***DESCRIPCION***

Quiero almacenar información sobre equipos ciclistas, los corredores que tiene en plantilla, duración de contratos, las alineaciones que presentan en cada carrera, puestos en los que quedan en la general (y en cada etapa si fuera una carrera por etapas) y los puntos UCI que se otorgan.

* Cada equipo ciclista tiene una serie de corredores en plantilla. Cada equipo tiene un nombre que es único y además es de una nacionalidad concreta y es necesario guardar la abreviación del nombre puesto que este se utiliza mucho en las retransmisiones de televisión. También seria conveniente guardar su marca de bici pues esta suele ser un patrocinador del equipo e interviene en contratos con corredores, además de los puntos UCI que ha conseguido a lo largo del año
* De los ciclistas queremos almacenar tanto su nombre como su apellido, junto con el país donde nacieron, su fecha de nacimiento y la altura y el peso pues esto puede ayudar a diferenciar que tipo de corredor es (escalador, croner, etc.). También nos interesa el nº de carreras que han corrido a lo largo del año.
* Los equipos pueden contratar a varios corredores durante un mismo periodo de tiempo (por ejemplo, pueden tener 3 corredores con contrato hasta 2025 y 4 con contrato hasta 2027) por lo que nos interesa conocer esas relaciones que tienen los equipos y los corredores almacenando también el periodo durante el cual están vinculados.
* A parte, tenemos las carreras, de las cuales queremos almacenar su nombre, el país por donde transcurren y la escala UCI que tienen, puesto que esto determina la cantidad de puntos UCI que se entregan en ella. Hay dos tipos de carrera, las carreras de un día, también llamadas clásicas, y las carreras por etapas. De las carreras de un día nos interesa conocer la distancia total que se recorre (normalmente cuanta más distancia se recorre más dura es la carrera), y de las carreras por etapas queremos almacenar el numero de etapas que tienen junto con la distancia que se recorre entre todas las etapas.
* Cada carrera pasa por una serie de municipios. Estos municipios tienen unos códigos postales que pueden ser varios según el tamaño de la ciudad. Los municipios pertenecen a una provincia o distrito (se llama de una manera u otra dependiendo del país) y esas provincias pertenecen lógicamente a un solo país.
* Las carreras por etapas tienen varias etapas. El numero de una etapa no nos dice gran cosa por sí misma, pero si el numero de etapa junto con la carrera en la que esta. De cada etapa queremos almacenar también la distancia que recorre junto con la altimetría (metros de desnivel positivo (pues nos puede ayudar a diferenciar si se trata de una etapa contra el crono, llana, montaña, etc.).
* Para cada carrera, los equipos presentan una serie de corredores. Esas presentaciones son únicas. Esa presentación (cada corredor tiene una en cada carrera), es la que corre las etapas o la carrera de un día. Nos interesa conocer sus dorsales, puesto que también se muestran en las retransmisiones por televisión, junto con la posición en la que han acabado en la general y los puntos UCI que han conseguido al terminar en dicha posición.
* Dentro de cada carrera, varias presentaciones comparten equipo, y por lo tanto nos interesa conocer la estructura de jerarquías dentro de dicha carrera (puede variar según la especialidad de cada carrera), conociendo cuales son los lideres de cada corredor y los gregarios de cada corredor, si los tuvieran (en equipos de bajo nivel no suele haberlas).
* Las presentaciones que corren las carreras por etapas corren a su vez las etapas. Todas las presentaciones no tienen por qué correr etapas (pueden ser presentaciones de clásicas). Queremos guardar la posición en la que queda cada presentación en cada etapa junto con los puntos UCI que obtienen por queda en esos puestos, para poder calcular cuantos puntos UCI obtiene cada equipo en total en cada carrera.

***ESQUEMA CONCEPTUAL***

Para resumir en un esquema toda la información arriba explicada he utilizado el software DIA explicado en clase, construyendo el archivo Diagrama\_ProyectoBD.dia.

