



DEPARTAMENTO DE COMPUTACIÓN  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
UNIVERSIDAD DE CHILE  
CC3201 BASES DE DATOS

---

## HITO N° 3 - GRUPO N° 21 - SECCIÓN 1

---

# LA 33.

---

Integrantes:	Juan I. Valdivia Constanza Vicencio Montserrat Montero Patricio Espinoza
Profesor:	Matías Toro
Auxiliares:	Claudio Gutiérrez Cristián Salazar Daniel Radrigán Fran A. Zautzik Scarlett Plaza
Ayudantes:	Daniela Assael Diego Reyes Joel Riquelme Kathleen Kohler Nahuel Gómez Natalia Quinteros Paula Ovalle Sebastián Contreras

Fecha de entrega: 4 de Diciembre del 2023  
Santiago, Chile

# Índice de Contenidos

<b>1. Hito 0: Problema y fuente de datos.</b>	<b>1</b>
1.1. Propuesta Aceptada. . . . .	1
1.2. Bases de datos del proyecto. . . . .	1
1.3. Cambios tras retroalimentación. . . . .	1
<b>2. Hito 1: Modelo Entidad Relación.</b>	<b>2</b>
2.1. Modelo de Datos Entidad Relación. . . . .	2
2.2. Modelo Relacional. . . . .	3
2.3. Cambios tras retroalimentación. . . . .	4
<b>3. Hito 2: Implementación y carga de datos</b>	<b>5</b>
3.1. Carga de Datos. . . . .	5
3.1.1. Creación de Esquema y Tablas en SQL. . . . .	5
3.1.2. Evidencia de correcta carga de datos. . . . .	7
3.2. Cambios tras retroalimentación . . . . .	9
<b>4. Hito 3: Aplicación Web.</b>	<b>10</b>
4.1. Consultas Complejas . . . . .	10
4.1.1. Primera Consulta . . . . .	10
4.1.2. Segunda Consulta . . . . .	10
4.1.3. Tercera Consulta . . . . .	10
4.2. Optimización de las consultas . . . . .	11
4.2.1. Primera Consulta . . . . .	11
4.2.2. Segunda Consulta . . . . .	12
4.2.3. Tercera Consulta . . . . .	14
4.3. Medidas de Seguridad contra inyecciones SQL . . . . .	16
4.3.1. Input automático . . . . .	16
4.3.2. Usar sentencias y consultas parametrizadas . . . . .	16
4.4. Aplicación web . . . . .	17
4.4.1. Interfaz . . . . .	17
4.5. Resultados Consultas . . . . .	17
4.5.1. Consulta 1: Input Pais = Italy . . . . .	17
4.5.2. Consulta 2: Input Escuderia = Ferrari, Temporada = 2019 . . . . .	18
4.5.3. Consulta 3: Input Escuderia = Renault . . . . .	18
4.6. Conclusiones del Proyecto . . . . .	19

# 1. Hito 0: Problema y fuente de datos.

## 1.1. Propuesta Aceptada.

Desarrollar una aplicación web con la cual poder consultar acerca de datos históricos de la F1 entre los años 1985 y 2020, tales como el nivel de rendimiento de un piloto y/o escudería en cierto circuito o temporada, considerando factores como el clima, o victorias y datos de pilotos históricos de una escudería, país, edad, etc. en concreto.

## 1.2. Bases de datos del proyecto.

El proyecto utiliza las siguientes bases de datos:

- <https://www.kaggle.com/datasets/petalme/f1-drivers-dataset>
- <https://www.kaggle.com/datasets/jtrotman/formula-1-race-events>
- <https://www.kaggle.com/datasets/aadiltajani/fia-f1-19502019-data>
- <https://www.kaggle.com/datasets/prajwalsood/f1-race-by-race-19832021>

## 1.3. Cambios tras retroalimentación.

En un principio, el grupo quería utilizar la base de datos con las respuestas de la 10ma encuesta nacional de juventudes de la INJUV, para sesarrollar una app web que permita realizar consultas óptimas acerca de la relación entre problemas de vivienda, ciudad, región, educación entre otras entidades importantes a la hora de hablar acerca la situación de jóvenes de 15 a 29 años.

Sin embargo, se señaló lo complicado que sería generar un modelo entidad relación complejo con esta base de datos, por lo que se decidió cambiar el proyecto a la propuesta actual.

