



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico #1

Sistema de inscripción Mundial de Irlanda 2017

16 de junio de 2017

Bases de Datos

Grupo 7

Integrante	LU	Correo electrónico
Abdala Leila	950/12	abdalaleila@gmail.com
Bernaus Andres	699/10	andres.bernaus@hotmail.com
Gonzalez Alejandro	32/13	gonzalezalejandro1592@gmail.com
Romero Lucas	440/12	lucasrafael.romero@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



Índice

1. Introducción	2
2. Modelo	2
2.1. Inscripciones	2
2.2. Competencias, modalidades y resultados obtenidos	3
2.3. Arbitraje	3
2.4. Diagrama Entidad Relación	4
3. Modelo Relacional	5
3.1. Restricciones	6
4. Implementación de Funcionalidades	6
5. Conclusión	10

1. Introducción

Con motivo del Campeonato Mundial de Taekwondo ITF, se desea modelar e implementar la base de datos para el sistema de inscripciones del mismo.

El objetivo principal de esta base de datos será, en primer lugar, responder a las consultas relacionadas con el proceso de inscripción en sí. Es decir, obtener el listado de inscriptos de las escuelas, listado de inscriptos por categoría, categorías en las que participa cada competidor, lista de equipos por país, etc. El modelo también debe permitir obtener los resultados del torneo, como por ejemplo las medallas obtenidas por cada país, puntajes de cada una de las escuelas que participaron, etc.

2. Modelo

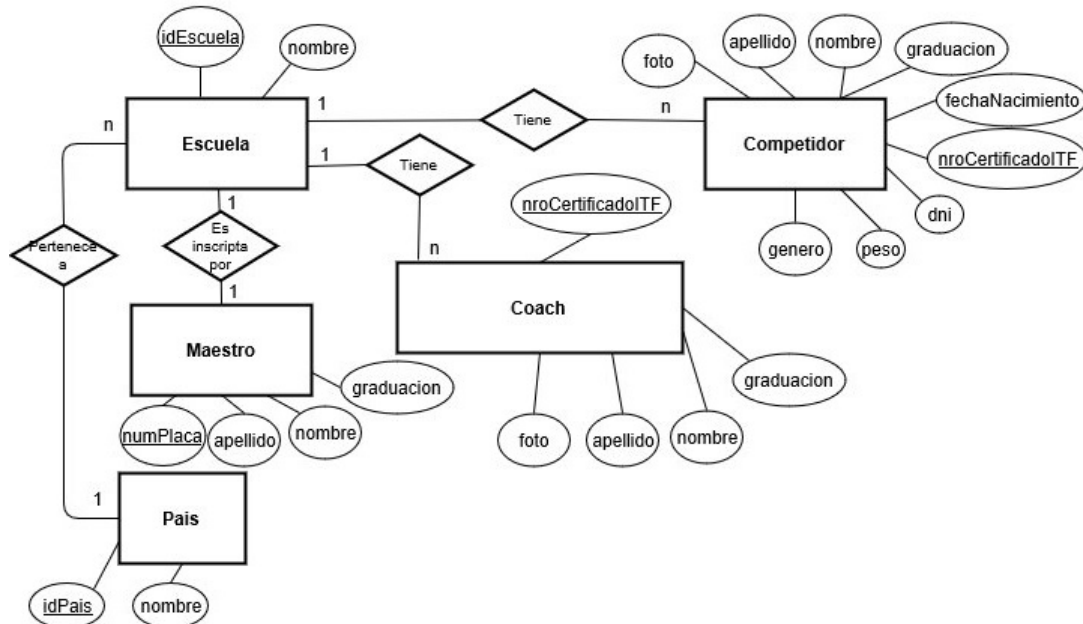
A continuación veremos en detalle los distintos aspectos del problema a resolver, y como decidimos modelarlos para cumplir con los requerimientos del problema.

2.1. Inscripciones

Para poder participar en el Torneo, los maestros de cada escuela deberán inscribir a sus alumnos y coach's en el certamen. A partir de esto, podemos deducir al menos 3 entidades: el **maestro** los alumnos inscriptos y los coach's. Sin embargo, podríamos obtener una mayor cantidad de entidades dependiendo de como se interprete a alguno de ellos (alumnos inscriptos, competidores). Tomemos por ejemplo al Maestro, este podría ser modelado como una entidad con sus datos personales, el nombre de su escuela, y país como atributos; o podríamos crear la entidad **Escuela** y que esta tenga como atributos nombre y país. Permittiéndonos acceder a estos datos de manera sencilla al momento de relacionarlas con sus competidores y resultados obtenidos en la competencia.