En primer lugar, almaceno los datos de los equipos y los corredores con los datos que me piden. Como el nombre del equipo me dicen que es único lo utilizo como clave en dicha entidad, y el nombre, apellido y nacionalidad lo utilizo como clave para distinguir cada corredor. Considero una relación ternaria N:M:P junto con otra entidad periodo (en la que considero como clave el año de inicio de vinculación y el año de fin de vinculación) puesto que me dicen que un equipo puede tener durante un periodo varios corredores, y los corredores pueden pertenecer durante distintos periodos a distintos equipos.

En segundo lugar, considero la entidad Carrera junto con los atributos que me han comentado, utilizando como clave el nombre de la carrera que es único. Como Me dicen que hay dos tipos de carrera (de un día y por etapas) empleo una jerarquía para diferenciarlas, y dentro de cada una de ellas pongo los atributos que me piden.

Mapa de colores

Descripción generada automáticamente con confianza media

Diagrama

Descripción generada automáticamente

Posteriormente considero la entidad presentación, en la que almaceno la carrera que corre cada corredor y con que equipo. De esta forma, como los corredores pueden estar en varias presentaciones (no tienen porque estar en algunas, puesto que pueden estar lesionados por ejemplo) considerando una relación 1: N está presente en, de igual forma los equipos presentan corredores en las carreras, y consideramos una relación 1: N con la diferencia de que aquí los equipos siempre deben presentar al menos 1 equipo en cada carrera. Así mismo las presentaciones corre 1 sola carrera (para cada carrera que corra un corredor considerare una presentación distinta), y la carrera la corren varias presentaciones resultando la relación 1: N corre.

También considero la relación pasa por entre carrera y municipios. Entiendo que es una relación N: M puesto que una carrera pasa por muchos municipios y por un municipio pueden pasar varias carreras también.

Interfaz de usuario gráfica, Mapa

Descripción generada automáticamente con confianza media

Ahora considero la relación tiene entre las entidades carrera por etapas y etapa, siendo esta última entidad débil en identificación por la información de que el numero de etapa solo tiene sentido como elemento diferenciador dentro de cada carrera.

A continuación, considero la relación N: M corre entre las entidades presentación y etapa. En la parte de etapa pongo un (0, n) puesto que la presentación no tiene porque correr una etapa, ya que esa presentación puede ser de una carrera de un día.

Por último, considero una reflexiva, es, en la entidad presentación en la que una presentación es líder de (0, n) gregarios, para así considerar los equipos en los que los corredores van por libre.

Diagrama

Descripción generada automáticamente

Mapa

Descripción generada automáticamenteComo ultimas entidades considero las referenciadas a los municipios, código postal, provincia y país. En las que considero la entidad Cod.postal puesto que me han dicho que un municipio puede tener varios códigos postales en función del tamaño. Podríamos haber omitido la tabla país poniéndolo como un atributo de la provincia, pero he preferido no hacerlo y tratarlo por partes.

***ESQUEMA LÓGICO RELACIONAL***

En primera instancia realizo el diseño de las entidades de tipo 1 (las que existen por si mismas y no tengo que añadirle datos de otras tablas).

Periodo (Año inicio vinculación, Año fin vinculación)

Corredor (Nombre, Apellido Completo, Nacionalidad, Altura, Peso, Fecha nacimiento, nº carreras corridas)

Equipo (Nombre, Abreviación, Marca de bici, Nacionalidad, nº puntos UCI totales)

Carrera (Nombre, Escala UCI, País)

País (Nombre)

Ahora hago el de las entidades de tipo 2, que relacionan entidades de tipo 1 entre sí, como por ejemplo la tabla Esta\_vinculado que relaciona equipo, corredor y periodo, la jerarquía anteriormente explicada de la entidad carrera.

