

# *Proyecto final INGP*



Iván Mañús Murcia

Jose M<sup>a</sup> Muela Bernabeu

Inteligencia de Negocio y Gestión de Procesos

Universidad de Alicante

# Índice

<b>Software necesario</b>	<b>2</b>
<b>Idea del proyecto y valor generado</b>	<b>3</b>
<b>Creación del esquema estrella</b>	<b>4</b>
<b>Schema Workbench</b>	<b>6</b>
<b>Pentaho-Data Integration</b>	<b>10</b>
Dimensión Fecha	13
Dimensión Pilotos	14
Dimensión Nacionalidad	15
Dimensión Carreras	16
Tabla de hechos	17
<b>Visualización y análisis de datos en Power BI</b>	<b>18</b>
<b>Bibliografía</b>	<b>25</b>

## Software necesario

Ante todo, especificar el software necesario para trabajar en esta práctica:

- **MySQL Workbench:** Para realizar el diseño lógico correspondiente a las especificaciones del enunciado.
- **Schema Workbench:** Software con el que podremos tratar los datos de las tablas de nuestro diseño lógico, elaborando un cubo con nuestro esquema.
- **Spoon:** Software incluido en el data-integration de Pentaho, con el cual realizamos un procesamiento de datos y puesta en sus respectivas tablas, comprobando posteriormente la población de datos en cada una de las tablas por medio de MySQL Workbench.
- **PowerBI:** Herramienta de visualización de datos una vez se conecta con la base de datos existente y disponemos de las tablas y tipos de datos de la misma.

## Idea del proyecto y valor generado

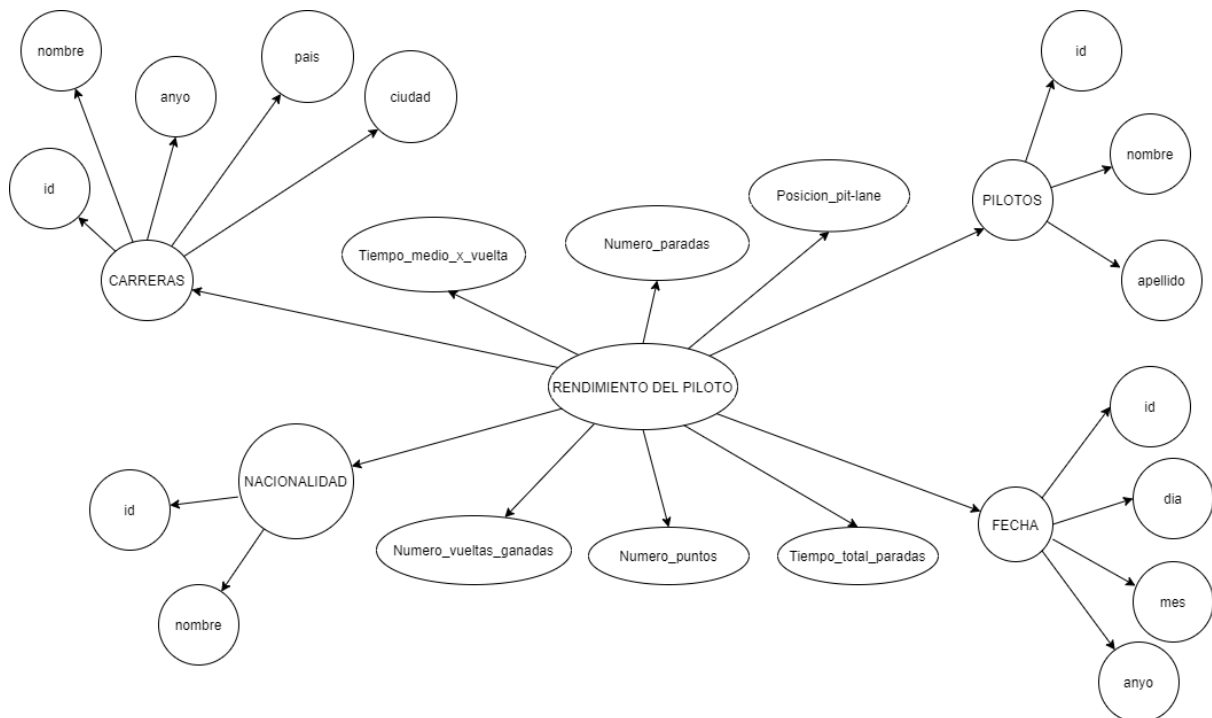
Para la idea de proyecto decidimos indagar en varias ideas y decidimos escoger una cuyo dataset fuese bastante completo, en diferentes archivos .csv y cuyo tema fuese de nuestro agrado y conociéramos bien. Teniendo en cuenta estos requisitos, nos decantamos por un dataset cuya información era acerca del mundo de la Fórmula 1, recogiendo los diferentes datos que abarca este deporte desde el año 1950 hasta la actualidad. Sin embargo, para aclarar el panorama, hemos decidido omitir ciertos archivos .csv que no veíamos imprescindibles para poder realizar un buen trabajo (como es el caso de los constructores/escuderías/equipos de F1). La idea era centrarnos en el rendimiento de un piloto en concreto (con nombre, apellidos y nacionalidad) en todas y cada una de las vueltas de cada carrera (con su nombre, ciudad y país) de las distintas temporadas que contienen los datos. Además de los datos típicos de un piloto, también haremos especial hincapié en otros datos como el tiempo medio por vuelta, el número de puntos ganados en una temporada y en la posición de parrilla (*pit-lane*) de cada piloto en cada carrera de todas las temporadas en las que disputan grandes premios.

El valor generado se dirige a los posibles clientes potenciales, en cuyo caso podrían ser las diferentes escuderías que participan en cada temporada de F1, como pueden ser Mercedes, Ferrari o McLaren entre otras. Uno de los aspectos a resaltar es, sin duda, que podremos apreciar, gracias a los datos recogidos por el rendimiento, qué pilotos son más rápidos en cuanto a tiempo por vuelta, cuáles puntúan más por temporada y quiénes ganan más vueltas en los respectivos grandes premios.

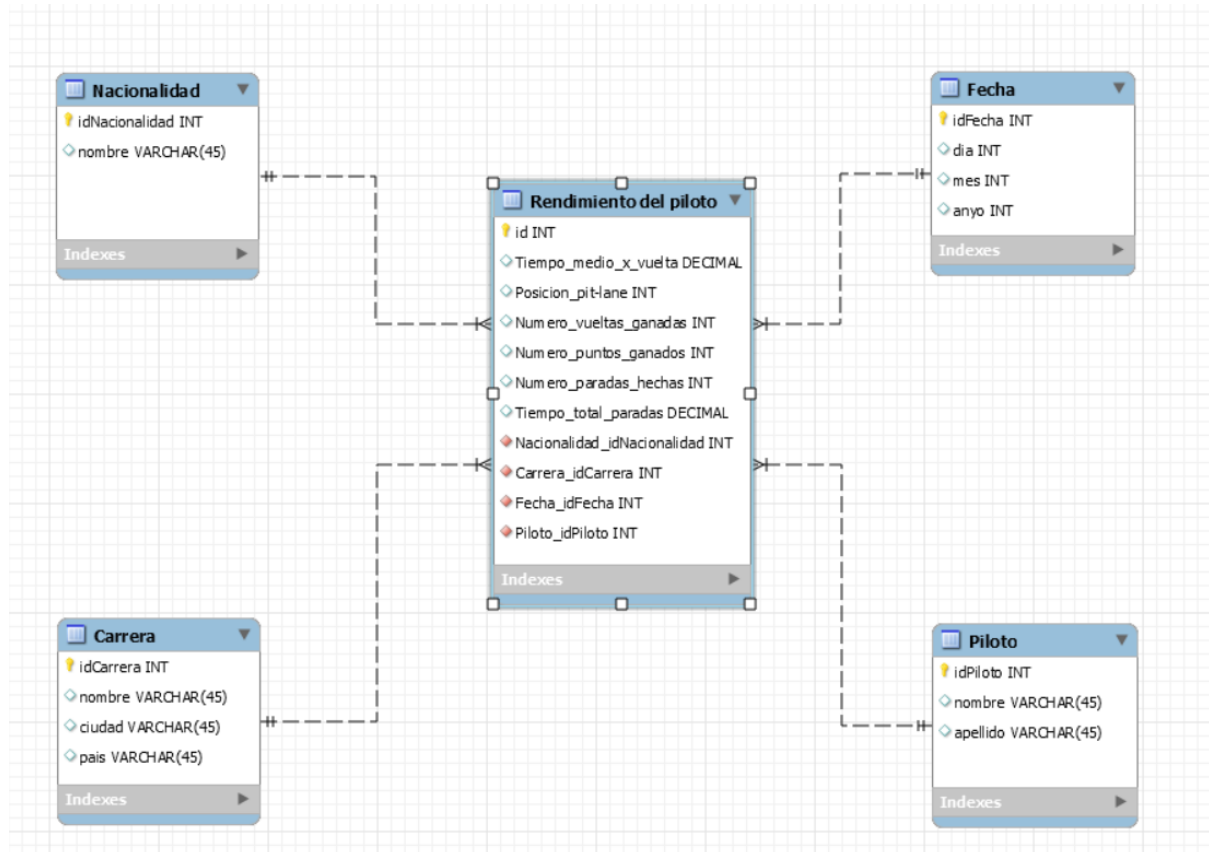
## Creación del esquema estrella

Una vez localizados todos los requisitos y estructurando las tablas con sus respectivos tipos de datos, hacemos las relaciones pertinentes entre las mismas para crear el diseño lógico. Acto seguido, determinaremos la tabla de hechos y sus dimensiones, haciendo uso de los archivos .csv proporcionados. Como los .csv contienen una cantidad alta de datos, nos vendrán bien para realizar el volcado de información más adelante.