Para los alumnos y coachs inscriptos decidimos modelarlos en las entidades **Competidor** y **Coach** respectivamente. Utilizando como clave primaria para ambos casos el numero de certificado ITF, que asumimos, es único para cada persona. Para el caso en que los Coachs participan a la vez como Competidores, simplemente nos aseguraremos de que los datos repetidos (nombre, certificado ITF, graduación, etc) sean consistentes en ambas entidades.

Obteniendo de momento el siguiente DER:



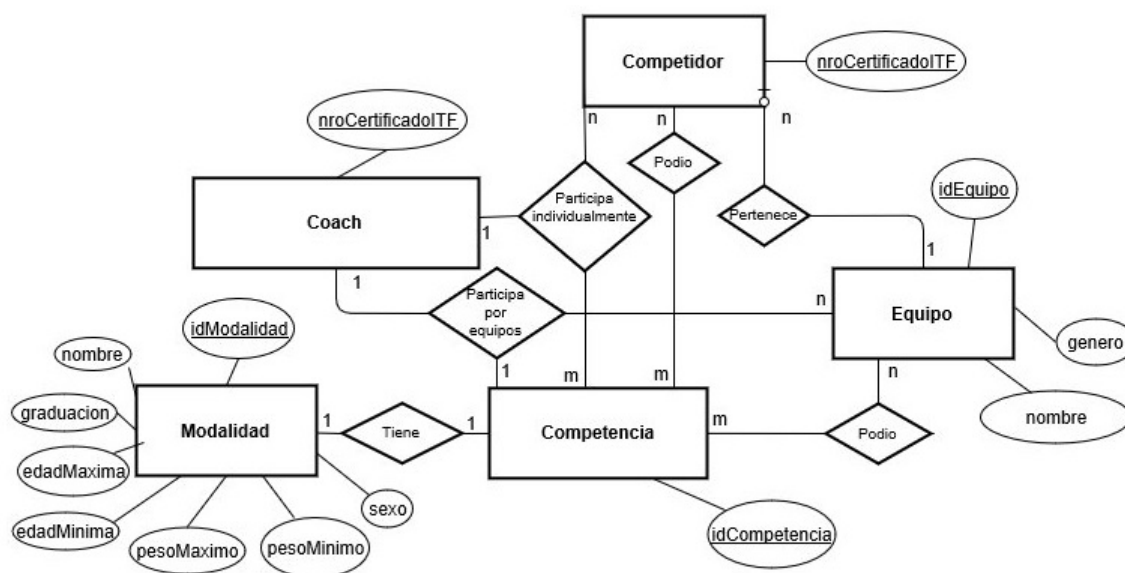
2.2. Competencias, modalidades y resultados obtenidos

El siguiente aspecto a modelar es la participación de los alumnos y Coachs en cada competencia. Dado que cada competidor debe estar acompañado por un Coach al momento de competir, decidimos modelarlo como una relación ternaria 1-n-m entre las entidades Coach, Competidor y Competencia respectivamente. Recordando siempre de chequear las restricciones que posee dicha competencia con respecto a la edad, peso, genero y graduación de sus integrantes; se creo también la entidad **Modalidad** la cual, además de tener las distintas disciplinas disponibles (Formas, Combate, Salto, Rotura de Potencia, etc) contendrá también las restricciones correspondientes para cada una de ellas, como por ejemplo edad mínima y máxima, peso mínimo y máximo, graduación, etc.

Para el caso de combates por grupo decidimos crear otra relación ternaria entre las entidades **Coach**, **Equipo**, y **Competencia**. La entidad Equipo se relacionara a su vez de forma parcial con la entidad Competidores, pues no todos los competidores participan en la modalidad de combate por grupos.

Se deberá además constatar que aquellos competidores que formen un equipo sean del mismo sexo, y pertenezcan a la misma escuela.

El ultimo aspecto a modelar son los resultados de los participantes de cada competencia. Para ello agregamos dos interrelaciones n a m entre las Competencias y sus competidores (o equipos), permitiendonos consultar las posiciones obtenidas en cada categoría y modalidad.



2.3. Arbitraje

Por ultimo, nuestro modelo debe ser capaz de obtener los árbitros de cada país que participaron en el torneo, y además la lista de árbitros que participaron como "Arbitro Central" en las modalidades de combate.

