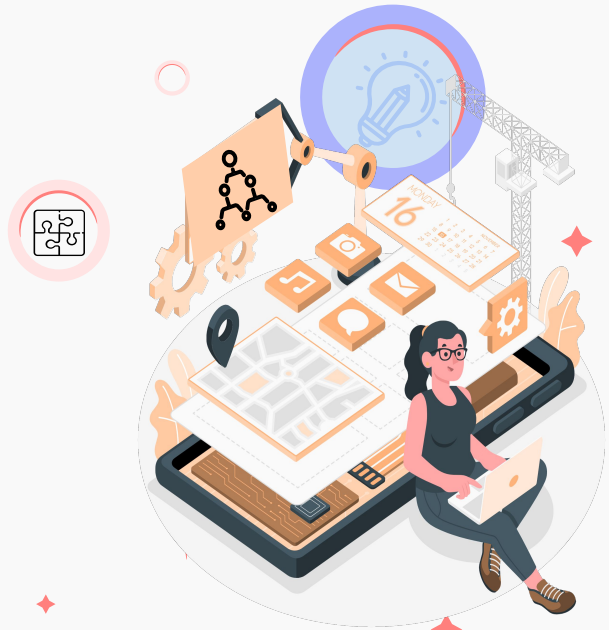
SISTEMAS INTENSIVOS EN DATOS 2

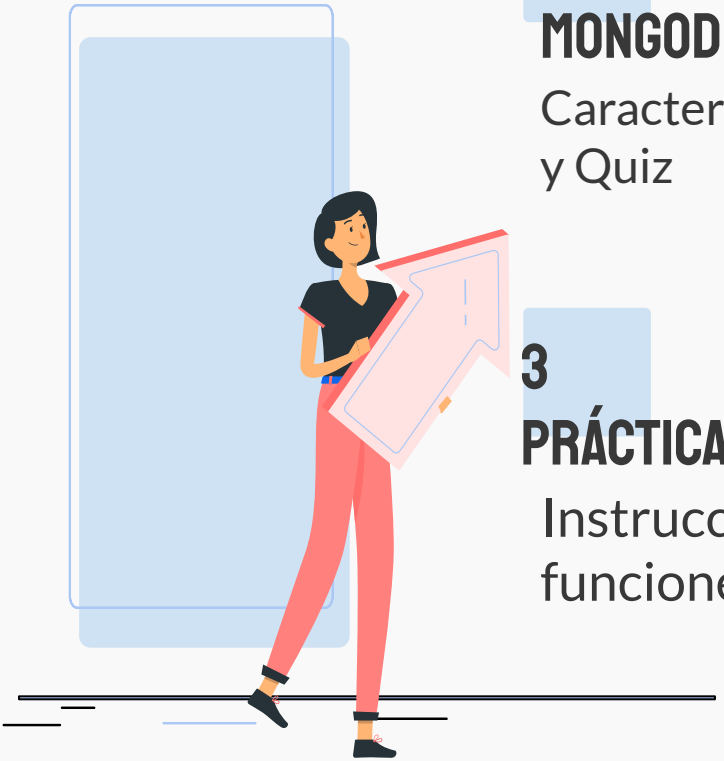
Mario Julián Mora
mario.mora@u.icesi.edu.co

Mónica Rojas
mmrojas@icesi.edu.co

UNIVERSIDAD ICESI

UNIDAD 4 - NO SQL





1

MONGODB

Características principales y Quiz

2

INSTRUCCIONES MONGODB

Recorderis instrucciones básicas

3

PRÁCTICA MONGODB

Instrucciones básicas y funciones de agrupación

4

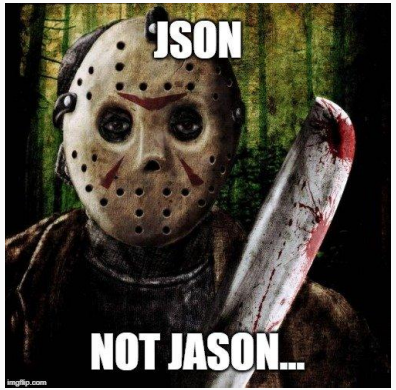
CIERRE

Próxima clase

mongoDB.