Partimos del siguiente diseño conceptual una vez localizadas todas las especificaciones del problema:



Una vez se determinan las tablas, sus tipos de datos y las relaciones entre ellas, creamos el modelo E/R por medio de la herramienta MySQL Workbench, quedando de ésta forma con el ejercicio en cuestión:



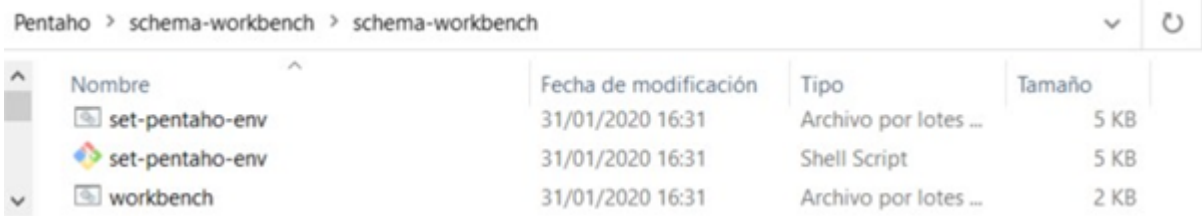
En este caso, consideramos como tabla de hechos la tabla **rendimiento\_piloto**, ya que tiene relación con el resto de tablas. Mientras que las demás son tablas de dimensiones.

Acto seguido, creamos la base de datos por medio de la opción **Forward Engineer** siguiendo estos pasos:

- Rellenamos los siguientes campos:
  - Connection
  - Hostname: **localhost**
  - Port: **3306**
  - Username: **root**
    - Password: **Store in Vault** ... y ponemos la contraseña del usuario.
- No creamos las claves ajenas (quitamos la opción de crearlas que por defecto está en crearlas).
- Decidimos si queremos exportar el código creado por las tablas a un script o a otro gestor de base de datos.
- Seleccionamos **Siguiente** con los valores por defecto hasta llegar a **Finalizar**. Una vez le damos a esa opción y configuramos la conexión con el servidor MySQL, se habrán creado todas las tablas vacías en la conexión MySQL utilizada.

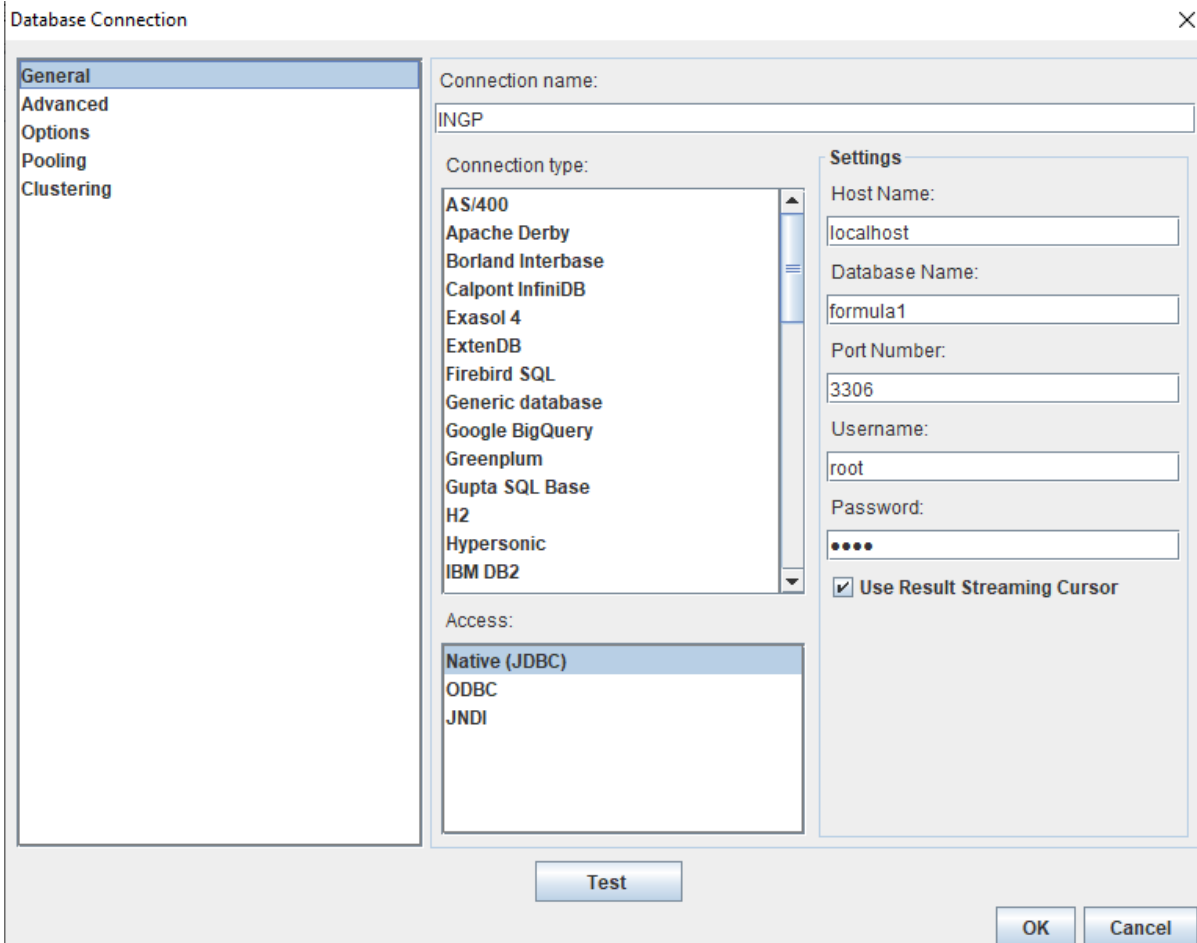
## Schema Workbench

Para ejecutar esta herramienta, hacemos doble clic en el archivo workbench.bat ubicado en la carpeta comprimida schema-workbench descargada desde la página de Pentaho.



Nombre	Fecha de modificación	Tipo	Tamaño
set-pentaho-env	31/01/2020 16:31	Archivo por lotes ...	5 KB
set-pentaho-env	31/01/2020 16:31	Shell Script	5 KB
workbench	31/01/2020 16:31	Archivo por lotes ...	2 KB

Una vez arrancado, procedemos a conectar con nuestra base de datos recientemente creada.



Database Connection

General  
Advanced  
Options  
Pooling  
Clustering

Connection name:  
INGP

Connection type:  
AS/400  
Apache Derby  
Borland Interbase  
Calpont InfiniDB  
Exasol 4  
ExtenDB  
Firebird SQL  
Generic database  
Google BigQuery  
Greenplum  
Gupta SQL Base  
H2  
Hypersonic  
IBM DB2

Access:  
Native (JDBC)  
ODBC  
JNDI

Settings  
Host Name:  
localhost  
Database Name:  
formula1  
Port Number:  
3306  
Username:  
root  
Password:  
.....  
☒ Use Result Streaming Cursor

Test OK Cancel

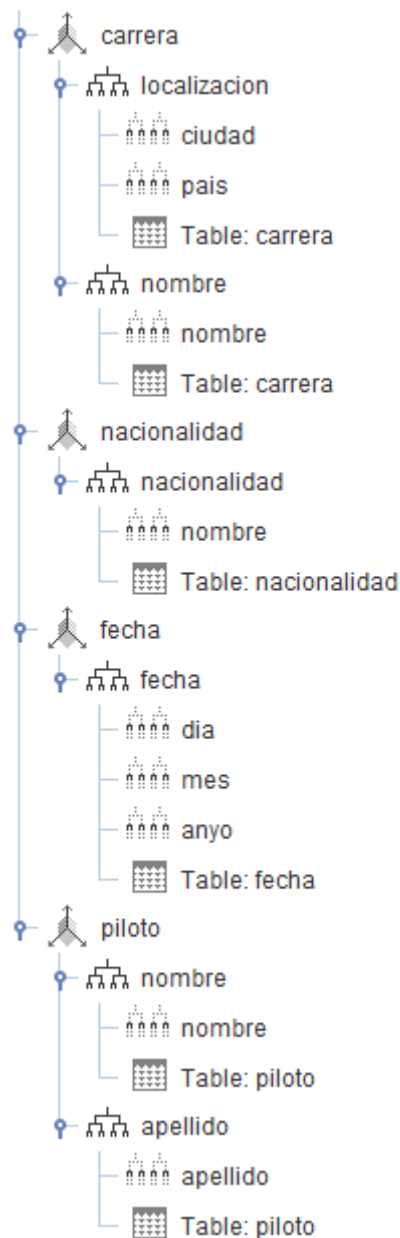
Con ello, crearemos un esquema partiendo de las **tablas de dimensiones** y finalizando con la **tabla de hechos**.

