicando el nombre de esta, el tipo de prueba (colectiva o individual), así como la fecha y el lugar donde se llevará a cabo.

CU6. Inscribirse en Prueba

Las inscripciones en las pruebas dependen del tipo de ésta: si se trata de una prueba de tipo colectiva será el mánager que representa a cada equipo quien se encargue de la inscripción. Sin embargo, si la prueba es de tipo individual corresponde a cada atleta realizar su propia inscripción en una prueba determinada.

CU7. Desarrollo de Prueba

En cada prueba participan entre 3 y 10 competidores (ya sean atletas individualmente o por equipos), cada uno de ellos compite por una calle determinada y tiene un dorsal distinto. Todo el desarrollo de la prueba siempre es controlado por un conjunto de 3 colegiados, quienes toman los datos de los resultados de las pruebas que arbitran y los validan (CU8).

CU8. Validar Resultados de Prueba

La validación de los resultados de cada prueba siempre es efectuada tras el desarrollo de ésta y llevada a cabo por los colegiados que han controlado el desarrollo de la misma. Determina finalmente la validación del podio de cada prueba, es decir, el oro para el atleta ganador, seguido de la plata y el bronce para los siguientes atletas

CU9. Gestión de Medallas

Toda la gestión de los metales que se entregan a los participantes ganadores del podio de cada prueba es gestionada por el personal de la dirección de la federación. Un usuario de perfil 'Directiva' podrá realizar todas las operaciones de gestión sobre las medallas, es decir, crear nuevos metales, modificar los datos de un metal ya existente o eliminar una medalla del sistema.

Ejercicios:

1. (máx 2,5 pts.) Implementar todas las clases del diagrama de clases de la Ilustración 1, junto a sus relaciones.

Todas las clases de entidad dispondrán de un constructor por defecto y otro que tenga tantos argumentos como atributos obligatorios de la entidad a la que corresponda, atendiendo al tipo de dato indicado en el diagrama. Así mismo, también deberán disponer de los correspondientes métodos getter y setter de cada campo. La implementación de las diferentes relaciones entre las clases podrá realizarse según el diseño que cada estudiante considere oportuno, pero deberán respetarse las cardinalidades mínima y máxima a cada lado de la relación y todas han de ser implementadas.

- Implementar la enumeración *Lugar* cuyos elementos se corresponden con los datos de la siguiente tabla:

	Enum Lugar		
Value	String nombre	String ubicacion	boolean airelibre
GIJONMESTAS	Las mestas	Gijón	true
GIJONCENTRO	Centro ciudad	Gijón	true
OVIEDOSANFRANCISCO	Parque San Francisco	Oviedo	true
AVILESPUERTO	Puerto	Avilés	true
AVILESPABELLON	Pabellón deportivo Avilés	Avilés	false
OVIEDOCENTRO	Centro ciudad	Oviedo	true

Esta enumeración se usa como campo de la clase *Prueba*. Será necesario ajustar la implementación de la esa clase para que concuerde con la enumeración *Lugar*.

2. (máx 1,5 pts.) Implementar una nueva clase abstracta de nombre *Metal*, que representa una medalla, por lo que será la superclase de las clases *Oro*, *Plata* y *Bronce*:

1. (2/3 del ejercicio) dispondrá de un constructor por defecto, y de getters y setters para cada campo siguiente:
 - un campo para la fecha de compra de la medalla (sólo fecha).
 - un campo booleano de nombre asignada y establecido por defecto a `false`.
 - un campo pureza de tipo `float` que corresponderá con el campo propio de las subclases.
 - un método abstracto `public abstract float[] cotasPurezaEfectiva()` que devuelve un array de 2 posiciones con los valores efectivos de pureza mínimo y máximo de una medalla. Este rango se calcula así: $\text{pureza_efectiva} = \text{pureza} \pm x\%$ siendo $x=5\%$ para los Oros, $x=7\%$ para las Platas y $x=3\%$ para los Bronces
2. (1/3 del ejercicio) La clase implementará la interfaz `Comparable`, de manera que se establecerá un orden de las medallas en función del valor de su pureza y, si hay empate, según la fecha de compra de la medalla.
3. (máx 1,5ptos.) Sobrescribir la función `toString` de la clase `Prueba` de forma que se muestre, en este orden: el identificador de la prueba, junto a su lugar y fecha de celebración, el nombre de la prueba y el de la competición a la que pertenece, si se trata de una prueba individual o no y, a través de un iterador sobre el campo para los `Participantes`, los datos de cada uno de éstos. Se indicará:

“Por la calle <calle> con el dorsal <dorsal> /t el equipo/atleta <idEquipo/idAtleta>.”

En el caso de tratarse de una competición por equipos, habrá que indicar también la composición (identificadores de los atletas) de ese equipo. Por último, si la prueba está cerrada, es decir, tiene un `Resultado`, indicar los datos de los colegiados de la prueba y el podio final de la prueba.
4. (máx 1,5ptos.) Implementar una función para el proceso de registro de un nuevo `Equipo`.
 1. (20% del ejercicio) Crear la base de datos relacional del sistema sólo para las entidades `Equipo`, `Atleta`, `Manager` y `Persona`, y adjuntar junto a la entrega el script de creación de la BD y de esas tablas en formato compatible con MySQL.
 2. (40% del ejercicio) Preguntar al usuario por la salida estándar por el nombre del equipo y establecer el año de inscripción al actual. Luego preguntar por los datos para el Mánager del equipo. A continuación, los datos de los atletas (entre 3 y 5 personas) que lo conforman, los cuales serán introducidos por la entrada estándar.
 3. (40% del ejercicio) Registrar en la base de datos creada toda la información sobre el registro, insertando un nuevo registro en la tabla `equipos`, otro en la tabla `managers`, así como nuevos registros de las tablas `personas` y `atletas`. Devolver al usuario el valor del `idequipo` resultante, junto con toda la información recuperada desde la base de datos.
5. (máx 3 ptos.) Implementar los siguientes métodos para importar y exportar la información de la Federación a través de la E/S con ficheros de caracteres y ficheros de bytes.
 - Método para exportar a un fichero de texto la información completa de todas aquellas medallas cuya pureza es mayor o igual a una dada, con la posibilidad de elegir sólo las que ya han sido asignadas o todas las medallas que cumplan el filtro de búsqueda. El informe deberá separar las medallas resultantes de la búsqueda y presentarlas agrupadas por tipo, para luego mostrar el `idMedalla`, pureza y fecha de compra de cada medalla.
 - Método para importar desde un fichero de bytes que contiene varios objetos `Colegiado`, aquellos que sean de una categoría en concreto (la cual se pasa como parámetro en forma

de cadena de caracteres). El método devolverá la colección de objetos de la entidad `Colegiado` que cumplan ese criterio.

- Método para exportar hacia un fichero binario una serie de objetos de la entidad `Atleta`, que cumplen unos criterios físicos (valor de peso y de altura dentro de unos rangos dados) de entre una colección de `Atletas` que se pasa como parámetro. El método preguntará al usuario por el rango de valores para el peso y para la altura y buscará entre los elementos de la colección aquellos que cumplen las restricciones, exportando únicamente esos al fichero binario.