## 2. Hito 1: Modelo Entidad Relación.

### 2.1. Modelo de Datos Entidad Relación.

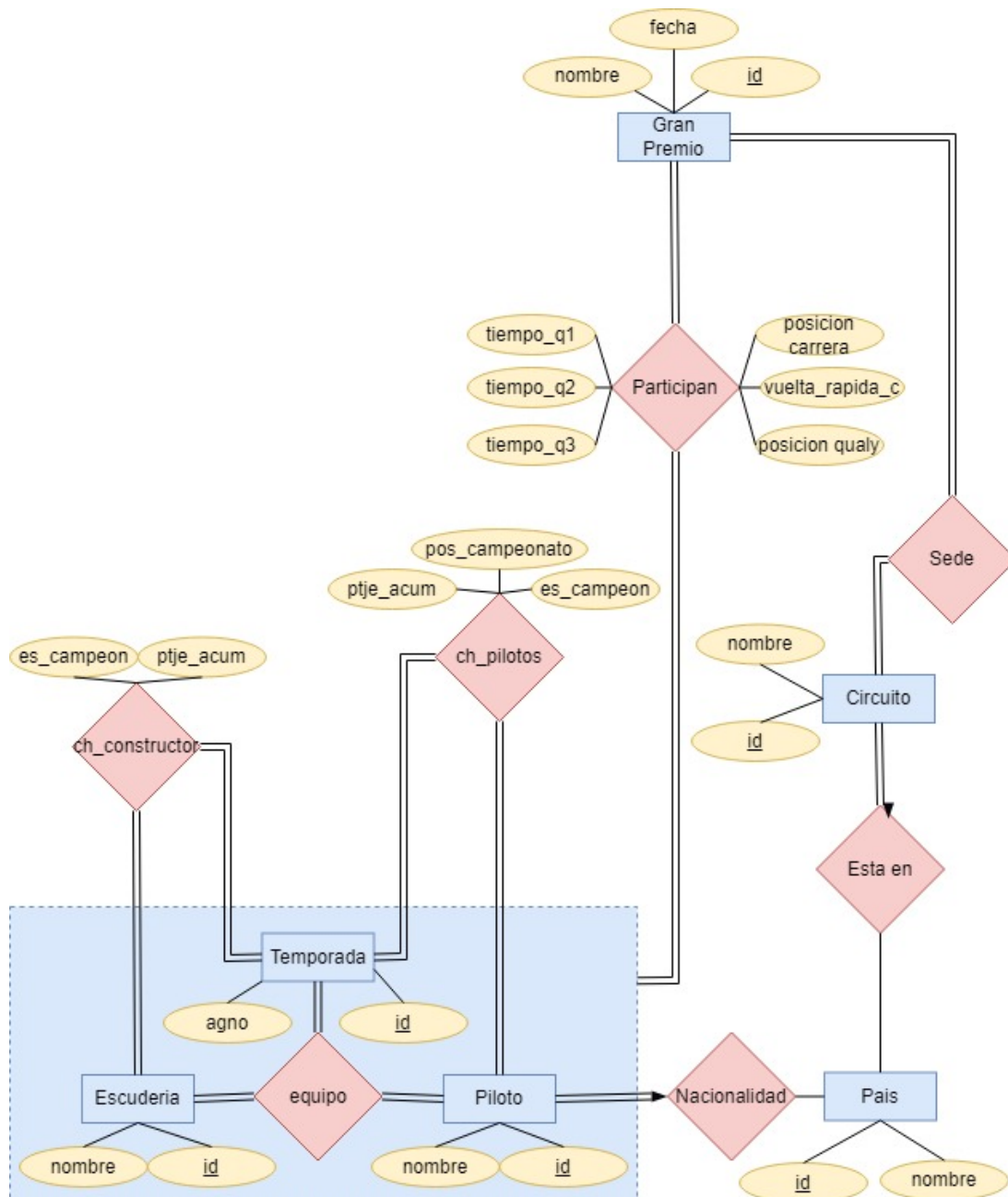


Figura 1: Modelo Relacional Final del Proyecto.

## 2.2. Modelo Relacional.

```
1 Temporada(id: Serial , agno: Int)
2
3 Circuito(id: Serial , nombre: Str)
4
5 Escuderia(id: Serial , nombre:String)
6
7 Piloto(id: Serial , nombre:String)
8
9 Pais(id: Serial , nombre:String)
10
11 Gran Premio(id: Serial , nombre: Str , fecha: Date)
12
13 Equipo(Es_id: Serial , Pi_id: Serial , T_id)
14     Es_id REF Escuderia(id)
15     Pi_id REF Piloto(id)
16     T_id REF Temporada(id)
17
18 Nacionalidad(Pi_id: Serial , Pa_id:Serial)
19     Pi_id REF Piloto(id)
20     Pa_id REF Pais(id)
21
22 EstaEn(Cir_id: Serial , Pa_id:Serial)
23     Cir_id REF Circuito(id)
24     Pa_id REF Pais(id)
25
26 Sede(Gp_id: Serial , Cir_id: Serial)
27     Gp_id REF GranPremio(id)
28     Cir_id REF Circuito(id)
29
30 Participan(EqE_id: Serial , EqP_id: Serial , EqT_id: Serial , Gp_id: Serial ,
31 posicion_carrera: Int , vuelta_rapida_c: Time, posicion_qualy: Int ,
32 tiempo_q1: Time, tiempo_q2: Time, tiempo_q3: Time)
33     (EqE_id, EqP_id, EqT_id) REF Equipo(Es_id, Pi_id, T_id)
34     Gp_id REF GranPremio(id)
35
36 Ch_Constructor(Es_id: Serial ,T_id:Serial , ptje_acumulado: Int , es_campeon:Bool)
37     T_id REF Temporada(id)
38     Es_id REF Escuderia(id)
39
40 Ch_Pilotos(T_id:Serial , Pi_id: Serial , ptje_acumulado:Int , es_campeon:Bool ,
41 pos_campeonato: Int)
42     Pi_id REF Piloto(id)
43     T_id REF Temporada(id)
```

### 2.3. Cambios tras retroalimentación.

En la retroalimentación del Hito 1, se señaló que no tenía sentido tener una entidad virtual entre Escudería y Piloto para hacer que participen en un Gran Premio, pues en ese entonces se había diseñado a Piloto como una entidad débil de Escudería.

Si bien, el modelo posteriormente fue modificado una gran cantidad de veces, se prefirió evitar las entidades débiles para hacer más sencilla la implementación de las tablas durante el Hito 2. De manera que, el equipo decidió que lo más conveniente era hacer una entidad virtual entre Temporada, Piloto y Escudería que caracterice a un determinado equipo que participe en un Gran Premio, además de agregar id's propias a cada elemento de las entidades.

