

Proyecto #2 - Grupo 2

Laboratorio Sistemas de Bases de Datos 2 - Sección A

Primer Semestre 2025

5 de mayo de 2025

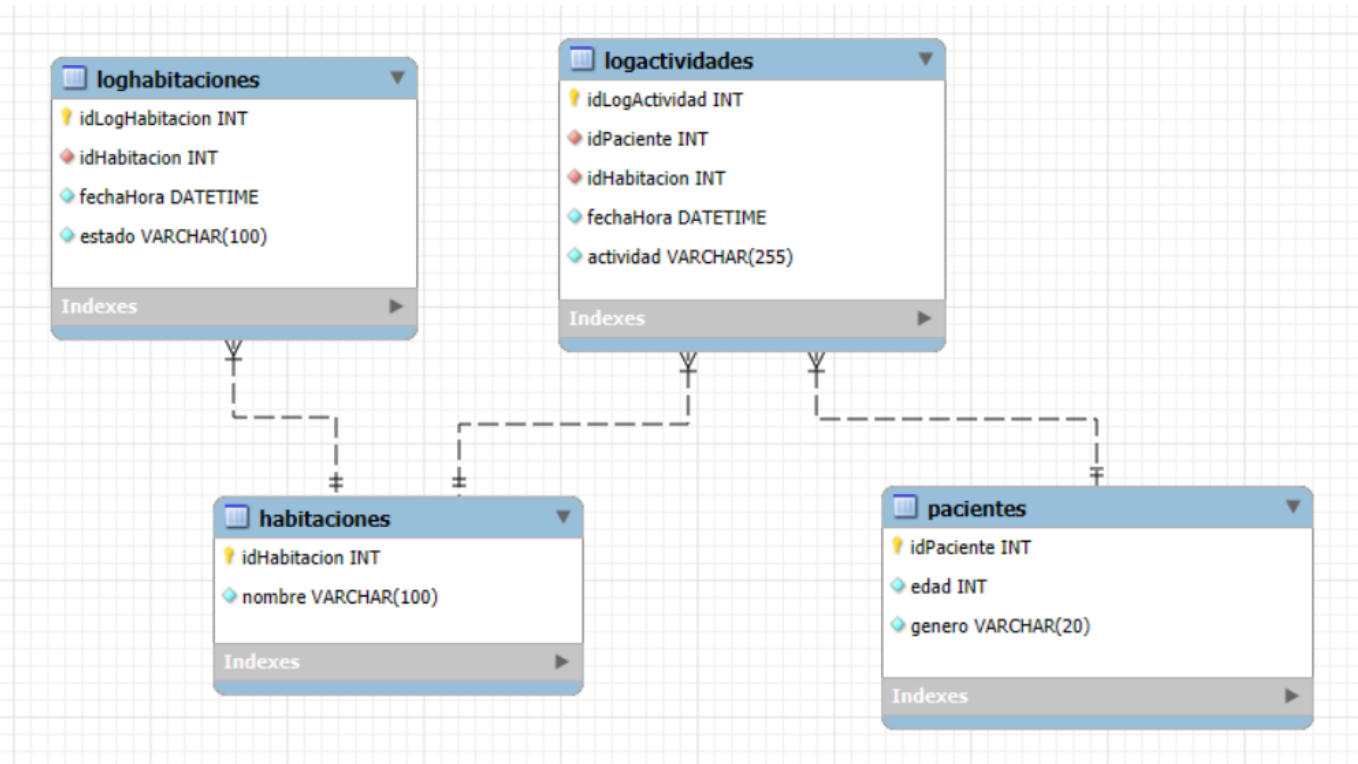
Miembros

Nombre	Carnet
Julio Alejandro Zaldaña Ríos	202110206
Edgar Mauricio Gómez Flores	2011-14340
Edgar Rolando Alvarez Rodriguez	202001144

Modelos de Bases de Datos

Se crearon modelos para basarse en la creación de las bases de datos.





MySQL - Diagrama Entidad Relación



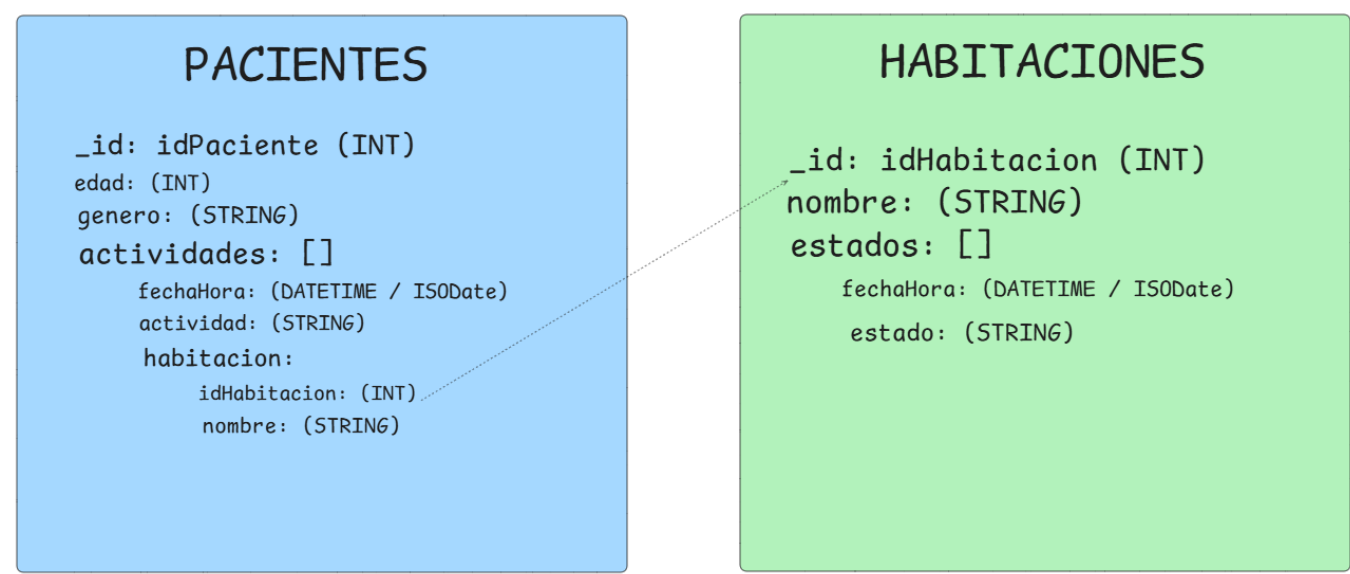
Para la base de datos relacional, se utilizó una estructura relacional de 4 tablas, que respeta las prácticas de normalización y relaciones explícitas mediante claves foráneas.

Se crearon cuatro tablas:



- 1. Pacientes
- 2. Habitaciones
- 3. Logactividades
- 4. Loghabitaciones

 Pacientes	 Habitaciones	 LogActividades	 LogHabitaciones
<p>idPaciente (PK): Identificador único del paciente.</p> <p>edad: Edad del paciente.</p> <p>genero: Género del paciente.</p>	<p>idHabitacion (PK): Identificador único de la habitación.</p> <p>habitacion: Nombre o descripción de la habitación.</p>	<p>id (PK): Clave primaria autoincremental.</p> <p>idPaciente (FK): Referencia al paciente asociado.</p> <p>idHabitacion (FK): Referencia a la habitación.</p> <p>fechaHora: Fecha y hora de la actividad.</p> <p>actividad: Descripción de la acción.</p>	<p>id (PK): Clave primaria autoincremental.</p> <p>idHabitacion (FK): Referencia a la habitación.</p> <p>fechaHora: Fecha y hora del estado.</p> <p>status: Estado de la habitación.</p>

MongoDB - Diagrama Colecciones



MongoDB permite un modelo más flexible basado en documentos. Aquí se optó por dos colecciones principales, estructuradas de manera jerárquica mediante subdocumentos y arrays embebidos:

 Pacientes	 Habitaciones
Cada documento representa un paciente individual, con su historial de actividades embebido.	Cada documento representa una habitación con su historial de estados.