Esta\_vinculado (Nombre\_equipo, Inicio\_periodo, Fin\_periodo, Nombre\_corredor, Apellido\_corredor, Nacionalidad\_corredor)

Fk1: Nombre\_equipo -> Equipo (nombre)

Fk2: Inicio\_periodo -> Etapa (Año inicio vinculación)

Fk3: Fin\_periodo -> Etapa (Año fin vinculación)

Fk4: Nombre\_corredor -> Corredor (nombre)

Fk5: Apellido\_corredor -> Corredor (apellido)

Fk6: Nacionalidad\_corredor -> Corredor (Nacionalidad)

Presentación (Id, Nombre\_Corredor, Apellido\_corredor, Nacionalidad\_corredor, Nombre\_equipo, Nombre\_carrera, posición, nº puntos UCI posición)

Fk1: Nombre\_equipo -> Equipo (nombre)

Fk2: Nombre\_Corredor -> Corredor (nombre)

Fk3: Apellido\_corredor -> Corredor (apellido)

Fk4: Nacionalidad\_corredor -> Corredor (nacionalidad)

Fk5: Nombre\_carrera -> Carrera (nombre

Carrera\_un\_dia (Nombre\_carrera, Distancia recorrida)

Fk1: Nombre\_carrera -> Carrera (nombre)

Carrera\_por\_etapas (Nombre\_carrera, nº etapas, Distancia total)

Fk2: Nombre\_carrera -> Carrera (nombre)

Provincia/Distrito (Nombre, Nombre\_país)

Fk1: Nombre\_país -> País (nombre)

A continuación, hago las entidades de tipo 3.

Etapa (Nombre\_carrera, nº, Distancia, Lugar salida, Lugar llegada, Altimetría)

Fk1: Nombre\_carrera -> Carrera\_por\_etapas (Nombre\_carrera)

Municipio (Nombre, Nombre\_provincia)

Fk1: Nombre\_provincia -> Provincia/Distrito (nombre)

Es (Id\_lider, Id\_gregario)

Fk1: Id\_lider -> Presentación (Id)

Fk2: Id\_gregario -> Presentación (Id)

Como ultimo paso realizo el diseño de las entidades tipo 4 que están relacionadas con algunas de tipo 3.

Termina (ID\_presentacion, nº\_etapa, Nombre\_carrera, posición, nº puntos UCI por posición)

Fk1: Nombre\_carrera -> Etapa (Nombre\_carrera)

Fk2: nº\_etapa -> Etapa (nº)

Fk3: ID\_presentacion -> Presentación (Id)

Pasa\_por (Nombre\_Municipio, Nombre\_carrera)

Fk1: Nombre\_Municipio -> Municipio (nombre)

Fk2: Nombre\_carrera -> Carrera (nombre)

Cod.postal (Código, Nombre\_Municipio)

Fk1: Nombre\_Municipio -> Municipio (Nombre)

Para el diseño en MySQL Workbench cambiaré la clave de la entidad corredor por un Id autogenerado, que nos hará mas sencillo relacionar las tablas con la entidad corredor, así como identificar con un solo campo a los corredores y no con los 3 anteriores.

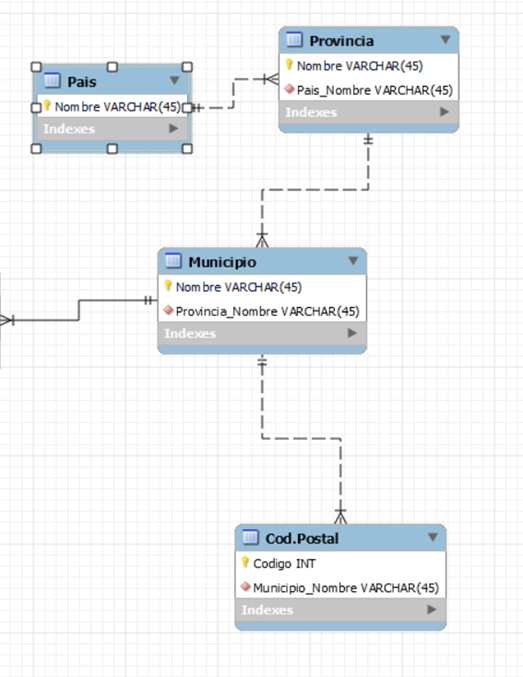
***DISEÑO EN MySQL Workbench***

Para realizar el diseño con esta herramienta he utilizado la traslación directa del diseño conceptual, y no a través del diseño lógico anterior. He creado el archivo Esquema\_ProyectoBD.mwb de la carpeta Diseño\_Proyecto

