



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Sistema de Inscripción Mundial de Irlanda 2017 (TP1)

Taekwon Do World Championship 2017

Bases de Datos  
Primer Cuatrimestre 2017

Integrante	LU	Correo electrónico
Lizana, Helen	118/08	hsle.22@gmail.com.ar
Mignanelli, Alejandro Rubén	609/11	minga_titere@hotmail.com
Sabarros, Ian	661/11	iansden@live.com
Sueiro, Diego	75/90	dsueiro@gmail.com



Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Asumimos</b>	<b>4</b>
<b>3. Desarrollo</b>	<b>5</b>
3.1. Entidad Modalidad . . . . .	5
3.2. Entidad Competencia . . . . .	5
3.3. Entidad Match . . . . .	6
3.4. Entidad Alumno . . . . .	6
<b>4. Modelo de Entidad Relación y Modelo Relacional</b>	<b>8</b>
4.1. Modelo de Entidad Relación . . . . .	8
4.2. Modelo Relacional . . . . .	9
<b>5. Entorno y Código</b>	<b>12</b>
<b>6. Conclusiones</b>	<b>12</b>

## 1. Introducción

En el presente trabajo se tratará de proveer un sistema de inscripciones para el Taekwon Do World Championship 2017, teniendo como finalidad, poder aprender a realizar una base de datos en base a un problema del mundo real. Para esto, primero se crearán un Modelo de Entidad Relación y un Modelo Relacional basadas en la especificación que se encuentra en el enunciado, los cuales usaremos para la creación de una base de datos. Una vez creada, se implementarán las siguientes funcionalidades:

- El listado de inscriptos en cada categoría para el armado de llaves
- El país que obtuvo mayor cantidad de medallas de oro, plata y bronce.
- Sabiendo que las medallas de oro suman 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.
- Dado un competidor, la lista de categorías donde haya participado y el resultado obtenido.
- El medallero por escuela.
- El listado de los árbitros por país.
- La lista de todos los árbitros que actuaron como árbitro central en las modalidades de combate
- La lista de equipos por país.

Para la resolución del problema, hemos decidido utilizar mySql.

## 2. Asumimos

En esta sección detallaremos asumisiones que no estan dichas en el enunciado:

- Los coachs son alumnos y que son inscriptos por el Maestro en caso que decidan competir.
- Sólo un coach acompaña al competidor o equipo por competencia.
- El número de placa de arbitro/instructor son únicas.
- La configuración de la modalidad lo realizara correctamente el desarrollo. Es decir, si el desarrollador agrega la modalidad Combate.<sup>en</sup>tonce configurará solo los campos de edad y género.
- La persona que incribió al alumno como competidor, fue el maestro de la escuela a la que pertenece.
- Cuando termina una competencia, el usuario del sistema ingresa la información del puesto que obtuvo el competidor/equipo en esa competencia.

### 3. Desarrollo

Cuando empezamos a diseñar el DER nos encontramos con varias dudas sobre algunas entidades y alguna de las relaciones entre ellas. Luego de hacer las consultas con nuestro Tutor tuvimos que realizar algunos cambios. A continuación describiremos los cambios más importantes.

#### 3.1. Entidad Modalidad

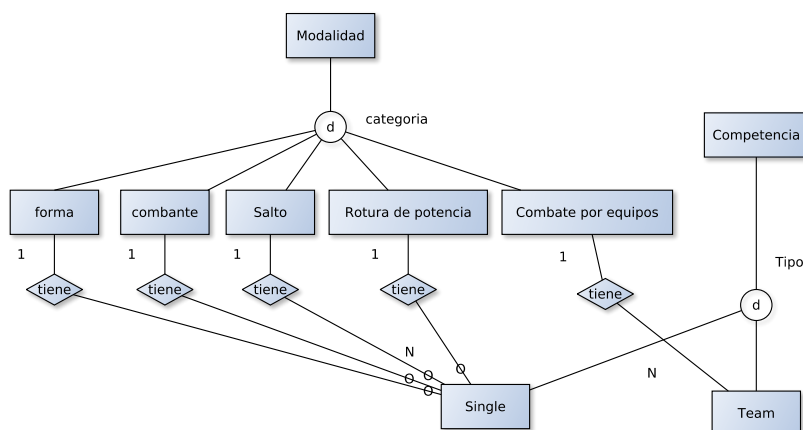
Entre los cambios más importantes, pueden destacarse aquellos relacionados con la entidad **Modalidad**, que por enunciado, solo puede ser de 5 valores posibles y dependiendo del tipo de modo, esta posee distintos atributos. Lo primero que se nos ocurrió fue tenerla como entidad padre y luego como entidades hijas (disjuntas) a cada tipo de modalidad(forma, salto, etc). Pero eso luego nos traía complicaciones para definir las relaciones entre Competencia y Modalidad, dado que había que poner muchas restricciones para que el modelo sea consistente con la especificación. Por ejemplo, si tuviésemos que registrar una competencia con una modalidad del tipo "Forma", una de las restricciones hubiese sido: una competencia debe estar relacionada con uno de los tipos de modalidad. También que solo está relacionado con el Tipo "Forma" y no puede estar relacionado con otro tipo. Y también había que aclarar que una competencia podría no estar en la modalidad "Forma".