Pacientes

Habitaciones

Campos:

- **_id** (idPaciente): Identificador del paciente.
- **edad**: Edad del paciente.
- **genero**: Género del paciente.
- **actividades**: Array de objetos con:
 - **fechaHora**: Fecha y hora de la actividad.
 - **actividad**: Descripción.
- **habitacion**: Subdocumento:
 - **idHabitacion**
 - **nombre** (de la habitación).

Campos:

- **_id** (idHabitacion): Identificador único.
- **nombre**: Nombre de la habitación.
- **estados**: Array de objetos con:
 - **fechaHora**: Momento del cambio.
 - **estado**: Descripción del estado.

Fases de Desarrollo

Correr Docker

```
$ docker compose -f docker-compose.yml up
```

Insertar datos

```
$ python3 -m venv .venv
$ source .venv/bin/activate
$ pip install pandas openpyxl sqlalchemy pymysql pymongo cryptography tabulate
$ python load.py
```

Analizar queries

```
$ python benchmark.py
```

API de Python para Consultas en ambas bases de datos.

Ingresa a la carpeta de fastapi y correr el contenedor de docker que contiene las dos bases de datos y la api desarrollada con fast api.

```
cd fastapi
docker compose up --build -d
```

Las consultas se hicieron con Postman, se tienen las respuestas de las consultas dentro de la carpeta endpoints, con las colecciones de los endpoints.

Consultas MYSQL

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

- Pediátrico: menores de 18 años
- Mediana edad: entre 18 y 60 años
- Geriátrico: mayores de 60 años

PROYECTO2_BASES2_MYSQL / /pacientes_por_categoria_edad

GET

http://3.147.45.23:6500/pacientes_por_categoria_edad

Send

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
--	-----	-------	-------------	-----	-----------

Body

Cookies

Headers (4)

Test Results

200 OK

374 ms

299 B

Save Response

{ } JSON

Preview

Visualize

```
1  [
2    {
3      "categoria_edad": "Geriátrico",
4      "total_pacientes": 34089
5    },
6    {
7      "categoria_edad": "Mediana edad",
8      "total_pacientes": 82254
9    },
10   {
11     "categoria_edad": "Pediátrico",
12     "total_pacientes": 37841
13   }
14 ]
```

2. Cantidad de pacientes que pasan por cada habitación

PROYECTO2_BASES2_MYSQL / /pacientes_por_habitacion

GEThttp://3.147.45.23:6500/pacientes_por_habitacion

Send

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

BodyCookiesHeaders (4)Test Results

200 OK313 ms1.27 KB

Save Response

JSONPreviewVisualize

```
1  [
2    {
3      "idHabitacion": 1,
4      "habitacion": "Sala de exámenes 1",
5      "total_pacientes": 2539
6    },
7    {
8      "idHabitacion": 2,
9      "habitacion": "Sala de exámenes 2",
10     "total_pacientes": 2054
11    },
12    {
13      "idHabitacion": 3,
14      "habitacion": "Sala de exámenes 3",
15      "total_pacientes": 1548
16    },
17    {
18      "idHabitacion": 4,
19      "habitacion": "Sala de exámenes 4",
20      "total_pacientes": 1029
21    },
22    {
23      "idHabitacion": 5,
24      "habitacion": "Sala de imágenes 1",
25      "total_pacientes": 1428
26    },
27  ]
```

PROYECTO2_BASES2_MYSQL / /pacientes_por_habitacion

GEThttp://3.147.45.23:6500/pacientes_por_habitacion

Send

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

BodyCookiesHeaders (4)Test Results

200 OK313 ms1.27 KB

Save Response

JSONPreviewVisualize

```
27  {
28    "idHabitacion": 6,
29    "habitacion": "Sala de procedimientos 1",
30    "total_pacientes": 1527
31  },
32  {
33    "idHabitacion": 7,
34    "habitacion": "Sala de procedimientos 2",
35    "total_pacientes": 1350
36  },
37  {
38    "idHabitacion": 8,
39    "habitacion": "Sala de procedimientos 3",
40    "total_pacientes": 1022
41  },
42  {
43    "idHabitacion": 9,
44    "habitacion": "Sala de procedimientos 4",
45    "total_pacientes": 760
46  },
47  {
48    "idHabitacion": 10,
49    "habitacion": "Recepción",
50    "total_pacientes": 6374
51  },
52  {
53    "idHabitacion": 11,
```