Como hicimos en la parte del diseño conceptual en primer lugar tratamos las entidades Equipo, Corredor y Periodo, junto con la relación N: M: P Esta\_vinculado que tienen. Nótese que aquí ya he modificado la clave de la entidad corredor, sustituyéndola por un Id como había comentado en el apartado anterior, lo que hace que la tabla Esta\_vinculado sea mucho mas sencilla de tratar, sin tantos atributos.

Diagrama

Descripción generada automáticamente



En segundo lugar, trato las tablas alrededor de Municipio, teniendo en cuenta que como todas las relaciones son del tipo 1: N tengo que llevarme las foreign keys a la otra entidad.

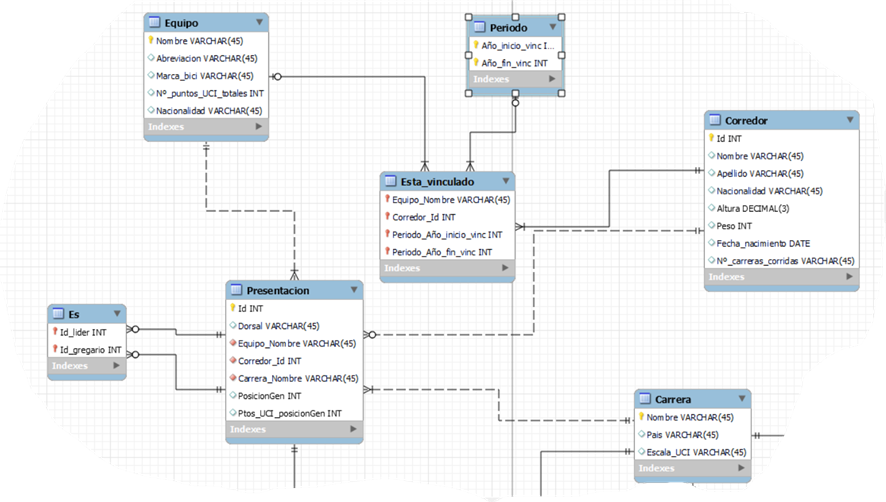
A continuación, trato la jerarquía existente en la entidad Carrera dividiéndola en esas dos entidades Carrera\_de\_un\_dia y Carrera\_por\_etapas. Nótese que no pongo obligatoriamente que una carrera sea de un día o por etapas y por eso se encuentra el circulo justo en la parte superior de las entidades de la jerarquía.

