

# Bases de Datos 2025

## Parcial II: NoSQL

Sergio Canchi, Agustín Curto, Ramiro Demasi,  
Diego Gimenez, Guido Ivetta, Iván Nieva

### Contexto

La base de datos **sample\_airbnb** contiene la siguiente colección:

Colección	Descripción
listingsAndReviews	Contiene documentos que representan los detalles de alojamientos y reviews de los usuarios acerca de los alojamientos disponibles en <a href="#">Inside AirBnB</a> .

Para cargar la base de datos **sample\_airbnb** ejecutar los siguientes comandos:

Instructivo para los que usan sus **computadoras personales**:

```
$ tar xzvf parcial2.tar.gz
$ cd parcial2/
$ mongorestore --host <host> --drop --gzip --db sample_airbnb sample_airbnb/
```

Instructivo para los que usan las **computadoras del laboratorio**:

El primer paso es obtener la imagen de mongo 8. Para ello, correr el siguiente comando:

```
$ docker pull mongo:8
```

**En caso de tener problemas**, descargar el archivo mongo8.tar.gz y usar el comando:

```
$ docker load -i mongo8.tar.gz
```

Ahora que ya tenemos la imagen, podemos levantar el contenedor y cargar los datos:

```
$ tar xzvf parcial2.tar.gz
$ cd parcial2/
$ docker run --name mongo-parcial2 \
  -p 27017:27017 -v mongod.conf:/etc/mongod.conf \
  -v ./sample_airbnb:/data/sample_airbnb -d mongo:8
$ docker exec -it mongo-parcial2 mongorestore \
  --drop --gzip --db sample_airbnb /data/sample_airbnb/
```

```
// (Opcional) para ejecutar el cliente
$ docker exec -it mongo-parcial2 mongosh
```

El siguiente es un fragmento de los datos de un alojamiento junto con una breve descripción (comentado con //) de algunos de los campos que vamos a usar en los ejercicios.

```
{
  _id: '10006546',                // identificador del alojamiento
  name: 'Ribeira Charming Duplex', // nombre del alojamiento
  ...
  property_type: 'House',         // tipo de la propiedad
  ...
  accommodates: 8, // número de huéspedes que puede alojar el alojamiento
  bedrooms: 3,     // número de dormitorios
  amenities: [      // amenity que provee el alojamiento
    'TV',
    'Cable TV',
    'Wifi',
    'Kitchen',
    'Paid parking off premises',
    ...
  ],
  ...
  address: {           // dirección del alojamiento
    ...
    market: 'Porto',
    country: 'Portugal', // país donde está localizado el alojamiento
    ...
  },
  ...
  review_scores: { // puntuaciones de los reviews
    ...
    review_scores_rating: 89 // rating global del alojamiento
  },
  reviews: [ // reviews sobre el alojamiento
    {
      _id: '58663741',
      date: ISODate("2016-01-03T05:00:00.000Z"),
      listing_id: '10006546',
      reviewer_id: '51483096',
      reviewer_name: 'Cátia',
      comments: 'A casa da Ana e do Gonçalo...'
    },
    ...
  ]
}
```

## Consignas

1. Calcular el rating promedio por país. Listar el país, rating promedio, y cantidad de rating. Listar en orden descendente por rating promedio. Usar el campo "review\_scores.review\_scores\_rating" para calcular el rating promedio.
2. Listar los 20 alojamientos que tienen las reviews más recientes. Listar el id, nombre, fecha de la última review, y cantidad de reviews del alojamiento. Listar en orden descendente por cantidad de reviews.  
HINT: \$first pueden ser de utilidad.
3. Crear la vista "top10\_most\_common\_amenities" con información de los 10 amenities que aparecen con más frecuencia. El resultado debe mostrar el amenity y la cantidad de veces que aparece cada amenity.
4. Actualizar los alojamientos de Brazil que tengan un rating global ("review\_scores.review\_scores\_rating") asignado, agregando el campo "quality\_label" que clasifique el alojamiento como "High" (si el rating global es mayor o igual a 90), "Medium" (si el rating global es mayor o igual a 70), "Low" (valor por defecto) calidad..  
HINTS: (i) para actualizar se puede usar pipeline de agregación. (ii) El operador \$cond o \$switch pueden ser de utilidad.
5. (a) Especificar reglas de validación en la colección listingsAndReviews a los siguientes campos requeridos: *name*, *address*, *amenities*, *review\_scores*, and *reviews* ( y todos sus campos anidados). Inferir los tipos y otras restricciones que considere adecuados para especificar las reglas a partir de los documentos de la colección.  
(b) Testear la regla de validación generando dos casos de fallas en la regla de validación y un caso de éxito en la regla de validación. Aclarar en la entrega cuales son los casos y por qué fallan y cuales cumplen la regla de validación. Los casos no deben ser triviales, es decir los ejemplos deben contener todos los campos especificados en la regla.

## Puntos a tener en cuenta

- Algunas consultas pueden resolverse con find y aggregate. En estos casos se puede resolver eligiendo alguno de estos métodos.
- Resolver las consultas sin vistas salvo que se lo pida explícitamente.
- Mostrar únicamente los campos pedidos en la consigna.
- Se piden que los campos que se devuelven sean valores escalares a menos que se pida los valores de los campos devueltos podrán ser documentos anidados, arreglos de escalares o arreglos de documentos.
- Buscar hacer la consulta de la forma más sencilla y eficiente posible.
- Se evaluará el correcto formato de las soluciones:
  - El código entregado debe ser legible.
  - Utilizar indentación de espacios de manera uniforme.

### **Detalle Importante**

El alumno puede usar sus notas personales e internet, pero queda prohibido utilizar modelos generativos (Chat-GPT, LLAMA, etc) y comunicarse con otras personas oral o digitalmente.

Romper alguna de estas dos restricciones invalida el examen

## Entrega

- Se entregará un archivo comprimido `soluciones.tar.gz` o `soluciones.zip` (con `soluciones.js` adentro) con las soluciones de los 5 ejercicios. Separar las soluciones mediante comentarios de Javascript (`/* */` o `//`).
- La entrega se hará mediante el [Aula Virtual](#) en el [correspondiente apartado](#).
  - Tendrán hasta las 17:00 para que se considere una entrega completa. La recomendación es empezar a subir el archivo a las 16:50 para evitar cualquier eventualidad.