PROYECTO2_BASES2_MYSQL / /pacientes_por_habitacion

GEThttp://3.147.45.23:6500/pacientes_por_habitacion

Send

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

BodyCookiesHeaders (4)Test Results

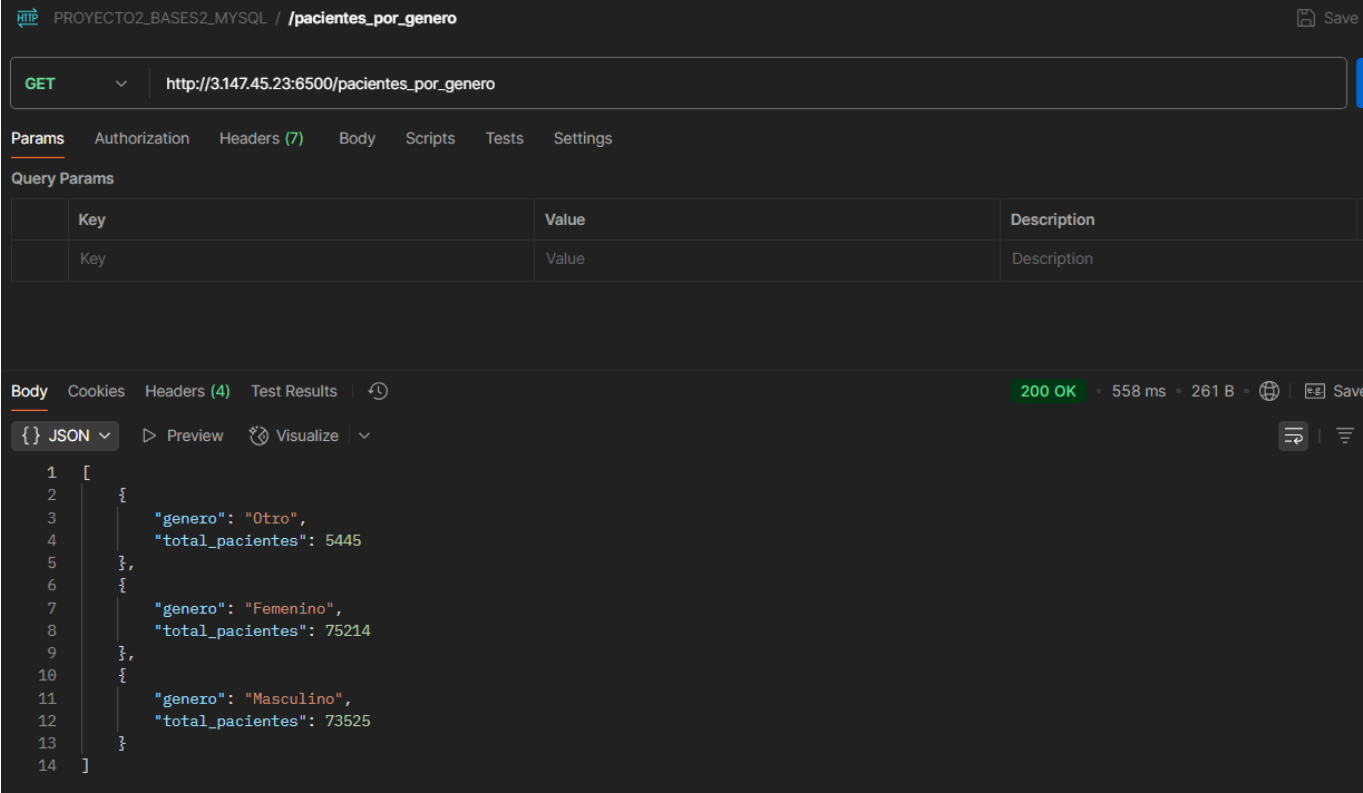
200 OK313 ms1.27 KB

Save Response

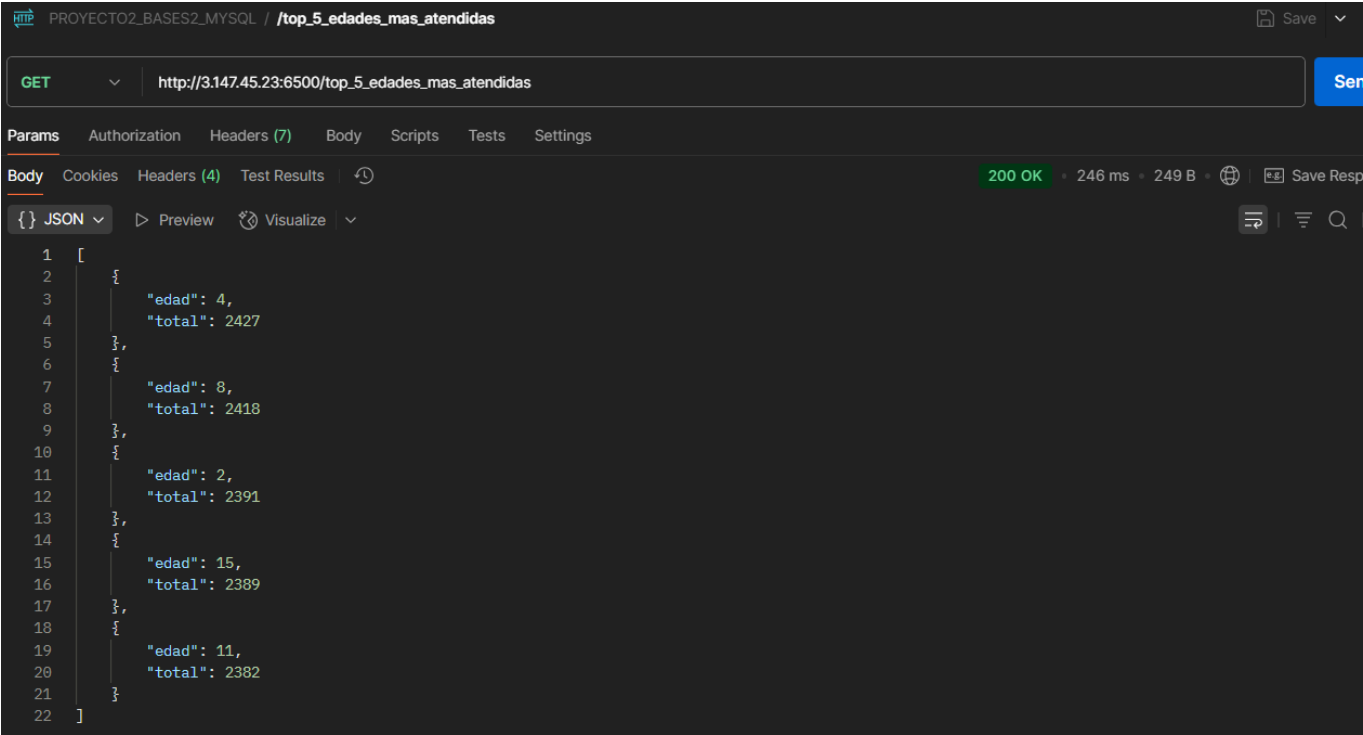
JSONPreviewVisualize

```
51  },
52  {
53    "idHabitacion": 11,
54    "habitacion": "Laboratorio",
55    "total_pacientes": 1683
56  },
57  {
58    "idHabitacion": 12,
59    "habitacion": "Estación de revisión 1",
60    "total_pacientes": 3562
61  },
62  {
63    "idHabitacion": 13,
64    "habitacion": "Estación de revisión 2",
65    "total_pacientes": 2155
66  },
67  {
68    "idHabitacion": 14,
69    "habitacion": "Estación de revisión 3",
70    "total_pacientes": 1092
71  },
72  {
73    "idHabitacion": 15,
74    "habitacion": "Estación de revisión 4",
75    "total_pacientes": 472
76  },
77  ]
```

3. Cantidad de pacientes que llegan a la clínica, agrupados por género



4. Top 5 edades más atendidas en la clínica



5. Top 5 edades menos atendidas en la clínica

PROYECTO2_BASES2_MYSQL / /top_5_edades_menos_atendidas

GET http://3.147.45.23:6500/top_5_edades_menos_atendidas

Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (4) Test Results 200 OK 310 ms 247 B Save

{ } JSON Preview Visualize

```
1  [
2    {
3      "edad": 90,
4      "total": 753
5    },
6    {
7      "edad": 76,
8      "total": 756
9    },
10   {
11     "edad": 79,
12     "total": 771
13   },
14   {
15     "edad": 77,
16     "total": 777
17   },
18   {
19     "edad": 83,
20     "total": 778
21   }
22 ]
```