Creamos las tablas de hechos por medio de la opción **Add Dimension**.

Para crear las jerarquías en cada dimensión usamos **Add Hierarchy**.

En cada jerarquía podemos crear niveles desde **Add Level**. Esta opción nos servirá para datos que agrupan varios tipos de datos (p.e la fecha, creando los niveles **día**, **mes** y **año**, los cuales son tres tipos de datos en uno).

Haciendo esto con todas las tablas de dimensiones que tenemos, debe de quedar una estructura de la siguiente forma:





Esto se realiza con todas las tablas de dimensiones y sus respectivos tipos de datos.

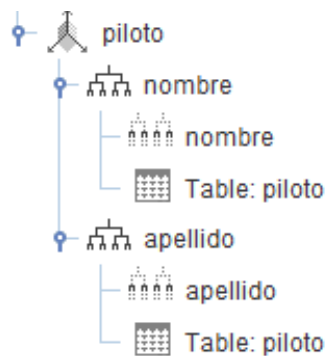
Una vez realizado todo este proceso para cada tabla, nos centramos en la **tabla de hechos** (rendimiento\_piloto), creando la misma por medio de la opción **Add cube** y añadiendo en el cubo:

- Crear una tabla para asociarlo con **rendimiento\_piloto** de la base de datos.
- Todas las tablas de dimensiones necesarias --> **Add Dimension Usage**.
- Añadir las medidas para los atributos de la tabla de hechos, utilizando la opción **Add Measure**.

La estructura que debe adoptar el cubo debe ser la siguiente:



En la siguiente imagen se aplica para cada una de las tablas de dimensiones aunque solo salga la imagen para la tabla **pilotos**.



Aquí añadimos todos los atributos de medidas para los tipos de datos de la tabla **rendimiento\_piloto**.

Solo ponemos el de un tipo de dato, pero se aplica para todos los que tiene la tabla de hechos (tiempo\_medio\_x\_vuelta, vueltas\_ganadas, puntos\_ganados, tiempo\_duracion y num\_paradas), siendo bien el agregador **sum** o bien **count**. Sin embargo, el **datatype** (**numeric**) se aplica a todas las medidas.

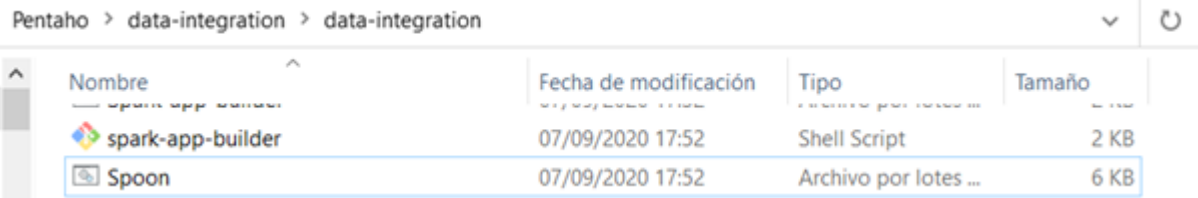
Attribute	
name	tiempo_medio_x_vuelta
description	
aggregator	avg
column	Tiempo_medio_x_vuelta
formatString	
datatype	Numeric
formatter	
caption	
visible	<input checked="" type="checkbox"/>

Attribute	
name	posicion_pitlane
description	
aggregator	count
column	Posicion_pit-lane
formatString	
datatype	Numeric
formatter	
caption	
visible	<input checked="" type="checkbox"/>

Attribute	
name	numero_puntos_ganados
description	
aggregator	sum
column	Numero_puntos_ganados
formatString	
datatype	Numeric
formatter	
caption	
visible	<input checked="" type="checkbox"/>

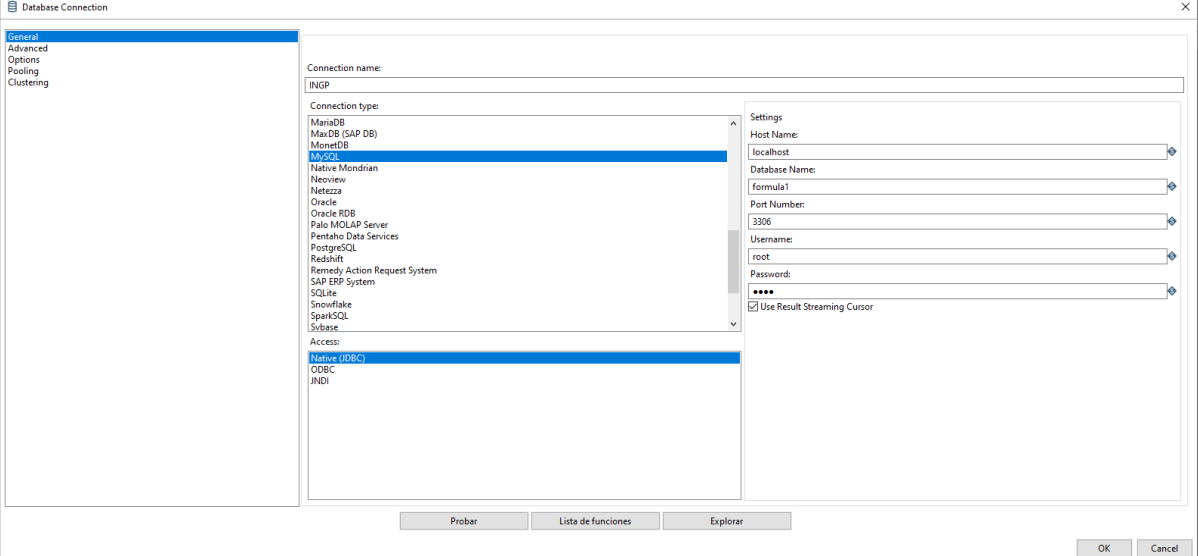
## Pentaho-Data Integration

Para iniciar **Spoon** tendremos que arrancar el servicio Spoon ubicado en la siguiente ruta:



Pentaho > data-integration > data-integration			
Nombre	Fecha de modificación	Tipo	Tamaño
spark-app-builder	07/09/2020 17:52	Shell Script	2 KB
Spoon	07/09/2020 17:52	Archivo por lotes ...	6 KB

En primer lugar, conectamos la BD configurando un conector como en el caso anterior:



Database Connection

Connection name: INGP

Connection type:

- MySQL
- Native Mondrian
- Neoview
- Netezza
- Oracle
- Oracle RDB
- Palo MOLAP Server
- Pentaho Data Services
- PostgreSQL
- Redshift
- Remedy Action Request System
- SAP ERP System
- SQLite
- Snowflake
- SparkSQL
- Sybase

Access:

- Native (ODBC)
- ODBC
- JNDI

Settings:

Host Name: localhost

Database Name: formula1

Port Number: 3306

Username: root

Password: root

☒ Use Result Streaming Cursor

Probar Lista de funciones Explorar

OK Cancel

Una vez ejecutamos todas las entradas y salidas para cada tabla, tiene que salir un tic en verde indicando que la transformación en Spoon ha funcionado correctamente.

Para ello, comenzamos creando las **entradas** (en archivos CSV) para cada una de las tablas.

CSV file input

Step name: CSV file input

Filename: C:\Users\IVAN\Desktop\UA42C\INGP\PRACTICA FINAL\CSV\drivers.csv Examinar...

Delimiter: , Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim ty
1	driverId	Integer	#	15	0	€	,	.	ningur
2	driverRef	String		18		€	,	.	ningur
3	number	String		2		€	,	.	ningur
4	code	String		3		€	,	.	ningur
5	forename	String		15		€	,	.	ningur
6	surname	String		13		€	,	.	ningur
7	dob	Date	yyyy-MM-dd			€	,	.	ningur
8	nationality	String		10		€	,	.	ningur
9	url	String		58		€	,	.	ningur

Y las **salidas** a tabla.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: ingp Editar... Nuevo... Wizard...

Esquema destino: formula1 Examinar...

Tabla destino: piloto Examinar...