MODELO DE DATOS – RDBMS VS MONGODB

RDBMS	MongoDB
Base de datos	Base de datos
Tabla	Colección
Índice	Índice
Fila	Documento JSON
Columna	Campo del documento
Join	Documentos embebidos y búsqueda



CARACTERÍSTICAS PRINCIPALES

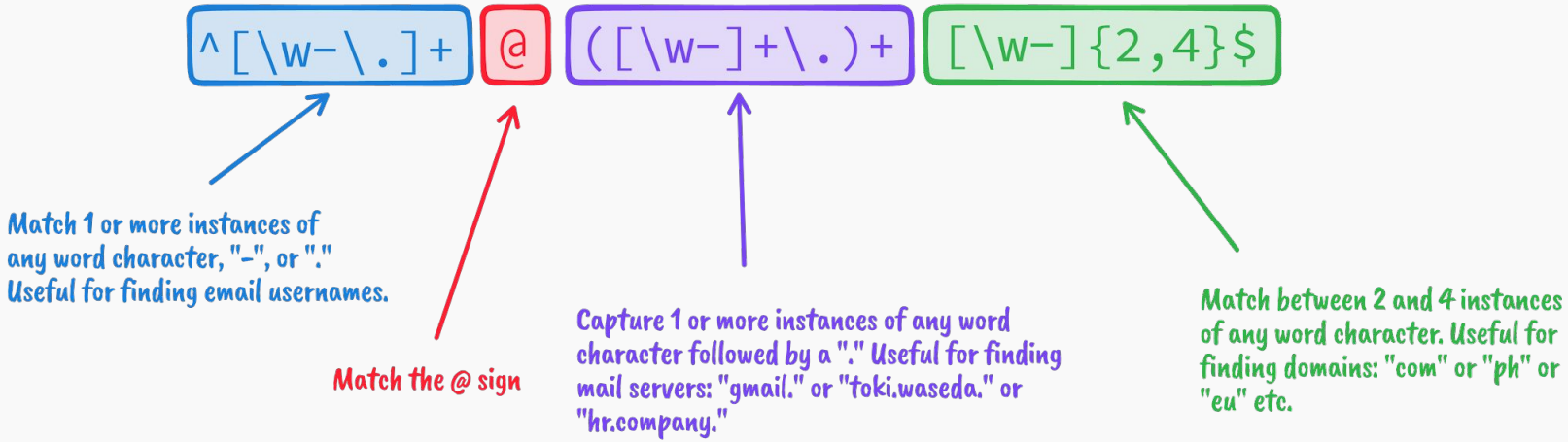
Consultas Ad-hoc

- MongoDB soporta la búsqueda por campos, consultas de rangos y expresiones regulares.
- Las consultas pueden devolver un campo específico del documento pero también puede ser una función JavaScript definida por el usuario.

Indexación

- Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios.
- El concepto de índices en MongoDB es similar a los encontrados en base de datos relacionales.

Expresiones Regulares



- Una expresión regular es una secuencia de caracteres que define un patrón de búsqueda.
- Es una herramienta utilizada por los programadores para buscar, validar y manipular texto de manera eficiente.
- Permite realizar tareas como la búsqueda de coincidencias de caracteres específicos, la validación de formatos y la extracción de información.
- Es útil en la programación para realizar operaciones complejas en texto de forma rápida y precisa.

CARACTERÍSTICAS PRINCIPALES

Agregación

- La función **MapReduce** puede ser utilizada para el procesamiento por lotes de datos y operaciones de agregación.
- Permite que los usuarios puedan obtener el tipo de resultado que se obtiene cuando se utiliza el comando SQL “group-by”.