6. Top 5 habitaciones más utilizadas

PROYECTO2_BASES2_MYSQL / /top_5_habitaciones_mas_utilizadas

GET http://3.147.45.23:6500/top_5_habitaciones_mas_utilizadas

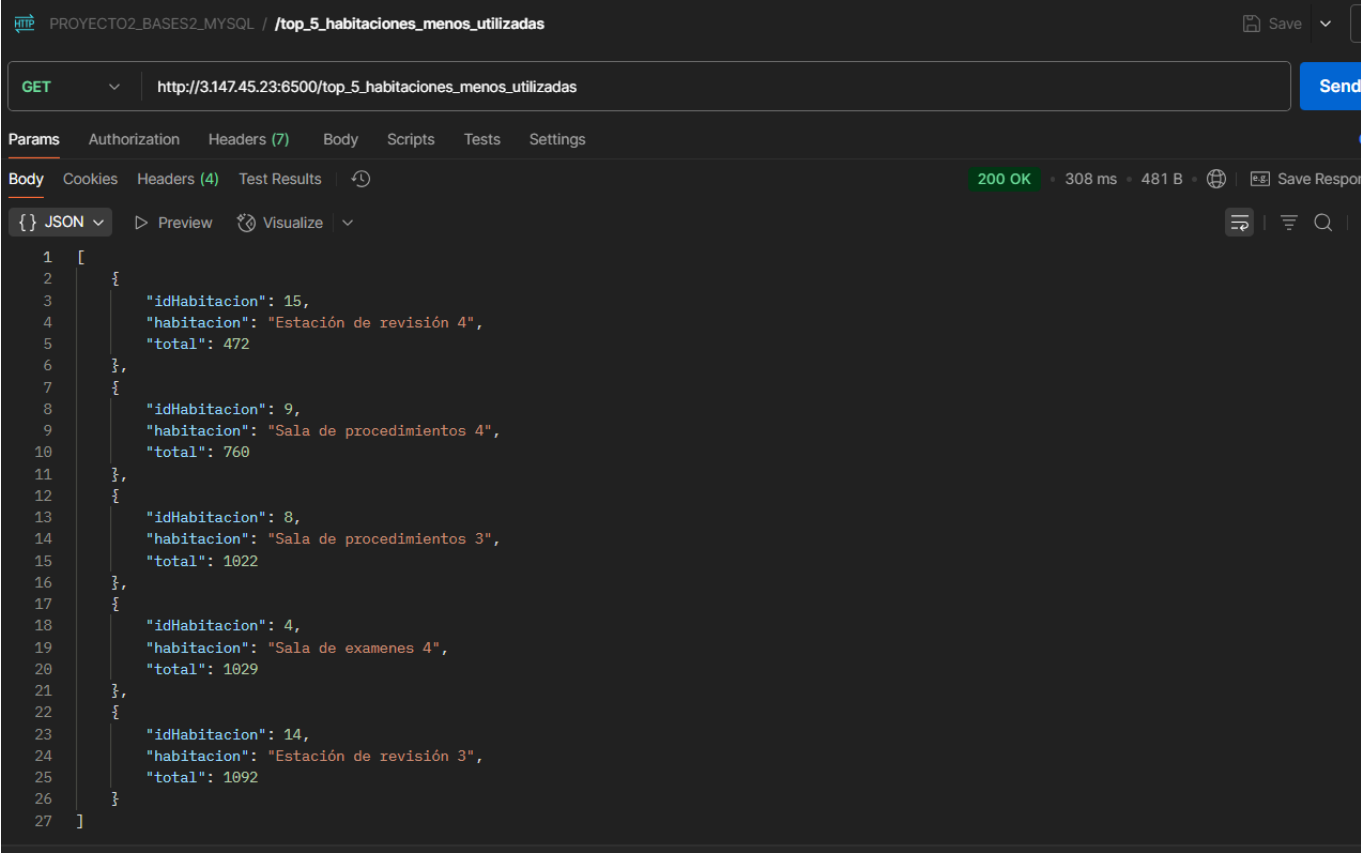
Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (4) Test Results 200 OK 398 ms 463 B Save Resp

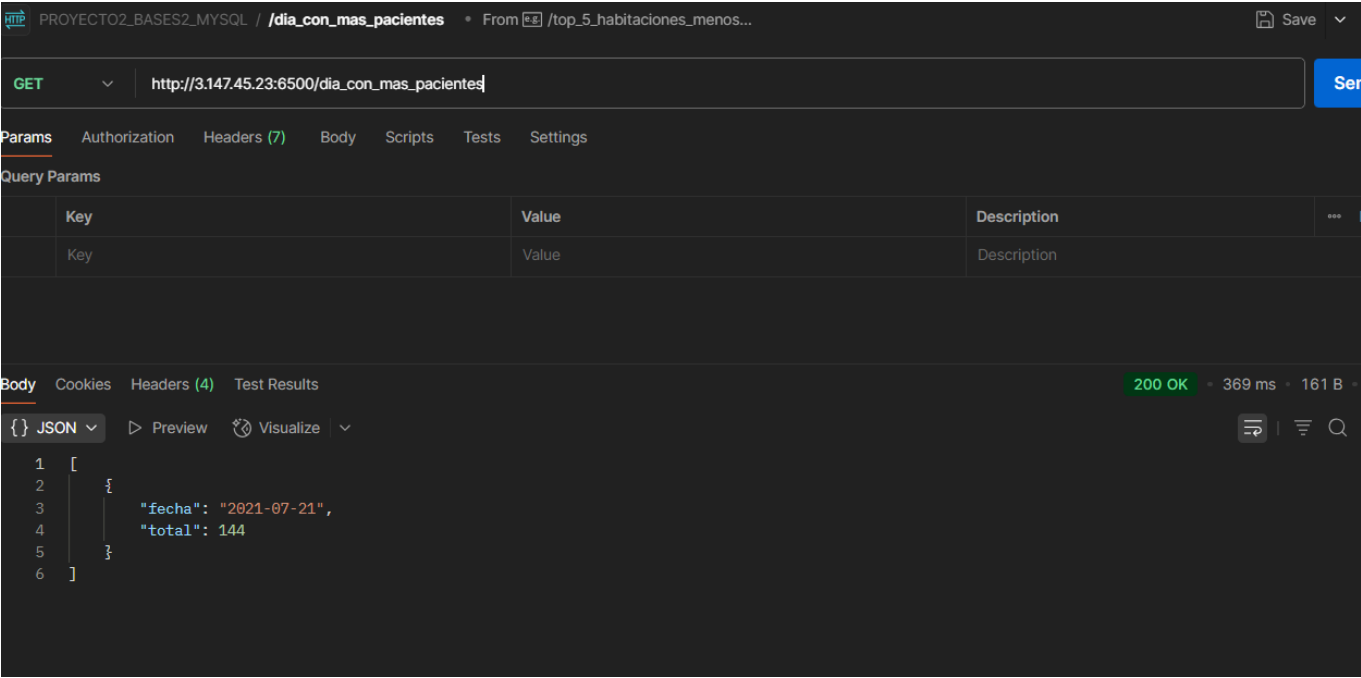
{ } JSON Preview Visualize

```
1  [
2    {
3      "idHabitacion": 10,
4      "habitacion": "Recepcion",
5      "total": 6374
6    },
7    {
8      "idHabitacion": 12,
9      "habitacion": "Estación de revisión 1",
10     "total": 3562
11   },
12   {
13     "idHabitacion": 1,
14     "habitacion": "Sala de exámenes 1",
15     "total": 2539
16   },
17   {
18     "idHabitacion": 13,
19     "habitacion": "Estación de revisión 2",
20     "total": 2155
21   },
22   {
23     "idHabitacion": 2,
24     "habitacion": "Sala de exámenes 2",
25     "total": 2054
26   }
27 ]
```

