

# BASE DE DATOS DE FÓRMULA 1

## ÍNDICE

1. Descripción
2. Modelo Entidad-Relación
3. Modelo Relacional
4. Carga de Datos
5. Consultas
6. Vistas
7. Funciones
8. Procedimientos
9. Conexión con AWS
10. Enlace GitHub
11. Conclusión



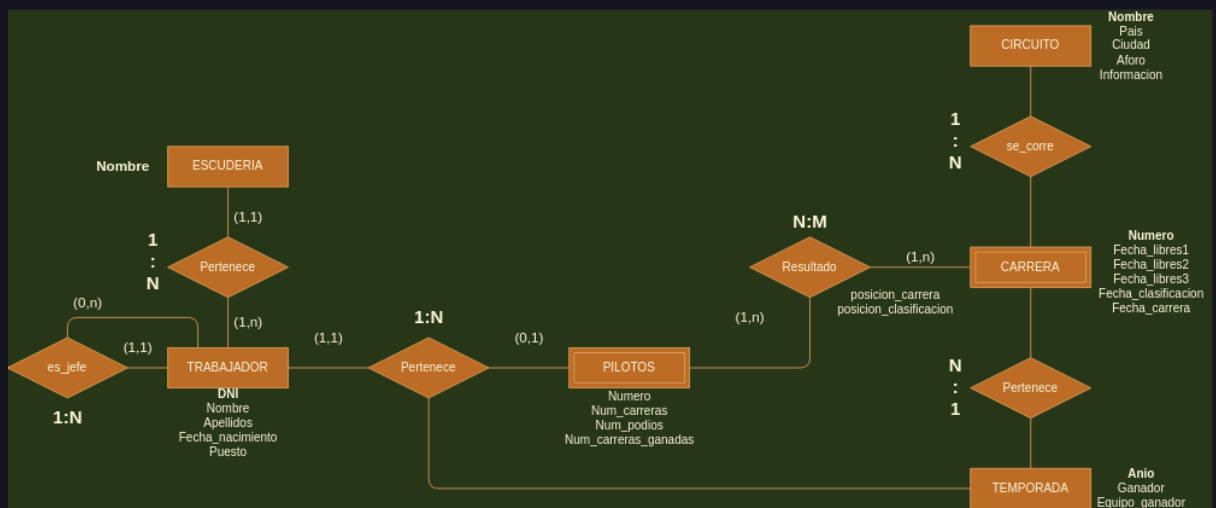
## 1. Descripción

Para este proyecto de Base de Datos he decidido realizar una base de datos con la historia de la competición automovilística de Fórmula 1.

Para hacer dicha base de datos he utilizado información sobre los pilotos que han corrido y corren actualmente en la competición, las carreras que se han disputado en cada temporada desde su inicio con el resultado de las mismas y en qué circuitos han sido. Además de toda la información relacionada con las escuderías.

## 2. Modelo Entidad-Relación

Explicado todo lo anterior, puedo mostrar el modelo Entidad-Relación resultante:

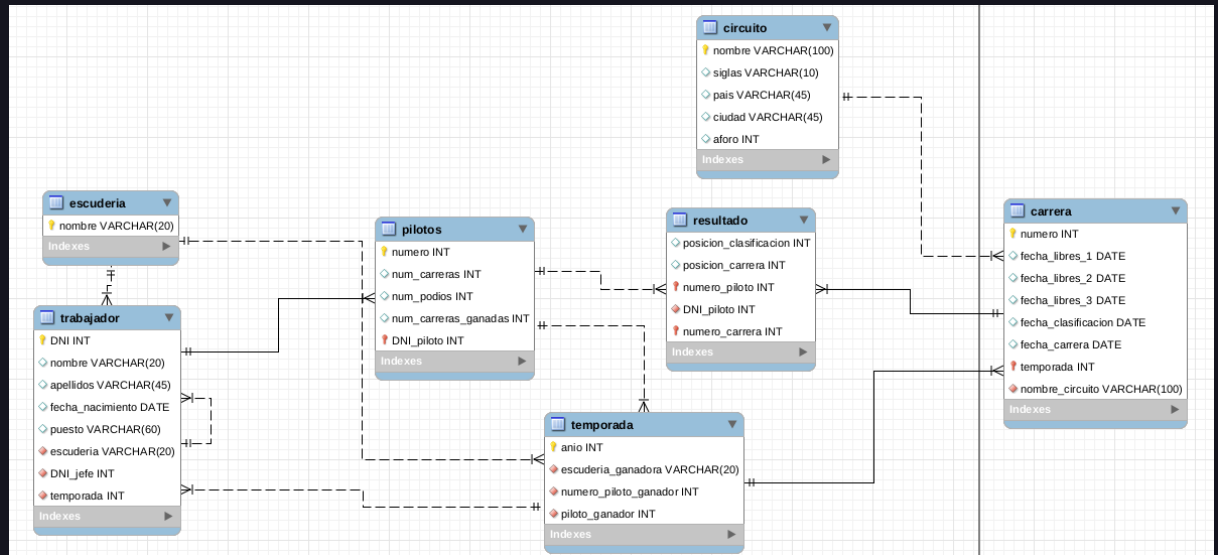


En este modelo podemos observar que tenemos 6 entidades (Pilotos, Escudería, Temporada, Carrera, Circuito y Trabajador). También podemos observar que:

- La entidad Pilotos es una herencia de la tabla Trabajador ya que un piloto es siempre un trabajador de la escudería pero un trabajador puede ser o no un piloto. En caso de que no lo sea, no haría falta guardar más información que la que está en la tabla Trabajador, por eso no está representado en el modelo expuesto arriba.
- La entidad Carrera es débil de la tabla Temporada ya que necesitamos del indicador de la misma para poder identificar qué carrera es, ya que hay muchas carreras 1, 2, 3, ..., lo que las diferencia es la temporada.
- Vemos una relación reflexiva en la tabla Trabajador ya que los trabajadores tienen un jefe que a su vez es trabajador de la escudería.
- Vemos una relación N:M que posteriormente será una tabla nueva en el modelo Relacional

### 3. Modelo Relacional

Una vez claro el modelo Entidad-Relación, pasamos al modelo Relacional:



En este modelo podemos observar que:

- Hay 7 tablas (las 6 de las entidades y la relación N:M del modelo Entidad-Relación).
- El nombre definitivo de todos los campos en cada tabla.
- Las claves foráneas de todas las tablas
  - DNI\_jefe en la tabla trabajador referenciando el DNI del trabajador
  - escuderia en la tabla trabajador referenciando la clave de escuderia
  - temporada en la tabla trabajador referenciando la clave de temporada
  - escuderia\_ganadora en la tabla temporada referenciando la escuderia
  - numero\_piloto\_ganador en la tabla temporada referenciando la clave de piloto (al ser una entidad débil de trabajador, la tabla temporada coge también la clave de trabajador)
  - temporada en la tabla carrera referenciando la temporada
  - nombre\_circuito en la tabla carrera referenciando la clave de circuito
- Las claves foráneas que también son primarias (DNI\_pilotos en la tabla pilotos, temporada en la tabla carreras, numero\_piloto y numero\_carrera en la tabla resultado).

## 4. Carga de Datos

Para la carga de datos he utilizado varias páginas. La página fundamental y desde la que he basado mis datos del proyecto ha sido Kaggle, donde encontré un archivo que contenía todos los pilotos, escuderías, resultados, ... de la Fórmula 1 hasta el año 2017. También he utilizado la página Generate Data para que me diera números aleatorios y así utilizarlos para rellenar los DNIs de los trabajadores, sin necesidad de tener que añadir 1 a 1 (este proceso lo tuve que repetir dos veces ya que Generate Data sólo te permite exportar 100 archivos y mi tabla trabajadores tiene 165). Modificando los datos obtenidos tanto del archivo de Kaggle y de Generate Data con la herramienta Excel (además de añadir a mano todos los datos relacionados con la temporada 2021) se culminaron los datos de mis tablas.