Diagrama

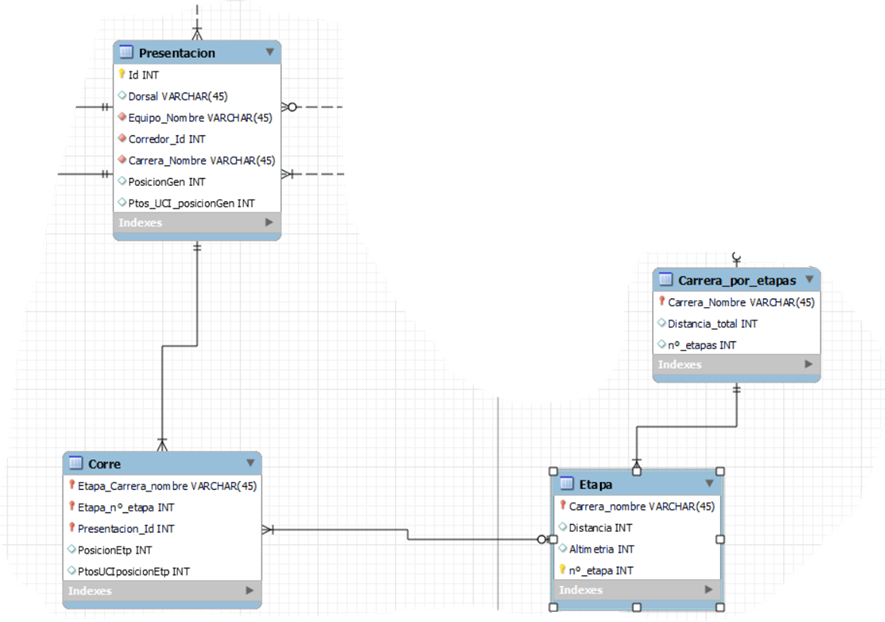
Descripción generada automáticamente

Ahora estudio la tabla Presentación. Al estar relacionada de forma 1: N con las tablas Equipo, Corredor, Carrera, tengo que llevarme las foreign keys a esa tabla. Nos podemos dar cuenta de lo conveniente que vuelve a ser poner una clave Id en la entidad corredor, que nos hace tratar de una manera mas sencilla la tabla Presentación.

Además, consideramos la entidad reflexiva es en la tabla presentación, en la que emparejamos (si tuviesen, de ahí los círculos vacíos en las flechas de la relación) los lideres junto con los gregarios en cada carrera.



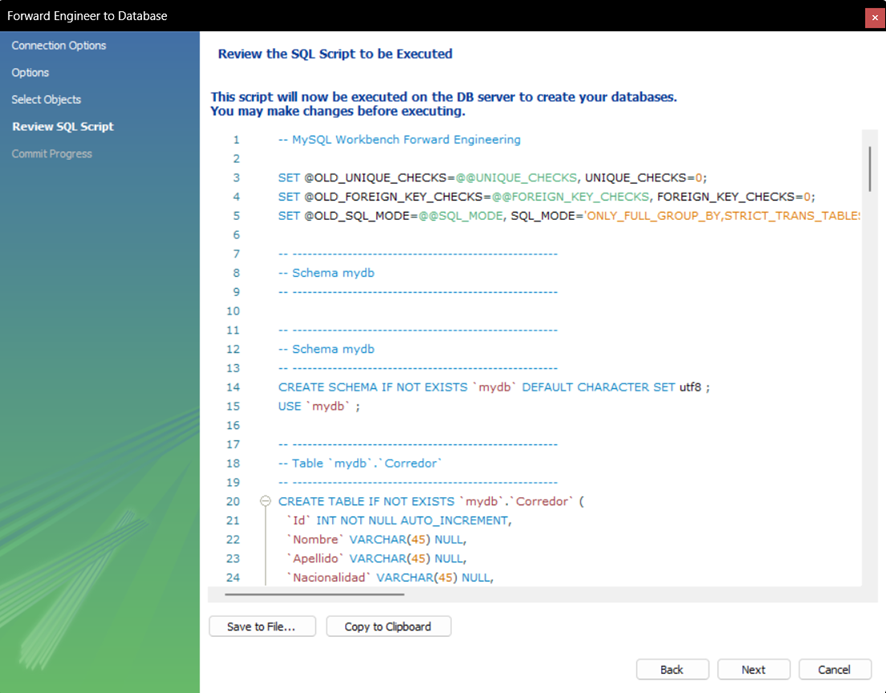
Por último, tenemos la tabla etapa, que al ser débil en existencia la clave esta formada por el nº de la etapa y la carrera en la que esta la etapa, siendo esta también foreign key. Posteriormente tenemos la relación N: M entre Presentación y Etapa, dándonos cuenta de que como no todas las presentaciones corren las etapas, tenemos ese circulo en la flecha que llega a la tabla Etapa.



***IMPLEMENTACION BASE DE DATOS RELACIONAL***

* **MySQL**

Para la creación de la base de datos en MySQL he utilizado el mismo modelo de MySQL Workbench en el apartado Database, pinchando en Forward Engineer, aceptando en todos los apartados que me salen, salvo en la ultima opción, en la que obtengo el código de creación de la base de datos.