Luego al consultarlo con el Tutor decidimos que no tenía tanta relevancia separar esa información en tantas entidades, si no que podríamos almacenar toda esa información en una sola entidad. Agregando como restricciones que si la entidad era de determinado modo, entonces tenía que tener valores no nulos los atributos correspondientes a ese modo.

#### 3.2. Entidad Competencia

Al principio pensamos esta entidad como una jerarquía, Competencia como entidad padre y Single y Team como entidades hijas. La entidad Single sería para las competencias entre competidores, y la entidad Team sería para las competencias entre equipos. Necesitábamos hacer esto para poder diferenciar que competencias eran de modalidad "combate por equipos" dado que tenía atributos característicos. Con esto teníamos cubierto esa diferencia pero al observar más en detalle la existencia de las entidad hijas, vimos que estas no tenían atributos característicos con respecto a su identidad padre, y que en definitiva al pasarlas a la base de datos solo guardarían los campos claves de los competidores/equipos y con un Id de competencia relacionado.

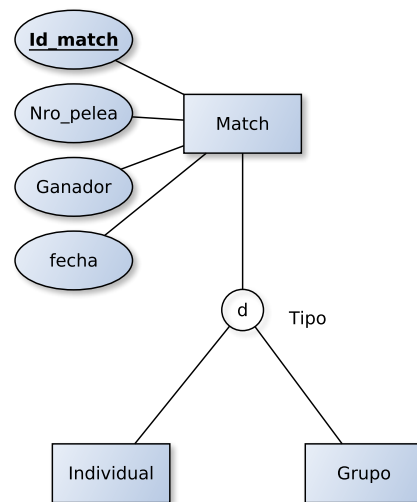
Para simplificar el modelo lo que hicimos fue eliminar estas entidades hijas y pasamos la relacion "participa" de Competidor a Competencia y lo mismo para equipo. También agregamos las restricciones de que una competencia esta relacionada a competidores o a equipos, no puede pasar ambas a la vez. Imagen de la version original de las entidades Modalidad y Competencia:



### 3.3. Entidad Match

Dado que el enunciado pedía obtener los ganadores de cada competencia: 1er, 2do y 3er lugar, pensamos en guardar la información de los encuentros y obtener de ellos el resultado de los ganadores. La entidad Match guardaría el id del competidor ganador y la fecha del encuentro. Para esto asumíamos que los dos últimos encuentros serían las del 3er puesto y la final. Aca nos volvimos a encontrar con el problema de diferenciar los encuentros entre competidores y los de equipos, por lo cual teníamos que crear una jerarquía para obtener MatchIndividual y MatchGrupo. Y al igual que en el caso de la entidad Competencia, estas entidades hijas carecían de atributos y sólo guardaban campos claves de otras entidades.

Luego de mostrárselo a nuestro benemérito Tutor, comprendimos que no había que guardar la información de los encuentros, y que sólo bastaba con que la competencia almacenara los Id de los competidores/equipos ganadores. Por eso decidimos eliminar la entidad Match, y guardamos en la relación ternaria Compite el número del puesto obtenido en la competencia.