Para cumplir con esto se definieron las entidades:

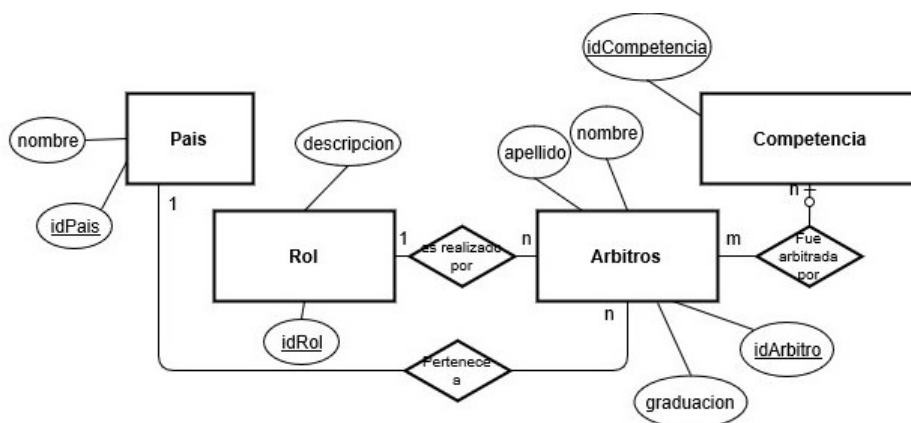
- **Arbitro** que contendrá toda la información de cada uno de los árbitros que participan, incluyendo el país al que pertenece.
- **Rol** que nos permitirá tener tabuladas todas las posibles funciones que pueden ejercer los árbitros del torneo. Es decir, arbitro central, presidente de mesa, juez o suplente.

Luego cada arbitro habrá realizado un rol determinado sobre un ring, formando entonces una relación 1 a n con la entidad rol; y además habrá participado en varias competencias. Lo que se

reduce a una relación n a m entre las entidades **Competencia** y **Arbitro**.

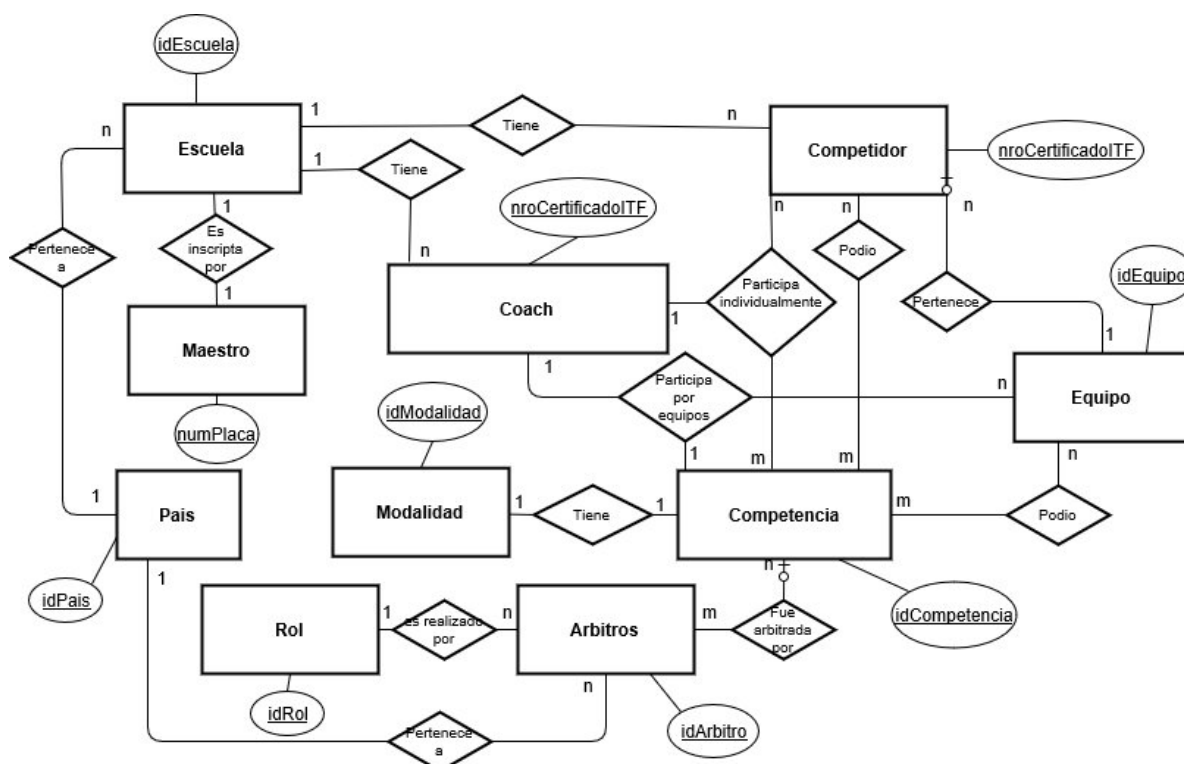
Si se quieren consultar los “Árbitros Centrales” de las modalidades de combate, lo único que debemos hacer es obtener los árbitros que ejercieron dicho rol, y comprobar si existe una entrada que los relacionen con alguna de las competencias con modalidad de combate.