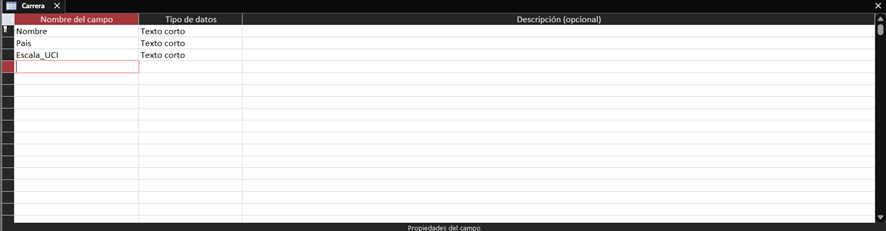


Lo que he hecho es copiar el código sql en el fichero BaseDatos\_ProyectoBD.sql que se encuentra en la carpeta Parte\_SQL donde he tenido que modificar una cosa, y es que siempre me salen en todas las foreign keys la palabra VISIBLE y pinchando en control + h y reemplazando todas las palabras VISIBLE por nada, ya obtengo la base de datos final que voy a utilizar y cargar a través de phpmyadmin.

Dentro del servidor de mi ordenador, pulso en el apartado importar y elijo el archivo que hemos construido antes dando para después pulsar en importar, creando así mi base de datos.

* **Microsoft Access**

Para la creación de las tablas en Access he modificado el método de creación, puesto que esta vez he seguido paso a paso el diseño lógico expuesto antes. De esta forma primero he creado las tablas tipo 1 como Corredor, Equipo, etc. Poniendo los campos correspondientes.



A continuación, he seguido con las tablas de tipo 2, es decir, aquellas que dependen en alguna foreign key de una tabla de tipo 1. En esta fase, he tenido que modificar un poco la creación de la tabla Esta\_vinculado ya que Access no acepta mas de una clave para una misma tabla, por lo que he decidido crear de clave un Id autogenerado que identifica cada vinculación.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para poner las foreign keys tengo que pulsar en asistente de búsqueda en Tipo de datos y eligiendo el campo de la tabla donde esta la foreign key correspondiente.

De la misma forma creo las tablas de las entidades tipo 3, que están relacionadas con una tabla de las anteriores al menos. Entre ellas se encuentra la tabla Etapa, que al ser débil en identificación tiene 2 claves, el nº de la etapa y el nombre de la carrera y por lo tanto he decidido crear un Id autogenerado para identificarlas poniéndola como clave.

Tabla

Descripción generada automáticamente

Por último, creo las tablas correspondientes a las entidades de tipo 4 que son las que más foreign keys suelen tener. En este apartado tengo que realizar la misma modificación que he explicado en los dos pasos anteriores a la tabla corre.



A continuación, una vez creadas todas las tablas siguiendo el sistema explicado, toca poblar de algunos datos dichas tablas. Como por ejemplo la tabla corredor

Tabla

Descripción generada automáticamente

O la tabla presentación

Tabla

Descripción generada automáticamente

Antes de crear los formularios podemos echar un vistazo al apartado de relaciones para ver si hemos construido de manera correcta, enlazando las tablas de la manera que se requería.

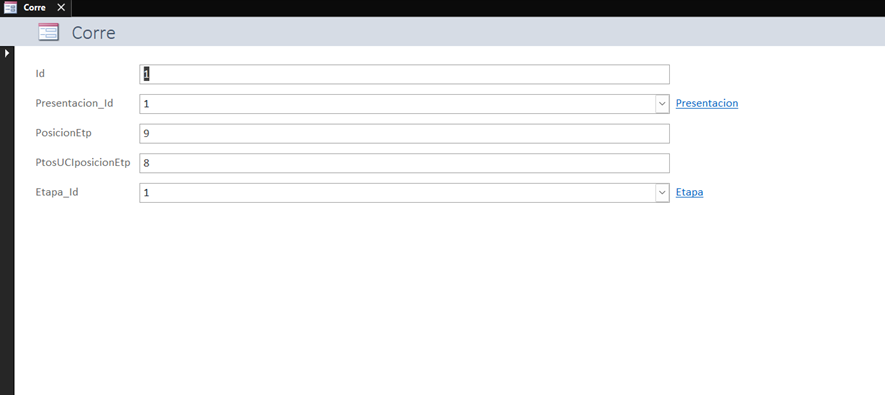
Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamentePara finalizar, nuestro último paso será la creación de formularios. Para ello dentro de cada tabla creada, en la vista de datos (no de diseño de la tabla) seleccionamos la opción crear formulario, una vez creado el formulario puedo adjuntar unos hipervínculos a otros formularios para que sea más fácil consultar información, por ejemplo, cuando tenemos el id de un corredor para saber de qué corredor se trata.

