Memoria Práctica Final

Miquel Àngel Montero Pazmiño

Enero 2025

Contents

1 Introducción

Este documento presenta el desarrollo así como los resultados obtenidos de realizar la práctica final de la asignatura de Sistema Gestors de Base de Dades.

Una aclaración inicial es que toda configuración inicial se realiza con los usuarios administrativos, es decir, la gestión de usuarios y la creación de áreas de almacenamiento así como la definición de las primeras bases de datos (RAW_IMPORT) se llevan a cabo con usuarios administrativos, ya que no se hace mención de que las áreas de almacenamiento deban ser gestionadas por ningún usuario ni que ninguno tenga la posibilidad de crear una base de datos como tal.

2 Infraestructura

En el siguiente apartado se pasarán a detallar tanto las versiones de los programas empleados así como la jerarquía de los ficheros adjuntos en la carpeta `Miquel_Angel_Montero_Pazmino_SQL`

2.1 Entorno de desarrollo

Versiones de los programas utilizados para la realización de esta práctica. Las configuraciones que se han realizado a estos programas aparecerán en sus respectivas secciones:

- SO: Ubuntu 24.04.1 LTS
- MySQL: mysql 8.0.40 ubuntu 24.04.1 for Linux
- PostgreSQL: psql 16.6 (Ubuntu 16.6-1.pgdg24.04+1)
- pgloader: 3.6.70f3557

2.2 Jerarquía ficheros

Para una mejor interpretación, los archivos SQL se han dividido en carpetas según que función cumplen.

- MYSQL: Contiene todos los archivos relacionados con el trabajo en MySQL
  - GESTMAT: Contiene todos los archivos relacionados con las operaciones de la base de datos GESTMAT, así como las definiciones y las consultas que se realizan sobre esta base de datos
  - `RAW_IMPORT`: Contiene todos los archivos SQL referentes a la importación local desde los csv así como las defincion de la propia base de datos y las vistas utilizadas
- POSTGRES: Contiene todos los archivos relacionados con el trabajo en postgreSQL
  - OLDGESTMAT: Contiene todos los archivos relacionados con las operaciones de la base de datos OLDGESTMAT, así como el archivo que define los parámetros para realizar la migración
  - PROD: Contiene todos los archivos SQL de la base de datos de PROD, así como las consultas que se realizan sobre la base de datos

3 Diseño Base de Datos

3.1 Modelo UML

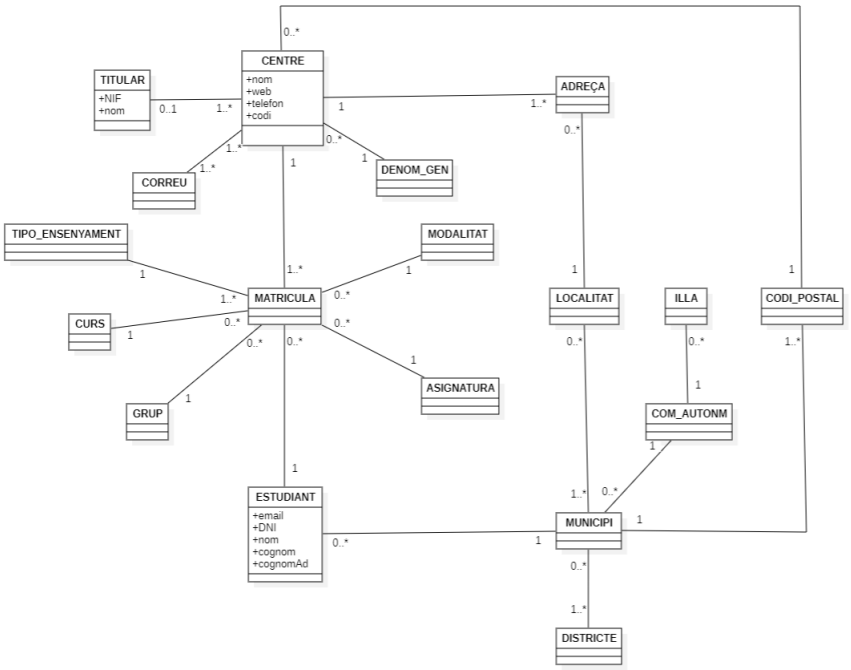

Figure 1: Imagen del diagrama de clases

Se puede apreciar la imagen que se ve a continuación, la cual es la representación del Modelo Relacional de la base de datos implementada. Más adelante se han añadido tablas adicionales correspondientes a las diferentes inconsistencias, las cuales no se reflejan en este modelo

3.2 Modelo Relacional

Se aprecia el modelo relacional indicando con colores las Foreign Keys para que sea más visual y fácil identificar las diferentes relaciones entre las tablas.

3.2.1 Suposiciones

- Encontramos varios centros que presentan múltiples direcciones, así como centros que comparten direcciones
- Se ha detectado que los códigos postales referentes a ESTUDIANT no corresponden con los códigos postales de la realidad, así como también distan de los códigos postales presentes en CENTRE, los cuales si presentan consistencias reales. Por ello, se ha decidido relacionar el ESTUDIANT con el MUNICIPI, el cual se cogerá como referencia para la distancia con el CENTRE que se pide más adelante
- Se ha decidido, a pesar de tener una clase CORR_ELECT, guardar como atributo el correo de los ESTUDIANT, ya que el tipo de relación que presentaban(1 - 1) se ha considerado como atributo.
- Se han evitado la generación de identificadores numéricos en varias tablas para evitar la duplicación de registros.
- Se han creado las tablas COM_AUT con la idea de que puedan haber ESTUDIANT procedentes de otras comunidades autónomas como alumnos de intercambio. Por este motivo, se crea asimismo la tabla ILLA debido a que no todas las comunidades autónomas cuentan con islas
- Por motivo de la inconsistencia de los códigos postales de ESTUDIANT, se ha asociado el DISTRIT con MUNICIPI

3.3 Normalización

Primera Forma Normal (1FN) Los datos son atómicos, es decir, no hay columnas con listas o valores repetidos dentro de celdas. Todas las tablas tienen claves primarias definidas que aseguran filas únicas.