En el caso de la consulta de árbitros por país, se puede realizar filtrando todo los elementos de la entidad Árbitros que pertenezcan al país deseado.



2.4. Diagrama Entidad Relación

A partir de todas los puntos presentados anteriormente obtenemos el siguiente DER: (Notar que únicamente mostraremos las entidades sus interrelaciones y claves primarias para observar todo el diagrama con mayor claridad)



3. Modelo Relacional

Una vez obtenido el diseño de la base de datos que deseamos modelar, el siguiente paso es comenzar a transformar dicho modelo en un esquema de datos lógico. Para ello utilizamos el Modelo Relacional, en el que cual reduciremos las distintas Entidades, con sus atributos e interrelaciones, a una estructura llamada esquema de relación. Esta estructura estará compuesta por un nombre y una lista de atributos, que se obtendrán a partir de las entidades y relaciones vistas en el DER.

- Pais(idPais, nombre)
PKCK{idPais}
- Escuela(idEscuela, nombre, idPais)
PKCK{idEscuela}
FK{idPais}
- Maestro(numPlaca, apellido, nombre, graduacion, idEscuela)
PK{numPlaca}
FK{idEscuela}
- Coach(nroCertificadoITF, apellido, nombre, graduacion, dni, idFoto, idEscuela)
PK{nroCertificadoITF}
CK{nroCertificadoITF,dni }
FK{idEscuela,idFoto}
- Equipo(idEquipo, nombre, generoEquipo)
PK{idEquipo}
- Competidor(nroCertificadoITF, apellido, nombre, fechaNacimiento, genero, graduacion, idEscuela, dni, peso, idFoto, idEquipo, esTitular)
PK{nroCertificadoITF}
CK{nroCertificadoITF,dni}
FK{idEscuela,idEquipo}
- Modalidad(idModalidad, edadMinima, edadMaxima, sexo, pesoMinimo, pesoMaximo, nombreModalidad, graduacion)
PK{idModalidad}
- Competencia(idCompetencia, idModalidad)
PK{idCompetencia}
FK{idModalidad}
- Rol(idRol, descripcion) PKCK{idRol}
- Arbitro(idArbitro, apellido, nombre, graduacion, idRol, idPais, idGrupoArbitro)
PK{idArbitro}
FK{idRol}
- Arbitraron(idCompetencia, idArbitro)
PK{(idCompetencia,idArbitro)}
FK{idCompetencia,idArbitro}
- PodioPorCompetidor(idCompetencia, nroCertificadoITF, posicion)
PK{(idCompetencia,nroCertificadoITF)}
FK{idCompetencia,nroCertificadoITF}
- PodioPorEquipo(idCompetencia, idEquipo, posicion)
PK{(idcompetencia,idEquipo)}
FK{idCompetencia,idEquipo}

- $\text{ParticipaIndividualmente}(\underline{\text{idCompetencia}}, \underline{\text{nroCertificadoITFCoach}}, \underline{\text{nroCertificadoITFCompetidor}})$
 $\text{PK}\bar{\text{CK}}\{(\text{idCompetencia}, \text{nroCertificadoITFCompetidor})\}$
 $\text{FK}\{\text{nroCertificadoITFCoach}, \text{nroCertificadoITFCompetidor}, \text{idCompetencia}\}$
- $\text{ParticipaEquipo}(\underline{\text{idCompetencia}}, \underline{\text{nroCertificadoITFCoach}}, \underline{\text{idEquipo}})$
 $\text{PK}\{(\text{idEquipo}, \text{nroCertificadoITFCoach})\}$
 $\text{CK}\{(\text{idEquipo}, \text{nroCertificadoITFCoach}), (\text{idEquipo}, \text{idCompetencia})\}$
 $\text{FK}\{\text{nroCertificadoITFCoach}, \text{idEquipo}, \text{idCompetencia}\}$

3.1. Restricciones

A continuación, describiremos las restricciones que se deben mantener a la hora de generar las distintas entidades y relaciones para así modelar correctamente el problema.

