

Universidad De San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Sistemas de Bases de Datos 2

## **PRACTICA 2**

**Juan Antonio Solares Samayoa**  
**Carnet 201800496**

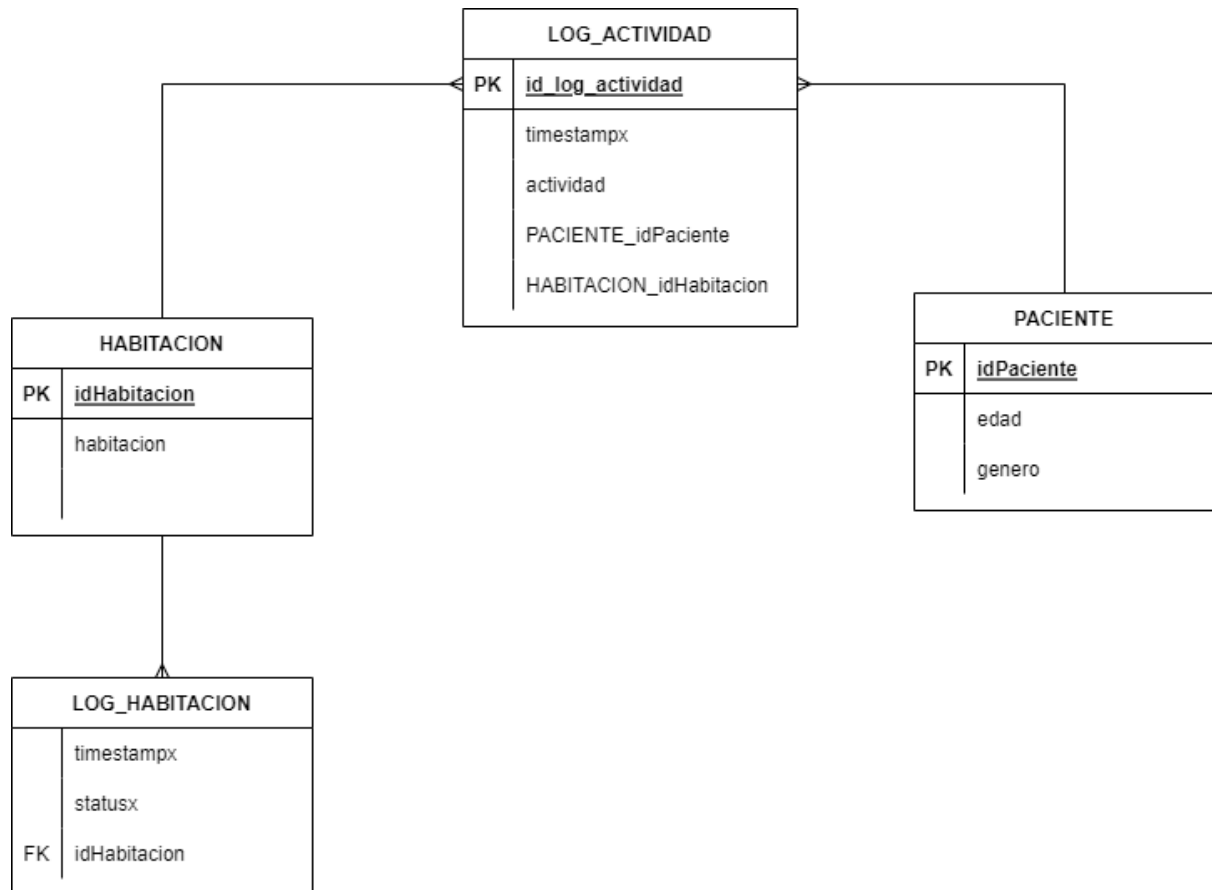
**Juan Diego Alvarado Salguero**  
**Carnet 201807335**

**Erick José André Villatoro Revolorio**  
**Carnet 201900907**

**21 de octubre de 2022**

# Consultas Mysql

## Explicación del modelo utilizado en la base de datos



Para el modelo de la base de datos de mysql se utilizó el siguiente modelo, Utilizando 4 tablas únicamente teniendo tablas con una relación de 1 a muchos como por ejemplo la tabla de “LOG ACTIVIDAD “ con las tablas de “HABITACIÓN” y “PACIENTE” ¿porque se manejo este modelo? para contestar esta interrogante es necesario comprender que el modelo de negocios planteado para este proyecto no requiere de un modelo con muchas tablas o con muchos más atributos, en pocas palabras no se necesita de un modelo muy complejo, otro motivo del porqué no utilizar un modelo tan complejo con muchas más tablas es por el simple hecho de que los archivos de entrada proporcionados no representan un grado de complejidad alto , dando como resultado que al utilizar un modelo de 4 tablas resulte ser bastante práctico para poder dar solución al problema planteado de una manera rápida y sencilla.