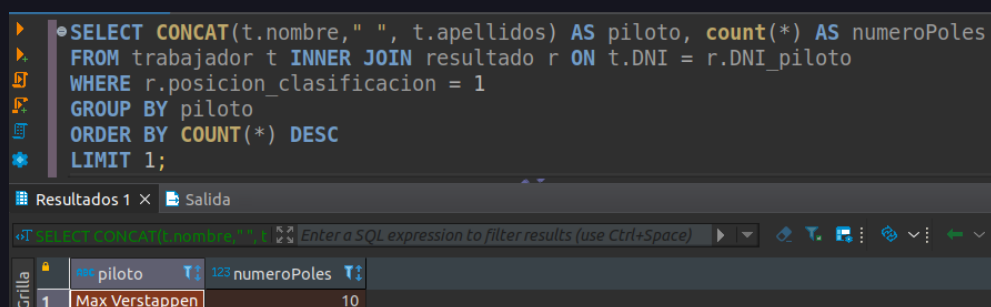
A la hora de realizar la carga masiva (importarlos en mi base de datos) lo hice de la siguiente manera:

1. Descargué todos los archivos que tenía en Excel con los datos con la terminación .csv
2. Me fui a mi proyecto en dbeaver y pulsé botón derecho encima de la tabla a la que quería añadirle los datos
3. Seleccioné en el desplegable la opción de "import data" y seleccioné el archivo .csv relacionado con esa tabla

## 5. Consultas

### 1. Piloto con más poles (posición 1 en la clasificación).

```
SELECT CONCAT(t.nombre," ", t.apellidos) AS piloto, count(*) AS numeroPoles
FROM trabajador t INNER JOIN resultado r ON t.DNI = r.DNI_piloto
WHERE r.posicion_clasificacion = 1
GROUP BY piloto
ORDER BY COUNT(*) DESC
LIMIT 1;
```

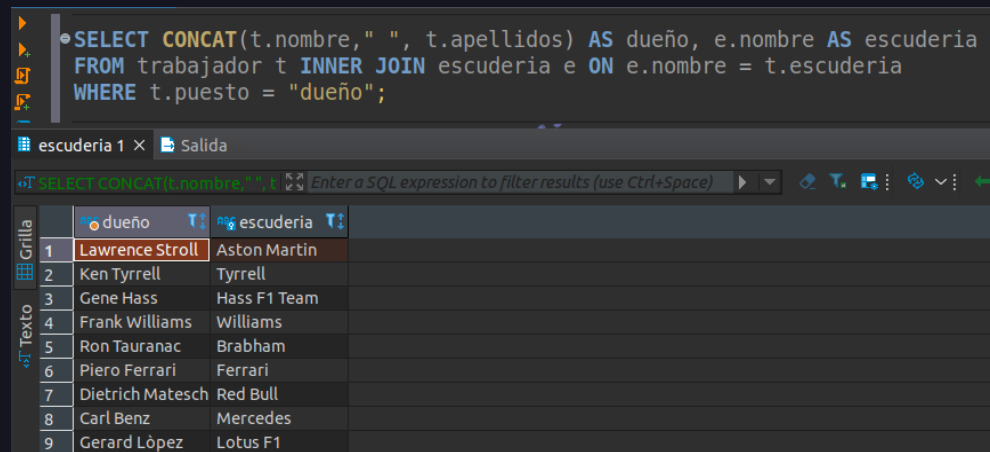


The screenshot shows a SQL IDE interface. The top pane contains the SQL query: `SELECT CONCAT(t.nombre," ", t.apellidos) AS piloto, count(*) AS numeroPoles FROM trabajador t INNER JOIN resultado r ON t.DNI = r.DNI_piloto WHERE r.posicion_clasificacion = 1 GROUP BY piloto ORDER BY COUNT(*) DESC LIMIT 1;`. The bottom pane shows the results in a table with two columns: 'piloto' and 'numeroPoles'. The first row shows 'Max Verstappen' with 10 poles.

	piloto	numeroPoles
1	Max Verstappen	10

## 2. Dueños de las escuderías.

```
SELECT CONCAT(t.nombre," ", t.apellidos) AS dueño, e.nombre AS escuderia
FROM trabajador t INNER JOIN escuderia e ON e.nombre = t.escuderia
WHERE t.puesto = "dueño";
```