Segunda Forma Normal (2FN) Cumple con 1FN y no tiene dependencias parciales, ya que todos los atributos no clave dependen completamente de la clave primaria. Incluso en tablas con claves compuestas, como R_CENTRE_CORREU, no hay columnas que dependan de una sola parte de la clave.

Tercera Forma Normal (3FN) Cumple con 2FN y no hay dependencias transitivas: los atributos no clave dependen únicamente de la clave primaria. Las relaciones entre tablas están estructuradas correctamente para evitar redundancias y transitividades innecesarias.

4 Trabajo en MySQL

4.1 Preparación previa

Antes de inciar con el trabajo especificado es importar notar que se piden la creación de areas de almacenamiento (TABLESPACE). Para ello, usaremos el engine InnoDB de MySQL el cual debe habilitarse en el archivo de configuración mysqld.cnf. Para realizar esta tarea hace falta des-comentar, o añadir en nuestro caso, la siguiente configuración:

- `innodb_file_per_table = 1`
- `innodb_data_home_dir = 1`
- `innodb_tmpdir = 1`

Una vez tengamos definida la configuración reiniciaremos la instancia de MySQL, en nuestro caso con el comando `(sudo) systemctl restart mysql` para hacer efectivos los cambios realizados en la configuración. Después de esto podemos generar ya nuestras TABLESPACE `DADES_TEMPORALS y PANDORA`.

Estas están definidas al inicio de los archivos
`SQL\MYSQL\RAW_IMPORT\Miquel_Angel_Montero_Pazmino_SCHEMA_RAW_IMPORT.sql`
y
`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_SCHEMA_GESTMAT.sql`
respectivamente

4.2 Definición de Usuarios

En MySQL se deben definir 2 usuarios

- `IMPORTADO_1`: Encargado de generar la base de datos `RAW_IMPORT` así como de insertar los datos procedentes de los archivos csv
- `TRANSFORMADOR_1`: Este usuario tiene permitdo acceder a la base de datos GESTMAT así como crear estructuras, insertar datos y realizar consultas, tanto a GESTMAT como a `RAW_IMPORT` para poder hacer el procesamiento de los datos

Cada usuario estará definido en el documento en el cual se define su base de datos, es decir, IMPORTADOR_1 se puede consultar su definición en el archivo:
`SQL\MYSQL\RAW_IMPORT\Miquel_Angel_Montero_Pazmino_SCHEMA_RAW_IMPORT.sql`
mientras que TRANSFORMADOR_1 queda definido en:
`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_SCHEMA_GESTMAT.sql`

4.3 Creación RAW IMPORT

Primeramente deberemos crear el area de almacenamiento donde deberemos guardar nuestra base de datos. Primeramente crearemos la TABLESPACE

"DADES_TEMPORALS" para continuar con la instanciación de la base de datos RAW_IMPORT.

Una vez creada la base de datos, deberemos ahora aportar los privilegios adecuados al usuario IMPORTADOR_1 para que pueda generar las estructuras dentro de la base de datos.

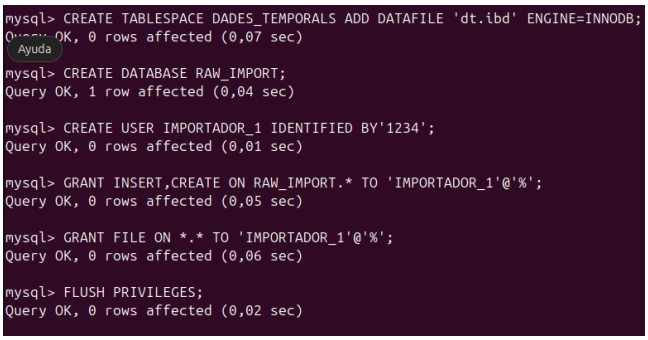

Figure 2: Captura de la creación de RAW_IMPORT y DADES_TEMPORALS

Tras realizar estos pasos, pasaremos a acceder a la base de datos con el usuario definido. Para ello accederemos desde nuestra terminal aplicando el siguiente comando `mysql -U IMPORTADOR_1 -P --local-infile`. Una vez dentro de la base de datos, definiremos las estructuras necesarias, definidas en el documento `SQL\MYSQL\RAW_IMPORT\Miquel_Angel_Montero_Pazmino_SCHEMA_RAW_IMPORT.sql`, el cual define las tabas CENTRE, ESTUDIANT y MATRICULA y cuyos atributos corresponden a las columnas de los csv que contienen los datos.

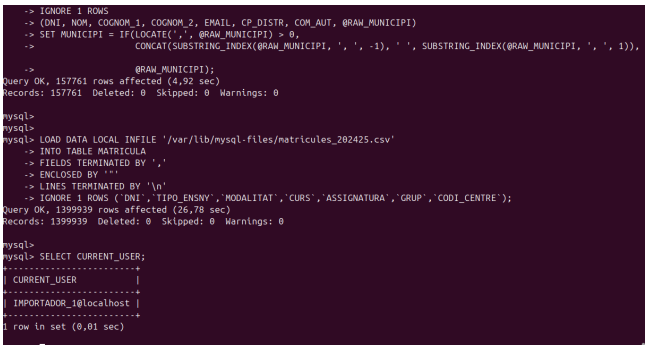

Figure 3: Captura de la importación de datos a RAW_IMPORT

Para la importación local, es necesario que los archivos se encuentren en una ubicación visible por el gestor MySQL, en este caso, el directorio donde se deben colocar los archivos es el siguiente `var/lib/mysql-files`.También será necesario indicar al sistema operativo que el propietario de estos archivos sea el usuario mysql. Una vez colocados los ficheros en el directorio correspondiente

se ha realizado la ejecución de las comandas correspondientes al archivo
`SQL\MYSQL\RAW_IMPORT\Miquel_Angel_Montero_Pazmino_SCHEMA_IMPORT_SCRIPT.sql`

del cual solo mencionar que en los datos referentes a ESTUDIANT se han encontrado municipios con nombres separados por comas, lo cual presentaba inconsistencias a la hora de importar los datos. Debido a esto, se ha procesan los datos de ESTUDIANT que en el campo municipio presenta una coma, eliminando y colocando los nombres de manera consistente.