7. Top 5 habitaciones menos utilizadas



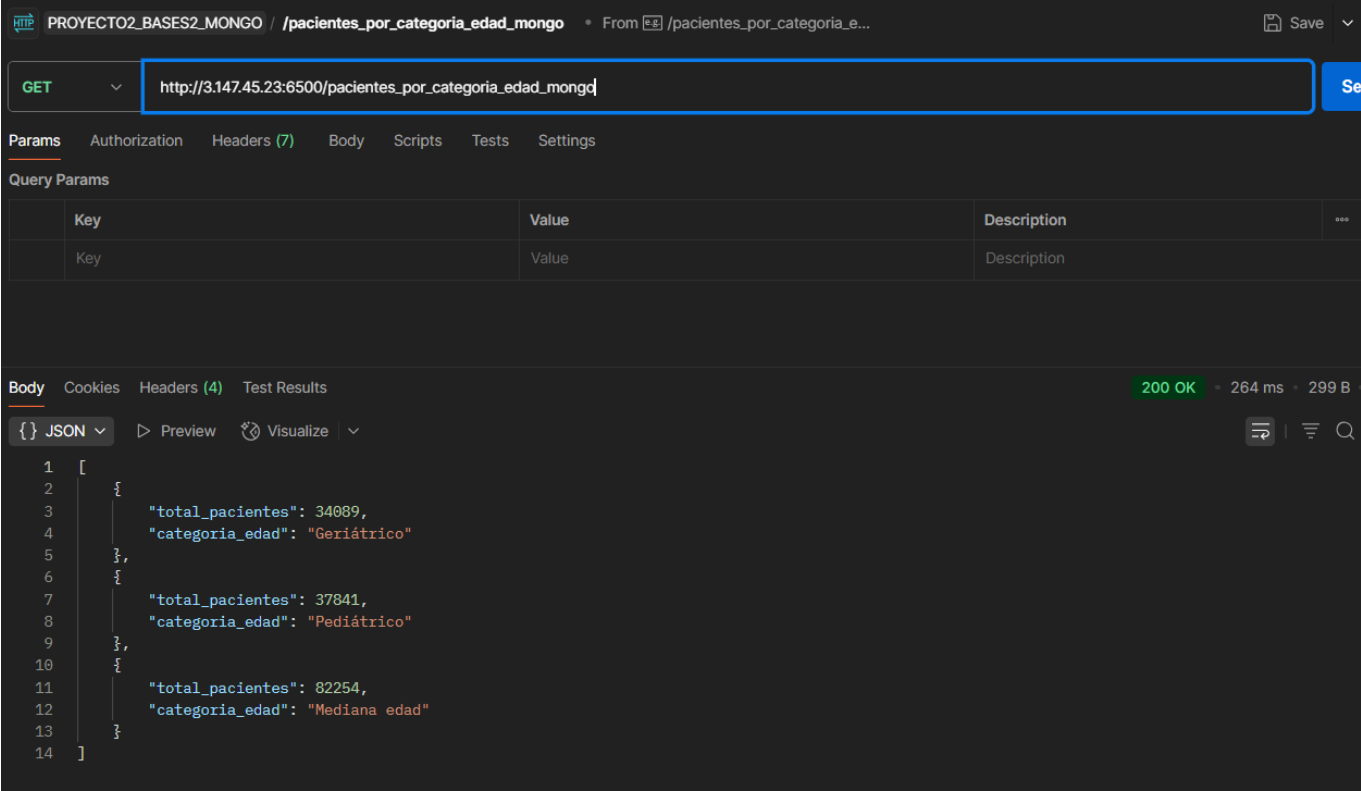
8. Día con más pacientes en la clínica



Consultas MongoDB

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

- Pediátrico: menores de 18 años
- Mediana edad: entre 18 y 60 años
- Geriátrico: mayores de 60 años



2. Cantidad de pacientes que pasan por cada habitación

PROYECTO2_BASES2_MONGO / /pacientes_por_habitacion_mongo

GET http://3.147.45.23:6500/pacientes_por_habitacion_mongo

Send

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Body

Cookies

Headers (4)

Test Results

200 OK

485 ms

1.02 KB

Save Response

{ } JSON

Preview

Visualize

```
1  [
2    {
3      "habitacion": "Recepcion",
4      "total_pacientes": 6374
5    },
6    {
7      "habitacion": "Estación de revisión 1",
8      "total_pacientes": 3562
9    },
10   {
11     "habitacion": "Sala de exámenes 1",
12     "total_pacientes": 2539
13   },
14   {
15     "habitacion": "Estación de revisión 2",
16     "total_pacientes": 2155
17   },
18   {
19     "habitacion": "Sala de exámenes 2",
20     "total_pacientes": 2054
21   },
22   {
23     "habitacion": "Laboratorio",
24     "total_pacientes": 1683
25   },
26   {
27     "habitacion": "Sala de exámenes 3",
```

PROYECTO2_BASES2_MONGO / /pacientes_por_habitacion_mongo

GET http://3.147.45.23:6500/pacientes_por_habitacion_mongo

Send

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Body

Cookies

Headers (4)

