

Diplomado en Big Data

Gestión de Datos Masivos

LABORATORIO MIGRACIONES DE DATOS CON MONGODB

Docente Evaluador: Ing. Alexander Ramírez Camargo

Presenta:

Ing. Jorge Alberto Gómez López

Ing. Guillermo Alexander Cornejo Argueta

1. Usando los modelos utilizados en el taller1 del modulo 2, hacer una inserción masiva de datos con Insert Many (al menos 1000 registros).
- Utilizando la pagina de <https://www.mockaroo.com/> se crean los mil registros para los siguientes modelos:
 - Usuarios
 - Películas
 - Carros
 - Compañías
 - Animales
 - TarjetasCredito

En donde carros y películas estan relacionadas, a continuación, se muestra parte del código de creación de dichas inserciones:

```
db.createCollection('usuarios');
db.usuarios.insertMany([
  {
    nombres: "Tiffani", apellidos: "Sunock", username: "tsunock8", email: "tsunock8@panel.net", genero: "F", edad: 85, fechaNacimiento: "3/25/2088", nacionalidad: "Sweden" },
    {
      nombres: "Corella", apellidos: "Gerhts", username: "cgerhts1", email: "cgerhts1@peapl.com", genero: "F", edad: 48, fechaNacimiento: "5/2/1998", nacionalidad: "Indonesia" },
    {
      nombres: "Joby", apellidos: "Plaster", username: "jplaster2", email: "jplaster2@iis.gd", genero: "F", edad: 78, fechaNacimiento: "3/22/1974", nacionalidad: "Poland" },
    {
      nombres: "Aray", apellidos: "Clouter", username: "aclouter3", email: "aclouter3@i63.com", genero: "F", edad: 16, fechaNacimiento: "2/13/1959", nacionalidad: "Bahamas" },
    {
      nombres: "Diamid", apellidos: "Barnhart", username: "dbarnhart4", email: "dbarnhart4@bravesites.com", genero: "M", edad: 3, fechaNacimiento: "4/28/2016", nacionalidad: "Brazil" },
    {
      nombres: "Federico", apellidos: "Oulet", username: "foulet5", email: "foulet5@comsenz.com", genero: "M", edad: 38, fechaNacimiento: "5/3/2819", nacionalidad: "China" },
    {
      nombres: "Hermy", apellidos: "Schultzeiss", username: "hschultzeiss6", email: "hschultzeiss6@ebay.com", genero: "M", edad: 89, fechaNacimiento: "2/24/2023", nacionalidad: "United States" },
    {
      nombres: "Cary", apellidos: "Lutckock", username: "clutckock7", email: "clutckock7@google.com.br", genero: "F", edad: 79, fechaNacimiento: "11/9/1997", nacionalidad: "Poland" },
    {
      nombres: "Jillene", apellidos: "Henfre", username: "jhenfre8", email: "jhenfre8@uol.com", genero: "F", edad: 61, fechaNacimiento: "11/30/1984", nacionalidad: "Dominican Republic" },
    {
      nombres: "Parsifal", apellidos: "Keightley", username: "pkeightley9", email: "pkeightley9@shah.com.au", genero: "M", edad: 8, fechaNacimiento: "10/8/2020", nacionalidad: "Syria" },
    {
      nombres: "Hildegard", apellidos: "Cisar", username: "hcisar", email: "hcisar@tumblr.com", genero: "F", edad: 36, fechaNacimiento: "3/29/2087", nacionalidad: "Sweden" },
    {
      nombres: "Nunziata", apellidos: "Long", username: "nlong", email: "nlong@liewriter.net", genero: "F", edad: 89, fechaNacimiento: "7/31/1906", nacionalidad: "China" },
    {
      nombres: "Sheela", apellidos: "Falconar", username: "sfalconar", email: "sfalconar@atena.ne.jp", genero: "F", edad: 28, fechaNacimiento: "3/17/1966", nacionalidad: "Bosnia and Herzegovina" },
    {
      nombres: "Donnica", apellidos: "Cundy", username: "dcundy", email: "dcundy@yellowbook.com", genero: "F", edad: 62, fechaNacimiento: "3/9/1966", nacionalidad: "China" },
    {
      nombres: "Gusta", apellidos: "Sergaunt", username: "gsergaunt", email: "gsergaunt@pinterest.com", genero: "F", edad: 68, fechaNacimiento: "8/21/2089", nacionalidad: "China" },
    {
      nombres: "Riva", apellidos: "Dywick", username: "dywick", email: "dywick@fing.com", genero: "M", edad: 56, fechaNacimiento: "8/19/2080", nacionalidad: "Russia" },
    {
      nombres: "Edna", apellidos: "Spenslove", username: "espenslove", email: "espenslove@chicagotribune.com", genero: "F", edad: 18, fechaNacimiento: "3/18/2018", nacionalidad: "Peru" },
    {
      nombres: "Cati", apellidos: "Menendes", username: "cmenendes", email: "cmenendes@gaol.com", genero: "F", edad: 19, fechaNacimiento: "10/6/1975", nacionalidad: "Japan" },
    {
      nombres: "Catharina", apellidos: "Chartland", username: "chartland", email: "chartland@quidoo.com", genero: "F", edad: 1, fechaNacimiento: "12/21/2087", nacionalidad: "Russia" },
    {
      nombres: "Tilla", apellidos: "Macdough", username: "tmacdough", email: "tmacdough@lyattite.com", genero: "F", edad: 69, fechaNacimiento: "7/31/2009", nacionalidad: "China" },
    {
      nombres: "Patrice", apellidos: "Bushell", username: "pbushell", email: "pbushell@qagsperso-orange.fr", genero: "F", edad: 76, fechaNacimiento: "3/14/2011", nacionalidad: "China" },
    {
      nombres: "Hayley", apellidos: "Galvin", username: "hgalvin", email: "hgalvin@flavors.me", genero: "F", edad: 32, fechaNacimiento: "7/29/2019", nacionalidad: "Mauritania" },
    {
      nombres: "Sandra", apellidos: "Dracey", username: "sdracey", email: "sdracey@blocal.com", genero: "F", edad: 33, fechaNacimiento: "10/19/1994", nacionalidad: "Brazil" },
    {
      nombres: "Marylyne", apellidos: "Liccontilli", username: "mlicontilli", email: "mlicontilli@bluewin.ch", genero: "F", edad: 68, fechaNacimiento: "10/22/1978", nacionalidad: "China" },
    {
      nombres: "Babette", apellidos: "Kalk", username: "bkalk", email: "bkalk@photobucket.com", genero: "F", edad: 108, fechaNacimiento: "5/1/1992", nacionalidad: "Costa Rica" },
    {
      nombres: "Olly", apellidos: "Rames", username: "orames", email: "orames@opera.com", genero: "F", edad: 74, fechaNacimiento: "5/2/2015", nacionalidad: "Brazil" },
    {
      nombres: "Powell", apellidos: "Plain", username: "pplain", email: "pplain@typepad.com", genero: "M", edad: 18, fechaNacimiento: "1/16/2011", nacionalidad: "Paraguay" },
    {
      nombres: "Farly", apellidos: "Almay", username: "falmay", email: "falmay@qdnread.com", genero: "M", edad: 11, fechaNacimiento: "12/18/2018", nacionalidad: "Argentina" },
    {
      nombres: "Amandine", apellidos: "Waylett", username: "awaylett", email: "awaylett@logovin.com", genero: "F", edad: 3, fechaNacimiento: "10/29/2083", nacionalidad: "Portugal" },
    {
      nombres: "Brent", apellidos: "Iglesia", username: "piglesia", email: "piglesia@macrmedia.com", genero: "M", edad: 21, fechaNacimiento: "1/9/1998", nacionalidad: "Mali" },
    {
      nombres: "Arlen", apellidos: "Tudball", username: "atudball", email: "atudball@posteros.com", genero: "M", edad: 55, fechaNacimiento: "5/13/1966", nacionalidad: "Indonesia" },
    {
      nombres: "Skipper", apellidos: "Medler", username: "smedler", email: "smedler@networkadvertising.org", genero: "M", edad: 92, fechaNacimiento: "5/24/1985", nacionalidad: "United States" },
    {
      nombres: "Natal", apellidos: "Alveque", username: "nalveque", email: "nalveque@megashock.us", genero: "M", edad: 71, fechaNacimiento: "12/23/1988", nacionalidad: "Philippines" },
    {
      nombres: "Gerik", apellidos: "Tadueus", username: "gtadueus", email: "gtadueus@bushy.com", genero: "M", edad: 35, fechaNacimiento: "3/12/1969", nacionalidad: "Palestinian Territory" },
    {
      nombres: "Tadeus", apellidos: "Du Hamel", username: "tduhamel", email: "tduhamel@tintervista.org", genero: "M", edad: 7, fechaNacimiento: "6/23/1967", nacionalidad: "Finland" },
    {
      nombres: "Sloan", apellidos: "Bissett", username: "sbissett", email: "sbissett@dailymotion.com", genero: "M", edad: 26, fechaNacimiento: "9/27/2089", nacionalidad: "Indonesia" },
    {
      nombres: "Haf", apellidos: "Borkis", username: "hborkis", email: "hborkis@bluepress.org", genero: "M", edad: 59, fechaNacimiento: "2/22/2087", nacionalidad: "Costa Rica" },
    {
      nombres: "Miquela", apellidos: "Magnin", username: "hmagnin", email: "hmagnin@kickstarter.com", genero: "F", edad: 1, fechaNacimiento: "9/9/1957", nacionalidad: "Russia" },
    {
      nombres: "Tan", apellidos: "Grinagh", username: "igrinagh", email: "igrinagh@yellowbook.com", genero: "M", edad: 75, fechaNacimiento: "6/12/1977", nacionalidad: "Russia" },
  ]
]);
```