- Los campos de idPais de los árbitros y escuelas corresponden a entradas validas en la entidad Pais.
- En las relaciones entre las competencias y los Competidores los valores de los nroCertificadoITF de los competidores y coaches deben ser elementos validos de las entidades Competidor y Coach.
- Al asignar un competidor a un equipo se debe chequear que sea del mismo sexo que el resto de sus compañeros de equipo y que pertenezcan a la misma escuela.
- Los Coachs que también participan como competidores tienen datos repetidos. Estos datos deben ser consistentes.
- Al crear una Competencia se deben tener asignados a todos los árbitros necesarios para el desarrollo de la misma. Es decir, se le debe asignar el rol correspondiente a cada uno de los árbitros; y se deberá controlar que no haya mas de un arbitro para los roles de presidente de mesa y arbitro central.
- Los competidores que participan en una competencia deben tener un peso, genero, edad y graduación consistente con la modalidad de dicha competencia.
- Los árbitros tienen la graduación suficiente para dirigir las competencias en las que participan.
- Los Coachs deben acompañar a participantes de su escuela.
- Hay un Coach por cada 5 alumnos inscritos en una escuela.
- Las modalidades utilizan sus restricciones conforme a lo establecido por el enunciado. Es decir que, por ejemplo, no deberá ser posible realizar restricciones sobre el peso, a los competidores inscritos en la modalidad de Formas.

4. Implementación de Funcionalidades

Una vez que terminado el modelo del problema, el siguiente paso fue comenzar a implementar la base de datos. Es decir generar en mySQL las distintas tablas, entradas y consultas necesarias para resolver el problema.

A continuación, mostraremos como se implementaron cada uno de los Store Procedures correspondientes a las consultas necesarias para cumplir con los requerimientos del punto 2 del enunciado:

- **Listado de inscritos en cada categoría.**
`SELECT nombre FROM Competidor comp INNER JOIN
(SELECT nroCertificadoITFCompetidor
FROM ParticipaIndividualmente p
INNER JOIN (select idCompetencia FROM (select idCompetencia, nombreModalidad FROM
Competencia c`

```
INNER JOIN Modalidad m ON c.idModalidad = m.idModalidad) t1 WHERE t1.nombreModalidad = 'mod1') t2
ON p.idCompetencia = t2.idCompetencia) t3
ON comp.nroCertificadoITF = t3.nroCertificadoITFCompetidor
```

■ **País que obtuvo mayor cantidad de medallas de oro, plata y bronce.**

```
SELECT t3.nombre FROM
(SELECT PA.nombre, COUNT(PA.nombre) AS cantMedallas from Pais PA
INNER JOIN (SELECT t1.nroCertificadoITF, E.idPais FROM Escuela E
INNER JOIN (SELECT C.nroCertificadoITF, C.idEscuela
FROM (select nroCertificadoITF from PodioPorCompetidor) P
INNER JOIN Competidor C ON P.nroCertificadoITF = C.nroCertificadoITF) t1
ON t1.idEscuela = E.idEscuela) t2
ON PA.idPais = t2.idPais GROUP BY PA.nombre) t3 ORDER BY t3.cantMedallas DESC
LIMIT 1
```

■ **Sabiendo que las medallas de oro van 3 puntos, las de plata 2 y las de bronce 1 punto, se quiere realizar un ranking de puntaje por país y otro por escuela.**

■ **Por país:**

```
CREATE PROCEDURE 'rankingPais'()
BEGIN
select P.nombre,
(count(case when PC.posicion = 1 then 1 else NULL end) + count(case when PE.posicion = 1 then 1 else NULL end)) *3 +
(count(case when PC.posicion = 2 then 1 else NULL end) + count(case when PE.posicion = 2 then 1 else NULL end)) *2 +
count(case when PC.posicion = 3 then 1 else NULL end) + count(case when PE.posicion = 3 then 1 else NULL end) as ranking from Pais P
left join Escuela E on P.idPais = E.idPais
left join Competidor C on C.idEscuela = E.idEscuela
left join Equipo Q on C.idEquipo = Q.idEquipo
left join PodioPorCompetidor PC on C.nroCertificadoITF = PC.nroCertificadoITF
left join podioPorEquipo PE on PE.idEquipo = Q.idEquipo

group by P.idPais
order by ranking desc
END
```