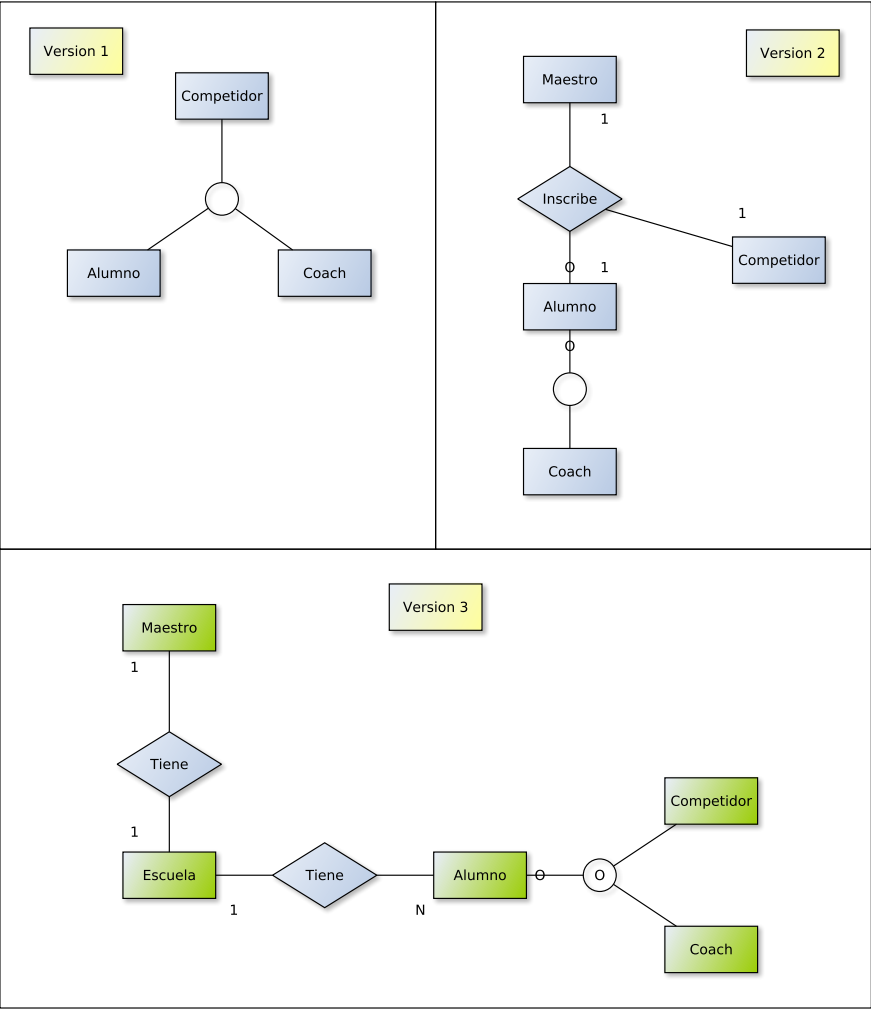
### 3.4. Entidad Alumno

Esta entidad sufrió varias modificaciones a lo largo del tp, mayormente causadas por sus relaciones con la entidad maestro, competidor y coach.

En un principio, pensábamos que un competidor podía ser alumno o ser coach, y de ahí deducíamos que el coach podía ser competidor, dando lugar a la versión 1. Como toda primer idea, producía una gran cantidad de errores, hasta el punto de no respetar el enunciado, puesto que no permitía un coach que no fuera competidor.

Luego pensamos en modelar el hecho de que el maestro inscribe al alumno. De ahí nació la terna Maestra, competidor, alumno. Para incluir al coach, lo hicimos sub entidad de alumno, permitiéndole participar como competidor, pero como no todo alumno es competidor, no obligándolo a ello. Pero una revolución implementativa desechó esta idea, que llamaremos versión 2, puesto que complicaba las queries y producía una tabla innecesaria.

