



ACCESO a DATOS – 2º DAM

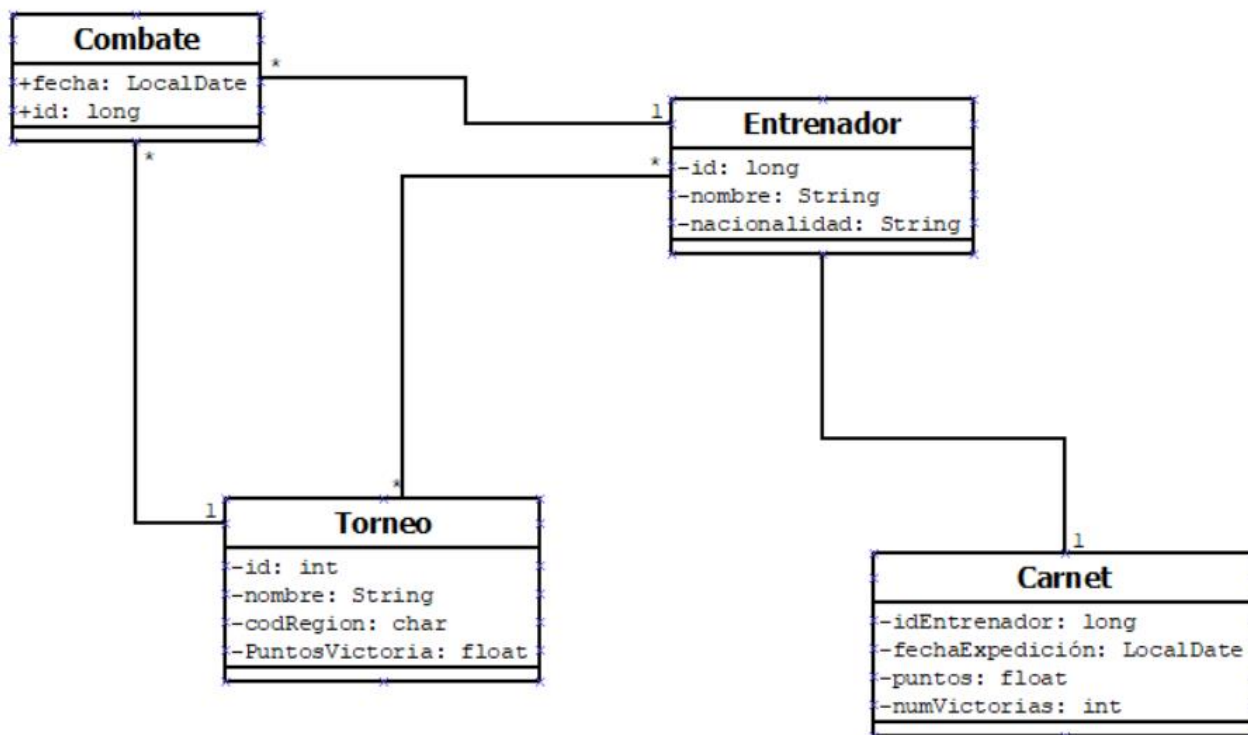
TAREA 2: Manejo de conectores

DESCRIPCIÓN:

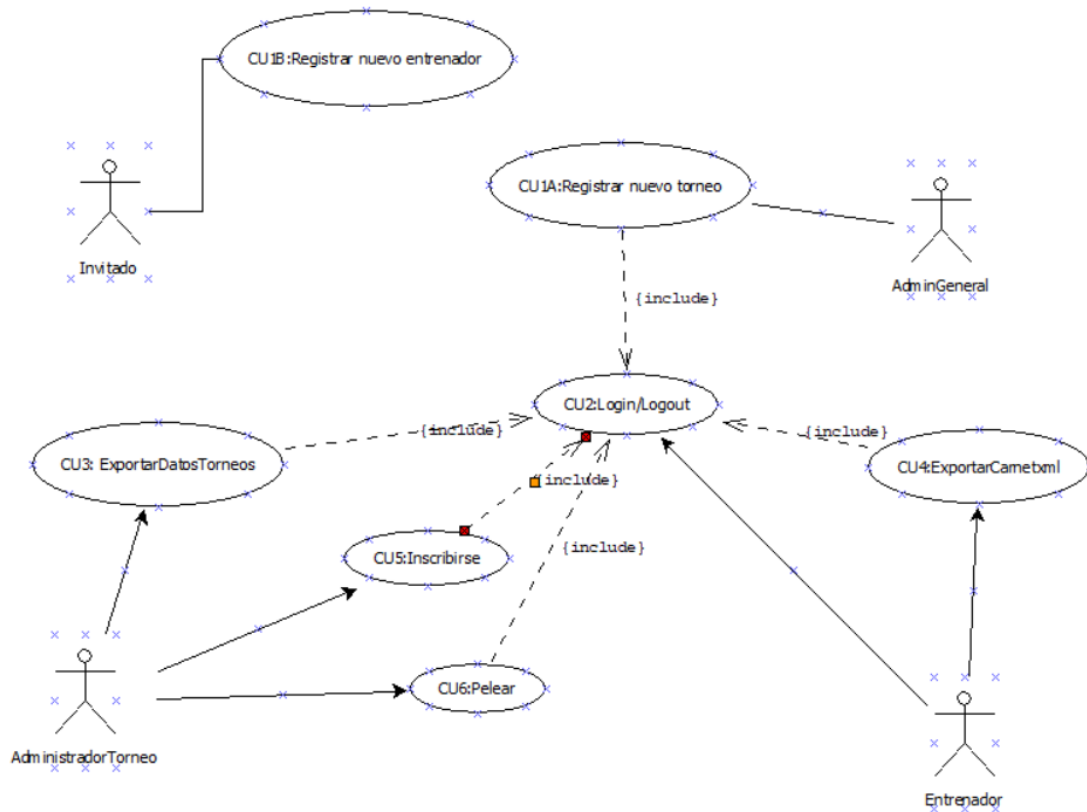
Se pretende implementar una aplicación Java que gestione la información relativa a combates que se realizan en diferentes gimnasios pokemon. Se tienen las siguientes especificaciones:

- Se desea almacenar la información de los **Entrenadores** (id, nombre y nacionalidad). Cada entrenador lleva su propio **Carnet** personal el cual almacena un id, la fecha de expedición del carnet, un campo que marca los puntos conseguidos por el entrenador hasta el día de hoy y el número de campeonatos que ha ganado.
- También se quiere tener registrada la información sobre los torneos que realiza un entrenador a lo largo de su carrera en los diferentes gimnasios. Para cada **torneo** se guarda su id, el nombre y el código de la región en la que está ubicado (un carácter propio para cada región) así como los puntos que otorga ganar (número aleatorio entre 50 y 100). Además, se quiere saber qué colección de entrenadores ha participado (será un máximo de 6) y quien lo gana, así como el conjunto de combates (serán 3) en que forman el torneo.
- En relación a los **Combates** deben reflejar la fecha en que se producen, que entrenadores participaron y quien lo gana y el torneo al que pertenece.

El modelo de clases de la aplicación es el siguiente:



acceso a diferentes funcionalidades del sistema. El diagrama de casos de uso inicialmente sería el siguiente:



De momento sólo se tendrán 4 perfiles de usuario:

- **Invitado:** es el perfil que no se autentica, por lo que sus funcionalidades son limitadas.
- **Administrador de torneo:** es la persona encargada de un torneo. Registrar los datos de los combates de los entrenadores en el torneo del que es responsable.
- **Entrenador:** es la persona que pelea en los torneos, realizando diferentes combates.
- **Administrador General:** es único en el sistema, podrá llevar a cabo tareas de administración tras introducir sus credenciales de acceso.

De momento sólo se estiman las siguientes funcionalidades:

- **CU1: Registrar:** es el proceso de alta en el sistema para un nuevo torneo para nuevo Entrenador.
 - A. **Nuevo Torneo:** el sistema permitirá la introducción de un nuevo torneo sólo al usuario Administrador general. Deberá dar el nombre y la región del nuevo torneo, así como el usuario de tipo administrador de torneo que será responsable del mismo (nombre y contraseña). El sistema comprobará que no exista ya ese torneo, informando en tal caso. Si todo va bien, el nuevo torneo quedará añadido al sistema habiéndole asignado un responsable administrador del mismo. También se crearán 3 combates asociados a dicho torneo con las fechas que se pidan por pantalla y se asignará la puntuación por ganar el torneo.
 - B. **Nuevo entrenador:** Un usuario invitado podrá, desde un torneo concreto, introducir sus datos (nombre y nacionalidad), para darse de alta en el sistema. Éste comprobará si ya existen esos datos, informando al usuario en tal caso. Si no, el invitado quedará dado de alta en el sistema y se le asignará un carné propio y único con la fecha de expedición actual y su primer torneo asociado.
- **CU2: Login/Logout:** es el proceso de autenticación/cierre de sesión en el sistema. El usuario introducirá sus credenciales de acceso al sistema y éste comprobará que son correctas (indicando error en caso contrario) para dar posteriormente acceso a las funcionalidades del sistema que le corresponda en función del perfil del usuario autenticado. De esta forma se guarda la sesión del usuario hasta el momento que cierra su sesión (Logout), cuando pasa a tener el perfil de invitado nuevamente.
- **CU3: Exportar datos torneo:** el responsable de un torneo podrá exportar los datos de los combates y entrenadores de su torneo. Se reflejarán los datos del torneo (id, nombre, región) la lista de los combates celebrados, así como los entrenadores que participaron con su nombre, id y si gana o no (Se mostrará por pantalla y se dará la opción de sacar por fichero).
- **CU4: Exportar Carnet en XML:** cada entrenador, tras autenticarse, podrá visualizar y exportar los datos de su propio carnet. Esto incluye: el id de entrenador, nombre y nacionalidad, fecha de expedición del carnet, puntos conseguidos y la colección de torneos (nombre y región) en los que ha participado hasta el día actual, indicando en qué fecha se realizó combates y si los gana o no. Esta información se exportará en formato XML, con la estructura del ANEXO I.
- **CU5: inscribirse:** un administrador de torneo recibe a un entrenador que llega a su torneo. Le pide su carnet de entrenador. Se asignarán los combates para ese entrenador en ese torneo (2 combates) y se rellenarán los datos de cada combate, actualizando los datos del entrenador. Si el torneo esta lleno se informara al entrenador de que no puede inscribirse.
- **CU6: pelear:** Un administrador de torneo ejecutara pelear. Se comprobará que el torneo tenga todos sus combates rellenos con entrenadores, si es así se resolverán los combates de manera aleatoria y el ganador del torneo (el que más combates gane, en caso de empate también aleatorio) se actualizarán los datos relativos a puntos conseguidos y al nº total de victorias de cada entrenador.

TAREA:

En primer lugar, deberá diseñarse un modelo relacional de tablas para persistir y recuperar la información relativa a todas las entidades anteriores, con sus correspondientes tipos de datos, valores por defecto, claves primarias y foráneas, así como otras restricciones. Deberá incluirse en la entrega el script sql con la creación de dicho modelo relacional, para poder ser creada una BD relacional MySQL de nombre `bdentrenadores_<nombreestudiante>`. Recuérdese que el proyecto Java ha de trabajar con el generador de proyectos Maven (a través de la inclusión de las dependencias que se estime necesarias en el fichero `pom.xml`).

Para esta tarea 2 se piden los siguientes apartados:

1. Preparar la BD de datos para que contenga, al menos, los siguientes datos iniciales:
 - 3 Torneos diferentes y 5 Entrenadores distintos (éstos con su correspondiente Carnet personal).
 - 1 Entrenador Con al menos 1 torneo terminado y las puntuaciones obtenidas.
 - 1 Torneo con todos los combates llenos listo para ejecutar pelear.
 - 1 Torneo con hueco para inscribir al menos un nuevo entrenador
 - Todos los datos que se manejen inicialmente deberán ser coherentes y no podrá haber inconsistencia en ellos. Esto incluye el disponer (de alguna forma) de un sistema de cálculo de la distancia entre 2 paradas para marcar ese dato en el Carnet del Peregrino. Incluir en la entrega esa estructura para las distancias con datos válidos.
 - Incluir en el script de creación de la BD relacional las sentencias para incorporar directamente los datos iniciales.
2. Refactorizar las clases del programa para que se trabaje por capas, distinguiendo, al menos, una capa propia para las entidades del sistema, otra para las clases DAO de acceso a datos y otra capa para la vista, es decir, el programa principal y otras clases para presentación de la información.
3. Implementar una clase propia para la realización de la Conexión a la BD relacional. Ésta tomará los datos de la conexión a partir de un fichero de propiedades (`properties`). Se trabajará con un pool de forma que habrá sólo una conexión a la BD mientras el programa esté ejecutándose.
4. Implementar las clases DAO para las 4 entidades marcadas en el diagrama de clases de forma que se permita realizar las siguientes operaciones:
 - Dar de alta un nuevo elemento de las entidades Entrenador, Carnet, Torneo y Combate.
 - Buscar un elemento por ID, para las entidades Entrenador, Carnet, Torneo y Combate.En ambos casos, estas operaciones deberán crear/recuperar las tuplas correspondientes en todas las tablas necesarias en cada caso, para persistir/tomar la información completa de todo el elemento.
5. Reimplementar los casos de uso CU1: Registrar (nuevo torneo y nuevo Entrenador), CU2: Login/Logout y CU4: Exportar Carnet XML para que ya trabajen con las funciones implementadas en las clases DAO,