Por otro lado para MongoDB no se hizo uso de ese modelo , debido a que MONGODB no es una base de datos relacional , es por eso que únicamente se usaron diferentes colecciones por cada archivo proporcionado.

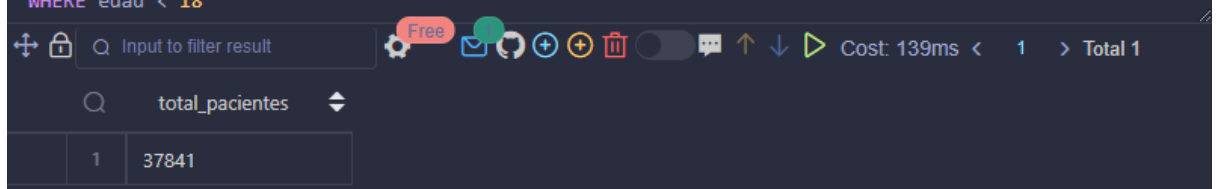
## Resultados de las consultas

1. Total de pacientes que llegan a la clínica por edad catalogados por las siguientes categorías

a. pediátrico

### MySQL

```
# 1. . Total de pacientes que llegan a la clínica por edad catalogados por las siguientes categorías
# a. Pediátrico: menores de 18 años
SELECT COUNT(*) total_pacientes
FROM PACIENTE
WHERE edad < 18
```



The screenshot shows the MySQL query results. The query is: `SELECT COUNT(*) total_pacientes FROM PACIENTE WHERE edad < 18`. The results table has one row with the value 37841 for the column total\_pacientes. The cost of the query is 139ms.

	total_pacientes
1	37841

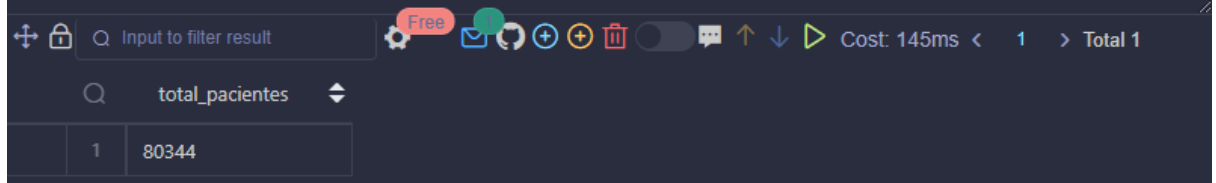
### Mongodb

```
> db.Paciente.find({
  "edad": {"$lt": 18}
}).count();
< 37841
```

b. Mediana edad

### MySQL

```
# b. Mediana edad: entre 18 y 60 años
SELECT COUNT(*) total_pacientes
FROM PACIENTE
WHERE edad >= 18 and edad < 60
```



The screenshot shows the MySQL query results. The query is: `SELECT COUNT(*) total_pacientes FROM PACIENTE WHERE edad >= 18 and edad < 60`. The results table has one row with the value 80344 for the column total\_pacientes. The cost of the query is 145ms.

	total_pacientes
1	80344

### Mongodb

```

> db.Paciente.find({
  "$and": [{
    "edad":{ "$gte" : 18 }
  },{ "edad":{ "$lt" : 60 }
  }]
}).count();
< 80344

```

c. Geriátrico

**Mysql**

```

# c. Geriatrico: mayores de 60 años
SELECT COUNT(*) as total_pacientes
FROM PACIENTE
WHERE edad >= 60

```

Free

Cost: 160ms < 1 > Total 1

	total_pacientes
1	35999

**Mongodb**

```

> db.Paciente.find({
  "edad":{ "$gte" : 60 }
})
).count();
< 35999

```

2. Cantidad de personas que pasan por cada habitación

**MySQL**

```
# 2. Cantidad de pacientes que pasan por cada habitacion
SELECT
    distinct
    habitacion,
    count(*) num_veces
from LOG_ACTIVIDAD
    JOIN HABITACION h
    on LOG_ACTIVIDAD.HABITACION_idHabitacion = h.idHabitacion
GROUP BY
    HABITACION_idHabitacion
ORDER BY num_veces
```

