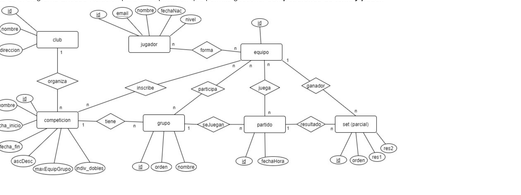
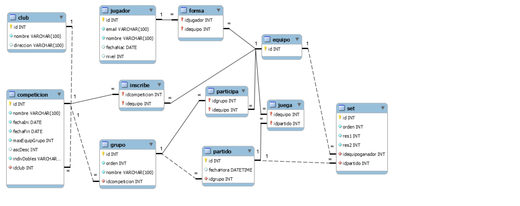
Acceso a Datos

Práctica. Competiciones Tenis/Pádel

Este es el diagrama Entidad-Relación para una aplicación que permita gestionar ***competiciones de tenis y pádel***:



A continuación se muestra el modelo relacional (tablas) obtenido a partir de D.E.R. anterior. Podés descargar el script de creación de la **base de datos tenispadel**.



1) Utilizando el plugin Hibernate Tools y partiendo de la base de datos anterior debés obtener las clases Java y archivos de mapeo Hibernate que manejará nuestra aplicación.

2) Todos los programas que compongan nuestra aplicación tendrán interfaz de usuario gráfico. Debés desarrollar los siguientes procesos:

MANTENIMIENTO DE DATOS MAESTROS (CRUD o ABMC)

Todos los procesos de este apartado seguirán un **operativa similar** a la del *Ejemplo CRUD géneros* (disponible en el aula virtual). La operativa es la siguiente:

- Se permite escribir en los campos del formulario excepto en la clave primaria (**id**), que será visible pero no editable.
- Al pulsar el botón de **búsqueda** aparecerá un listado con los registros de la BD filtrados según lo que haya escrito el usuario en los campos del formulario inicial. Por ejemplo, si en *Descripción* ha escrito **ta** en el listado (filtrado LIKE) podrán aparecer los géneros Fan**ta**sía, Fan**ta**stico, etc. Si el usuario no ha escrito nada se mostrarán todos los registros.
- Cuando se muestra el listado de registros se podrá:
  1. Seleccionar un registro, los datos se mostrarán en el formulario y se permitirá **guardar** y **borrar** (al borrar **siempre se pedirá confirmación** al usuario)
  2. Pulsar sobre Nuevo se volverá al formulario con todos sus campos vacíos permitiendo **guardar**.
  3. Cancelar, se vuelve al formulario.
- En el formulario, además de los botones Guardar, Borrar y Buscar, Siempre habrá un botón ***Limpiar*** o *Iniciar* que dejará el formulario como si se hubiese iniciado el proceso.
- Cuando se lea un String de un formulario siempre se pasará por trim() para quitar posibles espacios al comienzo o final de éste.
- Todas las fechas se leerán/mostrarán en formato **DD/MM/AAAA**.
- Al finalizar una acción (alta, baja, modificación) se informará al usuario del éxito o fallo de ésta mediante el correspondiente aviso, p.ej. mediante un *JoptionPane*.

Proceso Clubes

Datos: nombre, dirección.

Validaciones: nombre no puede estar vacío.

Búsquedas por: nombre y dirección.

Proceso Jugadores

Datos: email, nombre, fecha de nacimiento, nivel.

Al dar de alta un jugador se creará automáticamente un equipo individual formado por el propio jugador.

Validaciones: email y nombre no pueden estar vacíos. La fecha debe ser válida. Nivel > 0.

Búsquedas por: email y nombre.

Proceso Parejas (equipos)

Este proceso permitirá seleccionar dos jugadores ya existentes para formar un equipo de dobles (pareja).

Se permitirá borrar las parejas siempre que estas no estén inscritas en ninguna competición.

Datos: identificadores de los jugadores que forman la pareja.

Validaciones: Un jugador no puede formar pareja consigo mismo.

Búsquedas por: email y nombre.

GESTIÓN DE COMPETICIONES

Proceso Inscripciones

Este proceso comenzará permitiendo seleccionar un club y una competición. **Una vez seleccionado el club y la competición**:

Permitirá agregar o quitar equipos a la competición. Sólo se permitirán nuevas inscripciones si la competición aún no se ha iniciado (la fecha del sistema debe ser anterior a la fecha de inicio de la competición).

Validaciones: Si la competición es de dobles sólo se permitirá la inscripción de parejas (equipos formados por dos jugadores). Si la competición es individual sólo se permitirá la inscripción de equipos formados por un único jugador.