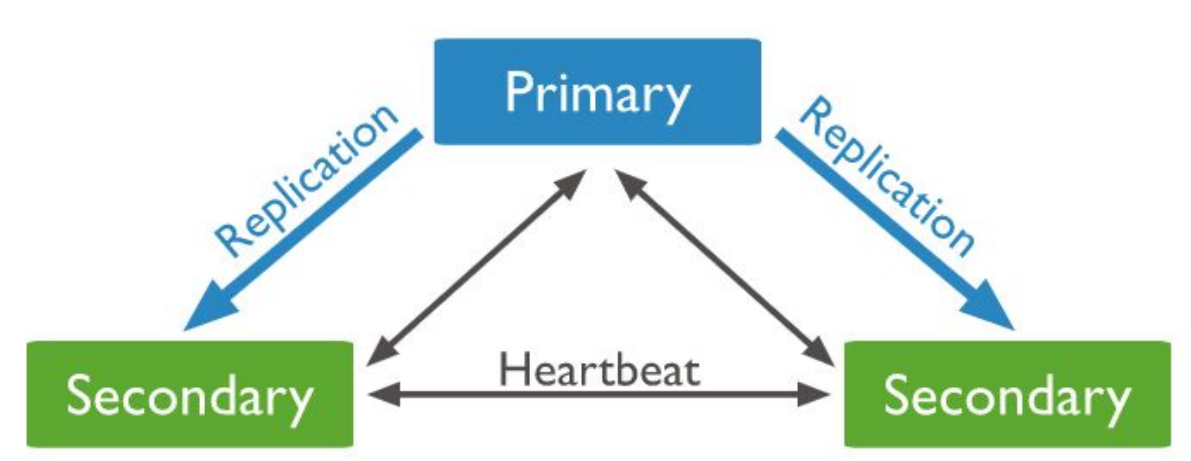
Javascript

- Permite ejecutar Javascript en el lado del servidor

CARACTERÍSTICAS PRINCIPALES

Replicación

- MongoDB soporta el tipo de **replicación maestro-esclavo.**
  - El maestro (primario) puede ejecutar comandos de lectura y escritura.
  - El esclavo (secundario) puede copiar los datos del maestro y sólo se puede usar para lectura
  - El esclavo (secundario) tiene la habilidad de poder elegir un nuevo maestro en caso de que se caiga el servicio con el maestro actual.



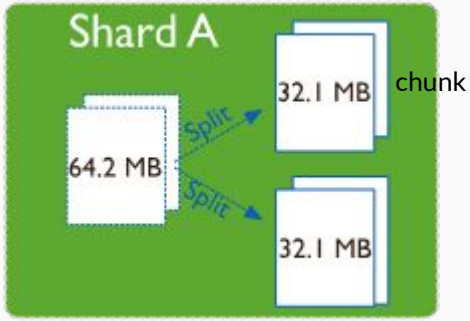
Heartbeats (latidos): Los miembros del conjunto réplica envían latidos (pings) cada dos segundos. Si un latido no regresa dentro de los 10 segundos, los otros miembros lo marcan como inaccesible.

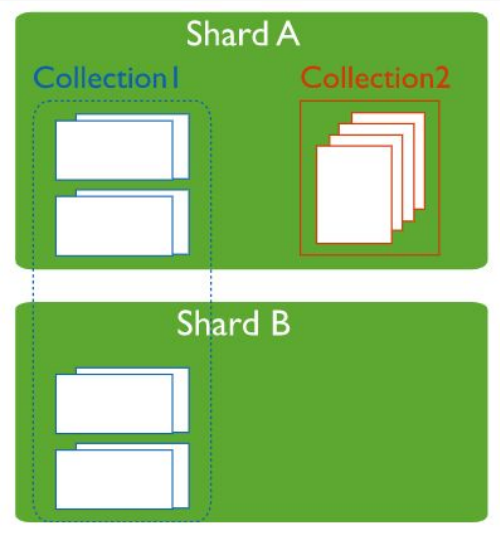
https://docs.mongodb.com/manual/replication/ +info

CARACTERÍSTICAS PRINCIPALES

Balanceo de carga

- MongoDB puede escalar de forma horizontal usando el concepto de “shard”.
  - El desarrollador elige una clave shard, la cual determina cómo serán distribuidos los datos en una colección. Los datos son divididos en rangos (basados en la clave shard) y distribuidos a través de múltiples shard.
  - Un shard es un maestro con uno o más esclavos. (max. 64MB)
- MongoDB tiene la capacidad de ejecutarse en múltiple servidores, balanceando la carga y/o duplicando los datos