■ **Por escuela:**

```
CREATE PROCEDURE 'rankingEscuela'()
BEGIN
select E.nombre,
(count(case when PC.posicion = 1 then 1 else NULL end) + count(case when PE.posicion = 1 then 1 else NULL end)) *3 +
(count(case when PC.posicion = 2 then 1 else NULL end) + count(case when PE.posicion = 2 then 1 else NULL end)) *2 +
count(case when PC.posicion = 3 then 1 else NULL end) + count(case when PE.posicion = 3 then 1 else NULL end) as ranking
from Escuela E
left join Competidor C on C.idEscuela = E.idEscuela
left join Equipo Q on C.idEquipo = Q.idEquipo
left join PodioPorCompetidor PC on C.nroCertificadoITF = PC.nroCertificadoITF
left join podioPorEquipo PE on PE.idEquipo = Q.idEquipo

group by E.idEscuela
order by ranking desc
```


END

- **Dado un competidor obtener la lista de categorías donde haya participado y el resultado obtenido.**

```
CREATE PROCEDURE 'ParticipacionesPorCompetidor'(in certITF int)
BEGIN
select C.nroCertificadoITF, M.*, PC.posicion from Competidor C
join ParticipaIndividualmente PI on PI.nroCertificadoITFCompetidor = C.nroCertificadoITF
join Competencia X on PI.idCompetencia = X.idCompetencia
join Modalidad M on M.idModalidad = X.idModalidad
left join PodioPorCompetidor PC on PC.idCompetencia = X.idCompetencia

where C.nroCertificadoITF = certITF

END
```

- **Medallero por escuela.**

```
CREATE PROCEDURE 'medalleroPorEscuela'()
BEGIN
select E.nombre,
count(case when PC.posicion = 1 then 1 else NULL end) + count(case when PE.posicion =
1 then 1 else NULL end) as oro,
count(case when PC.posicion = 2 then 1 else NULL end) + count(case when PE.posicion =
2 then 1 else NULL end) as plata,
count(case when PC.posicion = 3 then 1 else NULL end) + count(case when PE.posicion =
3 then 1 else NULL end) as bronce
from Escuela E
join Competidor C on C.idEscuela = E.idEscuela
left join Equipo Q on C.idEquipo = Q.idEquipo
left join PodioPorCompetidor PC on C.nroCertificadoITF = PC.nroCertificadoITF
left join podioPorEquipo PE on PE.idEquipo = Q.idEquipo

group by E.idEscuela
order by oro desc, plata desc, bronce desc
END
```

- **Listado de árbitros por país.**

```
CREATE PROCEDURE 'arbitroPorPais'(in nombrePais varchar(45))
BEGIN
SELECT A.* FROM Arbitro A
JOIN Pais P on A.idPais = P.idPais
WHERE nombrePais = P.nombre;
END
```

- **La lista de todos los árbitros que actuaron como arbitro central en las modalidades de combate.**

```
CREATE PROCEDURE 'arbitrosCentrales'(in modalidad VARCHAR(45))
BEGIN
select A.* from Arbitro A
join Rol R on R.idRol = A.idRol
join Arbitraron AR on A.idArbitro = AR.idArbitro
join Competencia C on AR.idCompetencia = C.idCompetencia
join Modalidad M on M.idModalidad = C.idModalidad
where M.nombreModalidad = modalidad and R.descripcion = 'central'
END
```

- **La lista de equipos por país.**

```
CREATE PROCEDURE 'equiposPorPais'(in nombrePais varchar(45))
```

```
BEGIN
SELECT E.nombre FROM equipos E
JOIN Competidor C ON C.idEquipo = E.idEquipo
JOIN Escuela S ON C.idEscuela = S.idEscuela
WHERE nombrePais = P.nombre
END
```

5. Conclusión

Sobre este trabajo, queremos destacar algunas conclusiones a las que llegamos durante el transcurso del mismo.

En principio partimos de un DER con mas del doble de entidades que tiene el presentado en este trabajo. Si bien esto quizás simplificaba las consultas, también generaba un aumento exponencial en la cantidad de restricciones, ya para cada entidad que replicaba información debíamos constatar que dicha información replicada estuviera exactamente igual. Esta complejidad añadida no formaba parte del problema original que buscábamos resolver, por lo que optamos por reducir al mínimo la replica de información, a costa de complejizar las consultas a la base de datos. También notamos que esta nueva forma de almacenar la información sin replicamiento, nos permitía mantener y actualizar los datos de manera mas simple, ya que solo debíamos cambiar una sola posición.