Test Results

200 OK

485 ms

1.02 KB

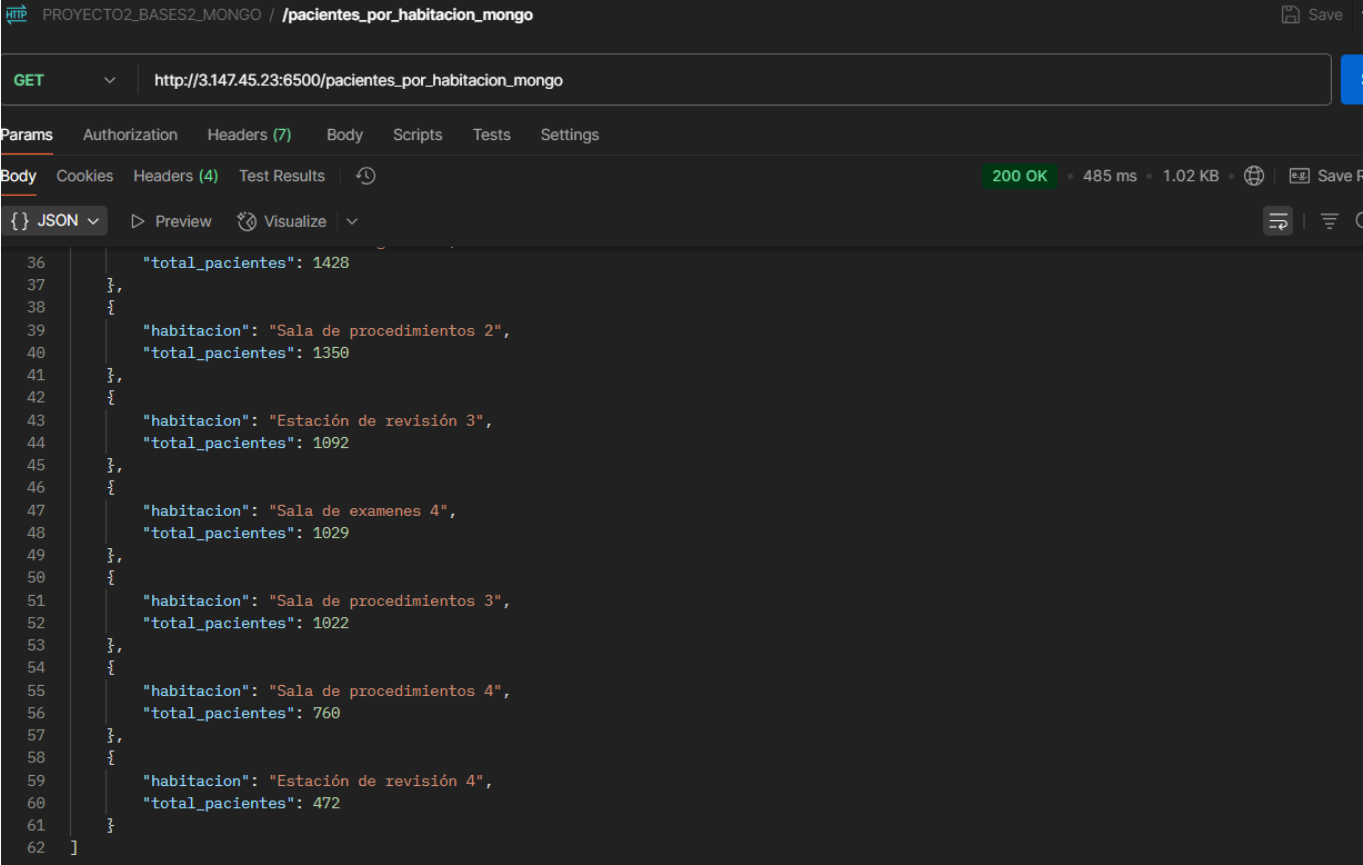
Save Response

{ } JSON

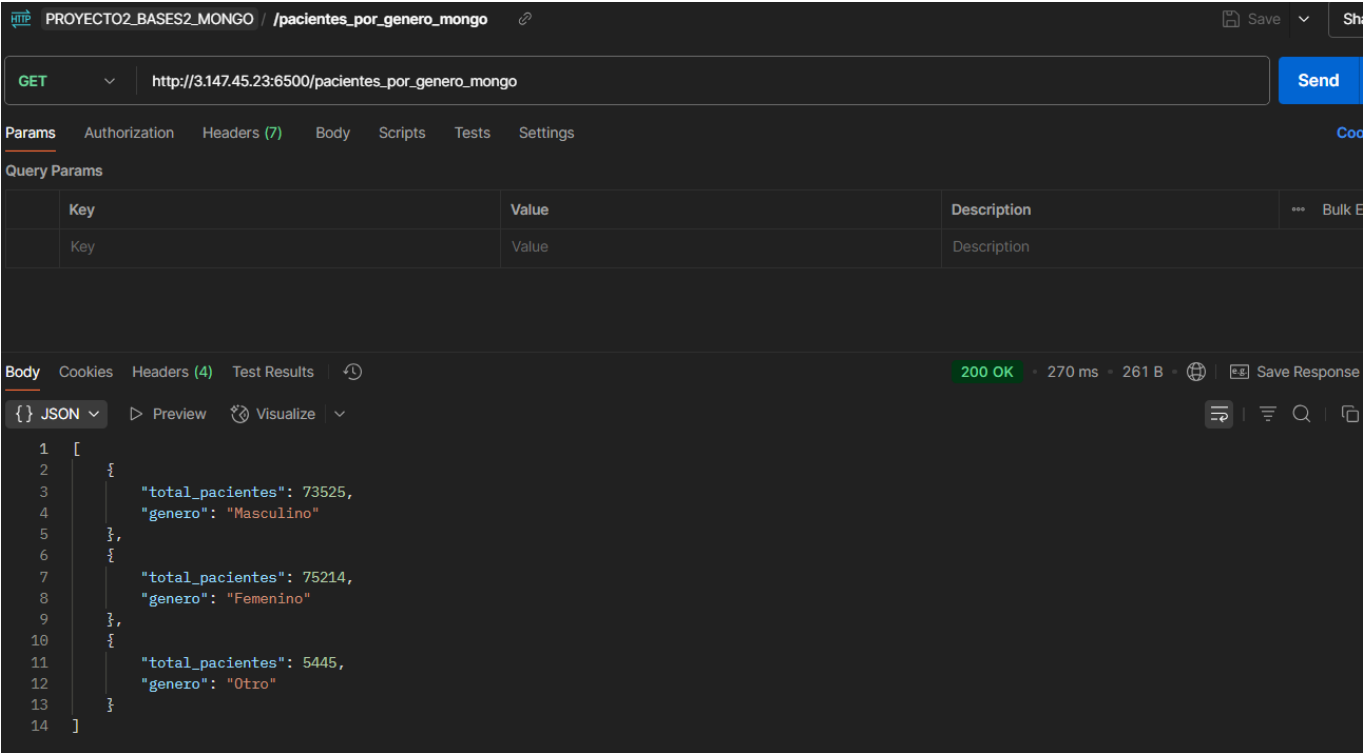
Preview

Visualize

```
26   {
27     "habitacion": "Sala de exámenes 3",
28     "total_pacientes": 1548
29   },
30   {
31     "habitacion": "Sala de procedimientos 1",
32     "total_pacientes": 1527
33   },
34   {
35     "habitacion": "Sala de imágenes 1",
36     "total_pacientes": 1428
37   },
38   {
39     "habitacion": "Sala de procedimientos 2",
40     "total_pacientes": 1350
41   },
42   {
43     "habitacion": "Estación de revisión 3",
44     "total_pacientes": 1092
45   },
46   {
47     "habitacion": "Sala de exámenes 4",
48     "total_pacientes": 1029
49   },
50   {
51     "habitacion": "Sala de procedimientos 3",
52     "total_pacientes": 1022
```



3. Cantidad de pacientes que llegan a la clínica, agrupados por género



4. Top 5 edades más atendidas en la clínica

PROYECTO2_BASES2_MONGO / /top_5_edades_mas_atendidas_mongo

GET http://3.147.45.23:6500/top_5_edades_mas_atendidas_mongo

Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (4) Test Results

200 OK 432 ms 299 B

JSON Preview Visualize

```
1  [
2    {
3      "total_pacientes": 2427,
4      "edad": 4
5    },
6    {
7      "total_pacientes": 2418,
8      "edad": 8
9    },
10   {
11     "total_pacientes": 2391,
12     "edad": 2
13   },
14   {
15     "total_pacientes": 2389,
16     "edad": 15
17   },
18   {
19     "total_pacientes": 2382,
20     "edad": 11
21   }
22 ]
```

