

# Base de datos NoSQL

## El boom de los datos en Internet



## Base de datos NoSQL

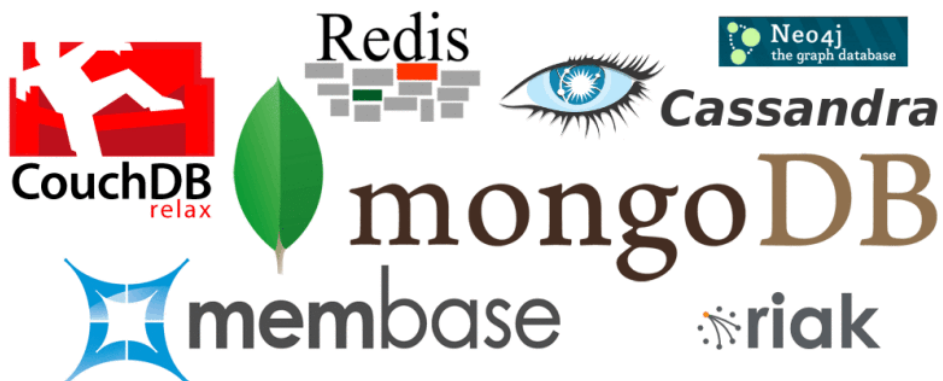
Conocidas también como “Not only SQL” son base de datos que no están construidas con el enfoque de las base relacionales

- No usan SQL como principal lenguaje de consulta
- Los datos almacenados no necesitan estructuras fijas como tablas
- No garantizan completamente ACID(atomicidad, consistencia, aislamiento y durabilidad)
- Escalan muy bien horizontalmente

## Base de datos NoSQL

Con el enfoque de ser escalables y eficientes

- No usan SQL como el principal lenguaje de consulta
- Los datos almacenados no necesitan estructuras fijas como tablas
- No garantizan completamente ACID(atomicidad, consistencia, aislamiento y durabilidad)
- Escalan muy bien horizontalmente



## Base de datos SQL

Se enfocan principalmente en mantener la consistencia y la seguridad de los datos almacenados.

- Usan SQL como el principal lenguaje de consulta.
- Las tablas son estructuras fijas
- Enfocados a garantizar ACID(atomicidad, consistencia, aislamiento y durabilidad)



# MongoDB

- Gestor de base de datos basado en documentos
- El mas conocido de los gestores de base de datos NoSQL
- Utiliza como lenguaje de consultas Javascript
- Los datos se guardan en formato tipo JSON, llamado BSON
- Las estructuras de los documentos no son fijas
- Se puede adicionar, eliminar y modificar libremente los campos

## Comparación entre MongoDB y gestores de base de datos relacionales

MongoDB	DB relacional
Base de datos	Base de datos
Colecciones	Tablas
Documentos	Filas
Campos	Columnas