Una vez ingresados todos los datos en la base de datos se mostrarán de la siguiente manera en mongodb compass:

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
animales	73.73 KB	1K	172.00 B	1	24.58 KB
carros	4.10 KB	1K	161.00 B	1	4.10 KB
compañías	4.10 KB	1K	81.00 B	1	4.10 KB
películas	86.02 KB	1K	164.00 B	1	24.58 KB
tarjetasCredito	4.10 KB	1K	261.00 B	1	4.10 KB
usuarios	95.30 KB	1K	209.00 B	1	24.58 KB

En donde se puede apreciar que cada modelo tiene ya 1000 registros cada uno.

2. Crear al menos 3 cursores en donde al menos cada cursor consulte 100 registros:

En las capturas mostradas a continuación solo se muestra una parte de los datos consultados.

Filtrar usuarios por nacionalidad y ordenar por edad y mostrar su username, género y edad:

```
var usuarios = db.usuarios.find({
  nacionalidad: "China"
}).sort({ edad: 1 }).pretty();

while (usuarios.hasNext()) {
  var usuario = usuarios.next();
  print("username: " + usuario.username + ", Genero: " +
usuario.genero+ ", Edad: " + usuario.edad );
}
```

```
> var usuarios = db.usuarios.find({
...   nacionalidad: "China"
... }).sort({ edad: 1 }).pretty();
>
> while (usuarios.hasNext()) {
...   var usuario = usuarios.next();
...   print("username: " + usuario.username + ", Genero: " + usuario.genero+ ", Edad: " + usuario.edad );
... }
username: cizakofr2, Genero: F, Edad: 1
username: acarrett4p, Genero: F, Edad: 2
username: mbuckstone2, Genero: M, Edad: 4
username: sridela, Genero: M, Edad: 4
username: mlegier61, Genero: F, Edad: 5
username: abrunger82, Genero: M, Edad: 5
username: maukland6e, Genero: F, Edad: 6
username: rgnewcockb5, Genero: F, Edad: 6
username: mgagghs, Genero: F, Edad: 7
username: cyukhnevichqd, Genero: F, Edad: 8
username: cthwaitej2, Genero: F, Edad: 9
username: ldowdonyk9, Genero: F, Edad: 9
username: abrusbyo5, Genero: F, Edad: 10
username: sbuckler9d, Genero: F, Edad: 11
username: ttebbse0, Genero: M, Edad: 11
username: tworttgb, Genero: F, Edad: 11
username: raloshkinfr, Genero: F, Edad: 11
username: achalklym1, Genero: M, Edad: 11
username: ngumbley1f, Genero: F, Edad: 12
username: sfirmagec6, Genero: M, Edad: 12
username: ilandere8, Genero: M, Edad: 13
username: gachaffyl9, Genero: F, Edad: 13
username: dtaceypk, Genero: F, Edad: 13
username: ntruwert8j, Genero: F, Edad: 14
username: esiddonsd0, Genero: M, Edad: 14
username: rbellisel, Genero: F, Edad: 14
username: tstiffkinsnd, Genero: M, Edad: 14
username: cpebmery4h, Genero: F, Edad: 15
username: bnorthidge8o, Genero: M, Edad: 15
username: jmcgiffie90, Genero: M, Edad: 15
username: mhorribinehp, Genero: F, Edad: 16
username: rcrasswellgw, Genero: M, Edad: 17
username: cdodgsonej, Genero: F, Edad: 18
username: btltmarshmb, Genero: F, Edad: 18
username: tseamarkco0, Genero: F, Edad: 18
username: kplaldein, Genero: M, Edad: 21
username: khorburgh13, Genero: M, Edad: 22
username: kharesnaipgg, Genero: F, Edad: 22
username: sbroadephu, Genero: M, Edad: 23
username: rhearleyqf, Genero: M, Edad: 23
username: landrichakg2, Genero: F, Edad: 24
username: brizzardinj, Genero: F, Edad: 24
username: fcamiller129, Genero: M, Edad: 25
username: rciricic, Genero: F, Edad: 26
username: epyburnn1, Genero: F, Edad: 26
username: ztrickett69, Genero: M, Edad: 27
username: bwiners6c, Genero: M, Edad: 27
username: tlodwickd, Genero: M, Edad: 27
username: gwollastone3, Genero: M, Edad: 27
username: ewhittcockt, Genero: F, Edad: 27
username: dkleinberenn6, Genero: M, Edad: 27
username: nconnerlydm, Genero: M, Edad: 29
username: bhircheh4, Genero: M, Edad: 29
username: foulet5, Genero: M, Edad: 30
username: rdahmke7s, Genero: F, Edad: 32
username: cserginsongd, Genero: F, Edad: 32
```