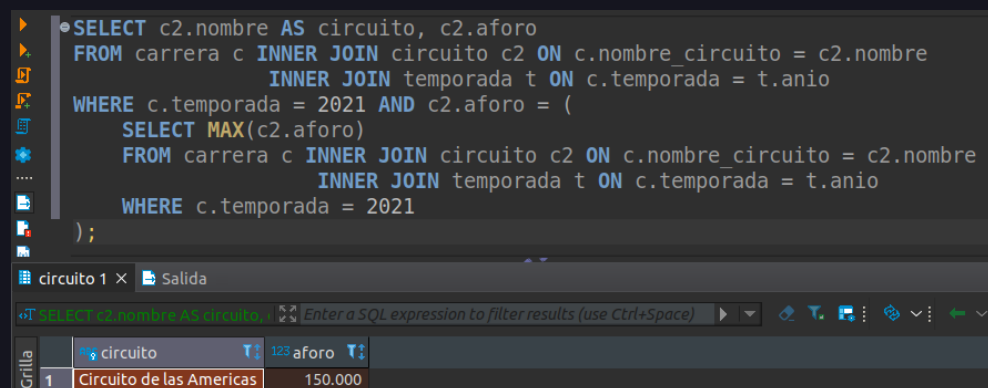


The screenshot shows a SQL IDE with a query editor at the top containing the SQL query for finding team owners. Below the editor, the results are displayed in a grid view with columns 'dueño' and 'escuderia'.

	dueño	escuderia
1	Lawrence Stroll	Aston Martin
2	Ken Tyrrell	Tyrrell
3	Gene Hass	Hass F1 Team
4	Frank Williams	Williams
5	Ron Tauranac	Brabham
6	Piero Ferrari	Ferrari
7	Dietrich Matesch	Red Bull
8	Carl Benz	Mercedes
9	Gerard López	Lotus F1

## 3. Circuito con más aforo en 2021

```
SELECT c2.nombre AS circuito, c2.aforo
FROM carrera c INNER JOIN circuito c2 ON c.nombre_circuito = c2.nombre
INNER JOIN temporada t ON c.temporada = t.anio
WHERE c.temporada = 2021 AND c2.aforo = (
    SELECT MAX(c2.aforo)
    FROM carrera c INNER JOIN circuito c2
    ON c.nombre_circuito = c2.nombre
    INNER JOIN temporada t ON c.temporada = t.anio
    WHERE c.temporada = 2021
);
```

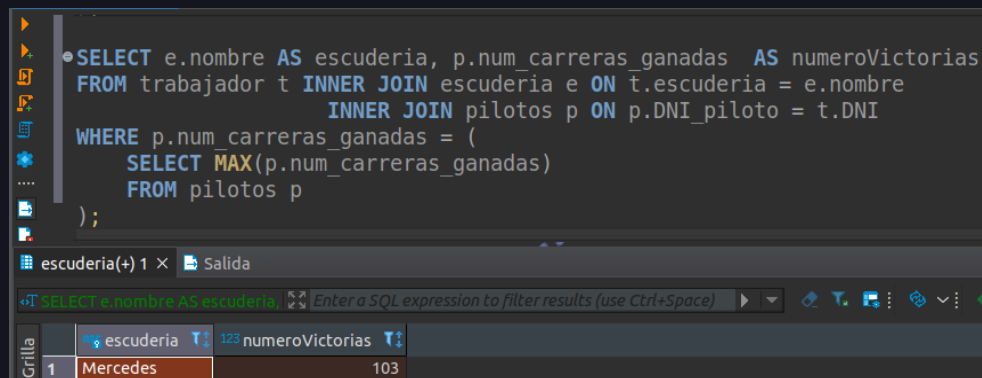


The screenshot shows a SQL IDE with a query editor at the top containing the SQL query for finding the circuit with the highest capacity in 2021. Below the editor, the results are displayed in a grid view with columns 'circuito' and 'aforo'.

	circuito	aforo
1	Circuito de las Americas	150.000

#### 4. Escudería con más victorias en 2021

```
SELECT e.nombre AS escuderia,  
       p.num_carreras_ganadas AS numeroVictorias  
FROM trabajador t INNER JOIN escuderia e ON t.escuderia = e.nombre  
              INNER JOIN pilotos p ON p.DNI_piloto = t.DNI  
WHERE p.num_carreras_ganadas = (  
       SELECT MAX(p.num_carreras_ganadas)  
       FROM pilotos p  
);
```