O para consultar las etapas a alas que nos referimos en la tabla corre.



De todas formas, no he expuesto todos los formularios ni todas las tablas con los datos, que lo puede revisar en el fichero Access\_ProyectoBD.accdb en la carpeta Parte\_Access.

***SQL***

* Para la creación de la base de datos he utilizado el propio documento de MySQL Workbench como expuse anteriormente para obtener el código DDL. Este código se encuentra en el documento BaseDatos\_ProyectoBD.sql que se encuentra en la carpeta Parte\_SQL.
* Para la inserción de los datos en la base de datos he utilizado los datos que he ido almacenando en el archivo Excel Datos\_ProyectoBD.xlsx que se encuentra en la carpeta Diseño\_Proyecto. En ese documento se encuentran todas las frases del tipo insert into…, que son las que he copiado y pegado en el fichero InsercionDatos\_ProyectoBD.sql que se encuentra en la carpeta Parte\_SQL.
* Para las consultas he creado el fichero Consultas\_ProyectoBD.sql que se encuentra en la carpeta Parte\_SQL. En ese fichero se encuentran las consultas de información acerca de la base de datos, junto con algunas acciones de actualización que pudiéramos hacer en un futuro como la inserción de más corredores y presentaciones.

***BD NO RELACIONAL***

* **Diseño**

El primer archivo que he creado ha sido Municipios.json, en el que he incorporado los nombres de los municipios, el país, provincia, códigos postales y carreras que pasan por dicho municipio. Lo he hecho de esta forma porque lo que nos interesa en especial es el municipio en concreto, con su ubicación (provincia, país), a través de que código o códigos postales podemos conocerlo y las carreras que pasan por él. Así no he creado un fichero para las provincias, puesto que estas no me aportan tanta información como los municipios.

El siguiente fichero que he creado ha sido Carreras.json, en el que hemos intentado agrupar los distintos atributos que tienen las carreras. Aquí la complicación ha sido que tenemos la jerarquía distinguiendo entre carreras de un día y las carreras por etapas. En este apartado la diferencia de información expuesta en el documento entre las carreras de un día y las carreras por etapas es la información del nº de etapas de las que consta la carrera por etapas, que esa información (incluido el campo, que ni siquiera existira) no aparece en las carreras de un día. También, para cada carrera he expuesto los municipios por los que pasa haciendo que la tabla Pasa\_por que teníamos en la base de datos relacional la tengamos ahora embebida en los dos ficheros.

Posteriormente me he dedicado a implementar la relación ternaria entre Periodo, Corredor y Equipo. Los he implementado en los ficheros Equipos.json, Corredores.json y Periodos.json. Para cada uno de esos ficheros, he incluido todos los atributos que les corresponden. A parte, en cada uno de los ficheros he puesto otro apartado para cada documento llamándolo vinculación. Así en el fichero Equipos.json nos encontramos los corredores junto con los años durante los cuales esta vinculado a dicho equipo, en el fichero Periodos.json nos encontramos que, para cada periodo de vinculación, el campo vinculación recoge pares equipo, corredor que están vinculados durante ese periodo, y en el fichero Corredores.json, para cada corredor en el apartado vinculaciones aparece los equipos junto con el periodo durante el cual han estado vinculados.

