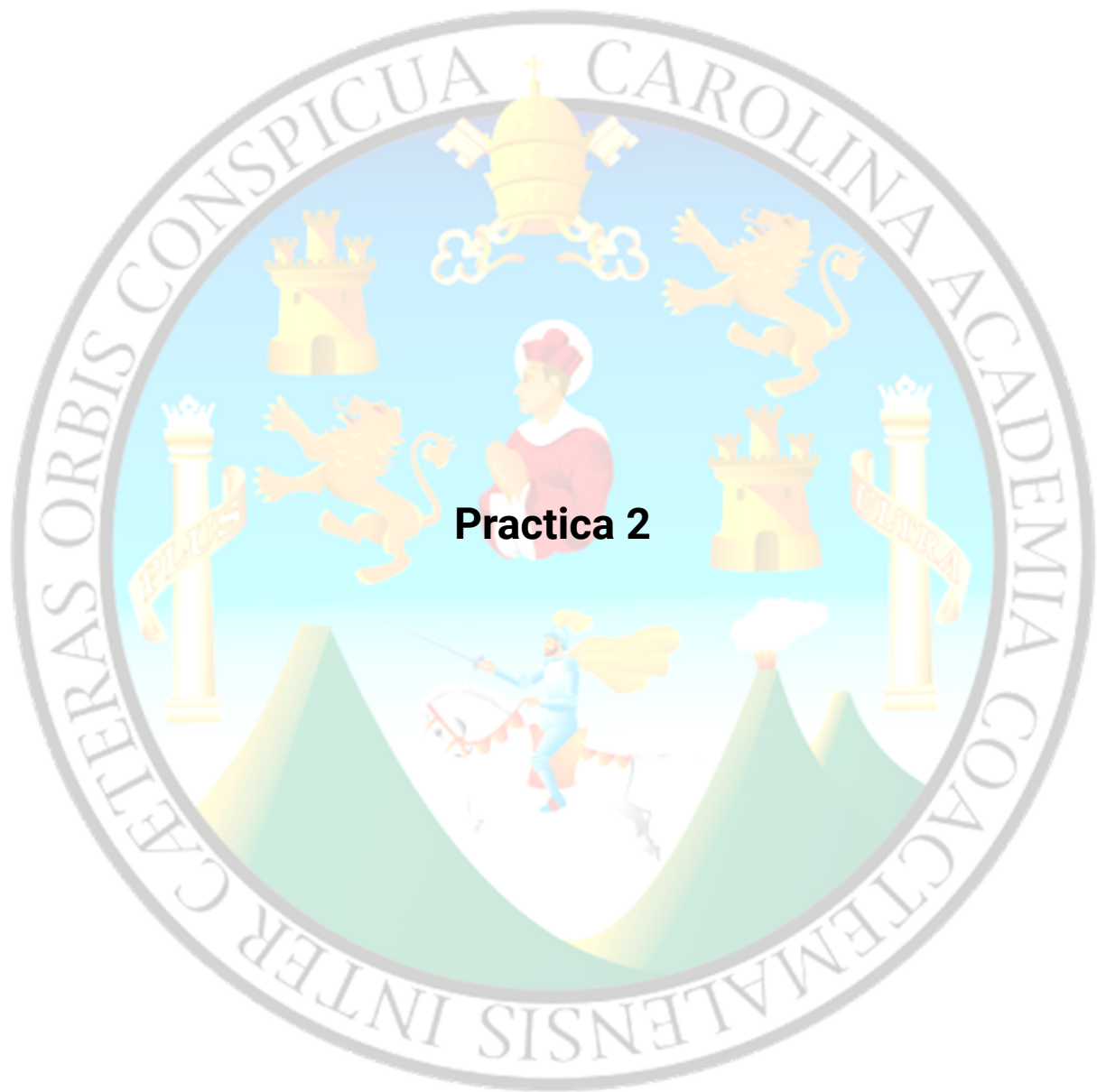


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

LABORATORIO BASES DE DATOS 2



Practica 2

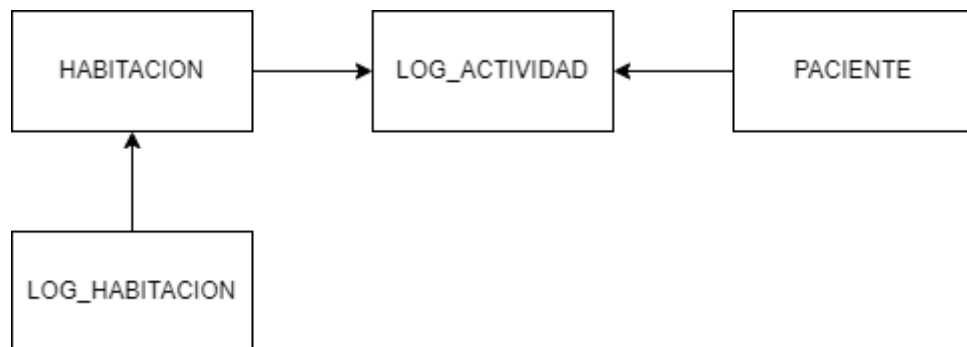
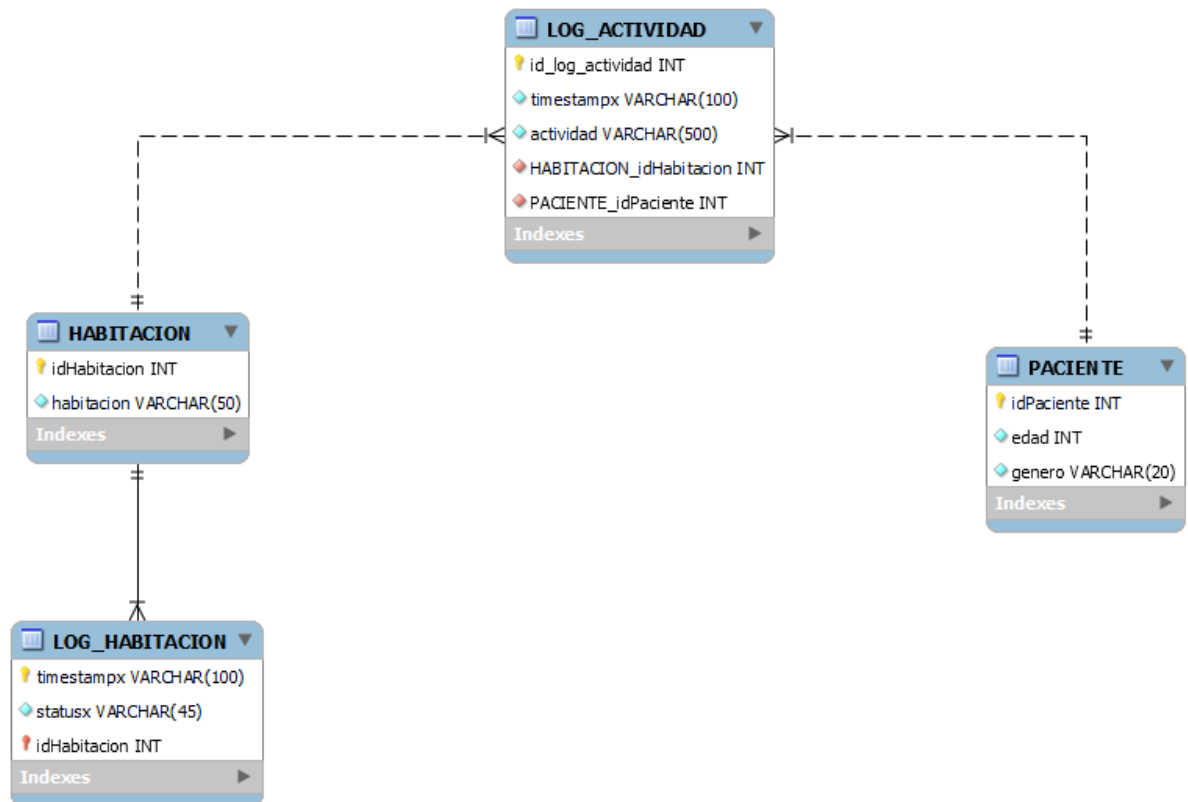
NOMBRES