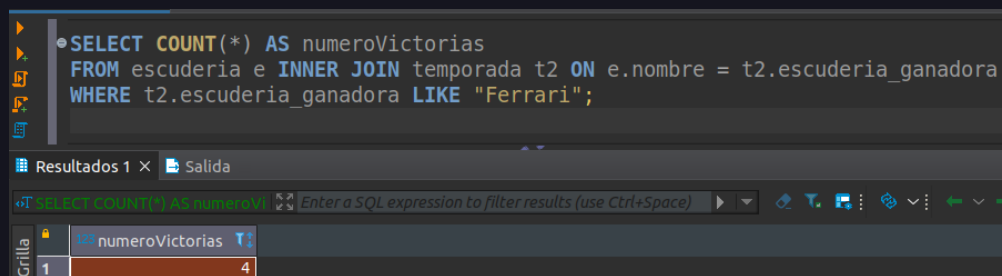


```
SELECT e.nombre AS escuderia, p.num_carreras_ganadas AS numeroVictorias  
FROM trabajador t INNER JOIN escuderia e ON t.escuderia = e.nombre  
              INNER JOIN pilotos p ON p.DNI_piloto = t.DNI  
WHERE p.num_carreras_ganadas = (  
       SELECT MAX(p.num_carreras_ganadas)  
       FROM pilotos p  
);
```

escuderia	numeroVictorias
Mercedes	103

#### 5. Número de temporadas ganadas por Ferrari

```
SELECT COUNT(*) AS numeroVictorias  
FROM escuderia e INNER JOIN temporada t2 ON e.nombre = t2.escuderia_ganadora  
WHERE t2.escuderia_ganadora LIKE "Ferrari";
```

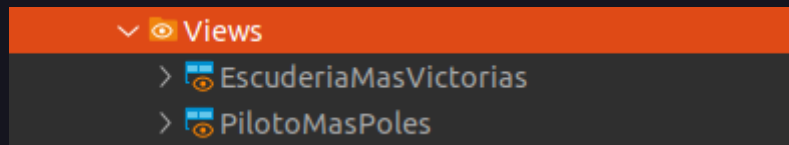


```
SELECT COUNT(*) AS numeroVictorias  
FROM escuderia e INNER JOIN temporada t2 ON e.nombre = t2.escuderia_ganadora  
WHERE t2.escuderia_ganadora LIKE "Ferrari";
```

numeroVictorias
4

## 6. Vistas

```
CREATE VIEW EscuderiaMasVictorias AS (  
  SELECT e.nombre AS escuderia, p.num_carreras_ganadas AS numeroVictorias  
  FROM trabajador t INNER JOIN escuderia e ON t.escuderia = e.nombre  
                                INNER JOIN pilotos p ON p.DNI_piloto = t.DNI  
  WHERE p.num_carreras_ganadas = (  
    SELECT MAX(p.num_carreras_ganadas)  
    FROM pilotos p)  
);  
  
CREATE VIEW PilotoMasPoles AS (  
  SELECT CONCAT(t.nombre, " ", t.apellidos) AS piloto,  
         count(*) AS numeroPoles  
  FROM trabajador t INNER JOIN resultado r ON t.DNI = r.DNI_piloto  
  WHERE r.posicion_clasificacion = 1  
  GROUP BY piloto  
  ORDER BY COUNT(*) DESC  
  LIMIT 1  
);
```



## 7. Funciones



### 1. Circuito con más aforo en un año dado

```
DELIMITER $$
CREATE FUNCTION mas_aforo(temporada INT) RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE resultado VARCHAR(100);

    SELECT MAX(c.aforo) INTO resultado
    FROM circuito c INNER JOIN carrera c1 ON c.nombre = c1.nombre_circuito
    WHERE c1.temporada = temporada;

    RETURN resultado ;
END $$
DELIMITER ;

SELECT mas_aforo(2021);
```

	ABC mas_aforo(2021) 
1	150000



### 2. Muestra el número de poles de un piloto recibiendo su número

```
DELIMITER $$
CREATE FUNCTION num_poles(piloto INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE resultado INT;

    SELECT COUNT(*) INTO resultado
    FROM pilotos p INNER JOIN resultado r ON p.numero = r.numero_piloto
    WHERE r.posicion_clasificacion = 1 AND r.numero_piloto = piloto
    GROUP BY r.numero_piloto ;

    RETURN resultado ;
END $$
DELIMITER ;

SELECT num_poles(44);
```

	123 num_poles(44) 
1	4



## 8. Procedimientos

### 1. Histórico de pilotos que han corrido para una escudería dada

```
DELIMITER $$
CREATE PROCEDURE historico_corredores(nomEscuderia VARCHAR(20))
BEGIN
    SELECT CONCAT(t.nombre, " ", t.apellidos) AS piloto
    FROM trabajador t
    WHERE t.escuderia = nomEscuderia;
END $$
DELIMITER ;

CALL historico_corredores ('Hass F1 Team');
```

	asc piloto
1	Mick Schumacher
2	Kevin Magnussen
3	Nikita Mazepin
4	Gene Hass
5	Günther Steiner