Además, desde la entrega del Hito 1 se añadió la entidad País y Circuito, permitiendo hacer mejores consultas sobre estos datos.

## 3. Hito 2: Implementación y carga de datos

### 3.1. Carga de Datos.

La base de datos del proyecto fue cargada en una maquina virtual otorgado por el equipo docente. El motor de bases de datos utilizado fue PostgreSQL.

#### 3.1.1. Creación de Esquema y Tablas en SQL.

Las sentencias SQL para crear el Esquema son:

```
1 CREATE SCHEMA F1
2 ALTER USER cc3201 SET search_path = F1;
```

Por su parte, las sentencias correspondientes a las tablas de las entidades del proyecto son:

```
1 CREATE TABLE Temporada (
2     id serial PRIMARY KEY,
3     agno INT
4 );
5 CREATE TABLE Circuito (
6     id serial PRIMARY KEY,
7     nombre VARCHAR(255)
8 );
9 CREATE TABLE Escuderia (
10    id serial PRIMARY KEY,
11    nombre VARCHAR(255)
12 );
13 CREATE TABLE Piloto (
14    id serial PRIMARY KEY,
15    nombre VARCHAR(255)
16 );
17 CREATE TABLE Pais (
18    id serial PRIMARY KEY,
19    nombre VARCHAR(255)
20 );
21 CREATE TABLE GranPremio (
22    id serial PRIMARY KEY,
23    nombre VARCHAR(255),
24    fecha DATE
25 );
```

Finalmente, las sentencias que crean las tablas de las relaciones del proyecto son:

```
1 CREATE TABLE Equipo (
2     Es_id bigint not null ,
3     Pi_id bigint not null ,
4     T_id bigint not null ,
5     PRIMARY KEY (Es_id, Pi_id, T_id),
6     FOREIGN KEY (Es_id) REFERENCES Escuderia(id),
7     FOREIGN KEY (Pi_id) REFERENCES Piloto(id),
8     Foreign Key (T_id) REFERENCES Temporada(id)
```

```
9 );
10 CREATE TABLE Piloto_Pais (
11     Pi_id bigint not null,
12     Pa_id bigint not null,
13     PRIMARY KEY (Pi_id, Pa_id),
14     FOREIGN KEY (Pi_id) REFERENCES Piloto(id),
15     FOREIGN KEY (Pa_id) REFERENCES Pais(id)
16 );
17 CREATE TABLE Temporada_Escuderia (
18     Es_id bigint not null,
19     T_id bigint not null,
20     ptje_acumulado FLOAT,
21     es_campeon BOOLEAN,
22     PRIMARY KEY (Es_id, T_id),
23     FOREIGN KEY (Es_id) REFERENCES Escuderia(id),
24     FOREIGN KEY (T_id) REFERENCES Temporada(id)
25 );
26 CREATE TABLE Temporada_Piloto (
27     T_id bigint not null,
28     Pi_id bigint not null,
29     ptje_acumulado FLOAT,
30     es_campeon BOOLEAN,
31     posicion_campeonato INT,
32     PRIMARY KEY (T_id, Pi_id),
33     FOREIGN KEY (Pi_id) REFERENCES Piloto(id),
34     FOREIGN KEY (T_id) REFERENCES Temporada(id)
35 );
36 CREATE TABLE Equipo_GranPremio (
37     EqEs_id bigint not null,
38     EqPi_id bigint not null,
39     EqT_id bigint not null,
40     Gp_id bigint not null,
41     posicion_carrera INT,
42     vuelta_rapida_c Time,
43     posicion_qualy INT,
44     tiempo_qualy_q1 Time,
45     tiempo_qualy_q2 Time,
46     tiempo_qualy_q3 Time,
47     PRIMARY KEY (EqEs_id, EqPi_id, Gp_id, EqT_id),
48     FOREIGN KEY (EqEs_id, EqPi_id, EqT_id) REFERENCES Equipo(Es_id, Pi_id, T_id),
49     FOREIGN KEY (Gp_id) REFERENCES GranPremio(id)
50 );
51 CREATE TABLE GranPremio_Circuito (
52     Gp_id bigint not null,
53     Cir_id bigint not null,
54     PRIMARY KEY (Gp_id, Cir_id),
55     FOREIGN KEY (Cir_id) REFERENCES Circuito(id),
56     FOREIGN KEY (Gp_id) REFERENCES GranPremio(id)
57 );
58 CREATE TABLE Circuito_Pais (
59     Cir_id bigint not null,
60     Pa_id bigint not null,
```



```

61 PRIMARY KEY (Cir_id, Pa_id),
62 FOREIGN KEY (Cir_id) REFERENCES Circuito(id),
63 FOREIGN KEY (Pa_id) REFERENCES Pais(id)
64 );

```

### 3.1.2. Evidencia de correcta carga de datos.

A continuación, se adjuntan varias capturas de querys que demuestran la correcta carga de cada tabla del modelo del proyecto, tanto para entidades como relaciones.

```

cc3201=# select * from pais limit 10;
 id | nombre
-----+-----
  1 | United Kingdom
  2 | Italy
  3 | Belgium
  4 | Netherlands
  5 | Thailand
  6 | France
  7 | Spain
  8 | United States
  9 | Brazil
 10 | Germany
(10 rows)

```

(a) País

```

cc3201=# select * from escuderia limit 10;
 id | nombre
-----+-----
  1 | Alfa Romeo
  2 | Talbot-Lago
  3 | Ferrari
  4 | Kurtis Kraft Offenhauser
  5 | Deidt Offenhauser
  6 | Maserati
  7 | Simca-Gordini
  8 | null
  9 | Maserati Milano
 10 | BRM
(10 rows)