Input to filter result
 Free
 Cost: 264ms < 1 >

<input checked="" type="checkbox"/>	Q	habitacion	num_veces
	1	Estación de revisión 4	538
	2	Sala de procedimientos 4	980
	3	Estación de revisión 3	1288
	4	Sala de procedimientos 3	1352
	5	Sala de exámenes 4	1387
	6	Sala de imagenes 1	1786
	7	Sala de procedimientos 2	1816
	8	Laboratorio	1876
	9	Sala de exámenes 3	2139
	10	Sala de procedimientos 1	2154
	11	Estación de revisión 2	2790
	12	Sala de exámenes 2	2968
	13	Sala de exámenes 1	3778
	14	Estación de revisión 1	5286
	15	Recepcion	37546

## Mongodb

```
C: > Users > javil > OneDrive - Facultad de Ingeniería de la Universidad de San Carlos de Guatemala > I
3 use(database);
4
5 db.getCollection('LogActividades').aggregate([
6
7     {
8         $group: {
9             _id: '$idHabitacion',
10            Habitacion: {
11                $first: '$Habitacion'
12            },
13            count: {
14                $sum: 1
15            }
16        },
17        {
18            $sort: {
19                _id: 1
20            }
21        }
22    ])

```

```
1 [
2
3     {
4         "_id": 1,
5         "Habitacion": "Sala de exámenes 1",
6         "count": 3778
7     },
8     {
9         "_id": 2,
10        "Habitacion": "Sala de exámenes 2",
11        "count": 2968
12    },
13    {
14        "_id": 3,
15        "Habitacion": "Sala de exámenes 3",
16        "count": 2139
17    },
18    {
19        "_id": 4,
20        "Habitacion": "Sala de exámenes 4",
21        "count": 1387
22    },
23    {
24        "_id": 5,
25        "Habitacion": "Sala de imágenes 1",
26        "count": 1786
27    },
28    {
29        "_id": 6,
30        "Habitacion": "Sala de procedimientos 1",
31        "count": 2154
32    },
33    {
34        "_id": 7,
35        "Habitacion": "Sala de procedimientos 2",
36        "count": 1816
37    },
38    {
39        "_id": 8,
40        "Habitacion": "Sala de procedimientos 3",
41        "count": 1352
42    },
43    ]

```

3. Cantidad de pacientes que que pasan por cada habitación, agrupados por género

## Mysql

```
# 3. Cantidad de pacientes que llegan a la clínica, agrupados por género
SELECT distinct genero, count(*) as num
FROM LOG_ACTIVIDAD
JOIN PACIENTE p on LOG_ACTIVIDAD.PACIENTE_idPaciente = p.idPaciente
GROUP BY genero

```

	genero	num
1	Masculino	31720
2	Femenino	33756
3	Otro	2208

## Mongodb

```

Guatemala > Documentos > Clases > Bases2 > BD2_Grupo_2 > Practica 2 > ConsultasMongo > Consulta3 > consulta.mongod
Currently connected to cluster0.pegwrr0.mongodb.net. Click here to change connection.

1 use('practica2');
2
3
4
5 const aggregation = [
6   {
7     $project: {
8       Genero: 1
9     }
10  },
11  {
12    $sortByCount: '$Genero'
13  }
14 ]
15
16
17 db.LogActividades.aggregate(aggregation);
18
19
20
21 [
22   {
23     "_id": "Femenino",
24     "count": 33754
25   },
26   {
27     "_id": "Masculino",
28     "count": 31732
29   },
30   {
31     "_id": "Otro",
32     "count": 2198
33   }
34 ]

```

#### 4. Top 5 edades más atendidas en la clínica

### Mysql

# 4. Top 5 edades más atendidas en la clínica

```

SELECT distinct edad, count(*) as num
FROM LOG_ACTIVIDAD
JOIN PACIENTE p on LOG_ACTIVIDAD.PACIENTE_idPaciente = p.idPaciente
GROUP BY edad
ORDER BY num DESC LIMIT 5

```

Free

Input to filter result

Cost: 196m

	edad	num
1	6	1872
2	2	1698
3	15	1664
4	11	1606
5	10	1592

### Mongodb

```

Je Guatemala > Documentos > Clases > Bases2 > BD2_Grupo_2 > Practica 2 > ConsultasMongo > Consulta4 > consulta.mongodb
Currently connected to cluster0.oegwrr0.mongodb.net. Click here to change connection.

1
2 use('practica2');
3
4
5 const aggregation = [
6   {
7     $group: {
8       _id: '$Edad',
9       count: {
10        $sum: 1
11      }
12    }
13  },
14  {
15    $sort: {
16      count: -1
17    }
18  },
19  {
20    $limit: 5
21  }
22 ]

23
24
25 db.LogActividades.aggregate(aggregation);
26