Es importante comentar que al llevarme los corredores a los distintos ficheros no he utilizado el Id como en la parte SQL, puesto que esto no representa nada en realidad, y no vamos a realizar joins entre unos ficheros y otros y he optado por poner el nombre completo del corredor que normalmente no se repite en ninguna carrera.

El siguiente fichero es Presentacion.json en el que incluyo todos los campos que aparecían en la parte SQL a excepción de que en vez de incluir el Id de cada corredor incluimos el nombre como hemos explicado antes. Una vez creada esa parte debo incluir la relación reflexiva Es. Para ello he creado dos campos que son arrays Lideres y Gregarios, en los que vamos incluyendo el nombre de los lideres y/o gregarios que tenga cada corredor. Posteriormente incluyo la información de la tabla Corre (llamada así en SQL), creando el campo Etapas, solo en aquellas presentaciones que corran carreras por etapas (en presentaciones que corran carreras de un día no creamos ningún campo). En dicho campo ponemos las etapas, posición y puntos UCI de la posición que consigue la presentación.

Por último, tenemos el fichero Etapas.json en el que guardamos la información de etapas en una carrera concreta junto con la distancia y altimetría de dicha etapa. Además, como tenemos la relación N: M Corre en el modelo relacional, he optado por embeber esa información también en este documento. Para ello he creado el campo Corredores, en el que coloco información del corredor que corre la etapa (no pongo el Id de la presentación por las mismas razones que vengo comentando de identificar mejor el corredor sin necesidad de joins) junto con la posición en la que acaba la etapa y los puntos UCI que obtiene por la posición.

* **Operaciones de consulta sobre la base de datos no relacional**

Para realizar las consultas he creado el fichero Consultas\_ProyectoBD.js que se encuentra en la carpeta Parte\_MongoDB. Ahí se incluyen una serie de consultas acerca de las bases de datos que hemos creado, además de algunas acciones de actualización que pudieran ser útiles en un futuro.

* **Generación de las bases de datos**

Para crear los ficheros que he comentado en la parte de diseño he optado por crear documentos en Notepad++ basándome en los ficheros con los que trabajamos en clase durante esta parte. En especial use el fichero de ciudades, el cual lo emplee para los ficheros Municipios.json y Carreras.json y que puede observar que tienen la misma estructura. Para el resto de ficheros, que son del tipo jsonArray emplee los ficheros de planetas y deportes.

Una vez conseguí la estructura que quería fui poblando los ficheros de documentos en los que incluía los datos de acuerdo con la estructura expuesta en el apartado de diseño.

Para finalizar, comento que he creado subcarpetas (las que he ido comentando a lo largo del trabajo) donde he ido colocando los ficheros que corresponden a esa parte. En la carpeta Diseño\_Proyecto puede encontrar el fichero Dia con el diagrama conceptual del trabajo, un documento Word con el diseño lógico del trabajo, que es el que copie y pegue en el apartado Diseño lógico de este documento, los datos que he cogido para insertar en las distintas bases de datos en el fichero Datos\_ProyectoBD.xlsx, y el esquema en MySQL Workbench a partir del que hemos construido nuestra base de datos relacional. En la carpeta Parte\_SQL puede encontrar el fichero BaseDatos\_ProyectoBD.sql donde está el código que he utilizado para crear las tablas de mi base de datos a través de phpmyadmin, el documento InsercionDatos\_ProyectoBD.sql donde se encuentra todos los insert para meter datos en la base de datos creada, el fichero Consultas\_ProyectoBD.sql donde hemos incluido todas las consultas acerca de nuestra base de datos y para finalizar hemos exportado nuestra base de datos que se encuentra en el fichero mydb.sql. El fichero de Access, Access\_ProyectoBD.accdb, con todas las tablas de nuestra base de datos, junto con algunos datos y los formularios anidados se encuentra en la carpeta Parte\_Access. Y la ultima parte de base de datos no relacional se halla en la carpeta Parte\_MongoDB, donde encontrara todos los documentos con los datos que he utilizado para realizar las consultas, a parte del documento Consultas\_ProyectoBD.js con todas las consultas de las bases de datos no relacional.