```

(b) Ecudería

```

cc3201=# select * from piloto limit 10;
 id | nombre
-----+-----
  1 | Kenny Acheson
  2 | Andrea de Adamich
  3 | Philippe Adams
  4 | Jack Aitken
  5 | Christijan Albers
  6 | Alexander Albon
  7 | Michele Alboreto
  8 | Jean Alesi
  9 | Jaime Alguersuari
 10 | Philippe Alliot
(10 rows)

```

(c) Piloto

```

cc3201=# select * from granpremio limit 10;
 id | nombre | fecha
-----+-----+-----
  1 | Australian Grand Prix | 2009-03-29
  2 | Malaysian Grand Prix | 2009-04-05
  3 | Chinese Grand Prix | 2009-04-19
  4 | Bahrain Grand Prix | 2009-04-26
  5 | Spanish Grand Prix | 2009-05-10
  6 | Monaco Grand Prix | 2009-05-24
  7 | Turkish Grand Prix | 2009-06-07
  8 | British Grand Prix | 2009-06-21
  9 | German Grand Prix | 2009-07-12
 10 | Hungarian Grand Prix | 2009-07-26
(10 rows)

```

(d) Gran Premio

```

cc3201=# select * from temporada limit 10;
 id | agno
-----+-----
  1 | 1950
  2 | 1951
  3 | 1952
  4 | 1953
  5 | 1954
  6 | 1955
  7 | 1956
  8 | 1957
  9 | 1958
 10 | 1959
(10 rows)

```

(e) Temporada

```

cc3201=# select * from circuito limit 10;
 id | nombre
-----+-----
  1 | Albert Park Grand Prix Circuit
  2 | Sepang International Circuit
  3 | Bahrain International Circuit
  4 | Circuit de Barcelona-Catalunya
  5 | Istanbul Park
  6 | Circuit de Monaco
  7 | Circuit Gilles Villeneuve
  8 | Circuit de Nevers Magny-Cours
  9 | Silverstone Circuit
 10 | Hockenheimring
(10 rows)

```

(f) Circuito

Figura 2: Evidencia de Correcta Carga de Datos de Entidades.

```
cc3201=# select * from equipo limit 10;
es_id | pi_id | t_id
-----+-----+-----
1 | 204 | 1
1 | 203 | 1
1 | 201 | 1
2 | 637 | 1
3 | 829 | 1
4 | 554 | 1
5 | 335 | 1
6 | 830 | 1
6 | 552 | 1
6 | 129 | 1
(10 rows)
```

(a) Equipo

```
cc3201=# select * from piloto_pais limit 10;
pi_id | pa_id
-----+-----
1 | 1
2 | 2
3 | 3
4 | 1
5 | 4
6 | 5
7 | 2
8 | 6
9 | 7
10 | 6
(10 rows)
```

(b) Piloto\_Pais

```
cc3201=# select * from temporada_escuderia limit 10;
es_id | t_id | ptje_acumulado | es_campeon
-----+-----+-----+-----
1 | 1 | 81 | t
6 | 1 | 13 | f
3 | 1 | 15 | f
2 | 1 | 19 | f
7 | 1 | 3 | f
9 | 1 | 2 | f
5 | 1 | 11 | f
4 | 1 | 13 | f
8 | 1 | 7 | f
3 | 2 | 74 | t
(10 rows)
```

(c) Temporada\_Escudería

```
cc3201=# select * from temporada_piloto limit 10;
t_id | pi_id | ptje_acumulado | es_campeon | posicion_campeonato
-----+-----+-----+-----+-----
1 | 204 | 30 | t | 1
1 | 203 | 27 | f | 2
1 | 201 | 24 | f | 3
1 | 637 | 13 | f | 4
1 | 829 | 11 | f | 5
1 | 554 | 9 | f | 6
1 | 335 | 6 | f | 7
1 | 830 | 5 | f | 8
1 | 552 | 4 | f | 9
1 | 129 | 4 | f | 9
(10 rows)
```

(d) Temporada\_Piloto

```
cc3201=# select * from granpremio_circuito limit 10;
gp_id | cir_id
-----+-----
1 | 1
2 | 2
3 | 17
4 | 3
5 | 4
6 | 6
7 | 5
8 | 9
9 | 20
10 | 11
(10 rows)
```

(e) GranPremio\_Circuito

```
cc3201=# select * from circuito_pais limit 10;
cir_id | pa_id
-----+-----
1 | 21
2 | 45
3 | 48
4 | 7
5 | 49
6 | 14
7 | 15
8 | 6
9 | 50
10 | 10
(10 rows)
```