Tamaño transacción (commit): 1000

Vaciar tabla ☐

Ignorar errores de inserción ☐

Specify database fields ☒

Main options Database fields

Repartir información en varias tablas ☐

Campo de partición:

Particionar información por mes ☒

Particionar información por días ☐

Utilizar actualización por lotes para inserciones ☒

El nombre de la tabla está definido en un campo? ☐

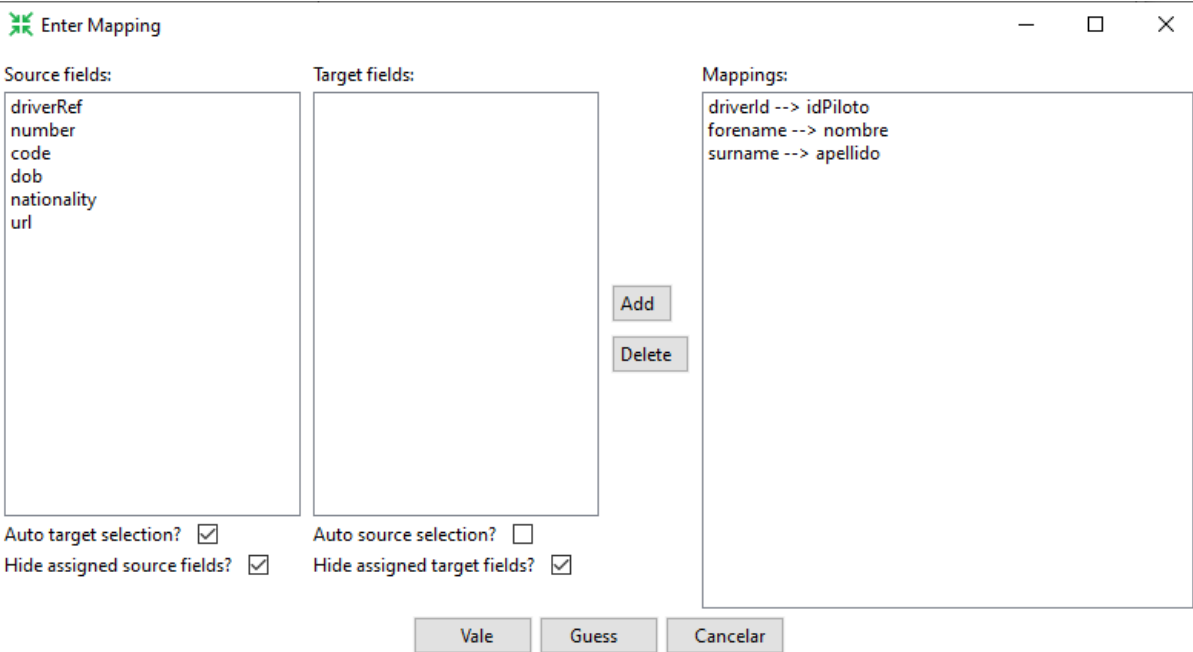
Campo que contiene el nombre de la tabla:

Almacena el campo con el nombre de tabla ☒

Incluye clave auto-generada ☐

Nombre del campo clave auto-generada:

Help Vale Cancelar SQL



Enter Mapping

Source fields:

- driverRef
- number
- code
- dob
- nationality
- url

Target fields:

Mappings:

- driverId --> idPiloto
- forename --> nombre
- surname --> apellido

Add

Delete

Auto target selection? ☒ Auto source selection? ☐

Hide assigned source fields? ☒ Hide assigned target fields? ☒

Vale Guess Cancelar

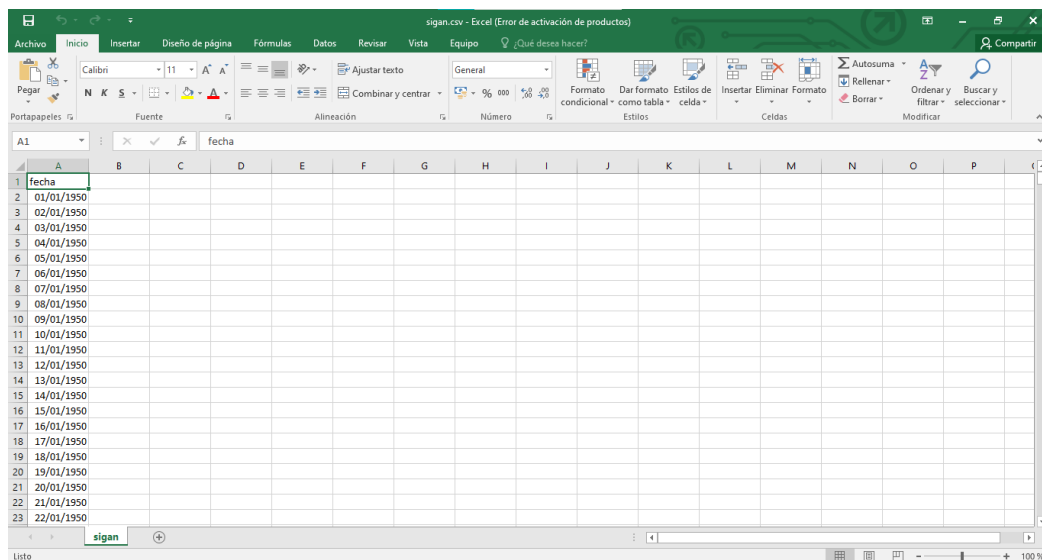
Y entremedias de entradas y salidas, hemos utilizado las diferentes funciones:

- **Ordenar filas:** Para ordenar los datos por un campo en concreto.
- **Memory group by:** Con ello extraemos varios valores de agregación. Utilizado para, entre otras, sacar el máximo de vueltas que gana cada piloto en cada carrera o también para, por medio de un sumatorio, obtener el tiempo total de paradas que ha invertido un piloto en una carrera exactas.
- **Filas únicas:** Con ello conseguimos eliminar los duplicados, logrando también que la tabla se reduzca notablemente en cuanto a número de filas. Se ha utilizado para, concretamente, la dimensión **Nacionalidad**, de forma que, aunque haya varios pilotos con la misma nacionalidad, sólo se identifican por una de la tabla.
- **Filtrar filas:** Para filtrar los datos de un campo en concreto. En nuestro caso, lo hemos utilizado para filtrar la posición en la que termina una vuelta cada piloto en cada carrera. Para ello, el dato de posición será 1, para indicar que el piloto ha ganado una vuelta. Una vez recopiladas todas las filas cuyo valor en ese campo es 1, hemos utilizado un sumatorio con **Memory group by**, de forma que nos calcule el número de vueltas ganadas de cada piloto en cada carrera.
- **Calculadora:** Permite calcular medidas y crear columnas nuevas. Usado en el cubo para dividir el tiempo total de la carrera entre el número de vueltas, para obtener la medida tiempo medio por vuelta.
- **Seleccionar/Renombrar valores:** Muy utilizado, usado sobre todo para eliminar columnas y cambiar tipos a las columnas.
- **Partir campos:** Utilizado para dividir las fechas(transformadas a string) en 3 columnas
- **Unión por clave:** Utilizado para juntar 2 tablas resultados mediante uno o más campos. Es semejante a **Multiway Join**

## Dimensión Fecha

Para insertar la dimensión fecha en la base de datos, tenemos que generar de alguna forma todas las fechas desde 1950.

En nuestro caso, hemos optado por generar un fichero de Excel:



Una vez tengamos todas las fechas desde 1950 hasta 2021, procedemos a importar dicho CSV desde PDI.

Renombramos el campo de date a string y partimos los campos en 3 columnas.




formula1.fecha: 23.729 filas en total (aproximadamente), limitado a 1.000

idFecha	dia	mes	anyo
1	1	1	1.950
2	2	1	1.950
3	3	1	1.950
4	4	1	1.950
5	5	1	1.950
6	6	1	1.950
7	7	1	1.950
8	8	1	1.950
9	9	1	1.950
10	10	1	1.950
11	11	1	1.950
12	12	1	1.950
13	13	1	1.950
14	14	1	1.950
15	15	1	1.950
16	16	1	1.950
17	17	1	1.950
18	18	1	1.950

## Dimensión Pilotos



En el caso de la dimensión pilotos, tenemos todos los datos en el CSV, tan solo mapeamos valores e insertamos en la base de datos.

 idPiloto	nombre	apellido
1	Lewis	Hamilton
2	Nick	Heidfeld
3	Nico	Rosberg
4	Fernando	Alonso
5	Heikki	Kovalainen
6	Kazuki	Nakajima
7	Sébastien	Bourdais
8	Kimi	Räikkönen
9	Robert	Kubica
10	Timo	Glock
11	Takuma	Sato
12	Nelson	Piquet Jr.
13	Felipe	Massa
14	David	Coulthard
15	Jarno	Trulli
16	Adrian	Sutil
17	Mark	Webber
18	Jenson	Button

## Dimensión Nacionalidad

El atributo de nacionalidad está en el CSV de pilotos.