Para favorecer el trabajo de inserción de la siguiente etapa, se ha procedido a generar las siguientes vistas:

- MAIL_LIST : Esta vista asocia todos los correos a un centro. Esta vista es necesaria cuando apreciamos la existencia de un campo llamado `EMAIL_LIST` el cual debemos tratar para no trabajar con campos multievaluados.
- VW_ADREC_LOCALITAT: Esta vista asocia todas las direcciones de la tabla CENTRE con un centro y una localidad. EL motivo de crear esta lista es evitar aquellas campos ADREÇA que presentan varias direcciones, incumpliendo así el nivel 1 de normalización de base de datos

Ambas sentencias se encuentran en el siguiente archivo:
`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_CREATE_VIEWS.sql`

4.4 Creación GESTMAT

La definición de la base de datos GESTMAT en MySQL se encuentra en el siguiente archivo:

Figure 4: Captura de la creación de la base de datos GESTMAT y la tablespace PANDORA

`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_SCHEMA_GESTMAT.sql`

Mientras que las sentencias para el traspaso de los datos de RAW_IMPROT a GESTMAT se puede consultar aquí:
`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_PASS_RAW_IMPORT_TO_GESTMAT.sql`

```
mysql> CREATE TABLE R_MUNICIPI_DISTRIT(
    ->     MUNICIPI INT NOT NULL,
    ->     DISTRIT INT NOT NULL,
    ->     FOREIGN KEY (MUNICIPI) REFERENCES MUNICIPI(ID),
    ->     FOREIGN KEY(DISTRIT) REFERENCES DISTRIT(ID),
    ->     CONSTRAINT ID PRIMARY KEY(MUNICIPI, DISTRIT)
    -> );
Query OK, 0 rows affected (0,13 sec)

mysql>
mysql> CREATE TABLE R_MUNICIPI_LOCALITAT(
    ->     MUNICIPI INT NOT NULL,
    ->     LOCALITAT INT NOT NULL,
    ->     FOREIGN KEY (MUNICIPI) REFERENCES MUNICIPI(ID),
    ->     FOREIGN KEY (LOCALITAT) REFERENCES LOCALITAT(ID),
    ->     CONSTRAINT ID PRIMARY KEY(MUNICIPI, LOCALITAT)
    -> );
Query OK, 0 rows affected (0,13 sec)

mysql> SELECT CURRENT_USER;
+-------------------+
| CURRENT_USER      |
+-------------------+
| TRANSFORMADOR_1@% |
+-------------------+
1 row in set (0,00 sec)

mysql>
```

Figure 5: Declaración de GESTMAT

```
mysql> INSERT INTO R_MUNICIPI_CP(MUNICIPI,CP)
    -> SELECT
    -> (SELECT
    ->         M.ID
    ->     FROM GESTMAT.MUNICIPI M
    ->     WHERE
    ->         CASE
    ->             WHEN C.MUNICIPI="CIUTADELLA" THEN M.NOM LIKE "CIUTADELLA %"
    ->             ELSE M.NOM=C.MUNICIPI
    ->             END LIMIT 1) AS MUNICIPI,
    ->             CONCAT('0',C.CP) AS CP
    ->     FROM
    ->         RAW_IMPORT.CENTRE C
    ->     GROUP BY C.MUNICIPI,C.CP;
Query OK, 151 rows affected (0,06 sec)
Records: 151  Duplicates: 0  Warnings: 0

mysql> SELECT CURRENT_USER;
+-------------------+
| CURRENT_USER      |
+-------------------+
| TRANSFORMADOR_1@% |
+-------------------+
1 row in set (0,00 sec)
```

Figure 6: Inserción de los datos de RAW_IMPORT a GESTMAT

El cual presenta algunas peculiaridades a destacar:

- Debido a la existencia de "CIUTADELLA" y "CIUTADELLA DE MENORCA" tanto en MUNICIPI como en LOCALITAT, se han tratado los registros relacionados con estos campos de manera especial
- Los registros de DISTRIT han sido obtenidos del campo CP_DISTR de la tabla RAW_IMPORT.ESTUDIANT recortando los 2 últimos caracteres

de este campo a excepción de uno, el cual presentaba un distrito de 3 dígitos.

- Los registros TITULAR presentan algunos campos que hacen referencia a que no cuentan con un titular. Para ello, se han omitido estos registros en la tabla TITULAR y se ha insertado un registro llamado BUIT, el cual se usará como referencia para todos aquellas relaciones que hacían referencia a la ausencia de titular

- Para los números de teléfono de CENTRE se a han realizado algunas configuraciones las cuales permiten eliminar caracteres, debido a que el número de teléfono se guarda como un INT

4.5 Consultas

Las consultas se encuentran en el siguiente archivo:
`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_SELECTS.sql`
A continuación se presentarán los tiempos de ejecución de las consultas:

```
| 7015941 |                629 |
| 7016207 |                666 |
| 7016219 |                638 |
| 7016438 |                598 |
| 7016530 |                802 |
| 7016542 |                594 |
| 7700015 |                519 |
| 7700027 |                862 |
| 7700039 |                214 |
| 7700040 |                801 |
| 7700209 |                855 |
| 7700234 |                770 |
| 7700349 |                877 |
| 7700386 |                758 |
+---------+--------------------+
212 rows in set (9,35 sec)

mysql> SELECT CURRENT_USER;
+------------------+
| CURRENT_USER     |
+------------------+
| TRANSFORMADOR_1@% |
+------------------+
1 row in set (0,01 sec)
```

Figure 7: Alumnos matriculados en FP

```
| 7003900 |                521 |
| 7700015 |                519 |
| 7015549 |                518 |
| 7013656 |                516 |
| 7015872 |                516 |
| 7002208 |                515 |
| 7015689 |                511 |
| 7001319 |                507 |
| 7007668 |                506 |
| 7015291 |                504 |
| 7007735 |                500 |
| 7014879 |                474 |
| 7007954 |                465 |
| 7015410 |                454 |
| 7006305 |                402 |
| 7008843 |                390 |
| 7002427 |                349 |
| 7008004 |                278 |
| 7008089 |                246 |
| 7700039 |                214 |
| 7007917 |                161 |
| 7003559 |                 75 |
| 7003602 |                 75 |
+---------+--------------------+
212 rows in set (12,00 sec)
```