```

```

1 {
2   "_id": 6,
3   "count": 1872
4 },
5 {
6   "_id": 2,
7   "count": 1698
8 },
9 {
10  "_id": 15,
11  "count": 1664
12 },
13 {
14  "_id": 11,
15  "count": 1606
16 },
17 {
18  "_id": 10,
19  "count": 1592
20 }
21
22 ]

```

## 5. Top 5 edades menos atendidas en la clínica

### Mysql

```

# 5. Top 5 edades menos atendidas en la clinica
SELECT edad, count(*) as num
FROM LOG_ACTIVIDAD
JOIN PACIENTE p on LOG_ACTIVIDAD.PACIENTE_idPaciente = p.idPaciente
GROUP BY edad
ORDER BY num ASC LIMIT 5

```

Free

Cost: 178

	edad	num
1	22	304
2	31	314
3	25	330
4	59	338
5	46	358

### Mongodb



```
users > javil > OneDrive - Facultad de Ingeniería de la Universidad de Sa
Currently connected to cluster0.oegwrr0.mongodb.net. Click he

use('practica2');

const aggregation = [
  {
    $group: {
      _id: '$Edad',
      count: {
        $sum: 1
      }
    }
  },
  {
    $sort: {
      count: 1
    }
  },
  {
    $limit: 5
  }
]

db.LogActividades.aggregate(aggregation);
```

```
1  [
2  {
3    "_id": 22,
4    "count": 304
5  },
6  {
7    "_id": 31,
8    "count": 314
9  },
10 {
11  "_id": 25,
12  "count": 330
13 },
14 {
15  "_id": 59,
16  "count": 338
17 },
18 {
19  "_id": 46,
20  "count": 368
21 }
22 ]
```

## 6. Top 5 habitaciones más utilizadas

## Mysql

```
# 6. Top 5 habitaciones más utilizadas
SELECT HABITACION_idHabitacion, count(*) as num
FROM LOG_ACTIVIDAD
    JOIN PACIENTE p on LOG_ACTIVIDAD.PACIENTE_idPaciente = p.idPaciente
GROUP BY HABITACION_idHabitacion
ORDER BY num DESC LIMIT 5
```

Free

Q Input to filter result

Cost: 269ms

		* HABITACION_idHabitacion int	num
	1	10	37546
	2	12	5286
	3	1	3778
	4	2	2968
	5	13	2790

## Mongodb

```
rs > javil > OneDrive - Facultad de Ingeniería de la Universidad de San
Currently connected to cluster0.oegwrr0.mongodb.net. Click here

use('practica2');

const aggregation = [
  {
    $group: {
      _id: '$idHabitacion',
      Habitacion: {
        $first: '$Habitacion'
      },
      count: {
        $sum: 1
      }
    }
  },
  {
    $sort: [
      {
        count: -1
      }
    ],
    $limit: 5
  }
]

db.LogActividades.aggregate(aggregation);
```

```
1  [
2  {
3    "_id": 10,
4    "Habitacion": "Recepcion",
5    "count": 37546
6  },
7  {
8    "_id": 12,
9    "Habitacion": "Estación de revisión 1",
10   "count": 5286
11  },
12  {
13    "_id": 1,
14    "Habitacion": "Sala de exámenes 1",
15    "count": 3778
16  },
17  {
18    "_id": 2,
19    "Habitacion": "Sala de exámenes 2",
20    "count": 2968
21  },
22  {
23    "_id": 13,
24    "Habitacion": "Estación de revisión 2",
25    "count": 2790
26  }
27 ]
```

## 7. Top 5 habitaciones menos utilizadas

### Mysql

```
# 7. Top 5 habitaciones menos utilizadas
SELECT HABITACION_idHabitacion, count(*) as num
FROM LOG_ACTIVIDAD
JOIN PACIENTE p on LOG_ACTIVIDAD.PACIENTE_idPaciente = p.idPaciente
GROUP BY HABITACION_idHabitacion
ORDER BY num ASC LIMIT 5
```