5. Top 5 edades menos atendidas en la clínica

PROYECTO2_BASES2_MONGO / /top_5_edades_menos_atendidas_mongo

GET http://3.147.45.23:6500/top_5_edades_menos_atendidas_mongo

Params Authorization Headers (7) Body Scripts Tests Settings

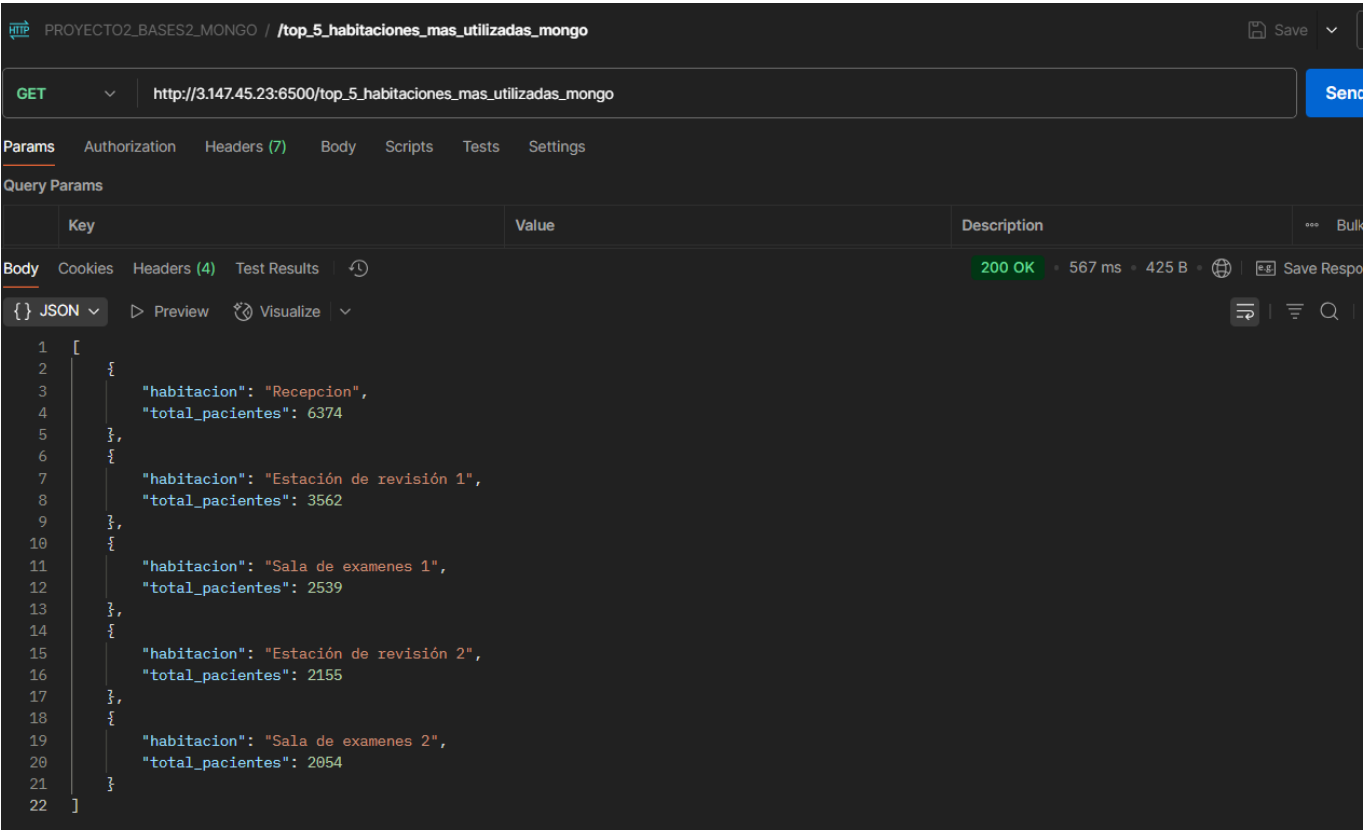
Body Cookies Headers (4) Test Results

200 OK 320 ms 297 B

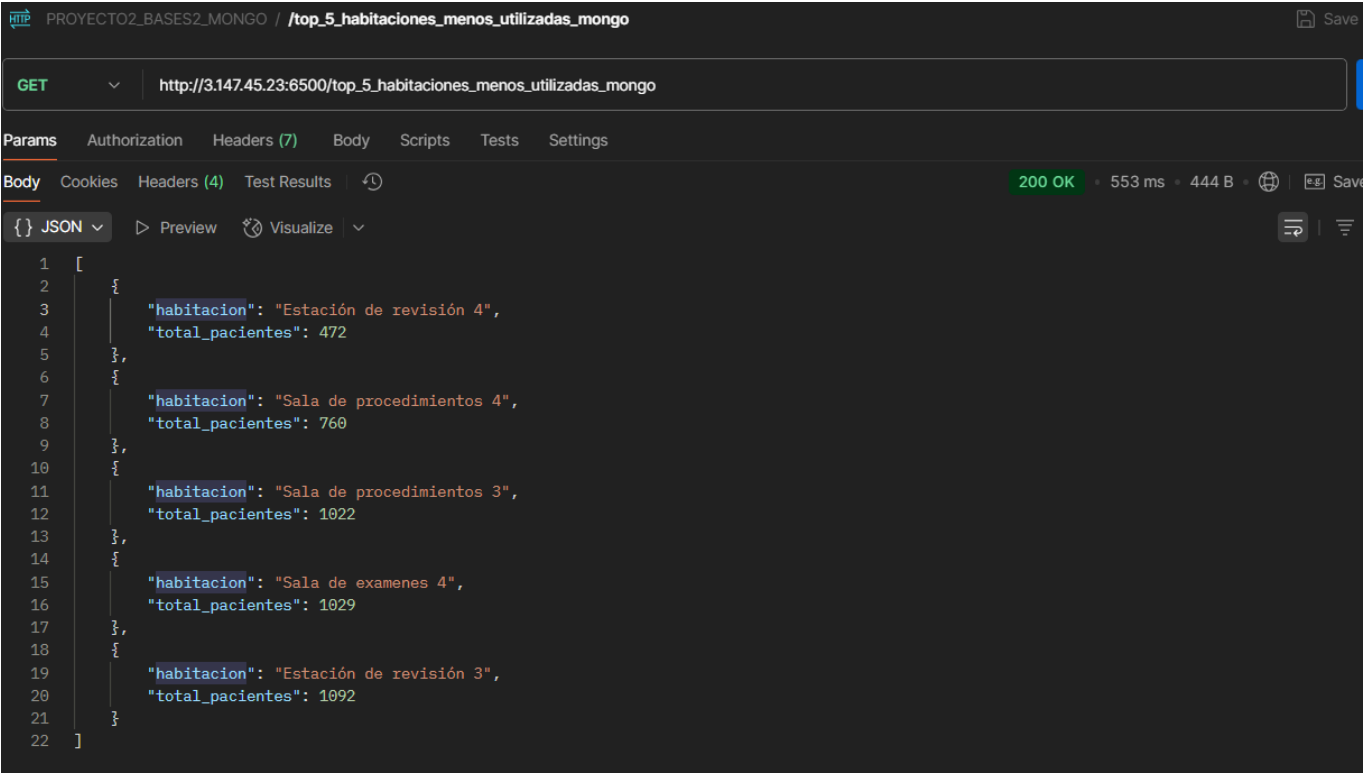
JSON Preview Visualize

```
1  [
2    {
3      "total_pacientes": 753,
4      "edad": 90
5    },
6    {
7      "total_pacientes": 756,
8      "edad": 76
9    },
10   {
11     "total_pacientes": 771,
12     "edad": 79
13   },
14   {
15     "total_pacientes": 777,
16     "edad": 77
17   },
18   {
19     "total_pacientes": 778,
20     "edad": 83
21   }
22 ]
```