# Modelado de datos en MongoDB

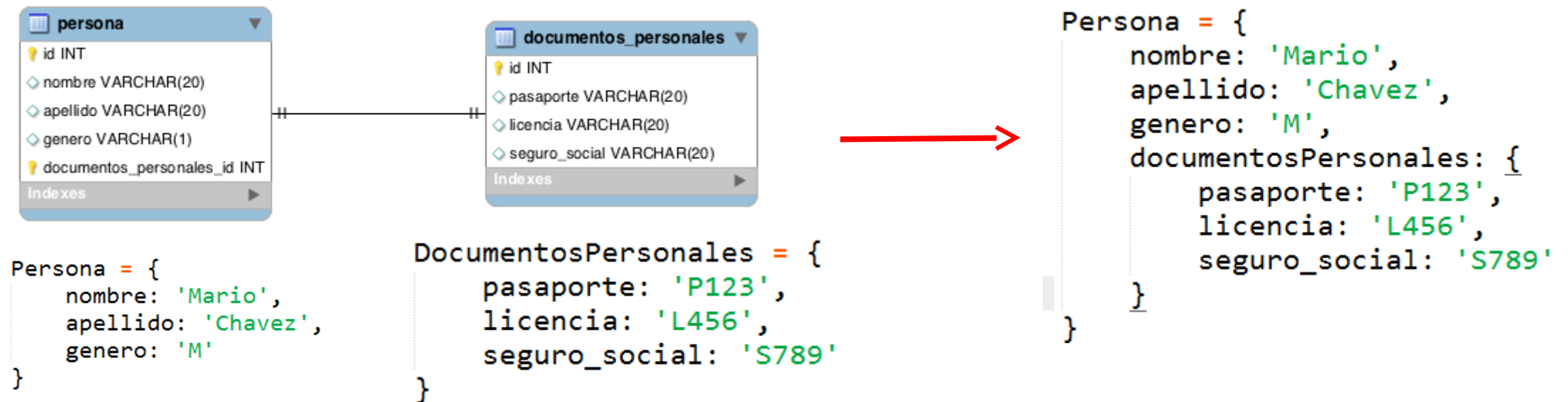
## Embeber

Consiste en incrustar un documento dentro de otro con el fin de que la relación sea directa

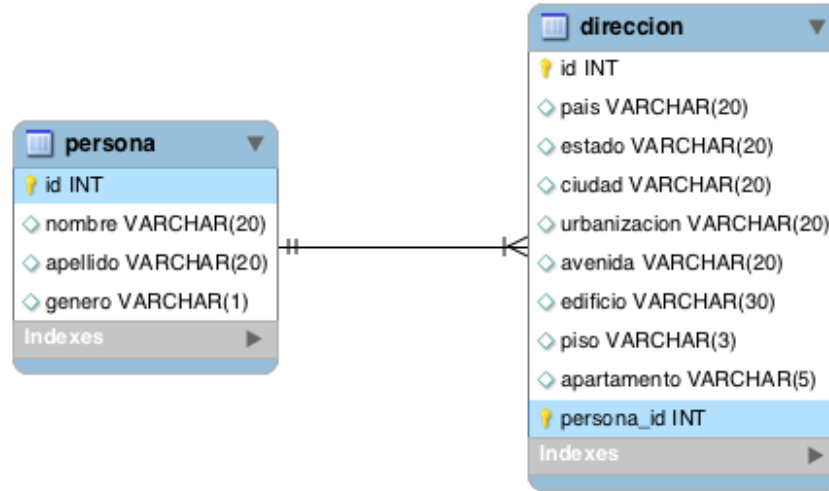
## Referenciar

Consiste en referenciar otro documento externo desde uno específico.

## Relaciones 1-1



# Relaciones 1-N



```
Persona = {  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M'  
}
```

```
Direccion1 = {  
  id: 1,  
  pais: 'Bolivia',  
  estado: '',  
  ciudad: 'La Paz',  
  urbanizacion: 'Miraflores',  
  avenida: ..,  
  edificio: ..,  
  piso: ..,  
  apartamento: ..  
}
```

```
Direccion2 = {  
  id: 2,  
  pais: 'Argentina',  
  estado: 'Buenos Aires',  
  ciudad: ...,  
  urbanizacion: ...,  
  avenida: ...,  
  edificio: ...,  
  piso: ...,  
  apartamento: ...  
}
```



```
Persona = {  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M',  
  direcciones: [1, 2]  
}
```

```
Persona = {  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M',  
  direcciones: [{  
    id: 1,  
    pais: 'Bolivia',  
    estado: ''  
    ...  
  }, {  
    id: 2,  
    pais: 'Argentina',  
    estado: 'Buenos Aires',  
    ...  
  }]  
}
```

# Relaciones N-N



```
id : 1
nombre: 'Elizabeth',
apellido: 'Blanco',
genero: 'F'
}
```

```
Persona2 = {
  id      : 2,
  nombre: 'Mario',
  apellido: 'Chavez',
  genero: 'M'
}
```

```
Direccion1 = {
  id: 1,
  pais: 'Bolivia',
  estado: '',
  ciudad: 'La Paz',
  urbanizacion: 'Miraflores',
  avenida: ..,
  edificio: ..,
  piso: ..,
  apartamento: ..
}
```