Figure 8: Densidad de centros

```
| Operaciones auxiliares de preparación del terreno, plantación y siembra de cultivos   | 7008223 | CENTRE D'EDUCACIÓ PE
RSONES ADULTES  CAMP RODÓ
| Dibujo artistico II                                                                    | 7015291 | NAU ESCOLA
                                                                                         |
| Gestión del montaje y del mantenimiento de instalaciones eléctricas                    | 7015941 | ECOLEA COLEGIO INTER
NACIONAL MALLORCA
| Embellecimiento de superficies                                                         | 7016219 | CENTRE DE FORMACIÓ,
INNOVACIÓ I DESENVOLUPAMENT DE LA FORMACIÓ PROFESSIONAL DE LES ILLES BALEARS_CFINFP_IB   |
| Mantenimiento instalaciones calorif                                                    | 7700040 | CENTRE DE PROFESSORS
 DE PALMA, JAUME CAÑELLAS MUT                                                            |
+----------------------------------------------------------------------------------------+---------+--------------------
----------------------------------------------------------------------------------------+
13 rows in set (19,33 sec)
```

Figure 9: Asignaturas con menos de 4 alumnos

```
| 99823539C | BERENICE                  | DE JUAN             |        7 |
| 99830553L | NICCOLAS AUGUSTO          | FERNANDEZ           |        7 |
| 99831487X | SARA                      | ROCAMORA            |        7 |
| 99837825T | FILIP                     | BATALLER            |       36 |
| 99841796S | DAYLIMAR ALEJANDRA        | SOUZA               |       44 |
| 99848162X | JOSÉ                      | CANET               |        7 |
| 99858506G | MANUEL                    | FRANCÉS             |       53 |
| 99870751J | ANDRÉS                    | HAN                 |        7 |
| 99878738L | ARIADNA                   | GARCÍA              |       34 |
| 99880051K | DIEGO                     | FUSTER              |       37 |
| 99888918X | LOLA                      | HARO                |       23 |
| 99904126S | PABLO                     | HERNÁNDEZ           |       20 |
| 99907865M | CELIA                     | SÁNCHEZ             |       26 |
| 99939005A | JOSÉ ÁNGEL                | GISBERT             |        7 |
| 99970219Y | HUGO                      | SÁNCHEZ             |       52 |
+-----------+---------------------------+---------------------+----------+
3977 rows in set (2,22 sec)
```

Figure 10: Alumnos no matriculados

```
mysql> SELECT COUNT(DNI) FROM ALUMNES_SENSE_MATRICULA;
+------------+
| COUNT(DNI) |
+------------+
|       3977 |
+------------+
1 row in set (0,91 sec)
```

Figure 11: Alumnos no matriculados COUNT

```
5882V | ÉRIK GÓMEZ                |
| SENCELLES                                                                                                |     7014375 | 8569
9837C | IRENE VÁZQUEZ             |
| SENCELLES                                                                                                |     7014375 | 8952
1673S | RAÚL GARCÍA               |
| SENCELLES                                                                                                |     7014375 | 9127
7324F | SAÚL JAIME                |
| SENCELLES                                                                                                |     7014375 | 9299
8774E | ISRAA CASTELLET           |
| SENCELLES                                                                                                |     7014375 | 9351
3369S | ROBERTO RUÍZ              |
| SENCELLES                                                                                                |     7014375 | 9365
6792X | JOEL BRAYAN HERRERO       |
| SENCELLES                                                                                                |     7014375 | 9705
0383C | ELIAS SELLÉS              |
| SENCELLES                                                                                                |     7014375 | 9759
5075A | MARÍA CARUSO              |
| SENCELLES                                                                                                |     7014375 | 9908
5551X | MIRIAM FERNÁNDEZ          |
+-----------------------------------------------------------------------------------------------------------+-------------+-----
------+----------------------------------+
3977 rows in set (1,71 sec)
```

Figure 12: Centros cercanos

4.6 Optimización

Como se observa en la ejecución del apartado anterior, las consultas realizadas presentan unos tiempos de ejecución muy superiores a los que se consideran óptimos. Para mejorar dichos resultados se ha focalizado en reducir las consultas directas a la tabla MATRICULA, la cual es la más atacada así como la que presenta mayor volumetría.

La estrategia principal utilizada ha sido la de reducir el tamaño de matricula mediante la creación de vistas materializadas. Dichas vistas deben ser creadas y actualizadas siempre que el sistema no se encuentre en uso, para mantener la disponibilidad del servicio en todo momento.

Se ha propuesto esta solución ya que se especifica que la inserción de nuevos datos se realiza de manera anual, por lo tanto estas vistas solo se verían actualizadas una vez al año

Las vistas materializadas a utilizar son las mismas vistas que se empleaban en las propias consultas, solo que ahora presentan las modificaciones presentes en el documento:

`SQL\MYSQL\GESTMAT\Miquel_Angel_Montero_Pazmino_OPTIMIZED_SELECTS.sql`

Los nuevos tiempos de ejecución son los siguientes:

```
mysql> EXPLAIN ANALYZE SELECT
    ->     VECT.CENTRE AS CENTRE,
    ->     VECT.USUARIS_MATRICULATS
    -> FROM
    ->     VW_EST_CENT_TE VECT
    -> WHERE VECT.TIPO_ENSNY="FP";
+------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------+
| EXPLAIN
                                                                       |
+------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------+
| -> Filter: (VECT.TIPO_ENSNY = 'FP')  (cost=85.7 rows=84.7) (actual time=6.94..9.07 rows=212 loops=1)
    -> Table scan on VECT  (cost=85.7 rows=847) (actual time=6.93..9 rows=847 loops=1)
 |
+------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------+
1 row in set (0,01 sec)
```