Proceso Competiciones

Este proceso comenzará permitiendo seleccionar un club. **Una vez seleccionado el club**:

Datos: nombre, fecha de inicio, fecha de fin, número máximo de participantes por grupo, número de equipos que ascienden o descienden, individual o dobles

Validaciones: nombre no puede estar vacío. Las fechas deben ser válidas y la fecha de fin posterior a la fecha de inicio. Número máximo de participantes no puede estar vacío y debe ser > 2. Obligatorio elegir individual o dobles.

Búsquedas por: nombre y club(el seleccionado al comienzo).

Proceso creación inicial de grupos

Este proceso comenzará permitiendo seleccionar un club y una competición. **Una vez seleccionado el club y la competición:**

Se mostrará el nombre de los jugadores o parejas inscritas y se permitirá lanzar la creación grupos. Se ordenarán los equipos por nivel (el nivel de una pareja es la suma de los niveles de ambos jugadores, si un jugador de la pareja no tiene nivel se considera que la pareja tampoco lo tiene). Los grupos se crearán empezando por los equipos de mejor nivel de forma secuencial (el mejor nivel es el 1). El nombre y orden un grupo recién creado será: **Grupo n (de orden n)**, siendo n=1, 2, 3, 4, … hasta agrupar a todos los equipos inscritos.

Los equipos de los que no se disponga nivel se agruparán en los últimos grupos de forma aleatoria.

Por ejemplo: Sea una competición con 4 participantes por grupo. Con estos equipos (nivel entre paréntesis) inscritos: E1 (3), E2(1), E3. E4(8), E5(2), E6, E7, E8(10), E9, E10(2), E11(3). Los grupos, teniendo en cuenta que algunos equipos no tienen nivel, podrían quedar así:

Grupo 1 (orden 1): E2 E5 E10 E1

Grupo 2 (orden 2): E11 E4 E8 E6

Grupo 3 (orden 3): E3 E9 E7

Una vez creados los grupos se crearán todos los partidos asociados al mismo (sin resultado aún). Cada equipo de un grupo jugará un partido con el resto de los equipos de su grupo.

Validaciones: Nunca se creará un grupo con un único participante (se mostrará un aviso indicando el equipo que no se puede agrupar).

Proceso promoción de grupos (crear nueva competición a partir de otra finalizada)

Este proceso comenzará permitiendo seleccionar un club y una **competición ya finalizada**. **Una vez seleccionado el club y la competición:**

Se pedirá el nombre y fechas (deben ser posteriores a la fecha de finalización de la competición original) de la nueva competición y se creará con los mismos atributos que la seleccionada (excepto el nombre y fechas).

No se requerirá inscripción de los equipos para este proceso. Los grupos de esta nueva competición se formarán a partir del número de equipos que ascienden o descienden *n* y los puntos obtenidos por cada equipo dentro de su grupo. Ver puntuación en el proceso *Tabla de clasificación de un grupo*.

Los *n* primeros equipos de cada grupo subirán al grupo superior (orden-1) y los *n* últimos equipos de cada grupo descenderán al grupo inferior (orden+1). Los *n* primeros jugadores del mejor grupo y los *n* últimos jugadores del peor grupo permanecerán en el mismo grupo.

Introducción resultados de partidos

Este proceso comenzará permitiendo seleccionar un club, una competición y un grupo. **Una vez seleccionado el club y la competición:**

Se permitirá seleccionar alguno de sus partidos para introducir la fecha y la hora a la que se ha jugado (si se deja en blanco tomar fecha y hora del sistema), y el resultado de cada set (o modificar un resultado introducido previamente). Un resultado se compone de dos o tres parejas de números, por ejemplo: *7/6 5/7 6/4*. En el caso de que se introduzca la fecha y hora del partido, esta debe estar dentro del periodo de la competición y ser anterior a hora del sistema.

Las combinaciones válidas en la pareja de números que conforman el resultado de un set son: 7/6 7/5 6/4 6/3 6/2 6/1 ó 6/0.

Cada pareja de números se refiere a un set. Gana el partido aquel jugador que gana dos sets (puede haber partidos de dos o de tres sets).

INFORMES

Tabla de clasificación de una competición

Este proceso comenzará permitiendo seleccionar un club y una competición. **Una vez seleccionado el club y la competición:**

Se mostrará la lista de todos los equipos participantes agrupados. Primero se mostrarán los grupos de menor orden (de más nivel a menos), dentro de cada grupo los equipos se ordenarán de mayor a menor puntuación. La puntuación se obtiene cada vez que se juega un partido. Se **otorgan 3 puntos al equipo ganador y 1 punto al equipo perdedor**. En caso de igualdad de puntos entre dos o más equipos, se clasificará primero a aquel equipo que tenga mejor diferencia setsGanados-setsPerdidos, si el empate a puntos y diferencia de sets persiste se clasificará primero a aquel equipo que tenga mejor diferencia juegosGanados-juegosPerdidos, para calcular las diferencias anteriores se tendrán en cuenta todos los partidos disputados durante la competición.