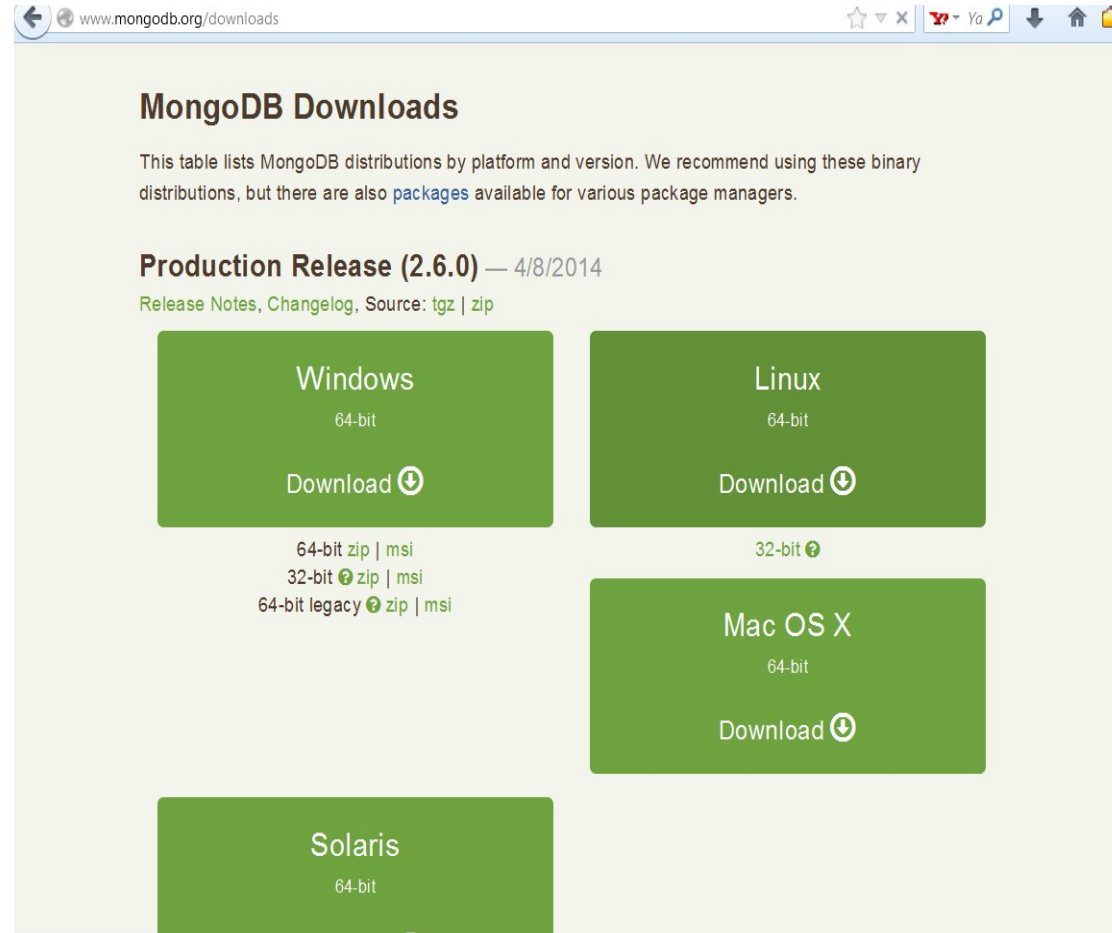
```
Direccion2 = {
  id: 2,
  pais: 'Argentina',
  estado: 'Buenos Aires',
  ciudad: ...,
  urbanizacion: ...,
  avenida: ...,
  edificio: ...,
  piso: ...,
  apartamento: ...
}
```



```
Persona1 = {
  id: 1,
  nombre: 'Elizabeth',
  apellido: 'Blanco',
  genero: 'F'
  direcciones : [{
    direccion_id : 1,
    viveAqui : true
  }, {
    direccion_id : 2,
    viveAqui : false
  }]
}
```

```
Persona2 = {
  id: 2,
  nombre: 'Mario',
  apellido: 'Chavez',
  genero: 'M'
  direcciones : [{
    direccion_id : 2,
    viveAqui : true
  }]
}
```

# MongoDB en acción



The screenshot shows the MongoDB Downloads page in a web browser. The browser's address bar displays 'www.mongodb.org/downloads'. The page has a light beige background. At the top, the title 'MongoDB Downloads' is followed by a paragraph: 'This table lists MongoDB distributions by platform and version. We recommend using these binary distributions, but there are also [packages](#) available for various package managers.' Below this, the 'Production Release (2.6.0) — 4/8/2014' is announced, with links for 'Release Notes', 'Changelog', and 'Source: tgz | zip'. The main content area features four green rectangular buttons for different operating systems: 'Windows 64-bit', 'Linux 64-bit', 'Mac OS X 64-bit', and 'Solaris 64-bit'. Each button has a 'Download' link with a download icon. Below the Windows button, additional links are provided: '64-bit zip | msi', '32-bit zip | msi', and '64-bit legacy zip | msi'. Below the Linux button, a '32-bit' link is visible.