Para ello tenemos que ordenar por la columna *nacionality*, usar la herramienta Filas únicas que borra los registros duplicados e insertar en nuestra base de datos el resultado:



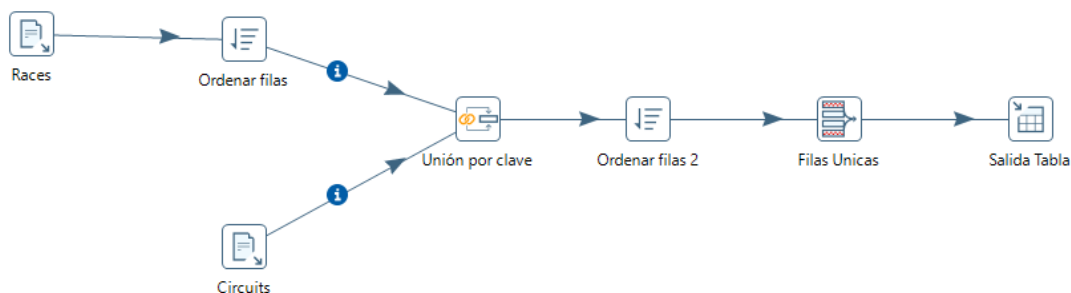
 idNacionalidad	nombre
854	American
855	American-Italian
856	Argentine
857	Argentine-Italian
858	Australian
859	Austrian
860	Belgian
861	Brazilian
862	British
863	Canadian
864	Chilean
865	Colombian
866	Czech
867	Danish
868	Dutch
869	East German
870	Finnish
871	French



## Dimensión Carreras

En esta dimensión tenemos que comprobar en qué ciudad y país se celebra cada carrera.

El CSV de carreras, tiene un identificador de circuito, por ello tenemos que unirlos por clave y borrar los registros repetidos como en la dimensión anterior y posteriormente introducirlo todo en la base de datos.



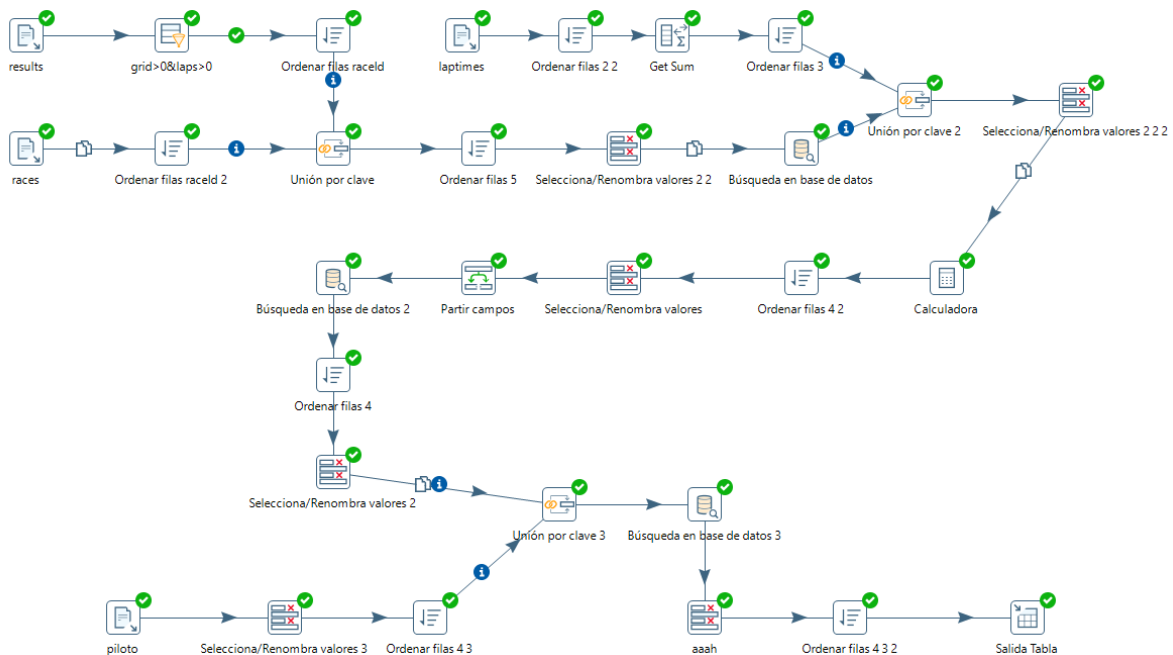
 idCarrera	nombre	ciudad	pais
1	70th Anniversary Grand Prix	Silverstone	UK
2	Abu Dhabi Grand Prix	Abu Dhabi	UAE
3	Argentine Grand Prix	Buenos Aires	Argentina
4	Australian Grand Prix	Melbourne	Australia
5	Austrian Grand Prix	Spielburg	Austria
6	Azerbaijan Grand Prix	Baku	Azerbaijan
7	Bahrain Grand Prix	Sakhir	Bahrain
8	Belgian Grand Prix	Spa	Belgium
9	Brazilian Grand Prix	SÃ£o Paulo	Brazil
10	British Grand Prix	Silverstone	UK
11	Caesars Palace Grand Prix	Nevada	USA
12	Canadian Grand Prix	Montreal	Canada
13	Chinese Grand Prix	Shanghai	China
14	Dallas Grand Prix	Dallas	USA
15	Detroit Grand Prix	Detroit	USA
16	Dutch Grand Prix	Zandvoort	Netherlands
17	Eifel Grand Prix	NÃ¼rburg	Germany
18	Emilia Romagna Grand Prix	Imola	Italy

## Tabla de hechos

El proceso ETL que seguimos para construir la tabla de hechos es bastante complejo, puesto que hay que unir varias salidas y funciones para ir construyendo poco a poco la tabla final que insertaremos en la base de datos.

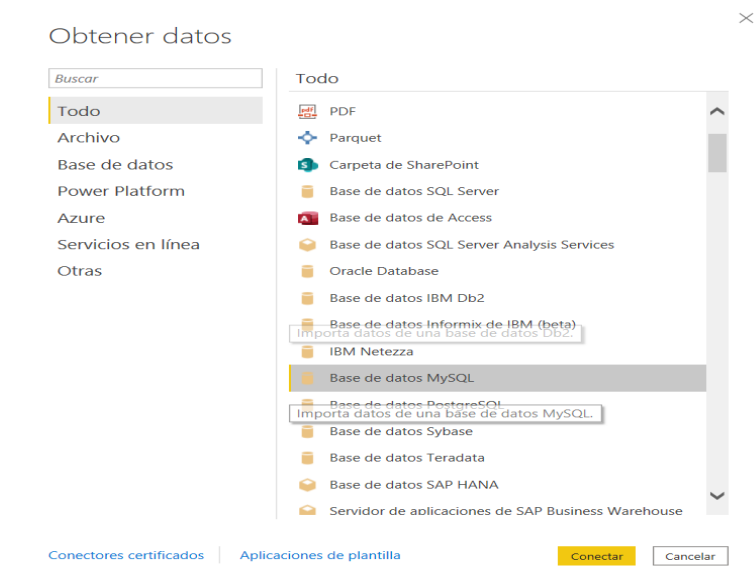
Para entrar un poco más en detalle, iremos paso a paso explicando en la medida de lo posible como hemos organizado todo este proceso ETL y darle un poco de sentido:

1. Empezamos con el CSV de resultados totales, con más de 20.000 registros donde hay datos de todas las carreras y resultados por piloto
2. Para que se pueda reconocer la carrera tenemos que unir dicho CSV con el CSV que contiene los datos de las carreras y así poder saber a qué carrera pertenece cada registro.
3. Luego, para saber el tiempo que ha tardado cada piloto en cada carrera, necesitamos hacer un sumatorio de todos los milisegundos tardados en una carrera.
4. Después debemos hacer la medida tiempo medio por vuelta, que se calcula usando la función o herramienta “Calculadora”, con la que tenemos que dividir el tiempo total entre las vueltas que ha dado ese piloto
5. Por último, sacamos todas las fechas de la base de datos y vamos casando los datos para que nos quede una única tabla resultado, que al final, junto a la nacionalidad del piloto extraída también de la base de datos tener el resultado final e insertarlo todo en la base de datos, en nuestra tabla de hechos.