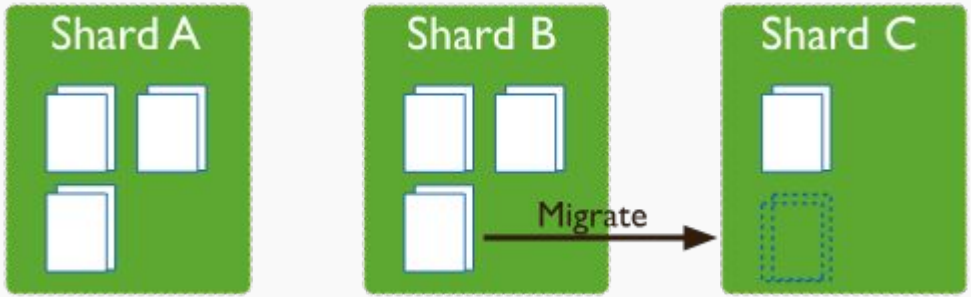
CHUNK: es una agrupación lógica de documentos que MongoDB usa para dividir la información entre los distintos *shards* de un *sharded cluster*.

+info

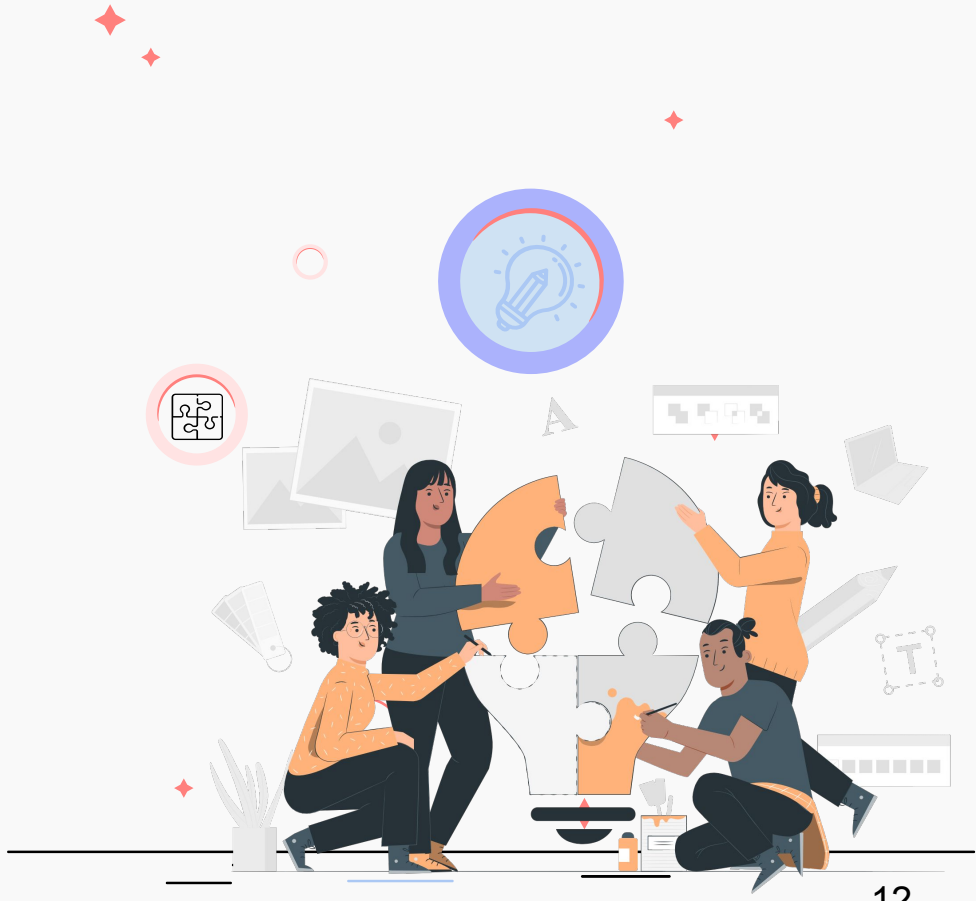
https://docs.mongodb.com/manual/reference/sharding/

CARACTERÍSTICAS PRINCIPALES

El balanceador es un proceso en segundo plano que gestiona las migraciones de chunks. Si la diferencia en el número de trozos entre el fragmento más grande y el más pequeño supera los umbrales de migración, el equilibrador comienza a migrar trozos por todo el clúster para garantizar una distribución uniforme de los datos.