### 2. Mostrar el número de poles de un piloto (dado su número) en una temporada dada (utilizando la función num\_poles) con el siguiente formato:

“piloto” ha conseguido XXXX poles en la temporada XXXX

```
DELIMITER $$
CREATE PROCEDURE poles_piloto(piloto INT)
BEGIN
    DECLARE resultado VARCHAR(100) DEFAULT "";

    SELECT CONCAT("EL PILOTO CON NÚMERO ", piloto, " HA HECHO ", num_poles
    (piloto), " POLES");

END $$
DELIMITER ;
```

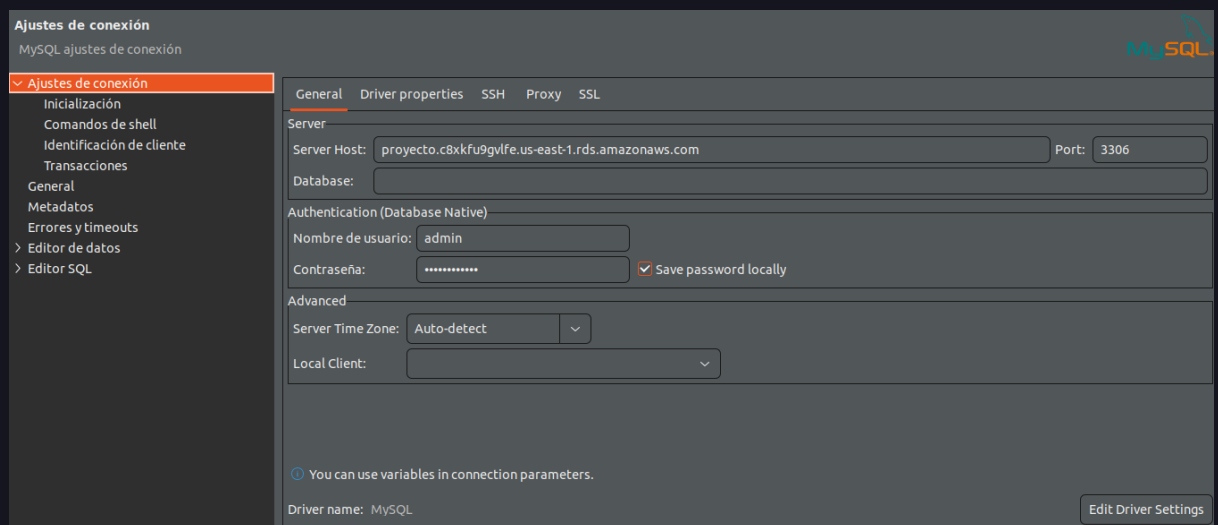
1	EL PILOTO CON NÚMERO 37 HA HECHO 3 POLES
---	--

## 9. Conexión con AWS

Para realizar la conexión con AWS, lo primero que tenemos que hacer es meternos en el laboratorio de Learner Lab para que así se nos quede guardada la instancia RDS que necesitamos.

A continuación, creamos con RDS, una base de datos MySQL que nos permitirá conectarla con nuestro gestor de bases de datos local (en nuestro caso DBeaver).

Una vez que tenemos la base de datos creada, la conectamos a nuestro gestor de base de datos copiando el enlace (EndPoint) y el puerto en dicho gestor, como en la imagen siguiente:



Una vez que lo tengamos conectado, ya solo tendremos que ejecutar el script de nuestro proyecto y listo.

## 10. Enlace GitHub

[https://github.com/manferser201/Base\\_de\\_Datos](https://github.com/manferser201/Base_de_Datos)

## 11. Conclusión

Este proyecto me ha ayudado a ver todas las dificultades que entraña el diseñar una base de datos, pero sobre todo, a comprender la complejidad que tiene el modificar una tabla o el añadir datos procurando que nada se dañe o de error.

Al haber tenido tantas dificultades al añadir los datos, me ha ayudado a saber utilizar mejor herramientas como Excel para poder modificar los datos a mi gusto.

Creo que es una experiencia muy positiva que ayuda a comprender mejor los conocimientos obtenidos de una forma más dinámica.