(f) Circuito\_País

```
cc3201=# select * from equipo_granpremio where eqt_id = 55 limit 10;
eqes_id | eqpi_id | eqt_id | gp_id | posicion_carrera | vuelta_rapida_c | posicion_qualy | tiempo_qualy_q1 | tiempo_qualy_q2 | tiempo_qualy_q3
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
3 | 22 | 55 | 104 | 1 | 00:01:21.046 | 1 | 00:01:20.089 | 00:00:00 | 00:00:00
145 | 501 | 55 | 104 | 5 | 00:01:22.929 | 2 | 00:01:20.62 | 00:00:00 | 00:00:00
3 | 675 | 55 | 104 | 2 | 00:01:21.361 | 3 | 00:01:20.637 | 00:00:00 | 00:00:00
73 | 11 | 55 | 104 | 16 | 00:01:22.881 | 4 | 00:01:20.645 | 00:00:00 | 00:00:00
146 | 658 | 55 | 104 | 4 | 00:01:22.66 | 5 | 00:01:20.715 | 00:00:00 | 00:00:00
146 | 100 | 55 | 104 | 3 | 00:01:22.671 | 6 | 00:01:20.786 | 00:00:00 | 00:00:00
130 | 600 | 55 | 104 | 19 | 00:01:23.365 | 7 | 00:01:20.877 | 00:00:00 | 00:00:00
145 | 582 | 55 | 104 | 7 | 00:01:22.246 | 8 | 00:01:20.888 | 00:00:00 | 00:00:00
154 | 762 | 55 | 104 | 10 | 00:01:22.855 | 9 | 00:01:21.027 | 00:00:00 | 00:00:00
130 | 144 | 55 | 104 | 6 | 00:01:22.889 | 10 | 00:01:21.049 | 00:00:00 | 00:00:00
(10 rows)
```

(g) Equipo\_GranPremio

Figura 3: Evidencia de Correcta Carga de Datos de Relaciones.

Notar que en el caso de la tabla “Equipo\_GranPremio”, las dos columnas: “tiempo\_qualy\_q2” y “tiempo\_qualy\_q3”; poseen el valor “00:00:0” porque aquellas tuplas corresponden a resultados por piloto de los Gran Premios ocurridos el año 2004, año en el que solo había Q1 en las Calificaciones.

```
cc3201=# select count(*) as Cantidad_de_Tuplas from equipo_granpremio;  
cantidad_de_tuplas  
-----  
                24869  
(1 row)
```

(a) Equipo\_GranPremio

```
cc3201=# select count(*) as Cantidad_de_Tuplas from equipo;  
cantidad_de_tuplas  
-----  
                3172  
(1 row)
```

(b) Equipo

Figura 4: Ejemplos de la Cantidad de Tuplas en la Base de Datos.

### 3.2. Cambios tras retroalimentación

En la retroalimentación del Hito 2, al grupo se le señaló principalmente el desorden y la poca efectividad de las capturas tomadas para evidenciar la correcta carga de datos. Error corregido, en este informe.

## 4. Hito 3: Aplicación Web.

### 4.1. Consultas Complejas

#### 4.1.1. Primera Consulta

```
1 SELECT nombre
2 FROM (SELECT cir_id
3        FROM f1_2.circuito_pais
4        WHERE pa_id = %s) AS cirid
5 JOIN f1_2.circuito ON cirid.cir_id = id
```

#### 4.1.2. Segunda Consulta

```
1 SELECT pi.nombre AS piloto , grpr.nombre AS grpremio
2 FROM f1_2.equipo_granpremio JOIN f1_2.piloto AS pi ON eqpi_id = pi.id
3 JOIN f1_2.escuderia AS esc ON eqes_id = esc.id
4 JOIN f1_2.temporada AS temp ON eqt_id = temp.id
5 JOIN f1_2.granpremio AS grpr ON gp_id = grpr.id
6 WHERE posicion_carrera = 1 AND esc.nombre = %s1 AND temp.agno = %s2;
```

#### 4.1.3. Tercera Consulta

```
1 "SELECT result.pm, result.nombre, result.agno
2 FROM
3     (SELECT eqgp.pm, esc.nombre, agno
4      FROM
5          (SELECT AVG(vuelta_rapida_c) AS pm, eqes_id, eqt_id
6           FROM f1_2.equipo_granpremio JOIN f1_2.granpremio ON gp_id = id
7           GROUP BY eqes_id, eqt_id) AS eqgp
8          , f1_2.temporada AS temp
9          , f1_2.escuderia as esc
10         WHERE eqgp.eqt_id = temp.id and eqgp.eqes_id = esc.id) AS result
11 WHERE result.nombre = %s
12 ORDER BY result.agno DESC";
```

## 4.2. Optimización de las consultas

Para esta consulta se utilizó un índice de Hash ya que permite que la consulta anidada se ejecute más rápido.

### 4.2.1. Primera Consulta

```
cc3201=# EXPLAIN ANALYZE SELECT nombre
cc3201=# FROM (SELECT cir_id
cc3201=# FROM f1_2.circuito_pais
cc3201=# WHERE pa_id = 2) AS cirid
cc3201=# JOIN f1_2.circuito ON cirid.cir_id = id;
QUERY PLAN
-----
Hash Join (cost=2.01..4.00 rows=4 width=21) (actual time=0.065..0.165 rows=4 loops=1)
  Hash Cond: (circuito.id = circuito_pais.cir_id)
  -> Seq Scan on circuito (cost=0.00..1.77 rows=77 width=25) (actual time=0.009..0.061 rows=77 loops=1)
  -> Hash (cost=1.96..1.96 rows=4 width=8) (actual time=0.026..0.028 rows=4 loops=1)
        Buckets: 1024 Batches: 1 Memory Usage: 9kB
        -> Seq Scan on circuito_pais (cost=0.00..1.96 rows=4 width=8) (actual time=0.009..0.018 rows=4 loops=1)
              Filter: (pa_id = 2)
              Rows Removed by Filter: 73
Planning Time: 0.442 ms
Execution Time: 0.194 ms
(10 rows)
```

Figura 5: Consulta pre-optimizacion

```
cc3201=# CREATE INDEX paid ON f1_2.circuito_pais USING hash (pa_id);
CREATE INDEX
```