INSTRUCCIONES BÁSICAS + FUNCIONES DE AGREGACIÓN



INSTRUCCIONES BÁSICAS

INSTRUCCIÓN	EJEMPLO
INSERT	db.users.insert({ nombre : "Monica", apellido : "Rojas"});
DISTINCT	db.users.distinct("genero");
FIND	db.users.find();
FIND (Selección)	<> db.users.find({nombre: {$ne : "Maria"}}); AND db.users.find({nombre:"Maria", apellido: "Andrade"}); OR db.users.find({$or:[{nombre:"Maria"},{apellido:"Fernandez"}]}); BETWEEN db.users.find({edad: {$gte:20,$lte:38}}); (entre 20 y 38) LIKE db.users.find({nombre: /Leon/}); (Contiene Leon) LIKE db.users.find({nombre: /^L/}); (Inicia con L) LIKE db.users.find({nombre: /o$/}); (Finaliza con o)
FIND (Proyección)	db.users.find({ }, { apellido : true,_id:0 });
SORT	db.users.find().sort({"nombre":1, "edad":-1})
COUNT	db.users.countDocuments({genero:'M'}); db.users.find({genero:'M'}).count();

OPERADORES

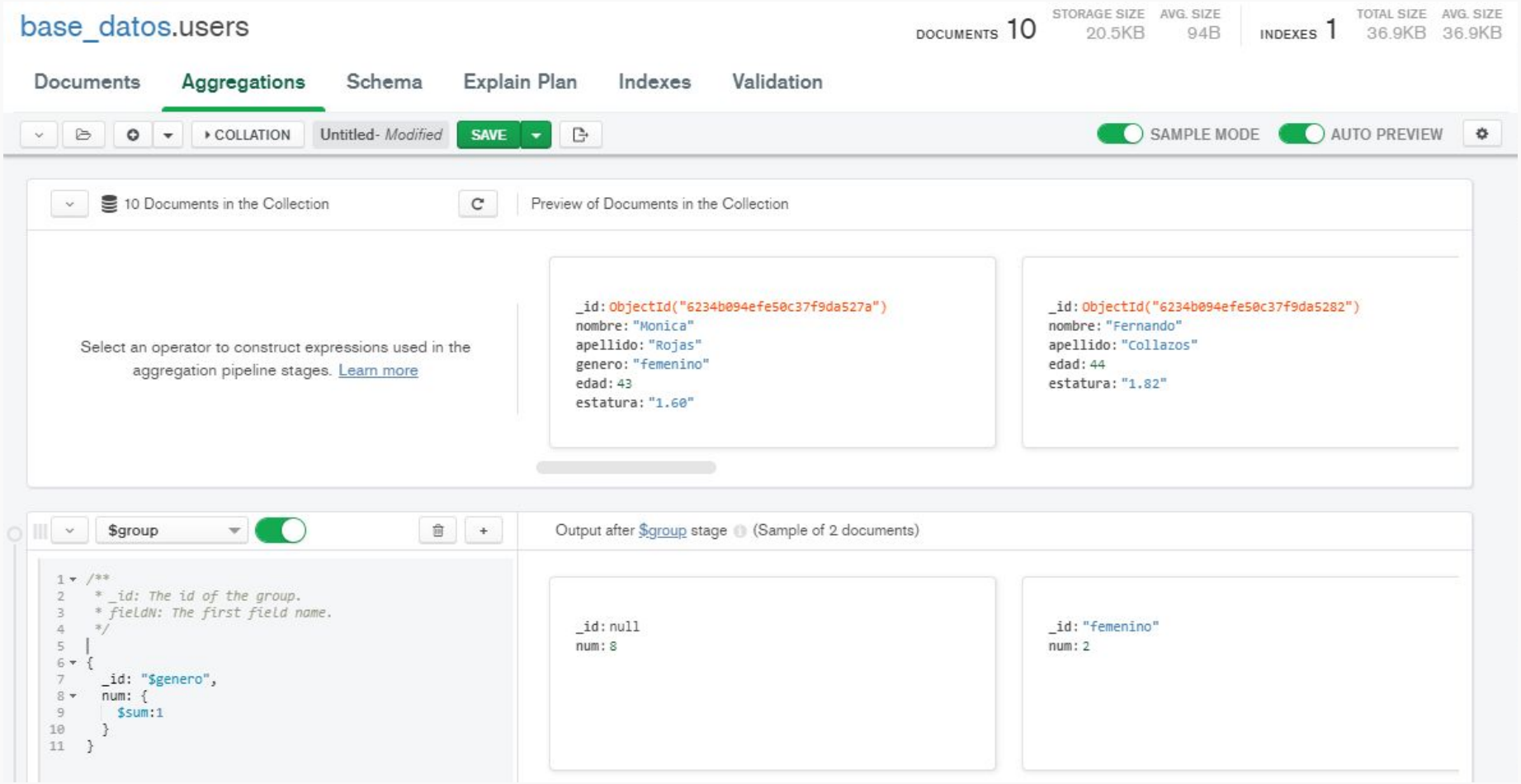
Operador	Expresión
Igual	$eq
Diferente	$ne
Mayor que	$gt
Mayor o igual que	$gte
Menor que	$lt
Menor o igual que	$lte
Existencia en array	$in
Inexistencia en array	$nin