Filtrar películas por genero múltiple y mostrar el nombre, fecha de estreno y género:

```
var peliculas = db.peliculas.find({
  genero: { $in: ["Comedy", "Drama"] }
}).pretty();
while (peliculas.hasNext()) {
  var pelicula = peliculas.next();
  print("Nombre: " + pelicula.nombre + ", Fecha de estreno: " +
pelicula.fechaEstreno+ ", Genero: " + pelicula.genero );
}
```

```
0 var peliculas = db.peliculas.find({
  genero: { $in: ["Comedy", "Drama"] }
  ...
}).pretty();
0 while (peliculas.hasNext()) {
  ...
  var pelicula = peliculas.next();
  ...
  print("Nombre: " + pelicula.nombre + ", Fecha de estreno: " + pelicula.fechaEstreno+ ", Genero: " + pelicula.genero );
  ...
Nombre: Brana, Fecha de estreno: 1/7/1954, Genero: Drama
Nombre: Molinski, fecha de estreno: 10/2/1927, Genero: Comedy
Nombre: Forrester, fecha de estreno: 7/21/2013, Genero: Drama
Nombre: Lewis, fecha de estreno: 1/6/2022, Genero: Drama
Nombre: Norton, fecha de estreno: 5/3/1954, Genero: Drama
Nombre: Gallo, fecha de estreno: 6/5/1959, Genero: Drama
Nombre: Davila, fecha de estreno: 11/30/2003, Genero: Comedy
Nombre: Lily, fecha de estreno: 11/20/1979, Genero: Drama
Nombre: Torle, fecha de estreno: 7/5/1974, Genero: Comedy
Nombre: Tobey, fecha de estreno: 3/22/1907, Genero: Comedy
Nombre: Jerald, fecha de estreno: 11/20/1957, Genero: Drama
Nombre: Eldin, fecha de estreno: 8/18/1999, Genero: Drama
Nombre: Jonas, fecha de estreno: 4/20/1955, Genero: Comedy
Nombre: Muffin, fecha de estreno: 4/29/1971, Genero: Comedy
Nombre: Sutton, fecha de estreno: 1/9/1985, Genero: Comedy
Nombre: Tracie, fecha de estreno: 8/6/1961, Genero: Drama
Nombre: Aguil, fecha de estreno: 5/1/1981, Genero: Drama
Nombre: Algrich, fecha de estreno: 3/31/2021, Genero: Comedy
Nombre: Leonard, fecha de estreno: 8/22/1985, Genero: Drama
Nombre: Robbin, fecha de estreno: 9/31/2017, Genero: Drama
Nombre: Milty, fecha de estreno: 9/6/1997, Genero: Drama
Nombre: Millery, fecha de estreno: 1/7/1959, Genero: Comedy
Nombre: Spence, fecha de estreno: 11/19/1953, Genero: Comedy
Nombre: Prissie, fecha de estreno: 5/18/2014, Genero: Drama
Nombre: Gregorio, fecha de estreno: 8/1/1966, Genero: Comedy
Nombre: Maryjo, fecha de estreno: 2/15/1999, Genero: Drama
Nombre: Ly, fecha de estreno: 2/16/1992, Genero: Drama
Nombre: Patty, fecha de estreno: 9/12/2015, Genero: Drama
Nombre: Colan, fecha de estreno: 1/31/2002, Genero: Drama
Nombre: Jaxaline, fecha de estreno: 2/9/1964, Genero: Comedy
Nombre: Quentin, fecha de estreno: 2/26/1953, Genero: Comedy
Nombre: Vici, fecha de estreno: 10/1/1976, Genero: Comedy
Nombre: Milne, fecha de estreno: 2/25/1965, Genero: Drama
Nombre: Rosilla, fecha de estreno: 9/4/1953, Genero: Drama
Nombre: Clayborne, fecha de estreno: 11/18/1968, Genero: Drama
Nombre: Lucila, fecha de estreno: 8/19/2013, Genero: Drama
Nombre: Miretita, fecha de estreno: 5/2/1922, Genero: Comedy
Nombre: Glenda, fecha de estreno: 9/2/2017, Genero: Drama
Nombre: Leone, fecha de estreno: 4/12/1981, Genero: Drama
```

