



Proyecto Programación #2

Gestor de Competiciones Gimnásticas

IES Francisco de los Ríos

1 Objetivo

Desarrollar un gestor de competiciones de gimnasia rítmica que permita crear las pruebas y los participantes, controlando el horario de cada actuación. Debe permitir crear y editar competiciones que estarán definidas por un nombre y una descripción, además de una fecha de inicio.

Dichas **competiciones** están compuestas de varias **pruebas**. Las pruebas se categorizan por el **tipo** (Individual o en Grupo), la **categoría** (Prebejamín, Benjamín, Alevín, Infantil, Junior y Senior) y el **aparato** (Mazas, Aro, Cinta, Cuerda y Manos Libres). **Por tanto, no puede haber dentro de una competición, dos pruebas del mismo tipo, misma categoría y mismo aparato.**

Cada prueba tendrá una **colección de participantes**. Cada **participante podrá "participar" en más de una prueba en una misma competición** y lo hará con **dorsales** distintos. Por ello, en realidad, cada prueba tendrá una lista de Participaciones.

Las **Participaciones** serán una clase **genérica** que contendrá el número de dorsal (único para cada prueba y para la competición), **el participante (qué podrá ser un grupo o una gimnasta)**, la **hora** de participación (no debe haber dos participaciones a la misma hora) y **unos puntos** obtenidos que se inicializa a 0.

La clase Prueba debe contar con un método **getWinner** que devolverá la participación(es) con mayor(es) puntuación(es) en cada momento. Devolverá una lista vacía si todas las participaciones están a 0 puntos.

Para modelar a los participantes, se contará con dos clases:

- **Gimnasta:** **dni**, **nombre**, **categoría** y **club**. A su vez, para posibles futuras ampliaciones del programa, como puede ser **gestión de jurado...**, esta clase heredará de **Persona** (**dni**, **nombre**, **teléfono** y **correo**).
- **Grupo:** **nombre**, **colección de gimnastas** y **club**. **(todos los miembros del grupo deben pertenecer al mismo club y misma categoría).**

los atributos
son
redundantes

Se pide generar una aplicación que mediante consola permita gestionar CRUD completo de sus componentes. Y permita persistir toda la información en sistema de ficheros (serializando o XML (preferiblemente)).

un grupo
solo tiene
hasta 5
participantes

Operaciones mínimas necesarias:

- Gimnastas: dar de alta, borrar, modificar y buscar por criterios.
- Grupos: crear, editar, insertar o borrar gimnastas y buscar por criterios.
- Competiciones: crear, editar, insertar pruebas y buscar por criterios.
- Pruebas: crear, editar, insertar participaciones y buscar por criterios.

IMPORTANTE ESTO

Todas las inserciones deben ir validadas.

No debe permitirse eliminar un componente que pertenece a otra relación. Es decir, si hay un gimnasta que ya ha participado en alguna prueba, no debe permitirse su borrado y debe ser comunicado dicho evento. (lo que en base de datos se denomina ON DELETE: RESTRICT).

2 Requisitos

- Se debe desarrollar el software mediante metodología SCRUM.
- Se debe emplear GIT como gestor de versiones.
- Se debe documentar el diagrama de clases, definiendo las interfaces, antes de comenzar el desarrollo.
- Se debe diseñar los casos de uso y flujo de pantallas, antes de comenzar el desarrollo.
- El ScrumMaster empleará una aplicación de gestor de tareas, como es TRELLO.
- Se empleará un IDE profesional, independientemente de la marca.
- Se empleará una JDK 1.8 de Oracle u OpenJDK (o cualquier otra distribución libre) para versiones posteriores.
- Se creará un proyecto Maven para el desarrollo del proyecto.
- Se desplegará la aplicación en un JAR para la realización de la demo.
- Se entregará el JAR y el PDF con la documentación que debe contener, como mínimo:
 - Alcance (descripción), Diseño (datos y diagrama de clases), Características reseñables del desarrollo, Enlace al repositorio Git y Tabla con porcentaje de desempeño de cada miembro del equipo.
- Se debe emplear: herencia, abstract, interfaces, enum, patrón singleton, colecciones, tratamiento de archivos, tratamiento y log de errores y patrón MVC.

3 Anexos

A pesar de que en este proyecto no vamos a emplear base de datos, se adjunta un posible modelo E-R para que sirva como base para el diseño del programa.