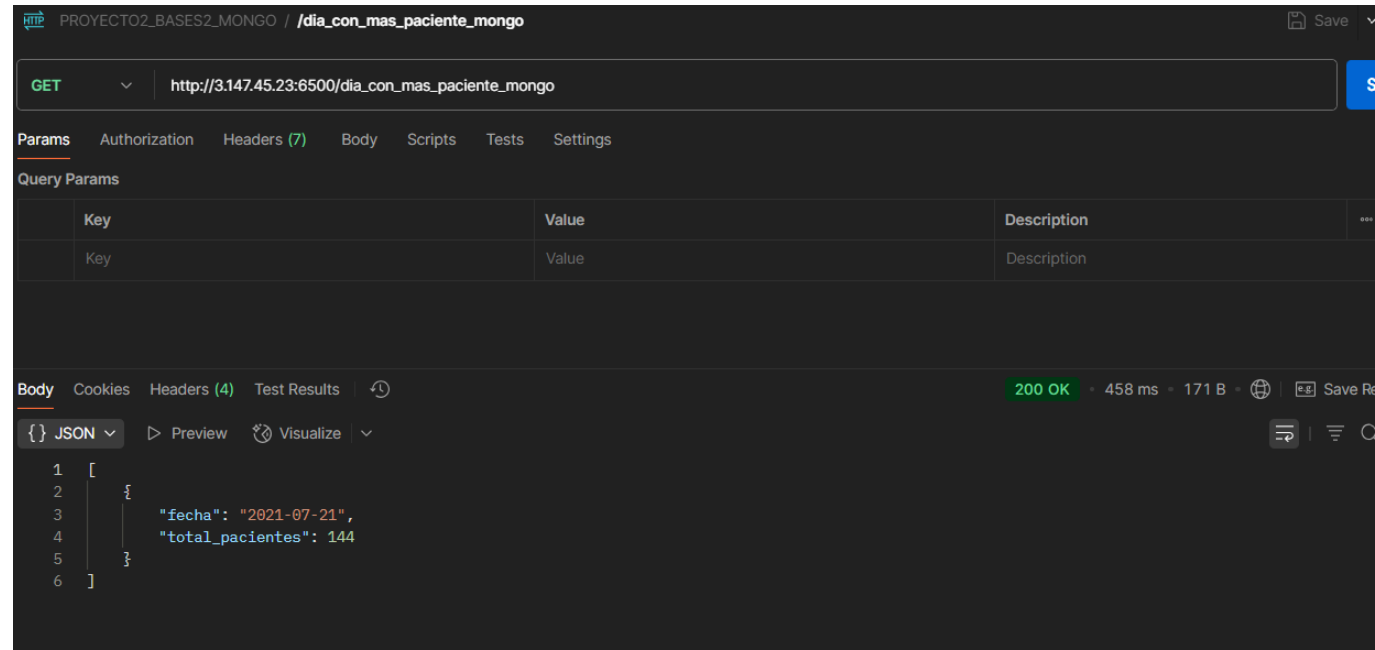
6. Top 5 habitaciones más utilizadas



7. Top 5 habitaciones menos utilizadas



8. Día con más pacientes en la clínica



Analisis de las bases de datos

Análisis Comparativo: MySQL vs MongoDB para el Sistema Hospitalario

Resultados del Benchmark

Consulta	MySQL avg (s)	MySQL med (s)	MySQL p90 (s)	Mongo avg (s)	Mongo med (s)	Mongo p90 (s)
pacientes_por_categoria_edad	0.12464	0.11436	0.16031	0.09668	0.09288	0.11228
pacientes_por_habitacion	0.28011	0.14348	0.17877	0.24636	0.23161	0.30955
pacientes_por_genero	0.11781	0.11600	0.13179	0.07527	0.07182	0.08891
top_5_edades_mas_atendidas	0.07410	0.07149	0.09201	0.06979	0.06617	0.07692
top_5_edades_menos_atendidas	0.07814	0.07563	0.09468	0.07102	0.06780	0.08749
top_5_habitaciones_mas_utilizadas	0.14063	0.13524	0.16370	0.27151	0.24641	0.36983
top_5_habitaciones_menos_utilizadas	0.13853	0.13301	0.17278	0.27523	0.24619	0.38146
dia_con_mas_pacientes	0.05585	0.05446	0.06401	0.24658	0.23194	0.30825

```
(venv) PS C:\EQUIPO\1S-2025\BASES-DE-DATOS\Lab\BD2_1S2025_Grupo2_Proyecto2> python .\load.py
Insertando datos en MySQL ...
Conexión a MySQL establecida.
Datos insertados en MySQL.
Insertando datos en MongoDB ...
Datos insertados en MongoDB.
(venv) PS C:\EQUIPO\1S-2025\BASES-DE-DATOS\Lab\BD2_1S2025_Grupo2_Proyecto2> python .\benchmark.py
Ejecutando benchmark (100 iteraciones por consulta)...
```

Estadísticas de ejecución (100 repeticiones):

	MySQL avg (s)	MySQL med (s)	MySQL p90 (s)	Mongo avg (s)	Mongo med (s)	Mongo p90 (s)
pacientes_por_categoria_edad	0.12464	0.11436	0.16031	0.09668	0.09288	0.11228
pacientes_por_habitacion	0.28011	0.14348	0.17877	0.24636	0.23161	0.30955
pacientes_por_genero	0.11781	0.116	0.13179	0.07527	0.07182	0.08891
top_5_edades_mas_atendidas	0.0741	0.07149	0.09201	0.06979	0.06617	0.07692
top_5_edades_menos_atendidas	0.07814	0.07563	0.09468	0.07102	0.0678	0.08749
top_5_habitaciones_mas_utilizadas	0.14063	0.13524	0.1637	0.27151	0.24641	0.36983
top_5_habitaciones_menos_utilizadas	0.13853	0.13301	0.17278	0.27523	0.24619	0.38146
dia_con_mas_pacientes	0.05585	0.05446	0.06401	0.24658	0.23194	0.30825

Basado en los resultados del benchmark realizado con 100 iteraciones por consulta, podemos observar lo siguiente:

1. Consultas más rápidas en MongoDB:

- **pacientes_por_categoria_edad**: MongoDB 22.4% más rápido (0.09668s vs 0.12464s)
- **pacientes_por_genero**: MongoDB 36.1% más rápido (0.07527s vs 0.11781s)
- **top_5_edades_mas_atendidas**: MongoDB 5.8% más rápido (0.06979s vs 0.0741s)
- **top_5_edades_menos_atendidas**: MongoDB 9.1% más rápido (0.07102s vs 0.07814s)

2. Consultas más rápidas en MySQL:

- **pacientes_por_habitacion**: MySQL 11.9% más rápido (0.28011s vs 0.24636s)
- **top_5_habitaciones_mas_utilizadas**: MySQL 48.2% más rápido (0.14063s vs 0.27151s)
- **top_5_habitaciones_menos_utilizadas**: MySQL 49.7% más rápido (0.13853s vs 0.27523s)
- **dia_con_mas_pacientes**: MySQL 77.4% más rápido (0.05585s vs 0.24658s)

Ventajas de MongoDB en este contexto

- Rendimiento superior en consultas de análisis demográfico**: MongoDB muestra claras ventajas en consultas relacionadas con categorías de edad y género de pacientes.
- Modelo de datos más natural**: La estructura documental de MongoDB se adapta mejor a los datos hospitalarios donde cada paciente tiene múltiples actividades asociadas, evitando joins complejos.
- Escalabilidad horizontal**: MongoDB ofrece mejor escalabilidad para crecimiento futuro de datos.

Ventajas de MySQL en este contexto

- Mejor rendimiento en consultas relacionadas con habitaciones**: MySQL muestra mejor desempeño en operaciones que involucran relaciones entre tablas (habitaciones y pacientes).

2. **Consistencia transaccional:** Para operaciones críticas que requieren ACID, MySQL sigue siendo mejor opcon.
3. **Consultas complejas:** Las consultas que requieren múltiples joins y operaciones relacionales son más eficientes en MySQL.

Conclusion Final

Para este caso en específico, se recomendaría implementar MongoDB como base de datos principal para la empresa hospitalaria, debido a:

1. La mayoría de las consultas analíticas (especialmente las relacionadas con pacientes) son más rápidas en MongoDB.
2. Los datos médicos son semi-estructurados, por eso el modelo documental funciona mejor.
3. Las operaciones más frecuentes (análisis por edad, género) son significativamente más rápidas.