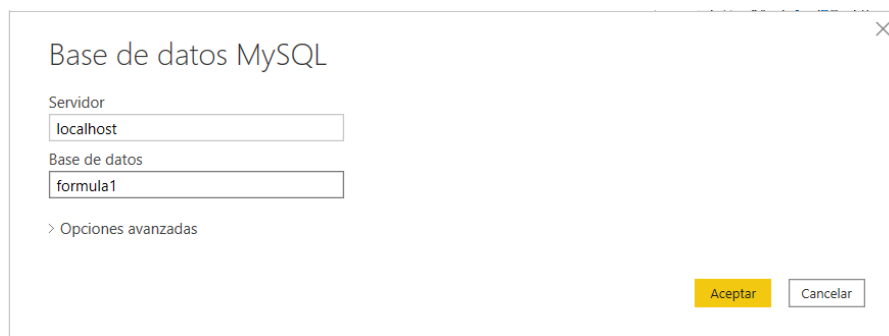


## Visualización y análisis de datos en Power BI

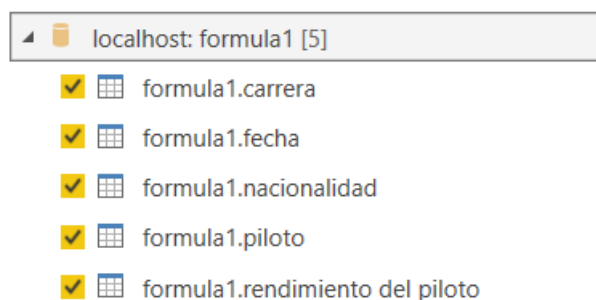
Abrimos Power BI Desktop y, antes de nada, seleccionamos la opción **Obtener Datos > Base de datos > MySQL**.



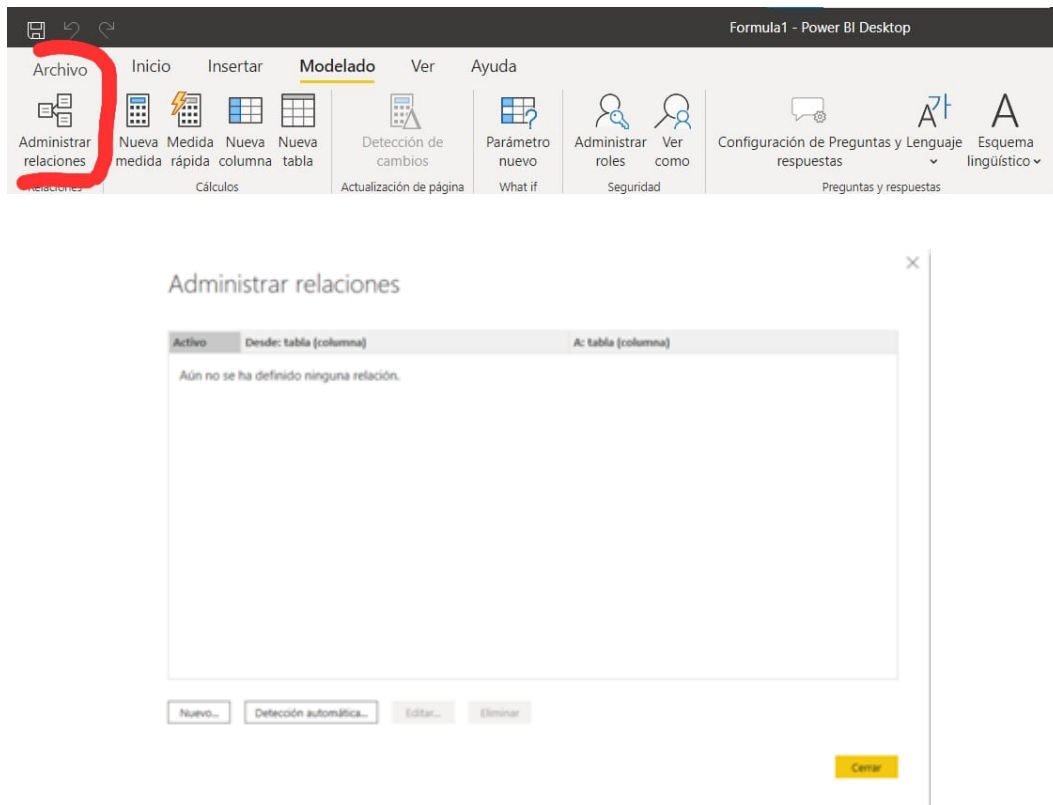
Ahora añadimos los datos de conexión y éste nos mostrará las tablas a continuación.



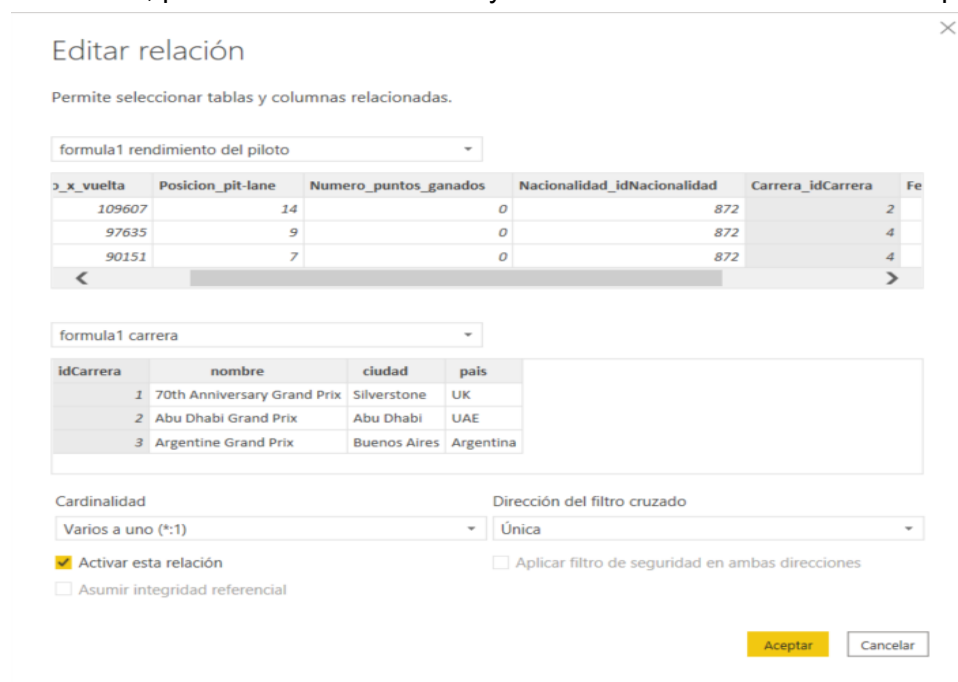
Seleccionamos todas las tablas y esperamos a que se carguen automáticamente.



Una vez hecho esto, exportamos la base de datos que creamos al principio en este programa desde **Administrar relaciones** (ubicado arriba a la izquierda de la ventana).



Le damos a **Nuevo** para crear las relaciones de la tabla de hechos con cada una de las tablas dimensionales, clicando sobre los tipos de datos id de cada tabla para hacer las pertinentes uniones, poniendo la cardinalidad y dirección del filtro cruzado correspondientes.



Le damos a **Aceptar** y aceptamos todas las relaciones (con todas las dimensiones) que hemos realizado previamente.

Administrar relaciones

Activo	Desde: tabla (columna)	A: tabla (columna)
<input checked="" type="checkbox"/>	formula1 rendimiento del piloto (Carrera_idCarrera)	formula1 carrera (idCarrera)
<input checked="" type="checkbox"/>	formula1 rendimiento del piloto (Fecha_idFecha)	formula1 fecha (idFecha)
<input checked="" type="checkbox"/>	formula1 rendimiento del piloto (Nacionalidad_idNacionalidad)	formula1 nacionalidad (idNacionalidad)
<input checked="" type="checkbox"/>	formula1 rendimiento del piloto (Piloto_idPiloto)	formula1 piloto (idPiloto)

Nuevo...

Detección automática...

Editar...