**MongoDB Downloads**

This table lists MongoDB distributions by platform and version. We recommend using these binary distributions, but there are also [packages](#) available for various package managers.

**Production Release (2.6.0) — 4/8/2014**  
[Release Notes](#), [Changelog](#), Source: [tgz](#) | [zip](#)

**Windows**  
64-bit  
Download

64-bit zip | msi  
32-bit zip | msi  
64-bit legacy zip | msi

**Linux**  
64-bit  
Download

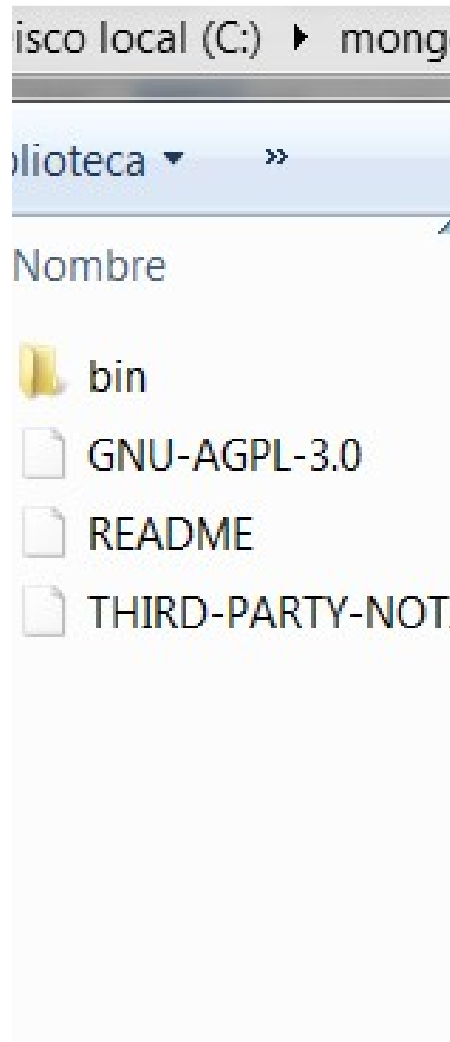
32-bit

**Mac OS X**  
64-bit  
Download

**Solaris**  
64-bit

<http://www.mongodb.org/downloads>

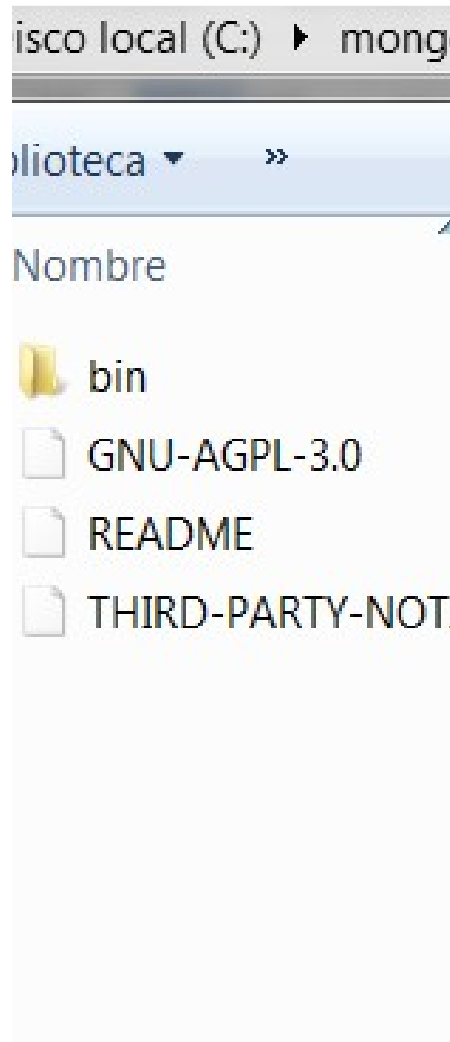
# Extraer el comprimido y navegar a la carpeta *bin*