FUNCIONES DE AGREGACIÓN – Contar

```
SELECT genero, count(*) FROM TABLE users
GROUP BY genero;
```

```
db.users.aggregate ([
                        {$group:
                            { _id: "$genero",
                              num: {$sum:1}
                            }
                        }
                    ])
```

```
Atlas Cluster0-shard-0 [primary] base_datos> db.users.aggregate([{$group: {_id: "$genero",num: {$sum:1}}}])
[ { _id: 'femenino', num: 2 }, { _id: null, num: 8 } ]
Atlas Cluster0-shard-0 [primary] base_datos>
```


base_datos.users
DOCUMENTS 10
STORAGE SIZE 20.5KB
AVG. SIZE 94B
INDEXES 1
TOTAL SIZE 36.9KB
AVG. SIZE 36.9KB
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
COLLATION
Untitled- Modified
SAVE
SAMPLE MODE
AUTO PREVIEW
10 Documents in the Collection
Preview of Documents in the Collection
Select an operator to construct expressions used in the aggregation pipeline stages. Learn more
_id: ObjectId("6234b094efe50c37f9da527a")
nombre: "Monica"
apellido: "Rojas"
genero: "femenino"
edad: 43
estatura: "1.60"
_id: ObjectId("6234b094efe50c37f9da5282")
nombre: "Fernando"
apellido: "Collazos"
edad: 44
estatura: "1.82"
$group
/**
* _id: The id of the group.
* fieldN: The first field name.
*/
{
_id: "$genero",
num: {
$sum:1
}
}
Output after $group stage (Sample of 2 documents)
_id: null
num: 8
_id: "femenino"
num: 2

FUNCIONES DE AGREGACIÓN

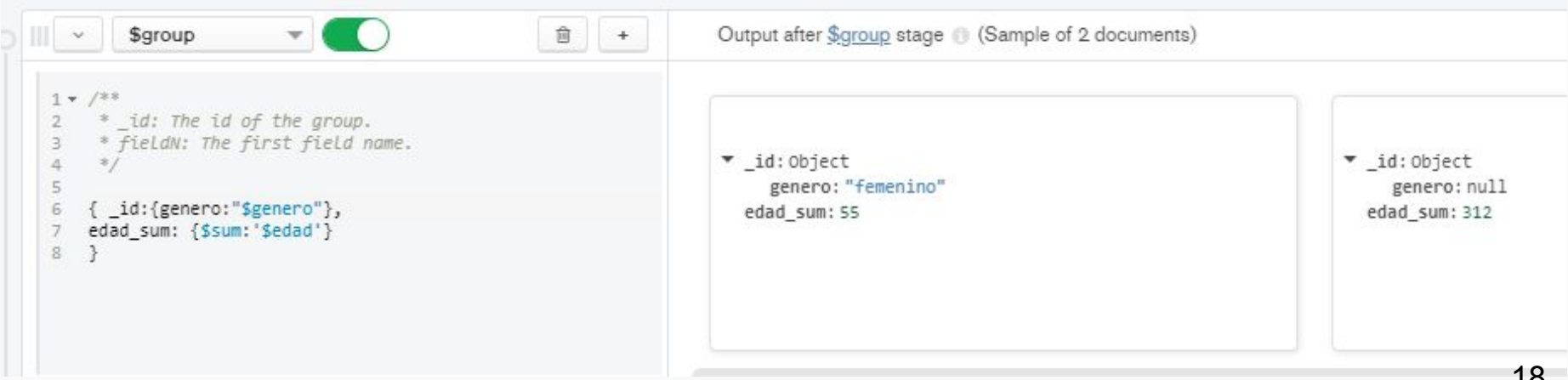
Ya hemos visto una función de agregación, $sum, pero hay muchas otras:

- $sum: suma (o incrementa)
- $avg : calcula la media
- $min: mínimo de los valores
- $max: máximo de los valores
- $push: inserta en un array un valor determinado
- $addToSet: inserta en un array los valores que digamos, pero solo una vez
- $first: obtiene el primer elemento del grupo, a menudo junto con sort
- $last: obtiene el último elemento, a menudo junto con sort

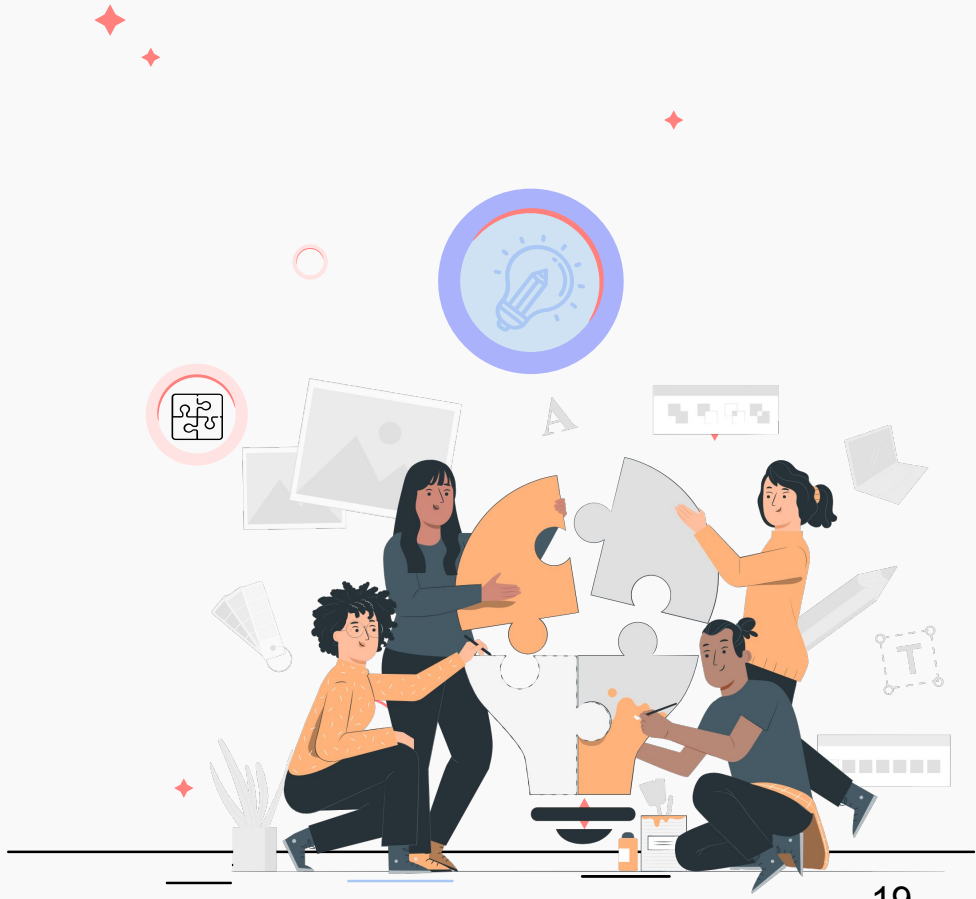
FUNCIONES DE AGREGACIÓN – Contar

```
SELECT genero, sum(edad) FROM TABLE users
GROUP BY genero;
```

```
db.users.aggregate([{$group: {_id: "$genero",edad_sum: {$sum:"$edad"}}}])
```



PRÁCTICA MONGODB TALLER ESTUDIANTES [LINK]



REFERENCIAS

Cielen, D., Meysman, A., & Ali, M. (2016). Introducing Data Science: Big Data. Machine Learning and More, Using Python Tools. Manning, Shelter Island, US, 322.

MOOC BigData: Sistemas gestores de bases de datos orientados a documentos III
https://www.youtube.com/watch?v=eYiebokW2hg

Manual MongoDB
https://docs.mongodb.com/manual/replication/