Figura 6: Creación de Índice

```
cc3201=# EXPLAIN ANALYZE SELECT nombre
cc3201=# FROM (SELECT cir_id
cc3201=# FROM f1_2.circuito_pais
cc3201=# WHERE pa_id = 2) AS cirid
cc3201=# JOIN f1_2.circuito ON cirid.cir_id = id;
QUERY PLAN
-----
Hash Join (cost=2.01..4.00 rows=4 width=21) (actual time=0.065..0.166 rows=4 loops=1)
  Hash Cond: (circuito.id = circuito_pais.cir_id)
  -> Seq Scan on circuito (cost=0.00..1.77 rows=77 width=25) (actual time=0.009..0.064 rows=77 loops=1)
  -> Hash (cost=1.96..1.96 rows=4 width=8) (actual time=0.026..0.028 rows=4 loops=1)
        Buckets: 1024 Batches: 1 Memory Usage: 9kB
        -> Seq Scan on circuito_pais (cost=0.00..1.96 rows=4 width=8) (actual time=0.009..0.018 rows=4 loops=1)
              Filter: (pa_id = 2)
              Rows Removed by Filter: 73
Planning Time: 0.366 ms
Execution Time: 0.197 ms
(10 rows)
```

Figura 7: Consulta post-optimizacion

### 4.2.2. Segunda Consulta

Se utilizó una vista materializada para reducir la complejidad de la consulta y evitar la repeticiones de patrones en esta. Asimismo, mediante la vista se garantiza una seguridad extra al solo dar acceso a un subconjunto de datos y no a todos ellos.

```
cc3201=# EXPLAIN ANALYZE SELECT pi.nombre, grpr.nombre
cc3201-# FROM f1_2.equipo_granpremio JOIN f1_2.piloto AS pi ON eqpi_id = pi.id
cc3201-# JOIN f1_2.escuderia AS esc ON eqes_id = esc.id
cc3201-# JOIN f1_2.temporada AS temp ON eqt_id = temp.id
cc3201-# JOIN f1_2.granpremio AS grpr ON gp_id = grpr.id
cc3201-# WHERE posicion_carrera = 1 AND esc.nombre = 'Ferrari' AND temp.agno = 2019;
                                QUERY PLAN

-----
Nested Loop  (cost=0.84..19.23 rows=1 width=33) (actual time=0.224..0.475 rows=3 loops=1)
  -> Nested Loop  (cost=0.56..18.88 rows=1 width=22) (actual time=0.212..0.446 rows=3 loops=1)
    -> Nested Loop  (cost=0.29..18.54 rows=1 width=16) (actual time=0.194..0.412 rows=3 loops=1)
      -> Seq Scan on temporada temp  (cost=0.00..1.89 rows=1 width=4) (actual time=0.020..0.022 rows=1 loops=1)
        Filter: (agno = 2019)
        Rows Removed by Filter: 70
      -> Nested Loop  (cost=0.29..16.65 rows=1 width=24) (actual time=0.169..0.379 rows=3 loops=1)
        -> Seq Scan on escuderia esc  (cost=0.00..7.67 rows=1 width=4) (actual time=0.011..0.067 rows=1 loops=1)
          Filter: ((nombre)::text = 'Ferrari'::text)
          Rows Removed by Filter: 373
        -> Index Scan using equipo_granpremio_pkey on equipo_granpremio  (cost=0.29..8.96 rows=1 width=32) (actual time=0.153..0.300 rows=3 loops=1)
          Index Cond: ((eqes_id = esc.id) AND (eqt_id = temp.id))
          Filter: (posicion_carrera = 1)
          Rows Removed by Filter: 39
        -> Index Scan using piloto_pkey on piloto pi  (cost=0.28..0.34 rows=1 width=18) (actual time=0.007..0.007 rows=1 loops=3)
          Index Cond: (id = equipo_granpremio.eqpi_id)
        -> Index Scan using granpremio_pkey on granpremio grpr  (cost=0.28..0.35 rows=1 width=23) (actual time=0.005..0.005 rows=1 loops=3)
          Index Cond: (id = equipo_granpremio.gp_id)
    Planning Time: 1.743 ms
    Execution Time: 0.540 ms
(20 rows)
```

Figura 8: Consulta pre-optimizacion

```
cc3201=# CREATE MATERIALIZED VIEW PilotoGRPR AS SELECT pi.nombre AS piloto , grpr.nombre AS grpremio, esc.nombre AS escu
adra, temp.agno AS temporada
FROM f1.equipo_granpremio JOIN f1.piloto AS pi ON eqpi_id = pi.id
JOIN f1.escuderia AS esc ON eqes_id = esc.id
JOIN f1.temporada AS temp ON eqt_id = temp.id
JOIN f1.granpremio AS grpr ON gp_id = grpr.id
WHERE posicion_carrera = 1;
SELECT 1035
```

Figura 9: Creación de Vista

```
cc3201=# EXPLAIN ANALYZE SELECT piloto, grpremio
cc3201=# FROM PilotoGRPR
cc3201=# WHERE escuadra = 'Ferrari' AND temporada = 2019;
               QUERY PLAN
-----
Seq Scan on pilotogrpr (cost=0.00..26.52 rows=5 width=33) (actual time=0.172..0.178 rows=3 loops=1)
  Filter: (((escuadra)::text = 'Ferrari'::text) AND (temporada = 2019))
  Rows Removed by Filter: 1032
Planning Time: 0.249 ms
Execution Time: 0.205 ms
(5 rows)
```