Luis Andres de la Peña Pineda
Angel Oswaldo Arteaga Garcia
Karen Lisbeth Morales Marroquin

CARNET:

201900450
201901816
201908316

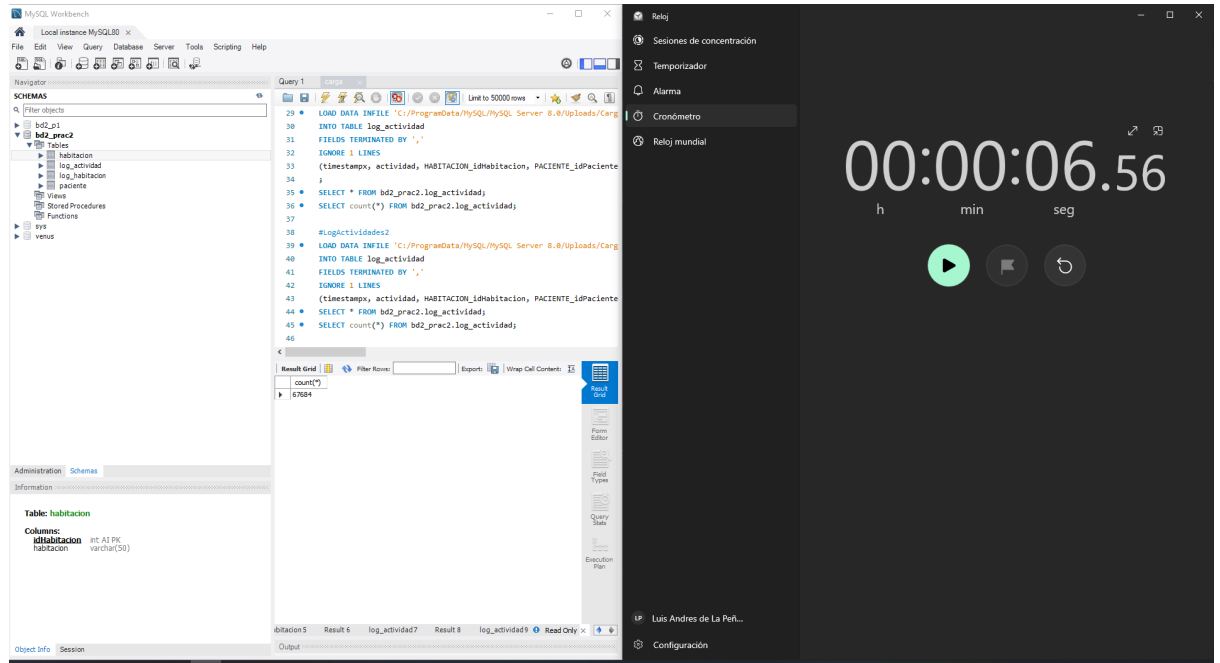
Modelos



Capturas

Cargas:

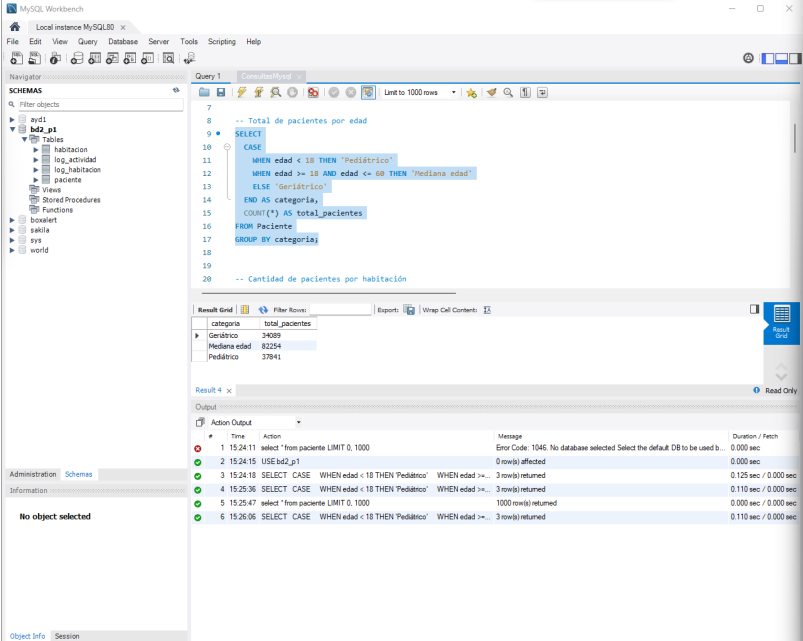
1. MySQL:



2. MongoDB:

Consulta 1:

1. MySQL:



The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema, including tables like 'habitation', 'log_actividad', 'paciente', 'bowling', 'sakila', and 'world'. The main window shows a SQL query titled 'Query 1' with the following code:

```
-- Total de pacientes por edad
SELECT
CASE
WHEN edad < 18 THEN 'Pediatrico'
WHEN edad >= 18 AND edad <= 60 THEN 'Mediana edad'
ELSE 'Geriatrico'
END AS categoria,
COUNT(*) AS total_pacientes
FROM Paciente
GROUP BY categoria;
```

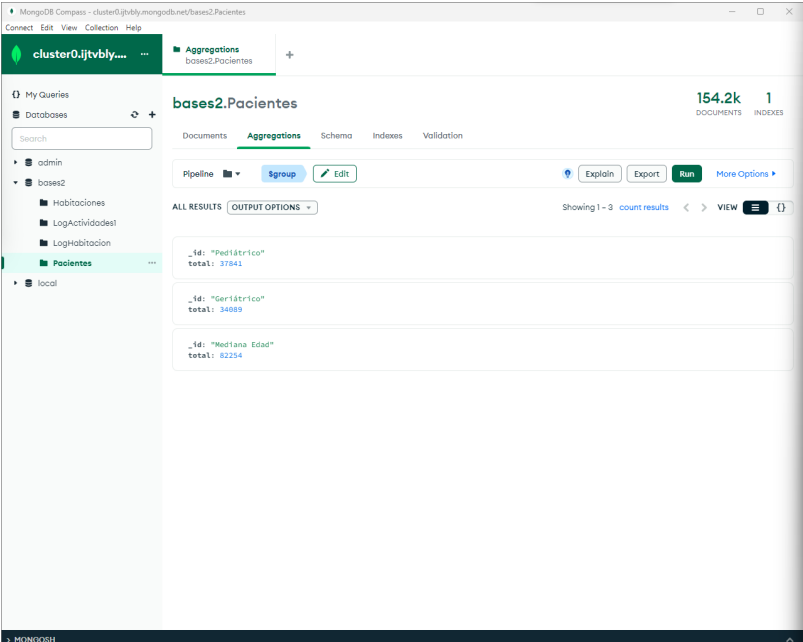
The 'Result Grid' shows the following data:

categoria	total_pacientes
Geriatrico	3409
Mediana edad	82254
Pediatrico	37841

The 'Output' tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	15:24:11	select * from paciente LIMIT 0, 1000	Error Code: 1046: No database selected Select the default DB to be used b...	0.000 sec
2	15:24:15	USE bd2_p1	0 row(s) affected	0.000 sec
3	15:24:18	SELECT CASE WHEN edad < 18 THEN 'Pediatrico' WHEN edad >= 18 AND edad <= 60 THEN 'Mediana edad' ELSE 'Geriatrico' END AS categoria, COUNT(*) AS total_pacientes FROM Paciente GROUP BY categoria;	3 row(s) returned	0.125 sec / 0.000 sec
4	15:25:36	SELECT CASE WHEN edad < 18 THEN 'Pediatrico' WHEN edad >= 18 AND edad <= 60 THEN 'Mediana edad' ELSE 'Geriatrico' END AS categoria, COUNT(*) AS total_pacientes FROM Paciente GROUP BY categoria;	3 row(s) returned	0.110 sec / 0.000 sec
5	15:25:47	select * from paciente LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec
6	15:26:06	SELECT CASE WHEN edad < 18 THEN 'Pediatrico' WHEN edad >= 18 AND edad <= 60 THEN 'Mediana edad' ELSE 'Geriatrico' END AS categoria, COUNT(*) AS total_pacientes FROM Paciente GROUP BY categoria;	3 row(s) returned	0.110 sec / 0.000 sec

2. MongoDB:



The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure, including collections like 'Habitaciones', 'LogActividades', 'LogHabitacion', and 'Pacientes'. The main window shows the 'bases2.Pacientes' collection with an aggregation pipeline. The pipeline is defined as:

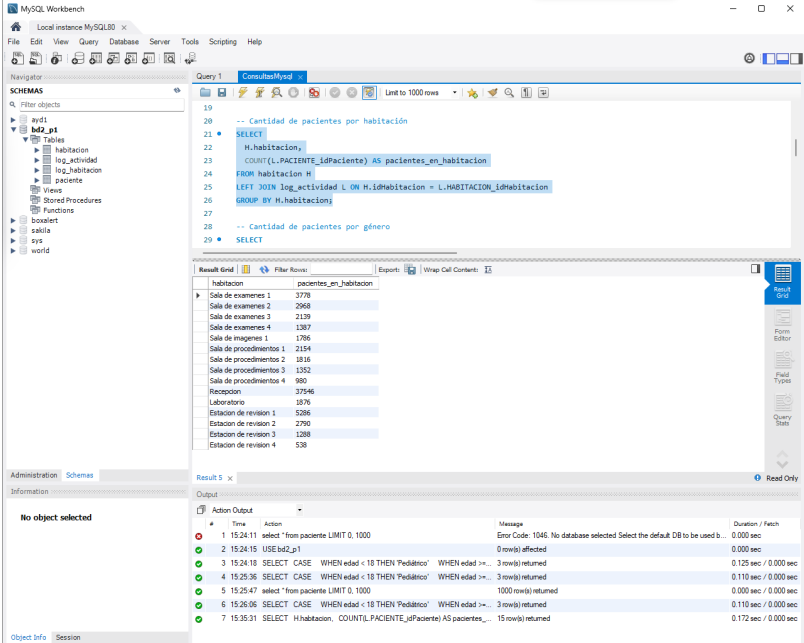
```
{
  "$group": {
    "_id": "$categoria",
    "total": { "$sum": "$total_pacientes" }
  }
}
```

The 'ALL RESULTS' tab shows the following data:

_id	total
"Pediatrico"	37841
"Geriatrico"	3409
"Mediana Edad"	82254

Consulta 2:

1. MySQL:



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

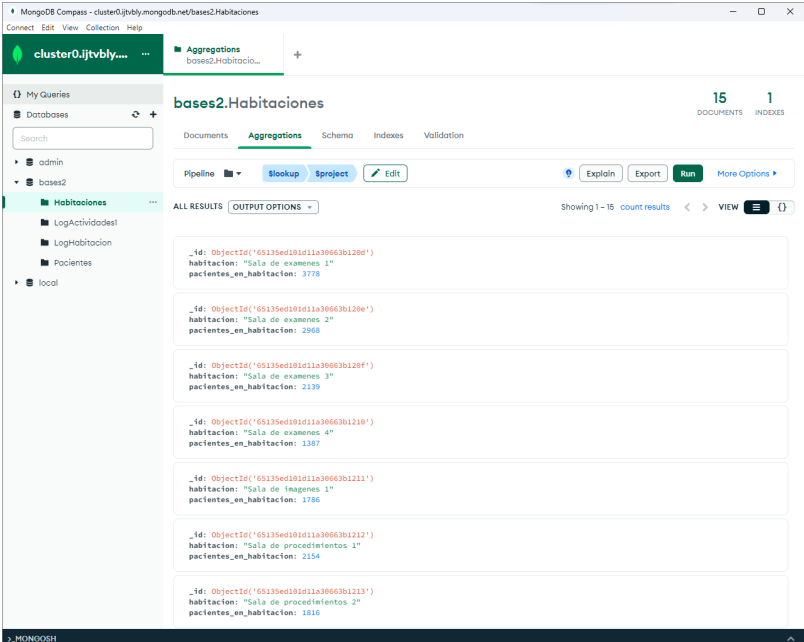
```
-- Cantidad de pacientes por habitación
21 SELECT
22   H.habitacion
23   COUNT(L.PACIENTE_idPaciente) AS pacientes_en_habitacion
24 FROM habitacion H
25 LEFT JOIN log_actividad L ON H.idHabitacion = L.HABITACION_idHabitacion
26 GROUP BY H.habitacion
27
-- Cantidad de pacientes por género
28 SELECT
```

The Results grid shows the following data:

habitacion	pacientes_en_habitacion
Sala de exámenes 1	3778
Sala de exámenes 2	2968
Sala de exámenes 3	2139
Sala de exámenes 4	1387
Sala de imágenes 1	1786
Sala de procedimientos 1	2154
Sala de procedimientos 2	1816
Sala de procedimientos 3	1352
Sala de procedimientos 4	980
Recepcion	37546
Laboratorio	1876
Estacion de revision 1	5286
Estacion de revision 2	2790
Estacion de revision 3	1288
Estacion de revision 4	538

The Output tab shows the execution of the query, including the message: "Error Code: 1046: No database selected Select the default DB to be used b...".

2. MongoDB:



The screenshot shows the MongoDB Compass interface. The aggregation pipeline is defined as follows:

```
1 $match: { "habitacion": "Sala de exámenes 1" }
2 $project: { "pacientes_en_habitacion": "$count" }
```

The Results tab shows the following data:

_id	pacientes_en_habitacion
ObjectId('65135ed181d11a30663b129d')	3778
ObjectId('65135ed181d11a30663b129e')	2968
ObjectId('65135ed181d11a30663b129f')	2139
ObjectId('65135ed181d11a30663b1210')	1387
ObjectId('65135ed181d11a30663b1211')	1786
ObjectId('65135ed181d11a30663b1212')	2154
ObjectId('65135ed181d11a30663b1213')	1816

Consulta 3:

1. MySQL:

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
25 LEFT JOIN log_actividad L ON H.IdHabitacion = L.HABITACION_IdHabitacion
26 GROUP BY H.habitacion;
27
28 -- Cantidad de pacientes por género
29 SELECT
30     genero,
31     COUNT(*) AS total_pacientes
32 FROM Paciente
33 GROUP BY genero;
34
35 -- Top 5 edades más atendidas
36 SELECT
37     edad,
38     COUNT(*) AS total_atendidos
```

The results window shows the following data:

genero	total_pacientes
Otro	5445
Femenino	75214
Masculino	73525

Next to the MySQL Workbench is a digital clock showing 00:00:00.16.

2. MongoDB:

The screenshot shows the MongoDB Compass interface. The aggregation pipeline is defined as follows:

```
7 15.35.31 SELECT $habilitacion, COUNT($PACIENTE.$Paciente) AS paciente...
8 15.40.30 SELECT $genero, COUNT(*) AS total_pacientes FROM Paciente GRO... 3 rows returned
```

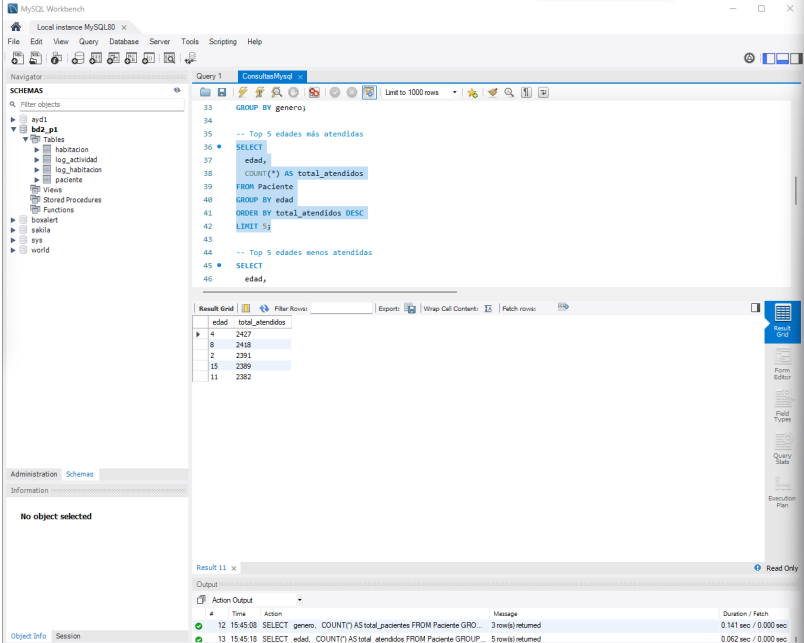
The results window shows the following data:

_id	total
"Masculino"	73525
"Femenino"	75214
"Otro"	5445

Next to the MongoDB Compass is a digital clock showing 00:00:00.42.

Consulta 4:

1. MySQL:



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

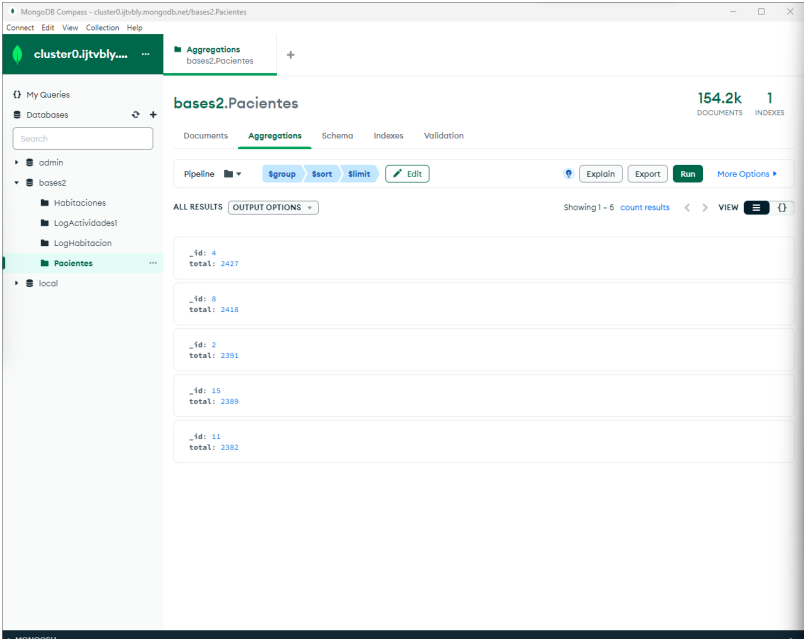
```
33 GROUP BY genero;
34
35 -- Top 5 edades más atendidas
36 SELECT
37   edad,
38   COUNT(*) AS total_atendidos
39 FROM Paciente
40 GROUP BY edad
41 ORDER BY total_atendidos DESC
42 LIMIT 5;
43
44 -- Top 5 edades menos atendidas
45 SELECT
46   edad,
```

The results window displays the following data:

edad	total_atendidos
4	2427
8	2418
2	2391
15	2389
11	2382

On the right, a digital clock shows 00:00:00.29.

2. MongoDB:



The screenshot shows the MongoDB Compass interface. The aggregation pipeline is defined as follows:

```
{
  "$group": {
    "_id": "$edad",
    "total": { "$sum": "$total_atendidos" }
  }
}
```

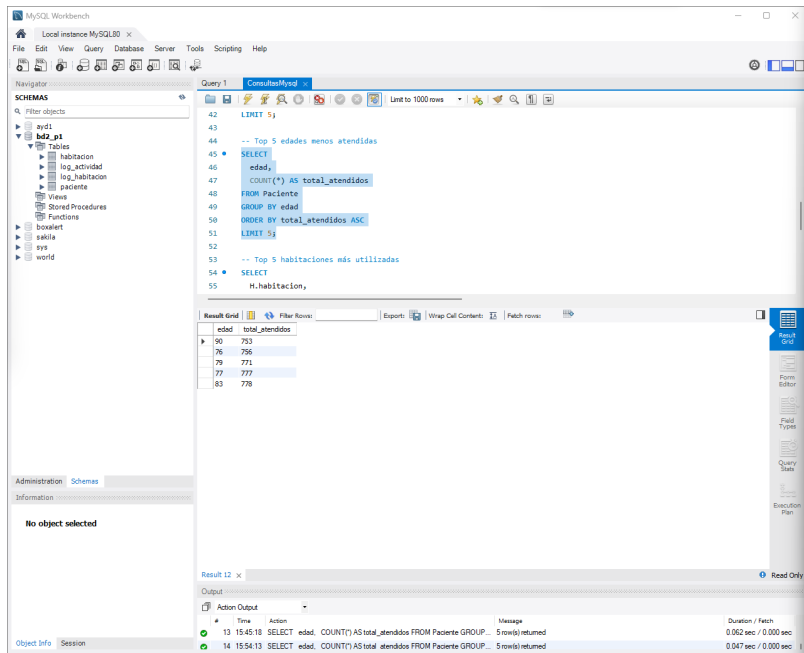
The results window displays the following data:

_id	total
4	2427
8	2418
2	2391
15	2389
11	2382

On the right, a digital clock shows 00:00:00.45.

Consulta 5:

1. MySQL:



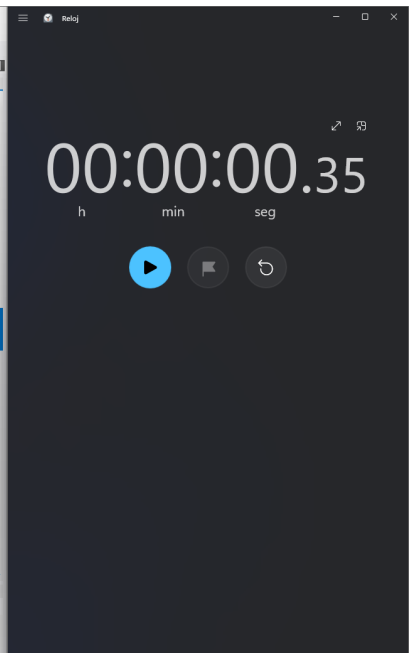
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
42 LIMIT 5)
43
44 -- Top 5 edades menos atendidas
45 SELECT
46   edad,
47   COUNT(*) AS total_atendidos
48 FROM Paciente
49 GROUP BY edad
50 ORDER BY total_atendidos ASC
51 LIMIT 5)
52
53 -- Top 5 habitaciones más utilizadas
54 SELECT
55   H.habitacion,
```

The results window displays the following data:

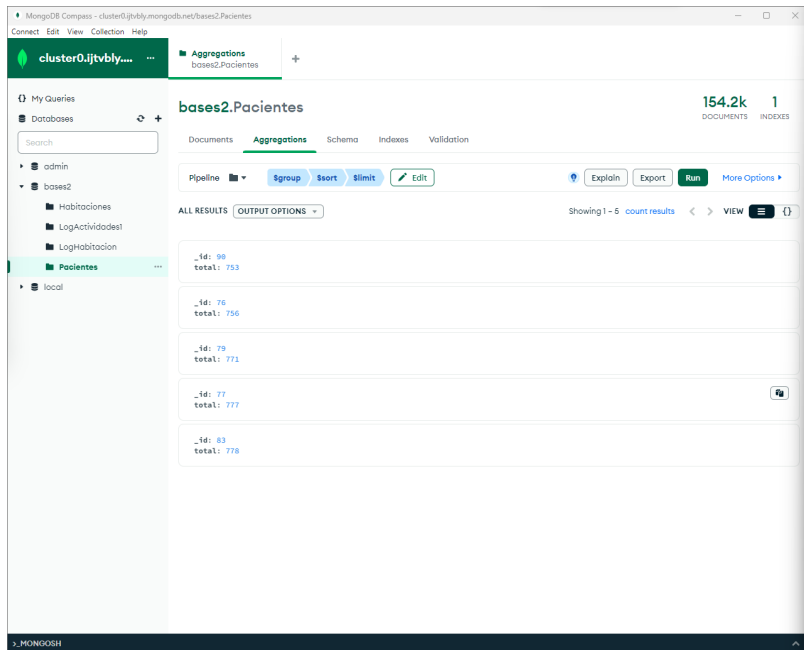
edad	total_atendidos
90	753
76	756
78	771
77	777
83	778

The status bar at the bottom indicates that 14 rows were returned.



The digital clock shows a time of 00:00:00.35, with units for hours (h), minutes (min), and seconds (seg).

2. MongoDB:



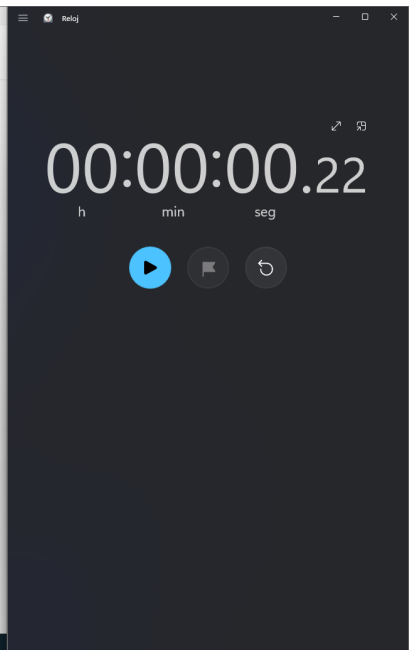
The screenshot shows the MongoDB Compass interface. The aggregation pipeline is defined as follows:

```
{
  "$group": {
    "_id": "edad",
    "total": "$COUNT"
  }
}
```

The results window displays the following data:

_id	total
90	753
76	756
78	771
77	777
83	778

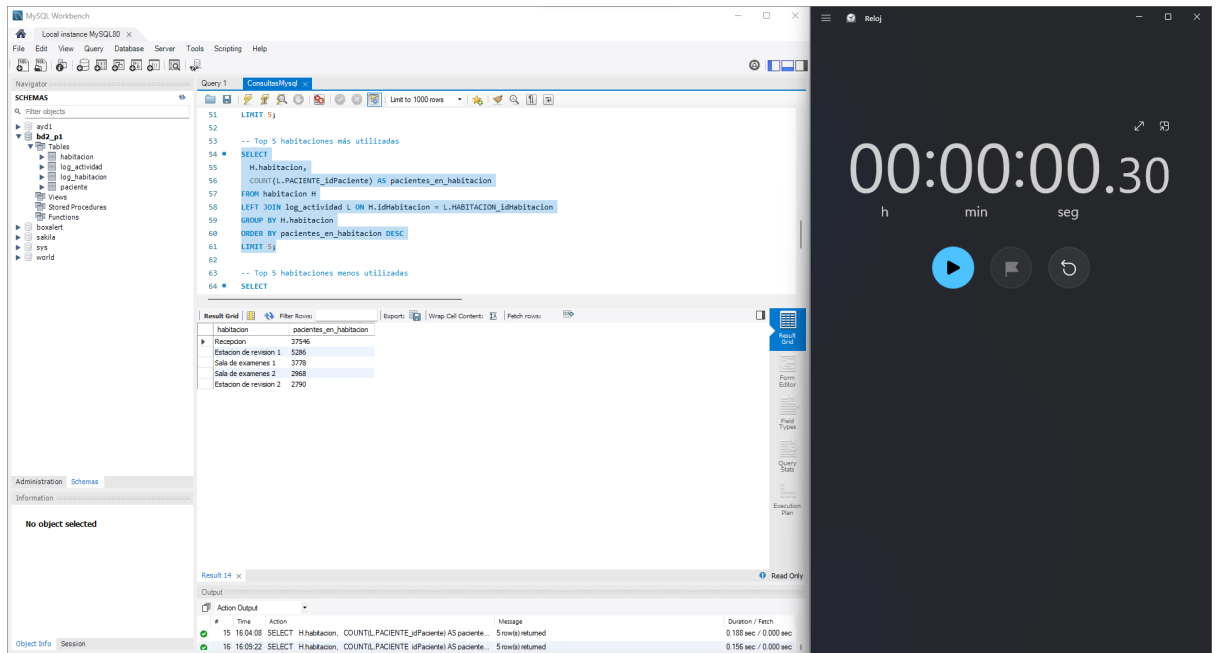
The status bar at the bottom indicates that 5 rows were returned.



The digital clock shows a time of 00:00:00.22, with units for hours (h), minutes (min), and seconds (seg).