Eliminar

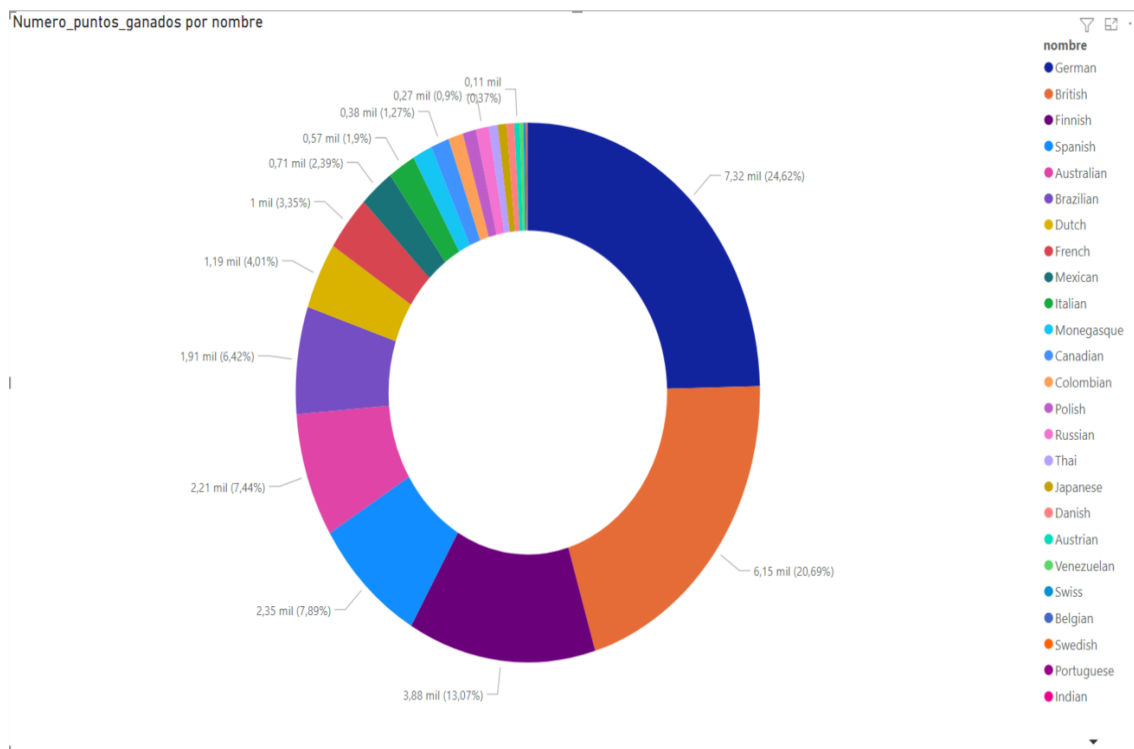
Cerrar

Para terminar este apartado, insertamos una serie de gráficos de barras, sectores, anillos y mapas, añadiendo en los mismos los datos de las medidas que contiene la tabla **rendimiento\_piloto**:

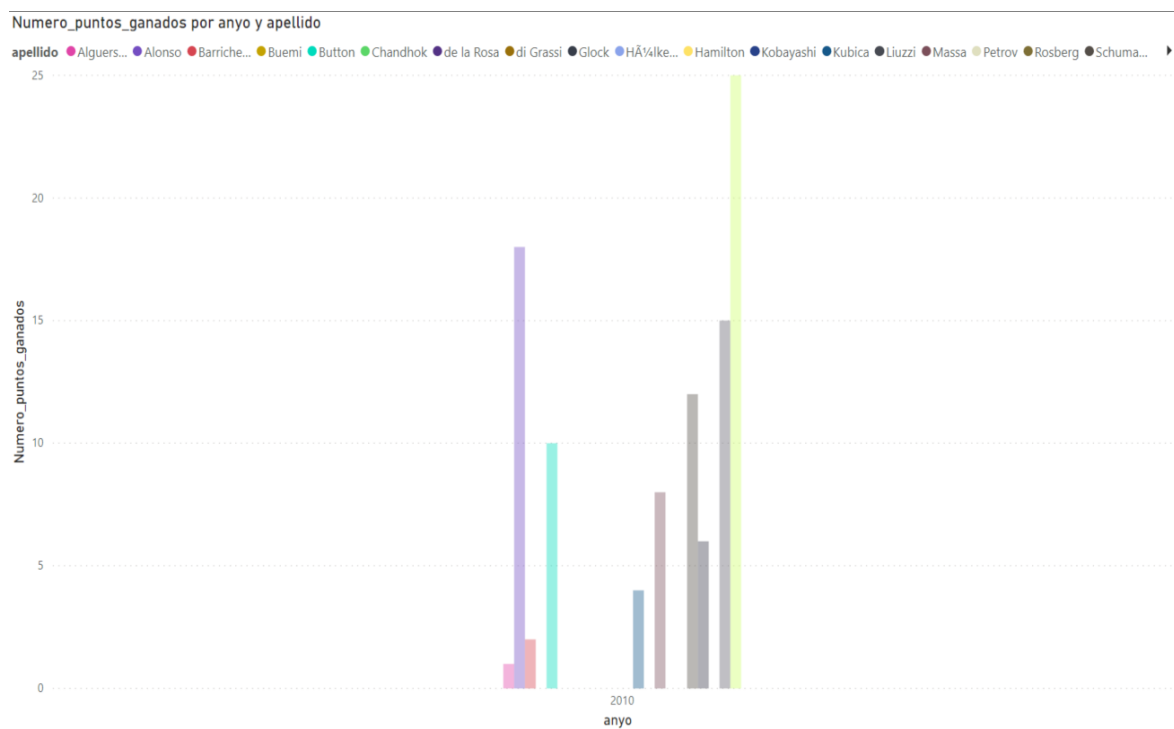
- **tiempo\_medio\_x\_vuelta**
- **posicion\_pit-lane**
- **num\_puntos\_ganados**

Ahora podremos ver, mediante una serie de herramientas de visualización de datos que nos aporta este software, cómo quedarían reflejados los datos contenidos en las diferentes tablas de nuestra base de datos.

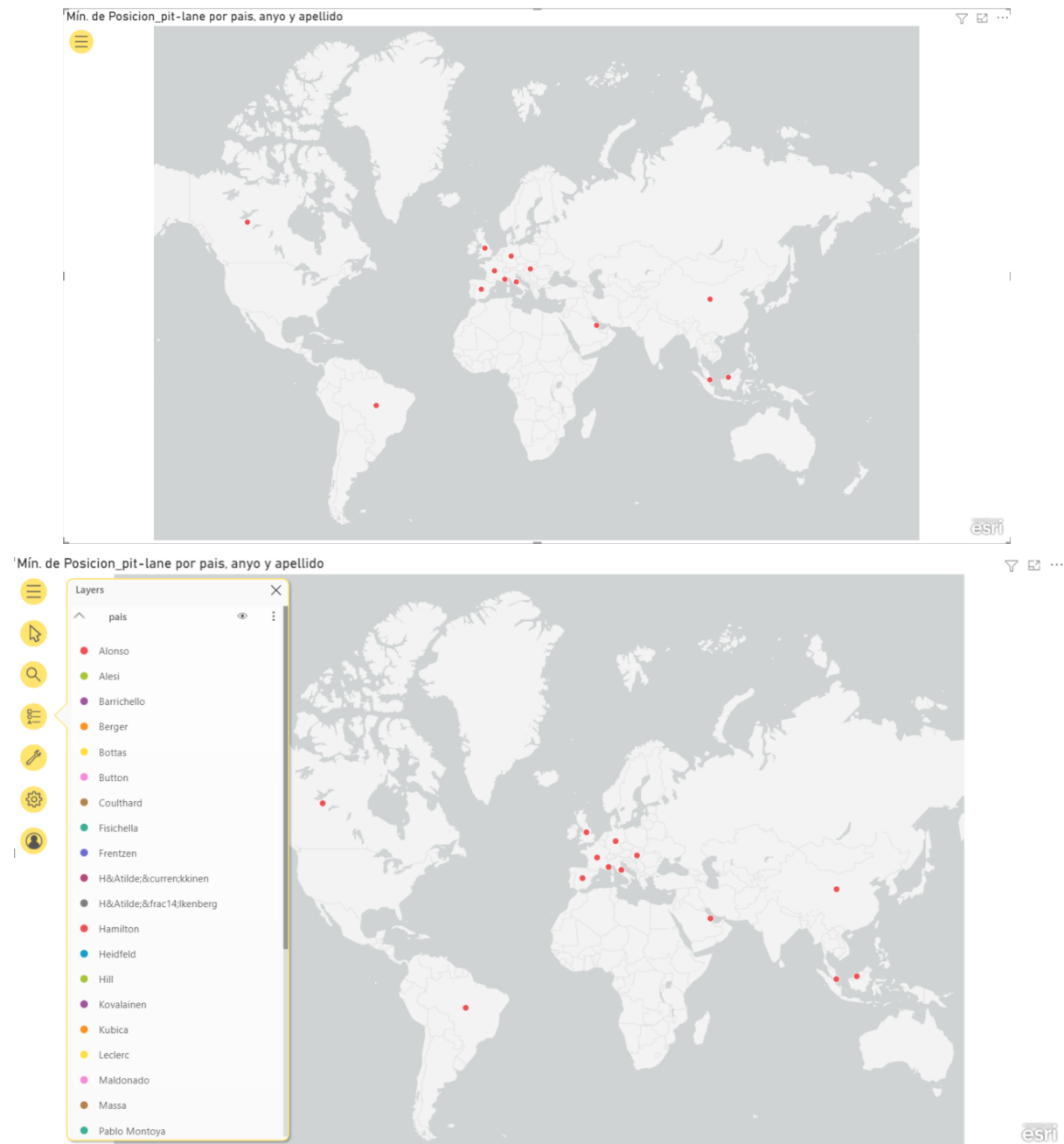
En este **gráfico de anillos** podemos visualizar los puntos que se han obtenido en todas las carreras de todas las temporadas (desde 1950 hasta 2021 inclusive) por medio de la nacionalidad a la que pertenezcan los pilotos que disputen esos grandes premios.