Figura 10: Consulta post-optimizacion

### 4.2.3. Tercera Consulta

Al igual que para la consulta anterior, se utilizó una vista materializada para reducir la complejidad de la consulta y evitar la repeticiones de patrones en esta. Asimismo, mediante la vista se garantiza una seguridad extra al solo dar acceso a un subconjunto de datos y no a todos ellos.

```
cc3201=# EXPLAIN ANALYZE SELECT result.pm, result.nombre, result.agno
cc3201=# FROM
cc3201=# (SELECT eqgp.pm, esc.nombre, agno
cc3201=# FROM
cc3201=# (SELECT AVG(vuelta_rapida_c) AS pm, eqes_id, eqt_id
cc3201=# FROM f1.equipo_granpremio JOIN f1.granpremio ON gp_id = id WHERE fecha>'2008-01-01'
cc3201=# GROUP BY eqes_id, eqt_id) AS eqgp, temporada AS temp, escuderia as esc
cc3201=# WHERE eqgp.eqt_id = temp.id and eqgp.eqes_id = esc.id) AS result
cc3201=# WHERE result.nombre = 'Ferrari';

QUERY PLAN

-----
Nested Loop  (cost=867.91..931.48 rows=2 width=31) (actual time=163.332..163.754 rows=13 loops=1)
  -> Hash Join  (cost=867.76..930.33 rows=7 width=35) (actual time=163.289..163.640 rows=13 loops=1)
    Hash Cond: (equipo_granpremio.eqes_id = esc.id)
    -> HashAggregate  (cost=860.08..891.16 rows=2487 width=32) (actual time=163.195..163.404 rows=152 loops=1)
      Group Key: equipo_granpremio.eqes_id, equipo_granpremio.eqt_id
      Batches: 1  Memory Usage: 145kB
      -> Hash Join  (cost=26.70..788.93 rows=7115 width=24) (actual time=79.865..139.535 rows=5345 loops=1)
        Hash Cond: (equipo_granpremio.gp_id = granpremio.id)
        -> Seq Scan on equipo_granpremio  (cost=0.00..696.69 rows=24869 width=32) (actual time=0.046..70.468 rows=24869 loops=1)
        -> Hash  (cost=22.76..22.76 rows=315 width=4) (actual time=0.678..0.681 rows=316 loops=1)
          Buckets: 1024  Batches: 1  Memory Usage: 20kB
          -> Seq Scan on granpremio  (cost=0.00..22.76 rows=315 width=4) (actual time=0.005..0.390 rows=316 loops=1)
      Filter: (fecha > '2008-01-01'::date)
      Rows Removed by Filter: 785
    -> Hash  (cost=7.67..7.67 rows=1 width=15) (actual time=0.081..0.084 rows=1 loops=1)
      Buckets: 1024  Batches: 1  Memory Usage: 9kB
      -> Seq Scan on escuderia esc  (cost=0.00..7.67 rows=1 width=15) (actual time=0.024..0.074 rows=1 loops=1)
      Filter: ((nombre)::text = 'Ferrari'::text)
      Rows Removed by Filter: 373
  -> Index Scan using temporada_pkey on temporada temp  (cost=0.14..0.16 rows=1 width=8) (actual time=0.005..0.005 rows=1 loops=1)
    Index Cond: (id = equipo_granpremio.eqt_id)
Planning Time: 0.515 ms
Execution Time: 163.886 ms
(23 rows)
```

Figura 11: Consulta pre-optimizacion

```
cc3201=# CREATE MATERIALIZED VIEW PilotoCircuito AS SELECT result.pm, result.nombre, result.agno
cc3201=# FROM
cc3201=# (SELECT eqgp.pm, esc.nombre, agno
cc3201=# FROM
cc3201=# (SELECT AVG(vuelta_rapida_c) AS pm, eqes_id, eqt_id
cc3201=# FROM f1.equipo_granpremio JOIN f1.granpremio ON gp_id = id WHERE fecha>'2008-01-01'
cc3201=# GROUP BY eqes_id, eqt_id) AS eqgp, temporada AS temp, escuderia as esc
cc3201=# WHERE eqgp.eqt_id = temp.id and eqgp.eqes_id = esc.id) AS result;
SELECT 152
```

Figura 12: Creación de Vista



```
cc3201=# EXPLAIN ANALYZE SELECT pm, nombre, agno FROM PilotoCircuito WHERE nombre = 'Ferrari';
                                QUERY PLAN
-----
Seq Scan on pilotocircuito  (cost=0.00..3.90 rows=13 width=34) (actual time=0.019..0.048 rows=13 loops=1)
  Filter: ((nombre)::text = 'Ferrari'::text)
  Rows Removed by Filter: 139
Planning Time: 0.183 ms
Execution Time: 0.080 ms
(5 rows)
```

Figura 13: Consulta post-optimizacion

## 4.3. Medidas de Seguridad contra inyecciones SQL

### 4.3.1. Input automático

Evitando que el usuario ingrese directamente con lo que ingresa a la base de datos.