Free

Cost: 259ms

	HABITACION_idHabitacion int	num
1	15	538
2	9	980
3	14	1288
4	8	1352
5	4	1387

## Mongodb

```
> Users > javil > OneDrive - Facultad de Ingeniería de la Universidad
Currently connected to cluster0.oegwrr0.mongodb.net. C

1 use('practica2');
2
3
4
5 const aggregation = [
6   {
7     $group: {
8       _id: '$idHabitacion',
9       Habitacion: {
10        $first: '$Habitacion'
11      },
12      count: {
13        $sum: 1
14      }
15    },
16    $sort: {
17      count: 1
18    },
19    $limit: 5
20  }
21 ]
22
23 db.LogActividades.aggregate(aggregation)
```

```
1 [
2   {
3     "_id": 15,
4     "Habitacion": "Estación de revisión 4",
5     "count": 538
6   },
7   {
8     "_id": 9,
9     "Habitacion": "Sala de procedimientos 4",
10    "count": 980
11  },
12  {
13    "_id": 14,
14    "Habitacion": "Estación de revisión 3",
15    "count": 1288
16  },
17  {
18    "_id": 8,
19    "Habitacion": "Sala de procedimientos 3",
20    "count": 1352
21  },
22  {
23    "_id": 4,
24    "Habitacion": "Sala de exámenes 4",
25    "count": 1387
26  }
27 ]
```

### 8. Día con más pacientes en la clínica

## Mysql

```
# 8. Día con más pacientes en la clínica
SELECT timestampx as fecha, count(*) n_pacientes
from LOG_ACTIVIDAD
GROUP BY timestampx
ORDER BY n_pacientes DESC LIMIT 1
```

	fecha	n_pacientes
1	7/21/2022	1476

## Mongodb

users > javil > OneDrive - Facultad de Ingeniería de la Universidad de San  
Currently connected to cluster0.oegwrr0.mongodb.net. Click here

```
use('practica2');
```

```
const aggregation = [
  {
    $sortByCount: '$timestamp'
  },
  {
    $limit: 1
  }
]
```

```
db.LogActividades.aggregate(aggregation);
```

```
1 [
2   {
3     "_id": {
4       "$date": "2022-07-21T06:00:00Z"
5     },
6     "count": 1476
7   }
8 ]
```

## Justificación

Este proyecto , se realizó con el propósito de verificar qué base de datos es mejor, para realizar todo tipo de consultas o análisis . Por una parte se tiene a MYSQL que se sabe que es una base de datos relacional bastante conocida por no procesar las tablas directamente sino que es posible utilizar procedimientos para procesar los datos de una manera más práctica y sencilla. Por otro lado se tiene a MONGODB que es de las bases de datos no relaciones más conocidas, en esta base datos de igual manera se puede realizar toda clase de consultas, análisis de datos , se pueden hacer búsquedas por campos , consultas de rangos entre muchas otras opciones. Teniendo en cuenta lo anterior sabiendo que los 2 motores de bases de datos son bastante útiles y capaces de realizar todo tipo de análisis , se realizaron 8 consultas idénticas por cada motor de base de datos teniendo resultados similares , entre ambas bases de datos cada una con sus características y atributos.

## Conclusión

- MongoDB es una base de datos , no relacional conocida por tener la capacidad de poder realizar todo tipo de consultas ya sea hacer una búsqueda por campos , consultas de rangos o utilizar expresiones regulares. Sabiendo que cada consulta tiene la capacidad de poder devolver un campo específico del documento o también una función de JS definida por desarrollador.
- Para una aplicación donde se requieren realizar operaciones de mayor complejidad , ya sea transacciones o diferentes tipos de operaciones que requieren un grado más alto de atención puede considerarse no usar MongoDB como base de datos , a pesar de que sí es una base de datos que tiene un tiempo de recuperación mínimo y que puede soportar grandes volúmenes de datos , no es recomendable para realizar transacciones de mayor complejidad.
- Utilizar MYSQL para un proyecto en donde se requiere realizar un mayor número de transacciones puede resultar de gran utilidad , ya que MYSQL se sabe que tiene la capacidad de poder realizar todo tipo de transacciones y operaciones de una manera veloz teniendo un gran rendimiento al momento de realizar las operaciones requeridas, a pesar de que una gran desventaja de utilizar MYSQL es que no resulta ser tan eficaz al momento de realizar cualquier tipo de modificación de escritura en la Base de datos.
- MongoDB puede considerarse , mucho más eficaz que MYSQL al momento de querer realizar inserciones en la base de datos esto debido a que al ser una base datos no relacional se sabe que tiene una escalabilidad horizontal y no vertical , considerando también que MONGODB para realizar la Auto Elasticidad no requiere de una interrupción como tal.
- Por otra parte al momento de querer consultar la información insertada , MYSQL resulta ser más eficiente que Mongoddb, esto por el simple hecho de

MYSQL es una base de datos relacional , con una tecnología madura que soporta totalmente el uso de JOINS en la base de datos.