Figure 13: Alumnos matriculas en FP Optimización

```
mysql> EXPLAIN ANALYZE SELECT     VEC.CENTRE AS CENTRE,     VEC.USUARIS_MATRICULATS FROM     VW_EST_CENT VEC;
+------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                  |
+------------------------------------------------------------------------------------------+
| -> Table scan on VEC  (cost=21.5 rows=212) (actual time=4.15..4.24 rows=212 loops=1)
 |
+------------------------------------------------------------------------------------------+
1 row in set (0,00 sec)
```

Figure 14: Densidad de centros Optimización

```
------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------+
| -> Nested loop inner join  (cost=8839 rows=13512) (actual time=1.78..59.3 rows=13 loops=1)
    -> Filter: (AXA.NUM_ESTUDIANTS < 4)  (cost=4110 rows=13512) (actual time=1.75..59 rows=13 loops=1)
        -> Table scan on AXA  (cost=4110 rows=40539) (actual time=0.029..51 rows=39724 loops=1)
    -> Single-row index lookup on C using PRIMARY (ID=AXA.CENTRE)  (cost=0.25 rows=1) (actual time=0.0214..0.0214 rows=1
 loops=13)
 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------+
1 row in set (0,07 sec)
```

Figure 15: Asignaturas con menos de 4 alumnos Optimización

```
mysql> EXPLAIN ANALYZE SELECT * FROM ALUMNES_SENSE_MATRICULA;
+--------------------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                                      |
+--------------------------------------------------------------------------------------------------------------+
| -> Table scan on ALUMNES_SENSE_MATRICULA  (cost=401 rows=3977) (actual time=0.0232..2.24 rows=3977 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------+
1 row in set (0,01 sec)
```

Figure 16: Alumnos no matriculados Optimización

```
mysql> EXPLAIN ANALYZE SELECT COUNT(DNI) FROM ALUMNES_SENSE_MATRICULA;
+-----------------------------------------------------------------------------------------+
| EXPLAIN                                                                                 |
+-----------------------------------------------------------------------------------------+
| -> Count rows in ALUMNES_SENSE_MATRICULA  (actual time=9.49..9.49 rows=1 loops=1)
 |
+-----------------------------------------------------------------------------------------+
1 row in set (0,01 sec)
```

Figure 17: Alumnos matriculas en FP Optimización

5 Trabajo en PostgreSQL

5.1 Preparación previa

Primeramente se debe crear el TABLESPACE PANDORA en un directorio el cual el usuario de postgres tenga acceso. En este caso dicho directorio es el _var_lib_postgresql_tablespaces el cual almacenará la base de datos GESTMAT. Dicha definición y la de los usuarios la podemos encontrar en :

`SQL\POSTGRES\OLDGESTMAT\Miquel_Angel_Montero_Pazmino_DEFINE_GESTMAT_DDBB.sql`

Se crearán también dentro de la base de datos GESTMAT los SCHEMAS OLDGESTMAT y PROD.

5.2 Definición de Usuarios

Los usuarios que se deben crear para trabajar en PostreSQL son los siguientes:

- UDATAMOVEMENT: Dicho usuario es el encargado de realizar la migración y debe tener acceso al esquema OLDGESTMAT, así como poder crear e insertar datos dentro de este esquema en la base de datos GESTMAT
- UCONSELLERIA: Este usuario debe tener control total sobre la base de datos GESTMAT y sobre los SCHEMAS OLDGESTMAT y PROD así como ser el dueño de la propia base de datos como se observa a continuación

```
 gestmat   | uconselleria | UTF8     | libc            | es_ES.UTF-8 | es_ES.UTF-8 |                |
     | =Tc/uconselleria           +
```

Figure 18: UCONSELLERIA como propietario de gestmat

5.3 Migración MySQL a PostgreSQL

Para realizar la migración de MySQL a PostgreSQL se ha utilizado la herramienta pgloader. Recalcar que la versión utilizada es la que se encuentra en el repositorio actual del github y no la que se obtiene mediante el comando apt get pgloader, el cual ofrece una versión desactualizada que generaba errores en la migración.

Para llevar a cabo esta migración se han utilizado los usuarios TRASFORMADOR_1 por la parte de MySQL y el UDATAMOVENT por parte de PostgreSQL. Para realizar la migración se ha hecho uso del siguiente archivo:

```
load database
from    mysql://TRANSFORMADOR\_1:1234@localhost/GESTMAT
into pgsql://udatamovement:12345@localhost:5432/gestmat

WITH include drop, create tables, no truncate, reset sequences, foreign keys

 SET maintenance\_work\_mem to '128MB', work\_mem to '12MB'

ALTER SCHEMA 'GESTMAT' RENAME TO 'oldgestmat'
```

en el cual se especifican los elementos que se quieren migrar así como los recursos de memorias dedicados a la migración. Para llevar a cabo la migración es necesario situarse en la carpeta pgloader_build_bin y se ejecuta el comando ._pgloader migracion.load(En el momento de la ejecución el archivo se encontraba en el mismo directorio, sino se tendría que colocar el path del archivo migrtacion.load). También será necesario inidcar que el usuario TRASNFORMADOR_1 cuenta con este tipo de identificación `IDENTIFIED WITH 'mysql_native_password'`

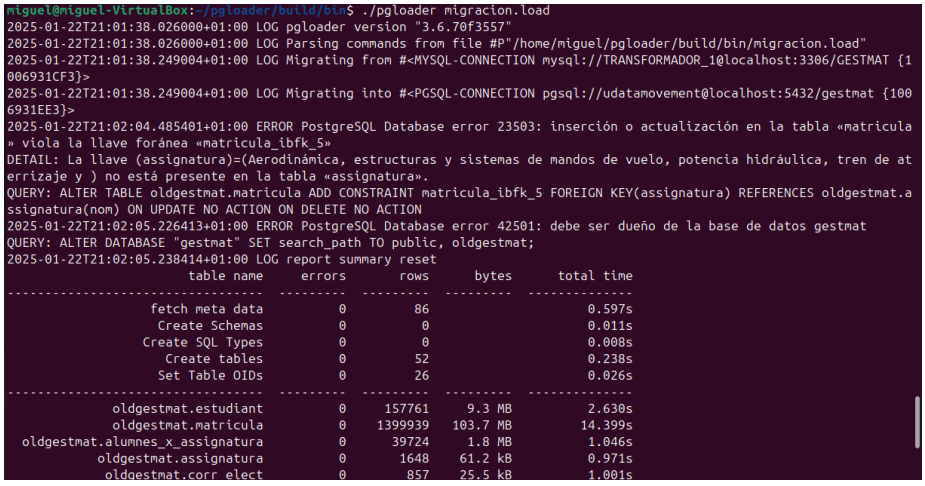

Figure 19: Resultados de la migración

En mi caso, se realizo la migración de manera positiva exceptuando el caso de que algunos registros de la tabla MATRICULA presentaban el campo ASSIGNATURA con valores que no se encontraban en la tabla ASSIGNATURA. Analizando este error en detenimiento, se observó como algunos valores de MATRICULA habían sufrido una ligera modificación añadiendo un espacio en blanco tras el nombre de ASSIGNATURA.