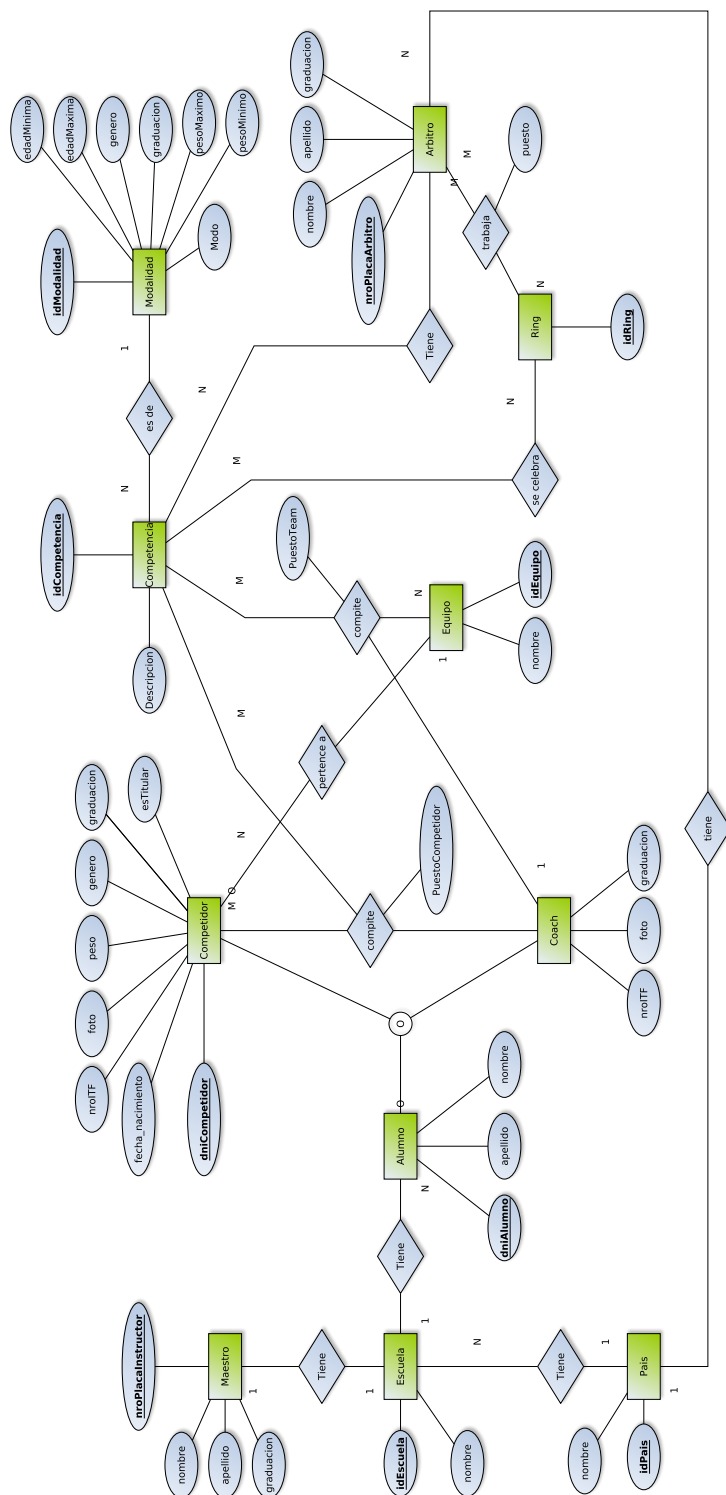
Esto dio paso a la versión 3, donde si bien no se ve a simple vista en el der que el maestro inscribe al alumno, se puede asumir semanticamente, y se puede obtener a través de escuela, y se considera a alumno como la entidad de las cuales subdividen coach y competidor, permitiendo todas las ventajas de la versión 2 y ahorrando una tabla.



#### 4. Modelo de Entidad Relación y Modelo Relacional

A continuación, se mostrarán el Modelo de Entidad Relación y el Modelo Relacional correspondientes a nuestra solución del problema.

### 4.1. Modelo de Entidad Relación





## 4.2. Modelo Relacional

**Escuela**(idEscuela, nombre, idPais, nroPlacaInstructor)

PK = CK = { idEscuela }

FK = { idPais, nroPlacaInstructor }

-----

**Pais**(idPais, nombre)

PK = CK = { idPais }

-----

**Maestro**(nroPlacaInstructor, nombre, apellido, graduación)

PK = CK = { nroPlacaInstructor }

-----

**Alumno**(dniAlumno, nombre, apellido, idEscuela)

PK = CK = { dniAlumno }

FK = { idEscuela }

*Restricciones:*

- un Alumno puede ser coach y competidor a la vez.

-----

**Coach**(dniAlumno, nroITF, foto, graduación)

PK = CK = FK = { dniAlumno }

*Restricciones:*

- Un alumno puede **no** estar en Coach
- habrá al menos un coach por cada 5 competidores

-----

**Competidor**(dniAlumno, nroITF, fechaNacimiento, género, graduación, peso, foto, esTitular, idEquipo)

PK = CK = { dniCompetidor }

FK = { idEquipo }

*Restricciones:*

- Un alumno puede **no** estar en Competidor.
- El Competidor puede **no** estar en Equipo.
- Si el Competidor está en Equipo, entonces si es titular, el campo esTitular tendrá un True, y si es suplente tendrá un false. En caso de no tener equipo, ese campo contendrá null.