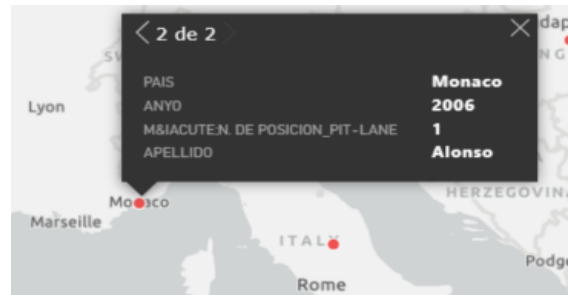
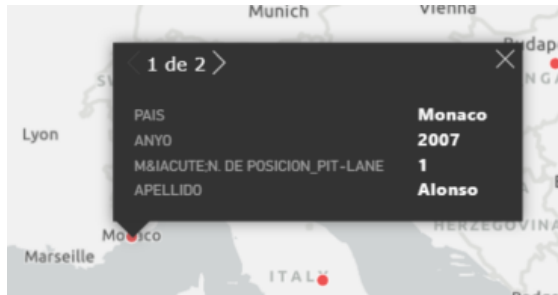
En este **gráfico de columnas agrupadas** podemos comprobar, mediante una serie de filtros, el número de puntos que han realizado los pilotos que hayan corrido el Gran Premio de Montmeló de la temporada 2010.



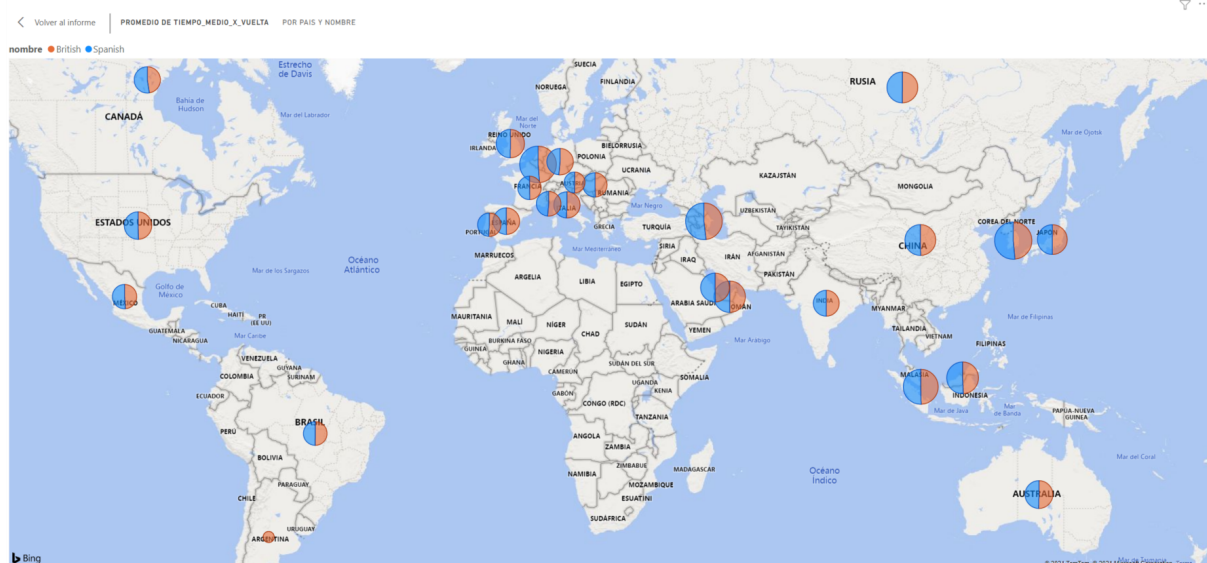
En este **mapa ArcGIS Maps** podemos visualizar, mediante una serie de puntos de color rojo, todos aquellos países en los que el piloto Fernando Alonso ha conseguido salir primero desde la parrilla de salida (*pit-lane*) desde la temporada 1950 hasta la vigente.



Cuando pasamos el ratón por encima de alguno de los puntos de color rojo, podemos comprobar los datos del **país** y **año** en los que el piloto Fernando Alonso ha quedado primero en la parrilla de salida (*pit-lane*). Si se da el caso de que en una misma carrera parte en la primera posición, se podrá visualizar de la siguiente manera. En este caso, ponemos como ejemplo el GP de Mónaco en el que Fernando Alonso parte primero en parrilla en los años 2006 y 2007.

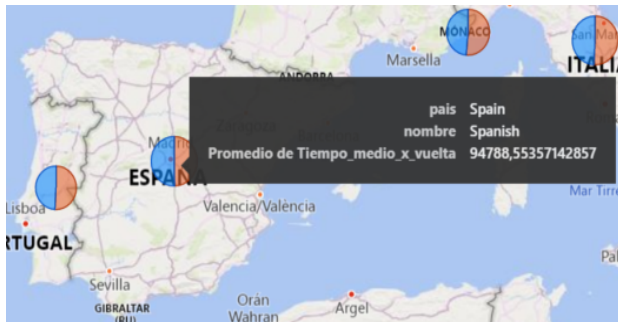


En este **mapa** podemos visualizar una comparativa del **tiempo\_medio\_x\_vuelta** en los diferentes países donde se disputan carreras de los pilotos españoles (color azul) y los pilotos británicos (color naranja).

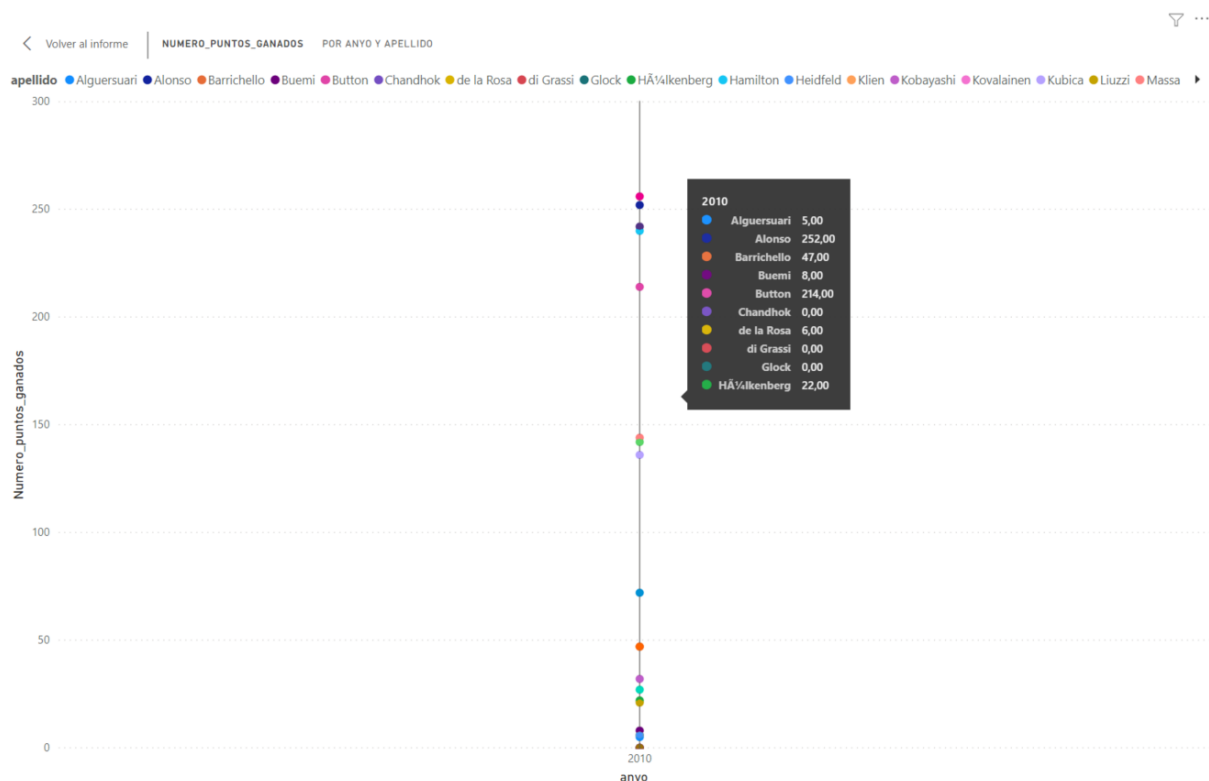




Cuando pasamos el ratón por encima de alguno de los sectores, podemos comprobar los datos correspondientes sobre los pilotos españoles y británicos sobre ese país en concreto (en este caso, España).



Por último, hemos optado por un **gráfico de líneas** en la que, por medio de algunos filtros, podemos determinar el número de puntos obtenidos en un año en concreto por parte de los pilotos que compiten en todos los grandes premios de esa temporada en cuestión. En el ejemplo que se puede visualizar a continuación hemos puesto como filtro el año 2010, viendo que esa temporada ganó Vettel con 256 puntos y el segundo fue Fernando Alonso con 252.



## Bibliografía

**Fuente de datos en Kaggle ↓**

<https://www.kaggle.com/rohanrao/formula-1-world-championship-1950-2020>