Para solucionar esta casuística, se realizo un UPDATE en matrícula cambiando los registros que presentaban valores de ASSIGNATURA incorrectos por los valores correspondientes.

La consulta se puede visualizar aquí:
`SQL\POSTGRES\OLDGESTMAT\Miquel_Angel_Montero_Pazmino_UPDATE_MATRICULA.sql`

5.4 Creación PROD

EL archivo para realizar el paso a PROD es el siguiente:

`SQL\POSTGRES\PROD\Miquel_Angel_Montero_Pazmino_SCHEMA_PROD.sql`

Los ajustes que se han realizado para la generación del schema prod han sido la sustitución de SERIALIZE en lugar de AUTO_INCREMENT y el uso de VARCHAR variable en los campos que no presentan una longitud fija. También ha sido necesario definir nombres diferentes en todas las CONSTRAINTS creadas mientras que en MySQL este aspecto no ha supuesto un problema.

Por último, es necesario que todas las tablas creadas se les indique explícitamente que schema van a pertenecer, en este caso `prod.table_name`

Una vex creado el SCHEMA adaptado solo falta introducir los datos de OLDGESTMAT a PROD. Dicho proceso se puede consultar en el siguiente archivo:

`SQL\POSTGRES\PROD\Miquel_Angel_Montero_Pazmino_PASS_OLDGESTMAT_TO_PROD.sql`

Para finalizar esta parte, será conveniente realizar la eliminación tanto de OLDGESTMAT como el usuario UDATAMOVEMENT.

```
gestmat=# DROP SCHEMA oldgestmat CASCADE;
NOTICE:  eliminando además 26 objetos más
DETALLE:  eliminando además tabla oldgestmat.adreca
eliminando además tabla oldgestmat.alumnes_sense_matric
eliminando además tabla oldgestmat.alumnes_x_assignatur
eliminando además tabla oldgestmat.assignatura
eliminando además tabla oldgestmat.centre
eliminando además tabla oldgestmat.codi_postal
eliminando además tabla oldgestmat.com_aut
eliminando además tabla oldgestmat.corr_elect
eliminando además tabla oldgestmat.curs
eliminando además tabla oldgestmat.denom_gen
eliminando además tabla oldgestmat.distrit
eliminando además tabla oldgestmat.estudiant
eliminando además tabla oldgestmat.estudiant_matriculat
eliminando además tabla oldgestmat.grup
eliminando además tabla oldgestmat.illa
eliminando además tabla oldgestmat.localitat
eliminando además tabla oldgestmat.matricula
eliminando además tabla oldgestmat.modalitat
eliminando además tabla oldgestmat.municipi
eliminando además tabla oldgestmat.r_centre_correu
eliminando además tabla oldgestmat.r_municipi_distrit
eliminando además tabla oldgestmat.r_municipi_localitat
eliminando además tabla oldgestmat.tipo_ensny
eliminando además tabla oldgestmat.titular
eliminando además tabla oldgestmat.vw_est_cent
eliminando además tabla oldgestmat.vw_est_cent_te
DROP SCHEMA
```

Figure 20: Eliminación de OLDGESTMAT

5.5 Consultas

Las consultas se encuentran en el siguiente archivo:

`SQL\POSTGRES\PROD\Miquel_Angel_Montero_Pazmino_SELECTS_POSTGRES.sql`

A continuación se presentarán los tiempos de ejecución de las consultas:

```
gestmat=# SELECT
    M.CENTRE,
    M.TIPO_ENSNY,
    COUNT(DISTINCT M.ESTUDIANT) AS NUM_ESTUDIANTS
FROM
    prod.MATRICULA M
WHERE
    M.TIPO_ENSNY='FP'
GROUP BY
    M.CENTRE,M.TIPO_ENSNY;
Duración: 2756,812 ms (00:02,757)
gestmat=#
```

Figure 21: Alumnos matriculados en FP Optimización

```
gestmat=# SELECT
    M.CENTRE,
    COUNT(DISTINCT M.ESTUDIANT) AS NUM_ESTUDIANTS
FROM
    prod.MATRICULA M
GROUP BY
    M.CENTRE
ORDER BY NUM_ESTUDIANTS DESC;
Duración: 2977,108 ms (00:02,977)
gestmat=#
```

Figure 22: Densidad de centros Optimización

```
gestmat=# SELECT
    M.ASSIGNATURA,
    M.CENTRE
FROM
    prod.MATRICULA M
GROUP BY
    M.ASSIGNATURA,M.CENTRE
HAVING
COUNT(DISTINCT M.ESTUDIANT) < 4;
                                assignatura                                 | centre
----------------------------------------------------------------------------+---------
 BASES DE DATOS                                                             | 7000649
 CIENCIAS APLICADAS II                                                      | 7002105
 DIBUJO ARTÍSTICO II                                                        | 7015291
 DISPENSACIÓN DE PRODUCTOS PARAFARM.                                        | 7003468
 DISPENSACIÓN DE PRODUCTOS PARAFARM.                                        | 7004679
 EMBELLECIMIENTO DE SUPERFICIES                                             | 7016219
 GESTIÓN DEL MONTAJE Y DEL MANTENIMIENTO DE INSTALACIONES ELÉCTRICAS        | 7015941
 IMPRESIÓN EN FLEXOLOGÍA                                                    | 7003471
 MANTENIMIENTO INSTALACIONES CALORIF                                        | 7700040
 MODELOS DE INTELIGENCIA ARTIFICIAL                                         | 7007929
 OPERACIONES AUXILIARES DE PREPARACIÓN DEL TERRENO, PLANTACIÓN Y SIEMBRA DE CULTIVOS | 7008223
 RECEPCIÓN Y LOGÍSTICA EN LA CLÍNICA DENTAL                                 | 7001150
 TÉCNICAS ADMINISTRATIVAS BÁSICAS                                           | 7008211
(13 filas)

Duración: 2638,302 ms (00:02,638)
```