Consulta 6:

1. MySQL:

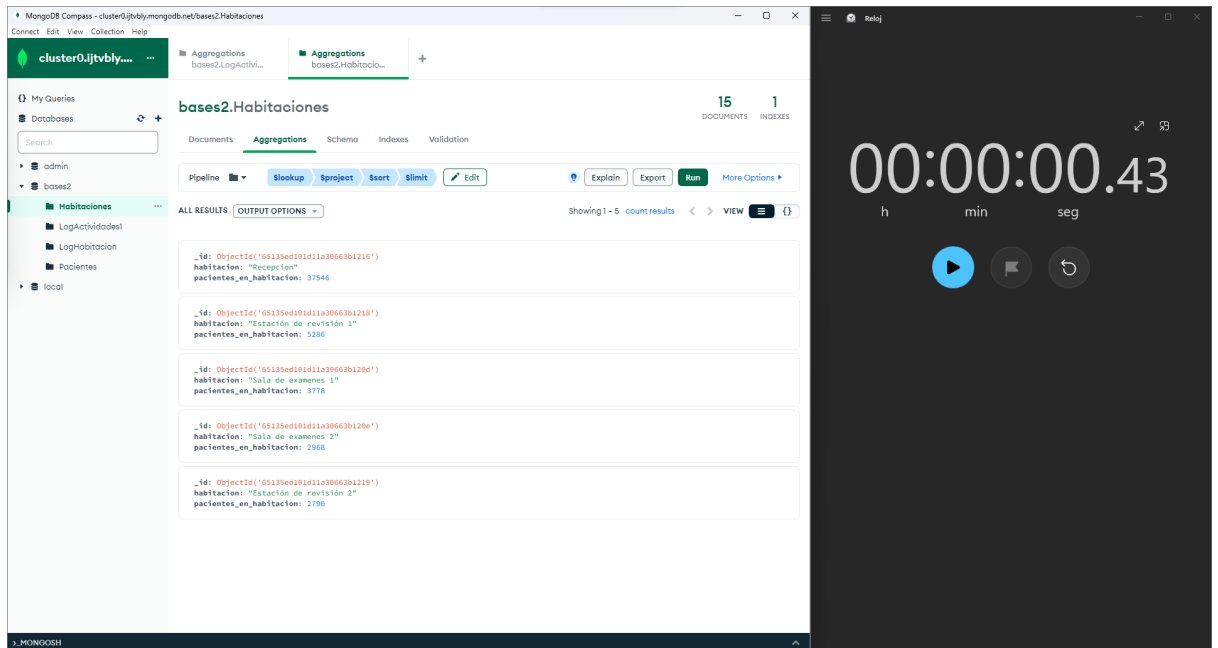


The screenshot shows the MySQL Workbench interface. The query editor contains a SQL query that uses a subquery to find the top 5 most used rooms and then joins it with the log table to find the top 5 most used rooms. The results are displayed in a table with columns 'habitacion' and 'pacientes_en_habitacion'.

```
51 LIMIT 5)
52
53 -- Top 5 habitaciones más utilizadas
54 SELECT
55   H.habitacion,
56   COUNT(L.PACIENTE_idPaciente) AS pacientes_en_habitacion
57 FROM habitacion H
58 LEFT JOIN log_actividad L ON H.idHabitacion = L.HABITACION_idHabitacion
59 GROUP BY H.habitacion
60 ORDER BY pacientes_en_habitacion DESC
61 LIMIT 5)
62
63 -- Top 5 habitaciones menos utilizadas
64 SELECT
```

habitacion	pacientes_en_habitacion
Recepcion	37546
Estacion de revision 1	5286
Sala de exámenes 1	3778
Sala de exámenes 2	2968
Estacion de revision 2	2790

2. MongoDB:



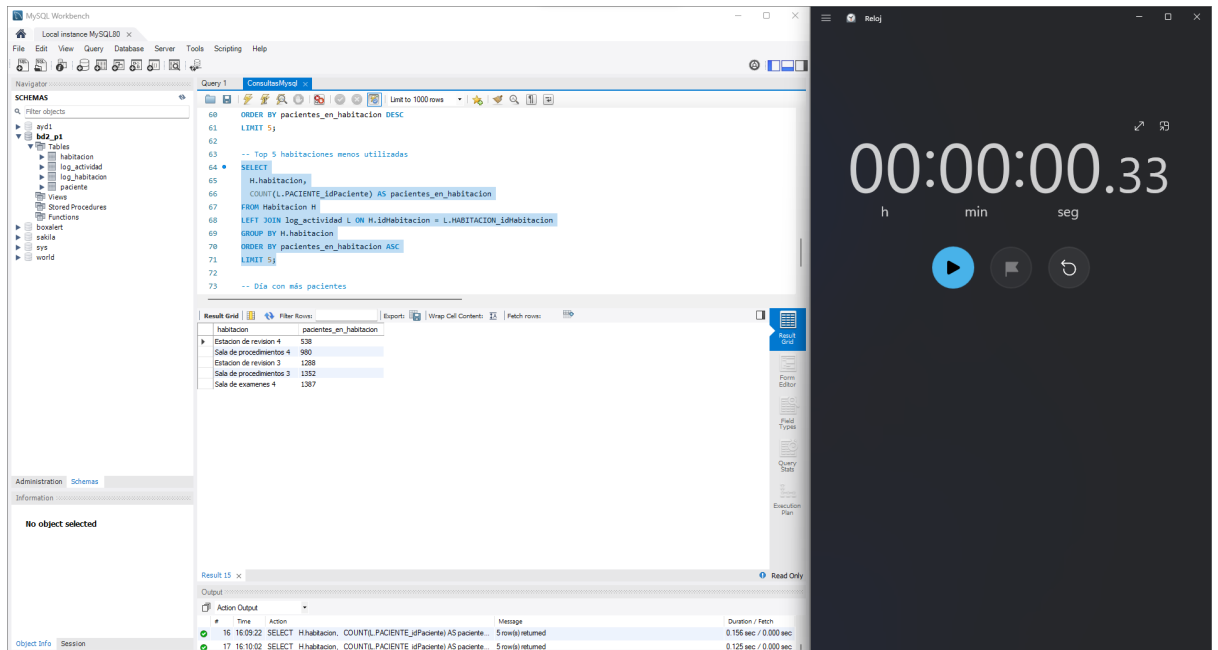
The screenshot shows the MongoDB Compass interface. The aggregation pipeline is defined in the 'Aggregations' tab. The pipeline consists of a single stage 'Aggregation' with a query that uses a subquery to find the top 5 most used rooms and then joins it with the log table to find the top 5 most used rooms. The results are displayed in a table with columns '_id', 'habitacion', and 'pacientes_en_habitacion'.

```
15 16:04:08 SELECT H.habitacion, COUNT(L.PACIENTE_idPaciente) AS paciente... 5 rows returned
16 16:09:22 SELECT H.habitacion, COUNT(L.PACIENTE_idPaciente) AS paciente... 5 rows returned
```

_id	habitacion	pacientes_en_habitacion
ObjectId('65135ed181d11a380663b1216')	Recepcion	37546
ObjectId('65135ed181d11a380663b1218')	Estacion de revision 1	5286
ObjectId('65135ed181d11a380663b129d')	Sala de exámenes 1	3778
ObjectId('65135ed181d11a380663b129e')	Sala de exámenes 2	2968
ObjectId('65135ed181d11a380663b1219')	Estacion de revision 2	2790