-----

**Equipo**(idEquipo, nombre)

PK = CK = { idEquipo }

*Restricciones:*

- Los equipos están conformados por 5 titulares y 3 suplentes, y todos deben ser de la misma escuela.
- 

**Competencia**(idCompetencia, descripcion, idModalidad)

PK = CK = { idCompetencia }

FK = { idModalidad }

-----

**compiteEnCompetenciaInd**(dniCompetidor, idCompetencia, dniCoach, puestoCompetidor)

PK = { (dniCompetidor, idCompetencia) }

CK = { (dniCompetidor, idCompetencia), (dniCompetidor, dniCoach), (idCompetencia, dniCoach) }

FK = { dniCompetidor, idCompetencia, dniCoach }

*Restricciones:*

- Los competidores que se registren en una competencia, deben cumplir con las exigencias de la misma, es decir, deben respetar la edad, el sexo, la graduación, o cualquier exigencia que la modalidad de la competencia exija.
  - Un coach que es competidor no podrá ser coach de si mismo.
- 

**compiteEnCompetenciaTeam**(idEquipo, idCompetencia, dniCoach, puestoTeam)

PK = { (idEquipo, idCompetencia) }

CK = { (idEquipo, idCompetencia), (idEquipo, idCompetencia), (idCompetencia, dniCoach) }

FK = { idEquipo, idCompetencia, dniCoach }

*Restricciones:*

- Los competidores de los equipos que se registren en una competencia, deben cumplir con las exigencias de la misma, es decir, deben respetar la edad, el sexo, la graduación, o cualquier exigencia que la modalidad de la competencia exija.
  - Un coach que es competidor no podrá ser coach del equipo al que pertenece.
- 

**Modalidad**(idModalidad, edadMínima, edadMáxima, género, graduación, pesoMínimo, pesoMáximo, modo)

PK = CK = { idModalidad }

*Restricciones:*

- Los modos determinan que atributos deben estar en null según lo pedido en el enunciado.

-----

**Ring**(idRing)

PK = CK = { idRing }

-----

**Arbitro**(nroPlacaArbitro, nombre, apellido, graduación, idPais)

PK = CK = { nroPlacaArbitro }

FK = { idPais }

-----

**arbitrosEnCompetencias**(nroPlacaArbitro, idCompetencia)

PK = CK = FK = { (nroPlacaArbitro, idCompetencia) }

*Restricciones:*

- La graduación de un arbitro debe ser siempre MAYOR a la graduación de la modalidad de la competencia.
- 

**ringsDeCompetencias**(idRing, idCompetencia)

PK = CK = FK = { (idRing, idCompetencia) }

*Restricciones:*

- Los rings de cada competencia deben poseer arbitros que tengan la capacidad de dirigirlas.
- 

**puestoArbitroEnRing**(nroPlacaArbitro, idRing, puesto)

PK = CK = FK = { (nroPlacaArbitro, idRing) }

*Restricciones:*

- En un Ring siempre habrá un presidente de mesa, un arbitro central, varios jueces y al menos tres suplentes.
-

## 5. Entorno y Código

La base ha sido desarrollada en mySql y se encuentra bajo el nombre taekwondo.sql en la carpeta SQL que se adjunta junto con el informe. Las queries están en la misma carpeta numeradas desde Ejercicio1.sql hasta Ejercicio8.sql .

## 6. Conclusiones

A lo largo del trabajo, nos encontramos con muchas cuestiones de decisión, en la clásica lucha claridad VS mantenimiento. Originalmente teníamos aproximadamente 24 entidades, que fueron recortadas a un total de 11, puesto que mantener la consistencia de tantas tablas resultaba mucho más costoso en nuestra opinión, que la pérdida de claridad al tener menos entidades que almacenaran la misma información. Por otro lado, pudimos apreciar de primera mano como el pasaje del DER al MR y del MR a la base no es tan directo como la teoría podría sugerir, puesto que al realizar la base, nos hemos encontrado que cierta información resultaban muy complicadas de ser accedidas mediante queries por la estructura de la base, lo cual termino provocando en cambios en esta, que luego escalaron a cambios en el MR y obviamente en el DER. Finalizando, también experimentamos la complejidad de armar la estructura de una base en un conjunto de 4 personas, puesto que la diferencia de opiniones sobre un mismo enunciado, impacto gravemente en la definición de entidades, lo cual repercute eventualmente en la generación de la base.

En conclusión, consideramos que se alcanzaron los objetivos propuestos en este trabajo, puesto que pudimos toparnos y resolver de una manera que consideramos correcta, un ejemplo de la vida real sobre bases de datos.