Mostrar las tarjetas de crédito que poseen sobregiro activado y mostrar tipo, empresa, limite de sobregiro y país:

```
var tarjetas = db.tarjetasCredito.find({
  sobregiro: true
}).pretty();
while (tarjetas.hasNext()) {
  var tarjeta = tarjetas.next();
  print("Tipo: " + tarjeta.tipo + ", Empresa: " + tarjeta.empresa+ ",
Limite Sobregiro: " + tarjeta.limiteSobregiro+ ", Pais: " + tarjeta.pais
);
}
```

```
0 var tarjetas = db.tarjetasCredito.find({
  sobregiro: true
  ...
}).pretty();
0 while (tarjetas.hasNext()) {
  ...
  var tarjeta = tarjetas.next();
  ...
  print("Tipo: " + tarjeta.tipo + ", Empresa: " + tarjeta.empresa+ ", Limite Sobregiro: " + tarjeta.limiteSobregiro+ ", Pais: " + tarjeta.pais );
  ...
Tipo: Switch, Empresa: Kerlake, Stroman and Green, Limite Sobregiro: undefined, Pais: Czech Republic
Tipo: Switch, Empresa: Schlomer and Sons, Limite Sobregiro: undefined, Pais: Poland
Tipo: Switch, Empresa: MacGoyer, Schmeier and McColynn, Limite Sobregiro: undefined, Pais: Philippines
Tipo: Jcb, Empresa: Abshire, Flatley and Larkin, Limite Sobregiro: undefined, Pais: Portugal
Tipo: Jcb, Empresa: Medhurst, Beatty and Goldner, Limite Sobregiro: undefined, Pais: Poland
Tipo: china-unionpay, Empresa: Matlica-Funk, Limite Sobregiro: undefined, Pais: Bosnia and Herzegovina
Tipo: Instapayment, Empresa: Lubedra-Collins, Limite Sobregiro: undefined, Pais: Indonesia
Tipo: Jcb, Empresa: Kim-Bosco, Limite Sobregiro: undefined, Pais: Dominican Republic
Tipo: Jcb, Empresa: Brakus, Satterfield and Orlison, Limite Sobregiro: undefined, Pais: Croatia
Tipo: visa, Empresa: Vanderport Inc, Limite Sobregiro: undefined, Pais: Ukraine
Tipo: visa-electron, Empresa: Langworth, Hayes and Ortiz, Limite Sobregiro: undefined, Pais: Philippines
Tipo: mastercard, Empresa: Kutich Inc, Limite Sobregiro: undefined, Pais: Georgia
Tipo: Jcb, Empresa: Bruen-Jast, Limite Sobregiro: undefined, Pais: Ukraine
Tipo: mastercard, Empresa: Lennon-Jast, Limite Sobregiro: undefined, Pais: Philippines
Tipo: maestro, Empresa: Labadie-Macejkovic, Limite Sobregiro: undefined, Pais: Brazil
Tipo: visa, Empresa: Gorczany-Rogahn, Limite Sobregiro: undefined, Pais: Ukraine
Tipo: Jcb, Empresa: Purdy, Orlson and Orlson, Limite Sobregiro: undefined, Pais: Uzbekistan
Tipo: americanexpress, Empresa: Parker, Weissnat and Gleason, Limite Sobregiro: undefined, Pais: Germany
Tipo: mastercard, Empresa: Hilley, Uno and Sanford, Limite Sobregiro: undefined, Pais: Russia
Tipo: maestro, Empresa: Medhurst-Kowacek, Limite Sobregiro: undefined, Pais: Czech Republic
Tipo: Laser, Empresa: Blick Inc, Limite Sobregiro: undefined, Pais: Jordan
Tipo: Jcb, Empresa: Mucisette, Stiedemann and Hoppe, Limite Sobregiro: undefined, Pais: Slovenia
Tipo: americanexpress, Empresa: Haley, Bernier and Jenkins, Limite Sobregiro: undefined, Pais: Thailand
Tipo: diners-club-mercad, Empresa: Cernell and Heller, Limite Sobregiro: undefined, Pais: United Kingdom
Tipo: diners-club-carte-blanche, Empresa: Klocko and Sons, Limite Sobregiro: undefined, Pais: China
Tipo: americanexpress, Empresa: Dare Group, Limite Sobregiro: undefined, Pais: Russia
Tipo: Jcb, Empresa: Heller Inc, Limite Sobregiro: undefined, Pais: Poland
Tipo: china-unionpay, Empresa: Volkman and Sons, Limite Sobregiro: undefined, Pais: France
Tipo: Switch, Empresa: Sogun, Heller and Koop, Limite Sobregiro: undefined, Pais: China
Tipo: Switch, Empresa: Schaden-Harvey, Limite Sobregiro: undefined, Pais: Guatemala
Tipo: mastercard, Empresa: Stamm, Runte and Hunte, Limite Sobregiro: undefined, Pais: Indonesia
Tipo: mastercard, Empresa: Cole, Grady and Ritchie, Limite Sobregiro: undefined, Pais: Indonesia
Tipo: Jcb, Empresa: Renner-Collier, Limite Sobregiro: undefined, Pais: Yemen
Tipo: americanexpress, Empresa: Wehner-Bentler, Limite Sobregiro: undefined, Pais: Morocco
```

3. Cree un índice para cada colección y visualícelo.

Se demostrará como se realizó el índice para una colección ya que el proceso es el mismo para las demás colecciones.

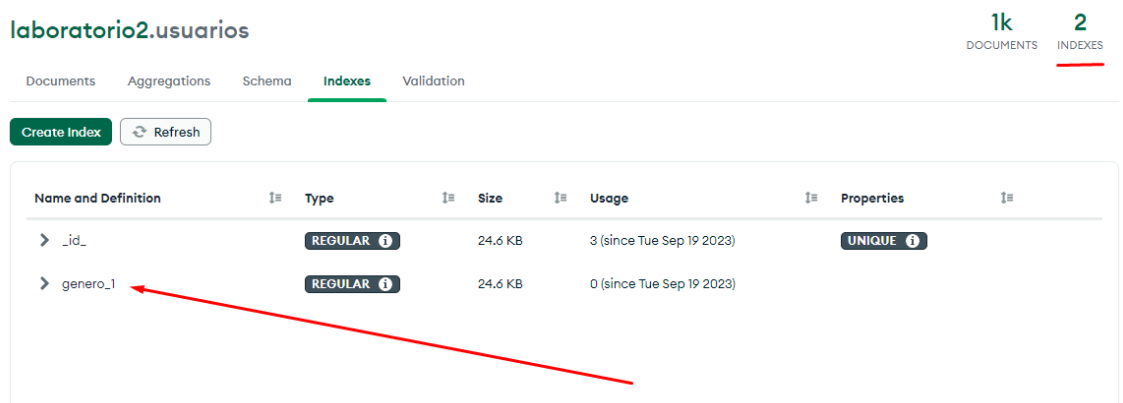
1. Para crear un índice a una colección se hace con el comando `db.usuarios.createIndex({genero: 1});`

```
> db.usuarios.createIndex({genero: 1});
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

2. Para poder ver el índice de la función utilizamos el comando `db.usuarios.getIndexes();`