Consulta 7:

1. MySQL:



The screenshot shows the MySQL Workbench interface with a query window and a results table. The query is as follows:

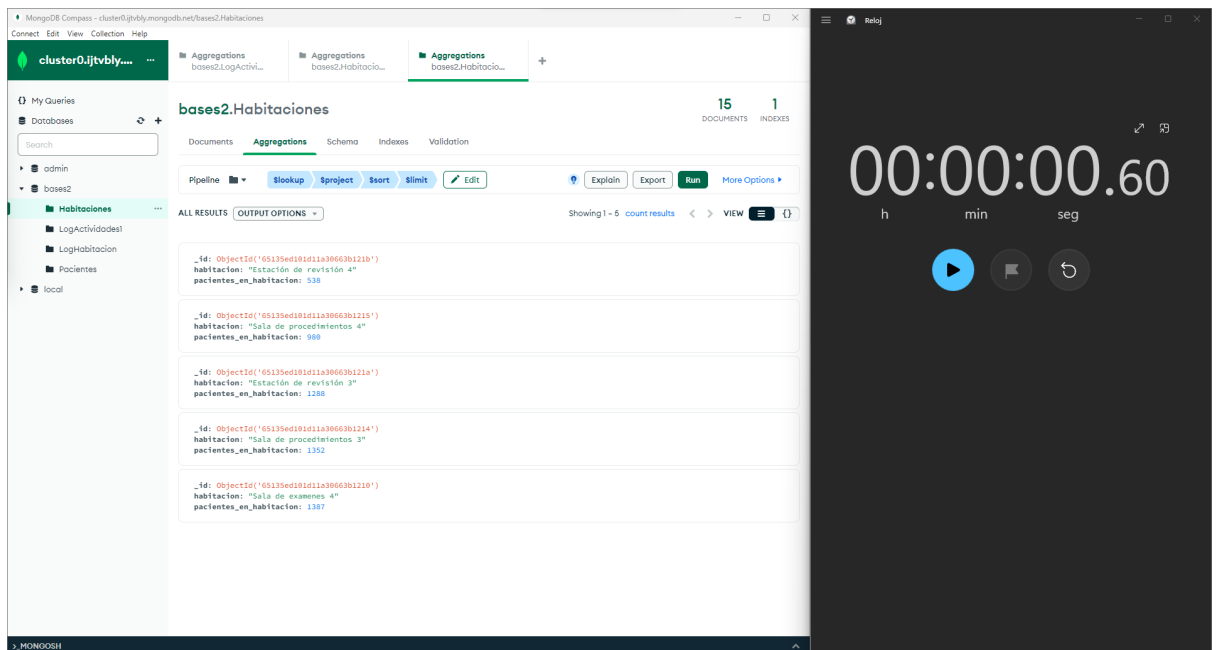
```
60 ORDER BY pacientes_en_habitacion DESC
61 LIMIT 5;
62
63 -- Top 5 Habitaciones menos utilizadas
64 SELECT
65     H.habitacion,
66     COUNT(L.PACIENTE_idPaciente) AS pacientes_en_habitacion
67 FROM Habitacion H
68 LEFT JOIN log_actividad L ON H.idHabitacion = L.HABITACION_idHabitacion
69 GROUP BY H.habitacion
70 ORDER BY pacientes_en_habitacion ASC
71 LIMIT 5;
72
73 -- Día con más pacientes
```

The results table shows the following data:

habitacion	pacientes_en_habitacion
Estacion de revision 4	538
Sala de procedimientos 4	990
Estacion de revision 3	1288
Sala de procedimientos 3	1352
Sala de exámenes 4	1387

Next to the MySQL Workbench is a digital clock showing 00:00:00.33.

2. MongoDB:



The screenshot shows the MongoDB Compass interface with an aggregation pipeline. The pipeline is as follows:

```
Pipeline
1. $lookup:
  from: pacientes_en_habitacion
  let: h = $hab
  pipeline:
    - $match: { "habitacion": "$hab" }
  as: p
2. $project:
  _id: "$hab"
  pacientes_en_habitacion: "$p.count"
3. $sort:
  pacientes_en_habitacion: -1
4. $limit:
  limit: 5
```

The results show the following data:

_id	pacientes_en_habitacion
Estacion de revision 4	538
Sala de procedimientos 4	990
Estacion de revision 3	1288
Sala de procedimientos 3	1352
Sala de exámenes 4	1387

Next to the MongoDB Compass is a digital clock showing 00:00:00.60.

Consulta 8:

1. MySQL:

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists databases including 'BD2_P1'. The 'Query' pane shows a SQL query that filters for a specific date and orders results by the number of records. The 'Result Grid' displays the query results, showing a single row for the date '2021-06-04' with a count of 1489. A 'World Clock' (Reloj mundial) window is overlaid on the right, showing a timer at 00:00.77. The 'Action Output' pane at the bottom shows the execution progress of the query.

totalPacientes	fecha
1489	2021-06-04

2. MongoDB:

The screenshot shows the MongoDB Compass interface. The 'My Queries' pane on the left shows a query for the 'bases2.LogActividades1' collection. The 'Query Results' pane shows the query results, displaying a single row for the date '2021-06-04' with a count of 1489. A 'World Clock' (Reloj) window is overlaid on the right, showing a timer at 00:00:00.33. The 'Query Results' pane also shows the execution progress of the query.

totalPacientes	fecha
1489	"2021-06-04"

Conclusión y justificación

La comparación de tiempos entre MySQL y MongoDB en las consultas realizadas es un aspecto crucial para determinar cuál de las dos bases de datos es más adecuada para las necesidades de la clínica médica. Estos resultados proporcionan información valiosa para elegir una base de datos sobre la otra. Analizando los datos de tiempo al ejecutar podemos observar que la mayoría son más rápidas en obtener los resultados en MySQL. podrías concluir que MySQL es más adecuado para las necesidades actuales, pero que MongoDB puede ser una opción valiosa a largo plazo a medida que la clínica crezca y sus necesidades evolucionen.

Repositorio

https://github.com/anddelap/BD2S22023_Grupo_1