Figure 23: Asignaturas con menos de 4 alumnos Optimización

```
Duración: 2638,302 ms (00:02,638)
gestmat=# SELECT
    E.DNI AS DNI,
    E.NOM AS NOM,
    E.COGNOM AS COGNOM,
    E.MUNICIPI AS MUNICIPI
FROM
    PROD.ESTUDIANT E
LEFT JOIN
    PROD.MATRICULA M ON E.DNI=M.ESTUDIANT
WHERE M.ESTUDIANT IS NULL
GROUP BY E.DNI;
Duración: 329,139 ms
```

Figure 24: Alumnos no matriculados

```
gestmat=# CREATE VIEW ALU_NO_MATR AS
SELECT
    E.DNI AS DNI,
    E.NOM AS NOM,
    E.COGNOM AS COGNOM,
    E.MUNICIPI AS MUNICIPI
FROM
    PROD.ESTUDIANT E
LEFT JOIN
    PROD.MATRICULA M ON E.DNI=M.ESTUDIANT
WHERE M.ESTUDIANT IS NULL
GROUP BY E.DNI;
CREATE VIEW
Duración: 65,282 ms
gestmat=# SELECT COUNT(DNI) FROM ALU_NO_MATR;
 count
-------
  3977
(1 fila)

Duración: 291,213 ms
```

Figure 25: Alumnos no matriculados Count

```
gestmat=# SELECT
    max(C.ID) AS CODI_CENTRE,
    max(concat(asm.nom,' ',asm.COGNOM)) as nom_estudaint,
    ASM.DNI AS ALUMNE
FROM
    PROD.ALU_NO_MATR ASM
JOIN
    PROD.MUNICIPI M ON ASM.MUNICIPI=M.ID
JOIN
    PROD.R_MUNICIPI_LOCALITAT RML ON RML.MUNICIPI=M.ID
JOIN
    PROD.ADRECA A ON A.LOCALITAT=RML.LOCALITAT
JOIN
    PROD.CENTRE C ON A.CENTRE=C.ID
group by ALUMNE;
Duración: 434,833 ms
```


Figure 26: centro x Cercanía

5.6 Optimización

La optimización en este caso, es la misma planteada en MySQL, de hecho, en Postgres si cuenta con la capacidad de crear vistas materializadas "reales". Haciendo uso de esta estructura veremos los resultados y tiempos nuevos de las consultas

```
gestmat=# EXPLAIN ANALYZE SELECT
    M.CENTRE AS CENTRE,
    M.NUM_ESTUDIANTS
FROM
    PROD.ALU_CENTRE_TE M
WHERE M.TIPO_ENSNY='FP';
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Seq Scan on alu_centre_te m  (cost=0.00..18.59 rows=212 width=12) (actual time=0.012..0.079 rows=212 loops=1)
   Filter: ((tipo_ensny)::text = 'FP'::text)
   Rows Removed by Filter: 635
 Planning Time: 0.049 ms
 Execution Time: 0.096 ms
(5 filas)

Duración: 1,844 ms
```

Figure 27: Alumnos matriculats en FP Optimizacion

```
gestmat=# EXPLAIN ANALYZE SELECT
    M.CENTRE AS CENTRE,
    M.NUM_ESTUDIANTS AS MATRICULATS
FROM
    prod.ALU_X_CENTRE M
ORDER BY MATRICULATS DESC;
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=12.31..12.84 rows=212 width=12) (actual time=0.071..0.079 rows=212 loops=1)
   Sort Key: num_estudiants DESC
   Sort Method: quicksort  Memory: 33kB
   ->  Seq Scan on alu_x_centre m  (cost=0.00..4.12 rows=212 width=12) (actual time=0.039..0.053 rows=212 loops=
 Planning Time: 0.650 ms
 Execution Time: 0.097 ms
(6 filas)

Duración: 1,176 ms
```

Figure 28: Densidad de centros Optimizacion

```
gestmat=# EXPLAIN ANALYZE SELECT
    AXA.ASSIGNATURA AS ASSIGNATURA,
    AXA.CENTRE AS CENTRE,
    C.NOM_CENTRE AS NOM_CENTRE
FROM
    prod.ALU_X_ASSIGN AXA
JOIN
    prod.CENTRE C ON AXA.NUM_ESTUDIANTS<4 AND C.ID=AXA.CENTRE;
                                                          QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=29.01..913.18 rows=16 width=63) (actual time=1.257..4.725 rows=13 loops=1)
   Hash Cond: (axa.centre = c.id)
   ->  Seq Scan on alu_x_assign axa  (cost=0.00..884.12 rows=16 width=39) (actual time=0.201..3.653 rows=13 loop
         Filter: (num_estudiants < 4)
         Rows Removed by Filter: 39677
   ->  Hash  (cost=18.45..18.45 rows=845 width=28) (actual time=1.048..1.049 rows=845 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 59kB
         ->  Seq Scan on centre c  (cost=0.00..18.45 rows=845 width=28) (actual time=0.002..0.940 rows=845 loops
 Planning Time: 0.344 ms
 Execution Time: 4.755 ms
(10 filas)

Duración: 7,455 ms
```