```
> db.usuarios.getIndexes();
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "laboratorio2.usuarios"
  },
  {
    "v" : 2,
    "key" : {
      "genero" : 1
    },
    "name" : "genero_1",
    "ns" : "laboratorio2.usuarios"
  }
]
```

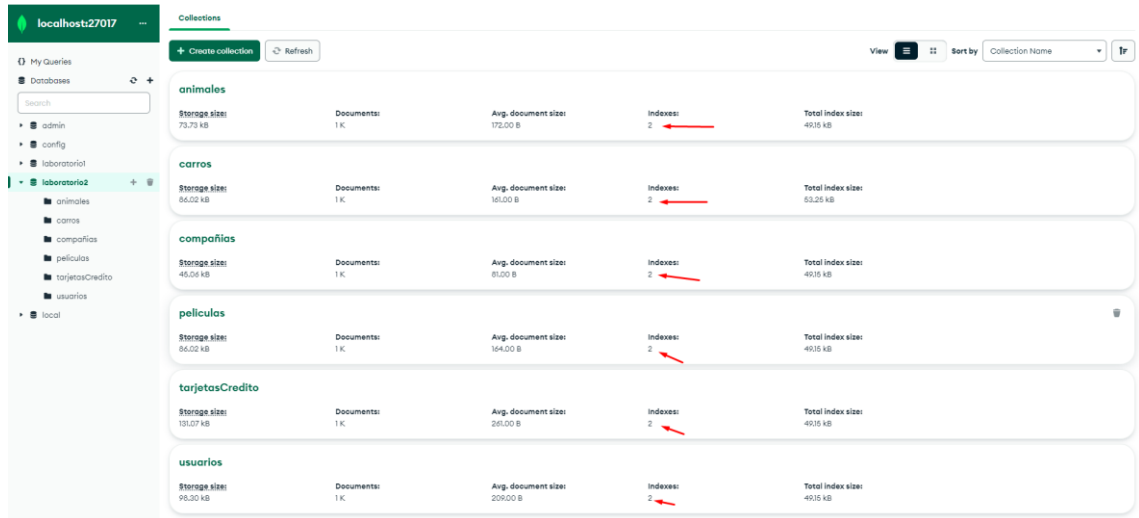
3. También se pueden ver dichos Índices creados en mongodb compass en la opción de Indexes.



The screenshot shows the MongoDB Compass interface for the `laboratorio2.usuarios` collection. The `Indexes` tab is selected, displaying a table of indexes. A red arrow points to the `genero_1` index.

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR	24.6 KB	3 (since Tue Sep 19 2023)	UNIQUE
> genero_1	REGULAR	24.6 KB	0 (since Tue Sep 19 2023)	

Una vez creados todos los índices en los modelos en mongoDB compas nos aparecerá de la siguiente manera:



The screenshot shows the MongoDB Compass interface for a database named 'laboratorio2' on 'localhost:27017'. The left sidebar lists the collections: animales, carros, compañías, películas, tarjetasCredito, and usuarios. The main panel displays a table of these collections with their respective storage and index statistics.

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
animales	73.75 KB	1 K	172.00 B	2	49.75 KB
carros	86.02 KB	1 K	161.00 B	2	83.25 KB
compañías	45.06 KB	1 K	81.00 B	2	49.75 KB
películas	86.02 KB	1 K	164.00 B	2	49.75 KB
tarjetasCredito	131.07 KB	1 K	261.00 B	2	49.75 KB
usuarios	98.30 KB	1 K	209.00 B	2	49.75 KB

4. Implemente el método limit en consultas.

Consulta 1:

Consultar 10 usuarios de nacionalidad china en orden de la menor edad a la mayor:

```
> db.usuarios.find({
...   nacionalidad: "China"
... }).sort({ edad: 1 }).limit(10).pretty();
{
  "_id" : ObjectId("650a4b9942cf1ad6b00e9193"),
  "nombres" : "Coretta",
  "apellidos" : "Izakof",
  "username" : "cizakofr2",
  "email" : "cizakofr2@cam.ac.uk",
  "genero" : "F",
  "edad" : 1,
  "fechaNacimiento" : "5/5/2007",
  "nacionalidad" : "China"
}

{
  "_id" : ObjectId("650a4b9942cf1ad6b00e8e6e"),
  "nombres" : "Alfi",
  "apellidos" : "Carrett",
  "username" : "acarrett4p",
  "email" : "acarrett4p@ameblo.jp",
  "genero" : "F",
  "edad" : 2,
  "fechaNacimiento" : "2/18/1974",
  "nacionalidad" : "China"
}

{
  "_id" : ObjectId("650a4b9942cf1ad6b00e8fbf"),
  "nombres" : "Maxy",
  "apellidos" : "Buckston",
  "username" : "mbuckstone2",
  "email" : "mbuckstone2@behance.net",
  "genero" : "M",
  "edad" : 4,
  "fechaNacimiento" : "3/15/1962",
  "nacionalidad" : "China"
}

{
  "_id" : ObjectId("650a4b9942cf1ad6b00e90c3"),
  "nombres" : "Shaw",
  "apellidos" : "Ride",
  "username" : "sridela",
  "email" : "sridela@ebay.com",
  "genero" : "M",
  "edad" : 4,
  "fechaNacimiento" : "8/25/1981",
  "nacionalidad" : "China"
}
```

Consulta 2:

Consultar 10 películas que sean del género de drama o comedia

```
> db.peliculas.find({
...   genero: { $in: ["Comedy", "Drama"] }
... }).limit(10).pretty();
{
  "_id" : ObjectId("650a4bb642cf1ad6b00e91b7"),
  "nombre" : "Heinrik",
  "fechaEstreno" : "10/2/1987",
  "director" : "Heinrik Sommerville",
  "genero" : "Comedy",
  "pais" : "Honduras",
  "empresa" : "Tagcat"
}
{
  "_id" : ObjectId("650a4bb642cf1ad6b00e91cb"),
  "nombre" : "Davina",
  "fechaEstreno" : "11/30/2003",
  "director" : "Davina Lillywhite",
  "genero" : "Comedy",
  "pais" : "Croatia",
  "empresa" : "Shuffletag"
}
{
  "_id" : ObjectId("650a4bb642cf1ad6b00e91d2"),
  "nombre" : "Torie",
  "fechaEstreno" : "7/5/1974",
  "director" : "Torie Provest",
  "genero" : "Comedy",
  "pais" : "Argentina",
  "empresa" : "Zazio"
}
{
  "_id" : ObjectId("650a4bb642cf1ad6b00e91d5"),
  "nombre" : "Toby",
  "fechaEstreno" : "3/27/1997",
  "director" : "Toby Garatty",
  "genero" : "Comedy",
  "pais" : "China",
  "empresa" : "Livefish"
}
{
  "_id" : ObjectId("650a4bb642cf1ad6b00e91e5"),
  "nombre" : "Jonas",
  "fechaEstreno" : "4/30/1955",
  "director" : "Jonas Boadby",
  "genero" : "Comedy",
}
```

Consulta 3:

Consultar 16 carros que sean del fabricante Honda.