de forma que el registro exitoso de un nuevo elemento Entrenador/Torneo será persistido en la BD relacional y, para recuperar todos los datos del Entrenador y de su Carnet o de los combates/torneos, se realizará a través de métodos que recuperen la información desde la base de datos.

6. Implementar el CU3: Exportar datos de torneo. El detalle de este caso de uso es el siguiente:

CU3: Exportar datos de Torneo
Precondiciones:
El usuario se encuentra ya autenticado con el perfil de administrador de torneo.
Postcondiciones:
Se muestra al usuario un informe con los datos sobre los combates realizadas por parte de los entrenadores en el torneo en cuestión.
Acciones:
1) El usuario accede a la pantalla de exportación de datos del torneo. 2) El sistema carga directamente la información básica de su torneo y la muestra por pantalla. 3) El sistema mostrará los datos del torneo (id, nombre y región) y lista de combates, mostrará, para cada uno de ellos, el id del combate, el nombre de los entrenadores, la fecha en que se realizó y cual gana. 4) El sistema preguntará si quiere exportar a fichero o volver a la pantalla principal.
Excepciones
1) Si el torneo seleccionado es inválido se avisa al usuario para que vuelva a introducir su selección.

NOTA: Se deberá trabajar a partir del proyecto de la tarea anterior, estructurando adecuadamente el nuevo código fuente, refactorizando el que ya había para que funcione todo correctamente.

ENTREGA:

La entrega se realizará a través del aula virtual del módulo en Educastur, y consistirá en un archivo comprimido (.zip o .rar) con el proyecto Java de la solución que deberá estar correctamente preparado para su despliegue y ejecución directamente, sin la necesidad de configuraciones añadidas, con una correcta compilación y ejecución del programa. Se tendrán en cuenta todos los mecanismos de seguridad que se añadan a la solución, con el fin de evitar cierres inesperados del programa.

ANEXO I:

- A) El formato con el que estarán dispuestos los datos del fichero *credenciales.txt* es el siguiente:

nombre contraseña perfil id

donde *perfil* podrá ser: o bien la cadena de caracteres "torneo", en cuyo caso el campo *id* siguiente será el identificador del campeonato del que es administrador/responsable, o bien "entrenador", en cuyo caso el campo *id* siguiente corresponderá al identificador del entrenador.

- B) Además, en el fichero binario *torneo.dat* se guardan los datos básicos de un objeto torneo: id, nombre, región, puntos victoria y el nombre del usuario administrador/responsable del torneo.

- C) El formato del fichero XML *nombre_entrenador.xml* para el carnet de un entrenador será así:

```
<carnet>
  <id>xxxx</id>
  <fechaexp>xx-xx-xxxx</fechaexp>
  <entrenador>
    <nombre>xxxxxxxx</nombre>
    <nacionalidad>xxxxxxxx</nacionalidad>
  </entrenador>
  <hoy>xx-xx-xxxx</hoy>
  <puntos>xx.x</puntos>
  <torneos>
    <torneo>
      <nombre>xxxxxx</nombre>
      <region>X</region>
      <combates>
        <combate>
          <id></id>
          <fecha></fecha>
          <victorio></victoria>
        </combate>
      </combates>
    </torneo>
    ( . . . )
  </ torneos >
</carnet>
```