Partidos jugados por un equipo

Este proceso comenzará permitiendo un equipo (individual o pareja). **Una vez seleccionado el equipo:**

Se mostrarán cronológicamente (de más recientes a más lejanos) los partidos que ha disputado.

Por cada partido se mostrará: jugadores del equipo contrario, resultado, fecha, nombre del grupo, competición y club.

Búsquedas por: nombre de jugador (para buscar equipos)

3) Al arrancar la aplicación se leerán del fichero **CFG.INI**, que debés incluir en el proyecto, los parámetros propios del tipo de persistencia que se quiera utilizar:

```
#Este es un fichero CFG.INI similar al que tendrás que utilizar
#Valores posibles de la propiedad tipoPersistencia: mysqlJDBC, hibernate)
tipoPersistencia=hibernate
#Propiedades para tipoPersistencia=mysqlJDBC
mysqlJDBC.servidor=localhost
mysqlJDBC.puerto=3306
mysqlJDBC.basedatos=competicionesFutbol
mysqlJDBC.usuario=root
mysqlJDBC.password=manager
#Propiedades para tipoPersistencia=hibernate
#El siguiente parámetro que será el que se pase al método configure() de Hibernate
hibernate.archivoCFG=xml/hibernate.cfg.xml
```

```
//Podemos utilizar un objeto Properties de la API de Java para cargar el archivo anterior:
Properties prop=new Properties();
//Carga del fichero CFG.INI (load, podéis indicar la ubicación de CFG.INI de dos formas. Sólo una de ellas)
prop.load(new FileReader("CFG.INI")); //CFG.INI se buscará en el path del proyecto Eclipse
prop.load(new InputStreamReader(PrincipalGUI.class.getResourceAsStream("CFG.INI"))); //CFG.INI se buscará en el mismo paquete que la clase PrincipalGUI

String tipoPer=prop.getProperty("tipoPersistencia");
```

4) Es **obligatorio** separar en diferentes clases el código que maneja el interfaz de usuario y el código que realiza el almacenamiento/recuperación de la información. Desde la clase que controla el GUI se interaccionará con el almacén de información a través de los métodos relacionados en el ***interface Persistencia*** que será implementado por una clase diferente para cada tipo de persistencia. A continuación se muestra un ejemplo de aplicación utilizando este interface:

```
public interface Persistencia {
//Los siguientes métodos son un ejemplo de lo que la clase GUI necesitaría para almacenar/recuperar información
//Cada alumno puede definir su propia lista de métodos que no tiene porque coincidir con la siguiente
        Jugador consultarJugador(String email);
        int guardarJugador(Jugador j); //devuelve la clave autogenerada
        ...
}

class PrincipalGUI{
public static void main(String[] args) {
//Cargamos CFG.INI
Properties prop=new Properties();
prop.load(new InputStreamReader(PrincipalGUI.class.getResourceAsStream("CFG.INI"))); //CFG.INI se buscará en el mismo paquete que la clase PrincipalGUI
String tipoPersistencia=prop.getProperty("tipoPersistencia");

Persistencia per;
switch (tipoPersistencia){
   case "mysqlJDBC":
        String servidor=prop.getProperty("mysqlJDBC.servidor");
        String baseDatos=prop.getProperty("mysqlJDBC.basedatos");
        String puerto=prop.getProperty("mysqlJDBC.puerto");
        String usuario=prop.getProperty("mysqlJDBC.usuario");
        String password=prop.getProperty("mysqlJDBC.password");
        per=new PersistenciaMySQL(servidor,puerto,baseDatos,usuario,password);
        break;
   case "hibernate":
        per=new PersistenciaHibernate(prop.getProperty("hibernate.archivoCFG"));
        break;
   default: //ERROR
   }
...
}

class PersistenciaMySQL implements Persistencia{
...
}

class PersistenciaHibernate implements Persistencia{
...
}
```

5) Instrucciones de entrega.

- El proyecto Eclipse debe nombrarse así: ***TenisPadelNombreApellidosDelAlumno***
- Además de la aplicación que desarrolléis el proyecto debe incluir estos archivos que tienen que ser importados en el proyecto Eclipse:
- Un script llamado **datos.sql** (se puede generar con *mysqldump -p -uroot tenispadel > datos.sql*) que contenga los datos de la BD utilizados durante la exposición de la práctica.
- **Opcionalmente un PDF** describiendo cualquier funcionalidad adicional que hayáis desarrollado respecto a lo descrito en la lista de procesos anterior.

Exportar el proyecto Eclipse (.zip) y subirlo al aula virtual.