```
> db.carros.find({
...   fabricante: "Honda"
... }).sort({modeloAnio: 1}).limit(16).pretty();
{
  "_id" : 260,
  "modelo" : "Civic",
  "fabricante" : "Honda",
  "modeloAnio" : 1988,
  "vinCarro" : "4USBT53545L940791",
  "fechaSalida" : "3/14/1965",
  "precio" : "$20507.39"
}
{
  "_id" : 402,
  "modelo" : "CR-X",
  "fabricante" : "Honda",
  "modeloAnio" : 1988,
  "vinCarro" : "1FTMF1EW7AK309387",
  "fechaSalida" : "8/30/1963",
  "precio" : "$55465.54"
}
{
  "_id" : 969,
  "modelo" : "Accord",
  "fabricante" : "Honda",
  "modeloAnio" : 1995,
  "vinCarro" : "JN8AZ2KR4AT701775",
  "fechaSalida" : "10/21/1961",
  "precio" : "$13231.88"
}
{
  "_id" : 36,
  "modelo" : "Insight",
  "fabricante" : "Honda",
  "modeloAnio" : 2000,
  "vinCarro" : "JTHBE1D21F5507544",
  "fechaSalida" : "11/6/1984",
  "precio" : "$36898.24"
}
{
  "_id" : 597,
  "modelo" : "CR-V",
  "fabricante" : "Honda",
  "modeloAnio" : 2000,
  "vinCarro" : "1N6AF0LY8EN354716",
}
```

Consulta 4:

Consultar 15 animales que tengan estatus de rescatado pero que no estén en peligro:

```
> db.animales.find({
...   rescatado: true,
...   enPeligro: false
... }).limit(15).pretty();
{
  "_id" : ObjectId("650b341b5598210e991be8f2"),
  "nombreComun" : "Pacific gull",
  "nombreCientifico" : "Gabianus pacificus",
  "pais" : "Indonesia",
  "anioRegistro" : 1986,
  "extinto" : false,
  "enPeligro" : false,
  "rescatado" : true
}
{
  "_id" : ObjectId("650b341b5598210e991be8f8"),
  "nombreComun" : "Caracal",
  "nombreCientifico" : "Felis caracal",
  "pais" : "Canada",
  "anioRegistro" : 1997,
  "extinto" : false,
  "enPeligro" : false,
  "rescatado" : true
}
{
  "_id" : ObjectId("650b341b5598210e991be8fc"),
  "nombreComun" : "Squirrel, indian giant",
  "nombreCientifico" : "Ratufa indica",
  "pais" : "Thailand",
  "anioRegistro" : 2007,
  "extinto" : false,
  "enPeligro" : false,
  "rescatado" : true
}
{
  "_id" : ObjectId("650b341b5598210e991be908"),
  "nombreComun" : "Booby, blue-footed",
  "nombreCientifico" : "Sula nebouxii",
  "pais" : "China",
  "anioRegistro" : 1997,
  "extinto" : true,
  "enPeligro" : false,
  "rescatado" : true
}
{
  "_id" : ObjectId("650b341b5598210e991be90c"),
  "nombreComun" : "Pygmy possum",
  "nombreCientifico" : "Acrobates pygmaeus",
  "pais" : "Mexico",
  "anioRegistro" : 2008,
```


5. Cree una nueva colección implementando un método validador

Para el siguiente ejemplo, se ha creado una nueva colección llamada “empleados”, dicha colección contiene documentos con las siguientes características:

Documento empleado:

- **usuario:** nombre del usuario para el empleado
- **password:** contraseña para la cuenta de usuario, la contraseña esta restringida por las siguientes características:
 - La contraseña debe estar compuesta de mínimo 8 caracteres
 - La contraseña debe contener por lo menos una letra
 - La contraseña debe contener por lo menos un número
 - La contraseña debe contener por lo menos un carácter especial
- **tipo:** tipo de usuario registrado en la colección, este campo está restringido por las siguientes características:
 - Solo existen 2 tipos de usuarios válidos: Admin, Usuario

Para implementar una colección que contenga un documento con dichas características, se hizo uso de validadores en mongodb

```
db.createCollection("empleados", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: [ "usuario", "password", "tipo" ],
      properties: {
        usuario: {
          bsonType: "string",
          description: "debe ser una cadena de caracteres (requerido)"
        },
        password: {
          bsonType: "string",
          minLength: 8,
          pattern: "^(?=.*[A-Za-z])(?=.*\\d)(?=.*[@$!%*#?&])[A-Za-z\\d@$!%*#?&]{8,}$",
          description: "debe ser una cadena de caracteres de por lo menos 8 caracteres, que contenga letras, numeros, caracteres especiales"
        },
        tipo: {
          bsonType: "string",
          enum: ['Admin','Usuario'],
          description: "Solo 2 tipos aceptados: Admin o Usuario"
        }
      }
    }
  }
})
```

A continuación, se realizan algunas observaciones acerca de la colección (y sus validadores) creados:

- **required:** En este campo se puede especificar la lista de campos que son requeridos al momento de querer insertar un documento.
- **password -> minLength:** propiedad donde se especifica el tamaño mínimo para el campo en cuestión (en este caso, una contraseña de minimo 8 caracteres)
- **password -> pattern:** propiedad donde se especifica el regex para verificar si el campo cumple con las reglas definidas por dicho regex.
- **tipo -> enum:** propiedad donde se especifica todos los valores que son válidos para un campo, en este ejemplo, se especifica que solamente Admin y Usuario son los dos unicos valores validos.

A continuación, se muestra un ejemplo cuando se quiere insertar un registro que no cumple con las condiciones estipuladas por el validador (contraseña no es por lo menos 8 caracteres, y no contiene una letra, un número, un carácter especial al mismo tiempo):

```
> db.empleados.insertMany([
... {usuario: "allanpoe", password: "123", tipo: "Admin"}
... ]);
2023-09-30T18:29:15.761-0600 E QUERY [js] BulkWriteError({
  "writeErrors" : [
    {
      "index" : 0,
      "code" : 121,
      "errmsg" : "Document failed validation",
      "op" : {
        "_id" : ObjectId("6518bd5baab0df0ad549a0d7"),
        "usuario" : "allanpoe",
        "password" : "123",
        "tipo" : "Admin"
      }
    }
  ],
  "writeConcernErrors" : [ ],
  "nInserted" : 0,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
}) :
BulkWriteError({
  "writeErrors" : [
    {
      "index" : 0,
      "code" : 121,
```

A continuación, se muestra un ejemplo de inserción que cumple con todas las reglas definidas por el validador:

```
> db.empleados.insertMany([
... {usuario: "allanpoe", password: "admin123$", tipo: "Admin"}
... ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6518bdbeaab0df0ad549a0d8")
  ]
}
>
```

6. Implemente un método Explain para alguna de las colecciones y sustente su uso

A continuación, se muestra un ejemplo donde se ejecuta el método explain para la siguiente query: Obtener el listado de películas cuyo país de origen son china o japon

```
> db.peliculas.explain().find({$or: [ {pais:'Japan'}, {pais:'China'}]});
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "taller2.peliculas",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "$or" : [
        {
          "pais" : {
            "$eq" : "Japan"
          }
        },
        {
          "pais" : {
            "$eq" : "China"
          }
        }
      ]
    },
    "queryHash" : "F860F5C3",
    "planCacheKey" : "7843AF48",
    "winningPlan" : {
      "stage" : "SUBPLAN",
      "inputStage" : {
        "stage" : "FETCH",
        "inputStage" : {
          "stage" : "IXSCAN",
          "keyPattern" : {
            "pais" : 1
          },
          "indexName" : "pais_1",
          "isMultiKey" : false,
          "multiKeyPaths" : {
            "pais" : [ ]
          },
          "isUnique" : false,
          "isSparse" : false,
          "isPartial" : false,
          "indexVersion" : 2,
          "direction" : "forward",
          "indexBounds" : {
            "pais" : [
              ["China", "China"],
              ["Japan", "Japan"]
            ]
          }
        }
      }
    },
    "rejectedPlans" : [ ]
  },
  "serverInfo" : {
    "host" : "lap-033",
    "port" : 27017,
    "version" : "4.2.24",
    "gitVersion" : "5e4ec1d24431fcdd28b579a024c5c801b8cde4e2"
  },
  "ok" : 1
}
```

El uso de explain nos permite comprender la manera en la que mongodb, de manera interna, busca ejecutar la query que en este caso nosotros hemos definido, de este modo nos permite conocer cualquier punto de optimización en caso de que se busque mejorar el tiempo de ejecución de dicha query. En el ejemplo para la consulta anterior, se pueden hacer las siguientes observaciones:

- La query se ejecutará en la colección “peliculas” (detallado en la sección namespace de la respuesta)
- Stage = FETCH nos indica que la query en cuestión debe devolver documentos de la colección como parte de su respuesta

- Stage = IXSCAN nos indica que, dentro del Stage = FETCH, para determinar si un documento debe ser incluido como parte de la respuesta, se hará a través un índice, en este caso el índice “pais_1” que es un índice que ordena de manera ascendente (keyPattern = pais : 1) los registros en base al campo “pais”
- IndexBounds = pais indica que, los valores de índices a buscar, son los valores “China” o “Japan”
- RejectedPlan = [] nos indica que, para ejecutar dicha consulta, no existen ningunas variantes que hayan resultado “menos optimas” para ejecutar dicha consulta.

7. Implemente un ejemplo del uso de réplicas con mongodb

Para el siguiente ejemplo de implementación de replicas en mongodb, se detallan las características / condiciones de dicha implementación:

- El sistema de replicas estará conformado por 2 servidores mongodb, uno fungiendo como db primaria (donde se podrán realizar operaciones de lectura y escritura) mientras que el segundo servidor fungirá como replica (que recibirá todos los cambios que provengan del servidor primario)
- Ambos servidores mongodb, se encontrarán dentro de la misma computadora física (localhost)
- A continuación, se detallan los aspectos técnicos de las instancias de mongodb:
 - Instancia “master”:
 - **Servidor:** localhost
 - **Puerto:** 27018
 - **Ruta física de almacenamiento de datos:**
C:/mongodb_instance/primary
 - **Nombre de replica:** replicaUsuarios
 - Instancia “slave”
 - **Servidor:** localhost
 - **Puerto:** 27019
 - **Ruta física de almacenamiento de datos:**
C:/mongodb_instance/slave1
 - **Nombre de replica:** replicaUsuarios

Paso 1: Levantamiento de instancias

Se ejecuta el siguiente comando para levantar la instancia mongodb primaria (instancia master)

mongod --replSet replicaUsuarios --dbpath "C:/mongodb_instance/primary" --port 27018

```
C:\Users\jorge.gomez>mongod --replSet replicaUsuarios --dbpath "C:/mongodb_instance/primary" --port 27018
2023-09-30T20:00:21.209-0600 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2023-09-30T20:00:21.213-0600 W ASIO [main] No TransportLayer configured during NetworkInterface startup
2023-09-30T20:00:21.215-0600 I CONTROL [initandlisten] MongoDB starting : pid=30448 port=27018 dbpath=C:/mongodb_instance/primary 64-bit host=lap-033
2023-09-30T20:00:21.215-0600 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2023-09-30T20:00:21.215-0600 I CONTROL [initandlisten] db version v4.2.24
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] git version: 5e4ec1d24431fcdd28b579a024c5c801b8cde4e2
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] allocator: tcmalloc
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] modules: none
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] build environment:
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] distmod: 2012plus
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] distarch: x86_64
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] target arch: x86_64
2023-09-30T20:00:21.216-0600 I CONTROL [initandlisten] options: { net: { port: 27018 }, replication: { replSet: "replicaUsuarios" }, storage: { dbPath: "C:/mongodb_instance/primary" } }
2023-09-30T20:00:21.218-0600 I STORAGE [initandlisten] Detected data files in C:/mongodb_instance/primary created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2023-09-30T20:00:21.218-0600 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=7566M,cache_overflow=(file_max=0M),session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress],
2023-09-30T20:00:21.240-0600 I STORAGE [initandlisten] WiredTiger message [1696125621:239378][30448:140719738136400],
txn-recover: Recovering log 2 through 3
2023-09-30T20:00:21.291-0600 I STORAGE [initandlisten] WiredTiger message [1696125621:291780][30448:140719738136400],
txn-recover: Recovering log 3 through 3
2023-09-30T20:00:21.336-0600 I STORAGE [initandlisten] WiredTiger message [1696125621:336629][30448:140719738136400],
```

Se ejecuta el siguiente comando en otra consola cmd, para levantar la instancia secundaria de mongodb (instancia "slave")

mongod --replSet replicaUsuarios --dbpath "C:/mongodb_instance/slave1" --port 27019