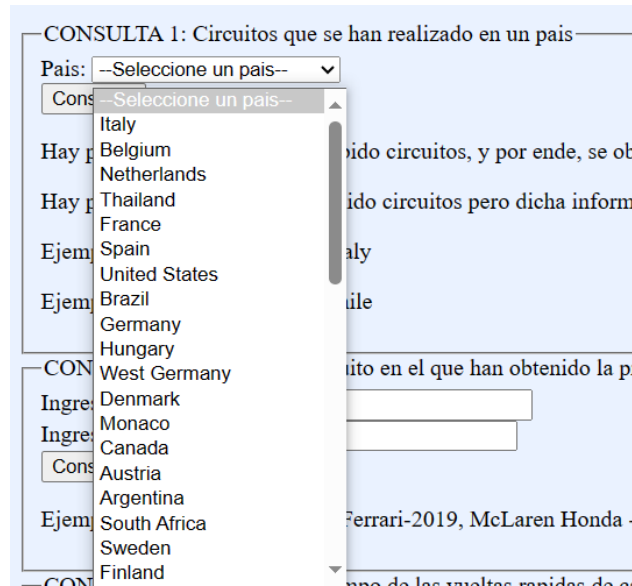


Figura 14: Input automático

### 4.3.2. Usar sentencias y consultas parametrizadas

Mediante los marcadores de posición se indica donde se sustituirá una cadena. Se tendrá que `:valor1` se sustituye por `GET['id']` de forma segura y al realizar `bindParam` se indica que la variable será de tipo `STRING`. Además de una desinfección de input mediante `filtervar`. Esto se realiza para cada consulta.

```
// Recibe el input
$variable1= $_GET['id'];
$cadena= filter_var($variable1, FILTER_SANITIZE_STRING);

// Ejecutar la consulta SQL
$query = "SELECT nombre FROM
        (SELECT cir_id
         FROM f1.circuito_pais
         WHERE pa_id = :valor1) AS cirid
        JOIN f1.circuito ON cirid.cir_id = id";
$stmt = $pdo->prepare($query);
$stmt->bindParam(":valor1", $cadena, PDO::PARAM_STR);
$stmt->execute([':valor1' => $cadena]);
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Figura 15: Aplicación de parametrización

## 4.4. Aplicación web

Link a la aplicación web: <https://grupo21.cc3201.dcc.uchile.cl/>

### 4.4.1. Interfaz

Interfaz con las consultas a realizar en sus respectivos inputs.

The screenshot shows a web browser window with the URL <https://grupo21.cc3201.dcc.uchile.cl/>. The page title is "F1 Data between 1985 and 2020". Below the title, the group name is "Grupo 21: Constanza Vicencio, Juan I. Valdivia, Monserrat Montero, Patricio Espinoza". A brief description of the project is provided. The interface contains three query sections:

- CONSULTA 1: Circuitos que se han realizado en un pais**
  - Input: Pais: Italy (dropdown menu)
  - Button: Consultar
  - Feedback: Hay paises en los que no han habido circuitos, y por ende, se obtiene 'No se encontraron resultados'.
  - Feedback: Hay paises que pueden haber tenido circuitos pero dicha informacion no se encontraba en la base de datos, y por ende, se obtiene 'No se encontraron resultados'.
  - Example: Ejemplo de pais con circuitos: Italy
  - Example: Ejemplo de pais sin circuitos: Chile
- CONSULTA 2: Pilotos y el circuito en el que han obtenido la primera posicion para una determinada escuderia en cierta temporada**
  - Input: Ingrese nombre escuderia: Ferrari
  - Input: Ingrese año temporada: 2019
  - Button: Consultar
  - Example: Ejemplo de escuderias y años: Ferrari-2019, McLaren Honda - 1989, Williams Renault - 1993
- CONSULTA 3: Promedio de tiempo de las vueltas rapidas de carrera de una escuadra cada temporada**
  - Input: Ingrese nombre escuderia: Renault
  - Button: Consultar
  - Example: Ejemplo de escuderias: Ferrari, Mercedes, Renault

Figura 16: Interfaz aplicación web

## 4.5. Resultados Consultas

### 4.5.1. Consulta 1: Input Pais = Italy

Resultados	
Nombre de los circuitos	
Autodromo Nazionale di Monza	
Autodromo Enzo e Dino Ferrari	
Pescara Circuit	
Autodromo Internazionale del Mugello	

Figura 17: Consulta 1

#### 4.5.2. Consulta 2: Input Escuderia = Ferrari, Temporada = 2019

Resultados	
Piloto	Circuito
Charles Leclerc	Belgian Grand Prix
Charles Leclerc	Italian Grand Prix
Sebastian Vettel	Singapore Grand Prix

Figura 18: Consulta 2

#### 4.5.3. Consulta 3: Input Escuderia = Renault

Resultados		
Tiempo vuelta mas rapida	Escuderia	Temporada
00:01:22.913176	Ferrari	2020
00:01:28.498952	Ferrari	2019
00:01:28.752595	Ferrari	2018
00:01:20.499875	Ferrari	2017
00:01:26.179429	Ferrari	2016
00:01:31.531684	Ferrari	2015
00:01:29.324263	Ferrari	2014
00:01:31.3625	Ferrari	2013
00:01:29.11725	Ferrari	2012
00:01:34.251579	Ferrari	2011
00:01:31.0365	Ferrari	2010
00:01:31.195478	Ferrari	2009
00:01:27.700222	Ferrari	2008
00:01:25.721118	Ferrari	2007
00:01:21.586972	Ferrari	2006
00:01:25.002737	Ferrari	2005
00:01:22.743417	Ferrari	2004
00:00:00	Ferrari	2003

Figura 19: Consulta 3: Input Escuderia = Renault

## 4.6. Conclusiones del Proyecto

Una de las mayores dificultades que tuvo el proyecto, fue la alta cantidad de errores durante el proceso de carga de datos provenientes de utilizar información de varias bases de datos de diferentes creadores. Se tuvo que adaptar los formatos repetidas veces para hacer calzar las llaves con las que obtener información de una tabla a otra. Esto, junto a la gran cantidad de datos con los que se trabajó, hizo que esta etapa fuese extensa, atrasando el resto del proyecto.