Figure 29: Asignaturas con menos de 4 alumnos Optimizacion

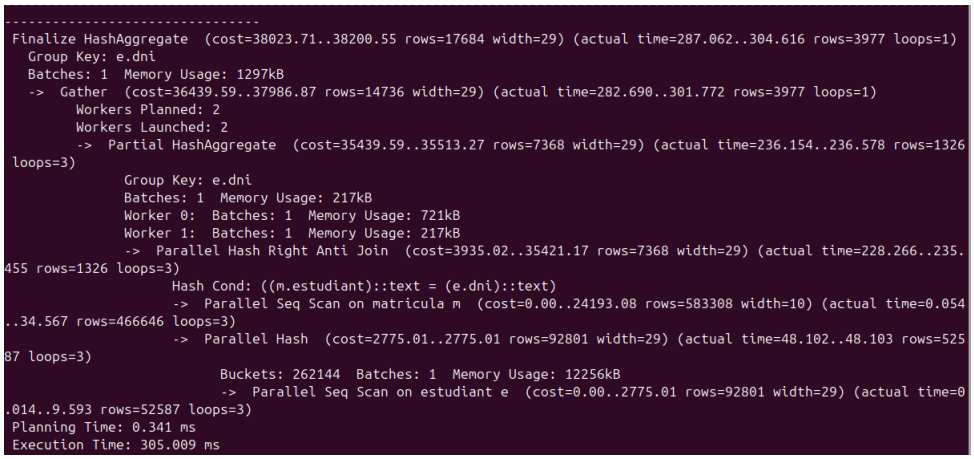

Figure 30: Alumnos no matriculado Optimización

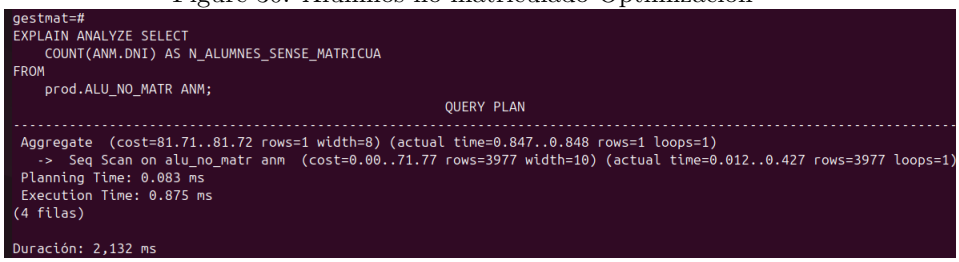

Figure 31: Alumnos no matriculado COUNT Optimización

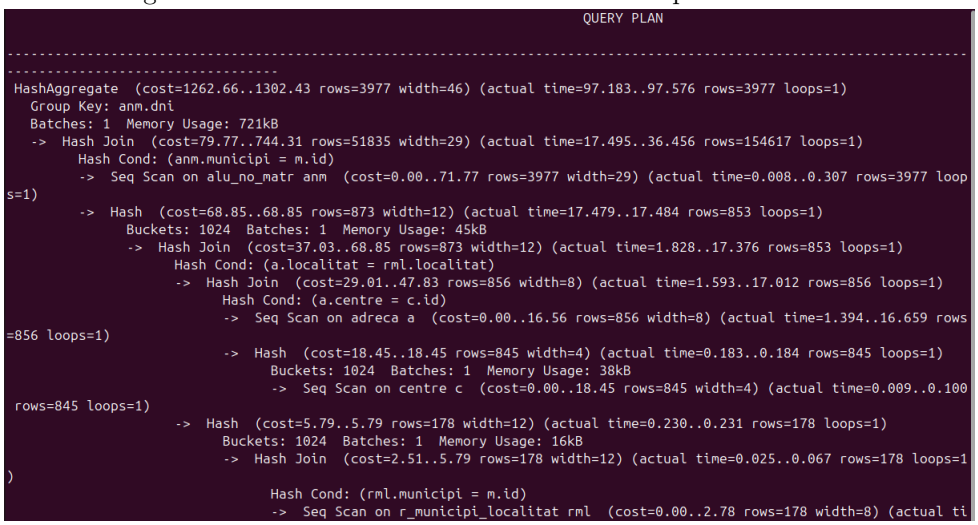

Figure 32: Centros x cervaní Optimización Parte 2

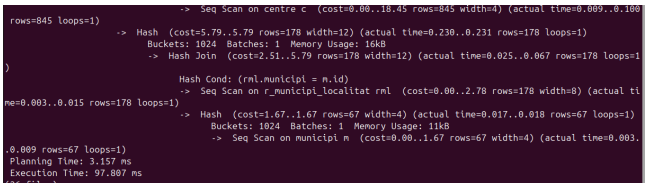

Figure 33: Centros x cercanía Optimización Parte 2

5.7 Comparativa MySQL vs PostgreSQL

- Vistas materializadas: PostgreSQL tiene soporte nativo con REFRESH MATERIALIZED VIEW.MySQL no las soporta nativamente, se emulan con tablas y triggers manuales.
- COUNT(DISTINCT): PostgreSQL lo maneja de forma eficiente, especialmente con índices. MySQL es menos eficiente en tablas grandes o consultas complejas.
- Joins complejos: PostgreSQL optimiza mejor los joins con paralelización y estadísticas actualizadas.MySQL requiere índices bien diseñados; puede ser más lento con múltiples tablas.
- ORDER BY:PostgreSQL usa paralelización y memoria (work_mem), siendo rápido con índices. MySQL funciona bien si las columnas están indexadas, pero puede usar disco si falta memoria.

6 Conclusión

La realización de este trabajo ha sido un proceso largo y con varios rollbacks debido a la gran variedad de casuísticas que se encontraban a lo largo del desarrollo, se ha convertido en un proyecto algo tedioso pero del que se ha aprendido y se han podido aplicar conocimientos tanto de programación como de conceptos propios de base de datos. El poder trabajar tanto en MySQL como en PostgreSQL también ha ayudado a ver las diferencias tanto de sintaxis como de ejecución y rendimiento que pueden tener los diferentes gestor de base de datos.

A nivel personal, espero haber reflejado la dedicación y el tiempo que se la dedicado a este proyecto, el cual es una de los más extensos que he realizado y del que considero se ha trabajado de manera correcta y me encuentro satisfecho con los resultados obtenidos y plasmados en este documento.

7 Bibliografía

- Documentación PostgreSQL
- Documentación MySQL
- Documentación pgloader
- ChatGPT - Herramienta de Apoyo para situaciones puntuales
- StackOverflow - Herramienta de Apoyo para situaciones puntuales