```
C:\Users\jorge.gomez>mongod --replSet replicaUsuarios --dbpath "C:/mongodb_instance/slave1" --port 27019
2023-09-30T20:01:19.715-0600 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2023-09-30T20:01:19.717-0600 W ASIO [main] No TransportLayer configured during NetworkInterface startup
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] MongoDB starting : pid=37104 port=27019 dbpath=C:/mongodb_instance/slave1 64-bit host=lap-033
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] db version v4.2.24
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] git version: 5e4ec1d24431fcdd28b579a024c5c801b8cde4e2
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] allocator: tcmalloc
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] modules: none
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] build environment:
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] distmod: 2012plus
2023-09-30T20:01:19.720-0600 I CONTROL [initandlisten] distarch: x86_64
2023-09-30T20:01:19.721-0600 I CONTROL [initandlisten] target arch: X86_64
2023-09-30T20:01:19.721-0600 I CONTROL [initandlisten] options: { net: { port: 27019 }, replication: { replSet: "replicaUsuarios" }, storage: { dbPath: "C:/mongodb_instance/slave1" } }
2023-09-30T20:01:19.722-0600 I STORAGE [initandlisten] Detected data files in C:/mongodb_instance/slave1 created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2023-09-30T20:01:19.722-0600 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=7566M,cache_overflow=(file_max=0M),session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress],
2023-09-30T20:01:19.747-0600 I STORAGE [initandlisten] WiredTiger message [1696125679:746876][37104:140719738136400],
txn-recover: Recovering log 2 through 3
2023-09-30T20:01:19.803-0600 I STORAGE [initandlisten] WiredTiger message [1696125679:802748][37104:140719738136400],
txn-recover: Recovering log 3 through 3
2023-09-30T20:01:19.856-0600 I STORAGE [initandlisten] WiredTiger message [1696125679:856124][37104:140719738136400],
txn-recover: Main recovery loop: starting at 2/4736 to 3/256
```

Paso 2: Configurando el replicaSet dentro de la instancia maestra

Se procede a conectarse a la instancia primaria de mongodb

mongo --port 27018

Dentro de la instancia primaria, se especifica la configuración del sistema de replicas, para este caso, se ha elegido la siguiente configuración:

```
config = { _id: "replicaUsuarios", members: [
  { _id: 0, host: "localhost:27018"},
  { _id: 1, host: "localhost:27019"}
]};
```

Esta configuración representa lo siguiente:

- El nombre del sistema de replicas se llama “replicaUsuarios”
- El sistema de replicas está conformado por 2 instancias, uno que se encuentra en el puerto 27018, otro que se encuentra en el puerto 27019

Se ejecuta la configuración anterior y se procede a ejecutar el comando: **rs.initiate(config)**

```
> config = { _id: "replicaUsuarios", members: [
... { _id: 0, host: "localhost:27018"},
... { _id: 1, host: "localhost:27019"}
... ]};
{
  "_id" : "replicaUsuarios",
  "members" : [
    {
      "_id" : 0,
      "host" : "localhost:27018"
    },
    {
      "_id" : 1,
      "host" : "localhost:27019"
    }
  ]
}
> rs.initiate(config)_
```

Se observa que, al momento de ejecutar el ultimo comando, la línea de comandos cambia el sufijo de la instancia:

```
}
> rs.initiate(config)
{ "ok" : 1 }
replicaUsuarios:SECONDARY>
```

Nos volvemos a conectar de nuevo a la instancia primaria, para que aparezca de manera correcta el sufijo “PRIMARY” (representando que estamos conectados a la instancia primaria de la réplica):

```
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
replicaUsuarios:PRIMARY> _
```

Paso 3: Verificaciones finales de la réplica creada

Si nos conectamos a la instancia secundaria, e intentamos ejecutar un comando (como por ejemplo “show dbs”), obtenemos el siguiente resultado:

```
replicaUsuarios:SECONDARY> show dbs
2023-09-30T20:11:52.011-0600 E QUERY [js] uncaught exception: Error: listDatabases failed:{
  "operationTime" : Timestamp(1696126309, 1),
  "ok" : 0,
  "errmsg" : "not master and slaveOk=false",
  "code" : 13435,
  "codeName" : "NotPrimaryNoSecondaryOk",
  "$clusterTime" : {
    "clusterTime" : Timestamp(1696126309, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:147:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:99:12
shellHelper.show@src/mongo/shell/utils.js:906:13
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
replicaUsuarios:SECONDARY> _
```

Si leemos el mensaje con atención, comenta que la instancia en la que estamos conectadas “Not master and slaveOk=false” indicando que esta instancia es nada más una réplica, y por ende no se pueden ejecutar operaciones dentro de dicha instancia (solo se pueden realizar operaciones en la instancia master).

Para terminar de corroborar que la réplica se ha configurado correctamente, realizamos una conexión a la instancia master, y creamos una colección, para poder observar el output de cada una de las instancias que conforman el replica set:

```
replicaUsuarios:PRIMARY> db.usuarios.insertOne({});
{
  "acknowledged" : true,
  "insertedId" : ObjectId("6518d6688405e1ea5bc7b2ab")
}
replicaUsuarios:PRIMARY>
```

Dentro de la consola de la instancia primaria, obtenemos el log de que una nueva colección ha sido creada:

```
2023-09-30T20:16:08.828-0600 I STORAGE [conn16] createCollection: test.usuarios with generated UUID: 0ac8c94b-a517-4787-af30-22427bc6dfba and options: {}
2023-09-30T20:16:08.836-0600 I INDEX [conn16] index build: done building index _id_ on ns test.usuarios
2023-09-30T20:17:09.487-0600 I QUERY [clientcursorsmon] Cursor id 294648868759700879 timed out, idle since 2023-09-30T20:07:05.514-0600
```

Mientras que, dentro de la consola de la instancia secundaria, obtenemos el mensaje que ha replicado la creación de la colección dentro de su instancia, de esta manera, comprobando que el sistema de replicas ha sido configurado correctamente:

```
2023-09-30T20:16:08.838-0600 I STORAGE [repl-writer-worker-0] createCollection: test.usuarios with provided UUID: 0ac8c94b-a517-4787-af30-22427bc6dfba and options: { uuid: UUID("0ac8c94b-a517-4787-af30-22427bc6dfba") }
2023-09-30T20:16:08.844-0600 I INDEX [repl-writer-worker-0] index build: done building index _id_ on ns test.usuarios
```