Nombre	Fecha de modifica...	Tipo	Tamaño
bsondump.exe	28/08/2012 15:56	Aplicación	6.533 KB
mongo.exe	28/08/2012 16:05	Aplicación	3.357 KB
mongod.exe	28/08/2012 14:55	Aplicación	6.583 KB
mongod.pdb	28/08/2012 14:55	Archivo PDB	66.427 KB
mongodump.exe	28/08/2012 15:01	Aplicación	6.563 KB
mongoexport.exe	28/08/2012 15:18	Aplicación	6.536 KB
mongofiles.exe	28/08/2012 15:50	Aplicación	6.548 KB
mongoimport.exe	28/08/2012 15:24	Aplicación	6.552 KB
mongooplog.exe	28/08/2012 15:43	Aplicación	6.534 KB
mongoperf.exe	28/08/2012 16:02	Aplicación	6.535 KB
mongorestore.exe	28/08/2012 15:11	Aplicación	6.551 KB
mongos.exe	28/08/2012 14:59	Aplicación	4.732 KB
mongos.pdb	28/08/2012 14:59	Archivo PDB	51.371 KB
mongostat.exe	28/08/2012 15:31	Aplicación	6.562 KB
mongotop.exe	28/08/2012 15:37	Aplicación	6.535 KB



# Extraer el comprimido y navegar a la carpeta *bin*



Nombre	Fecha de modifica...	Tipo	Tamaño
bsondump.exe	28/08/2012 15:56	Aplicación	6.533 KB
mongo.exe	28/08/2012 16:05	Aplicación	3.357 KB
mongod.exe	28/08/2012 14:55	Aplicación	6.583 KB
mongod.pdb	28/08/2012 14:55	Archivo PDB	66.427 KB
mongodump.exe	28/08/2012 15:01	Aplicación	6.563 KB
mongoexport.exe	28/08/2012 15:18	Aplicación	6.536 KB
mongofiles.exe	28/08/2012 15:50	Aplicación	6.548 KB
mongoimport.exe	28/08/2012 15:24	Aplicación	6.552 KB
mongooplog.exe	28/08/2012 15:43	Aplicación	6.534 KB
mongoperf.exe	28/08/2012 16:02	Aplicación	6.535 KB
mongorestore.exe	28/08/2012 15:11	Aplicación	6.551 KB
mongos.exe	28/08/2012 14:59	Aplicación	4.732 KB
mongos.pdb	28/08/2012 14:59	Archivo PDB	51.371 KB
mongostat.exe	28/08/2012 15:31	Aplicación	6.562 KB
mongotop.exe	28/08/2012 15:37	Aplicación	6.535 KB

- Crear carpeta: **C:\mongodb\data**
- Crear un archivo **C:\mongodb\bin\mongodb.config**
- Insertar en el anterior archivo **dbpath=C:\mongodb\data**

```
C:\mongodb\bin>mongodb --config c:\mongodb\bin\mongodb.config
Mon Apr 21 22:00:18
Mon Apr 21 22:00:18 warning: 32-bit servers don't have journaling enabled by
ault. Please use --journal if you want durability.
Mon Apr 21 22:00:18
Mon Apr 21 22:00:18 [initandlisten] MongoDB starting : pid=4808 port=27017 d
h=c:/mongodb/data 32-bit host=CronosPC
Mon Apr 21 22:00:18 [initandlisten]
Mon Apr 21 22:00:18 [initandlisten] ** NOTE: when using MongoDB 32 bit, you
limited to about 2 gigabytes of data
Mon Apr 21 22:00:18 [initandlisten] **      see http://blog.mongodb.org/post
7788967/32-bit-limitations
Mon Apr 21 22:00:18 [initandlisten] **      with --journal, the limit is 10
Mon Apr 21 22:00:18 [initandlisten]
Mon Apr 21 22:00:18 [initandlisten] db version v2.2.0, pdfile version 4.5
Mon Apr 21 22:00:18 [initandlisten] git version: f5e83eae9cfbec7fb7a07132192
d1b0c5207
Mon Apr 21 22:00:18 [initandlisten] build info: windows sys.getwindowsversio
jor=6, minor=0, build=6002, platform=2, service_pack='Service Pack 2') BOOST
_VERSION=1_49
Mon Apr 21 22:00:18 [initandlisten] options: { config: "c:\mongodb\bin\mongo
onfig", dbpath: "c:/mongodb/data" }
Mon Apr 21 22:00:18 [initandlisten] Unable to check for journal files due to
ost::filesystem::basic_directory_iterator constructor: El sistema no puede e
trar la ruta especificada: "c:\mongodb\data\journal"
Mon Apr 21 22:00:18 [initandlisten] waiting for connections on port 27017
Mon Apr 21 22:00:18 [websvr] admin web console waiting for connections on po
8017
```

# Lanzando el cliente y realizando consultas

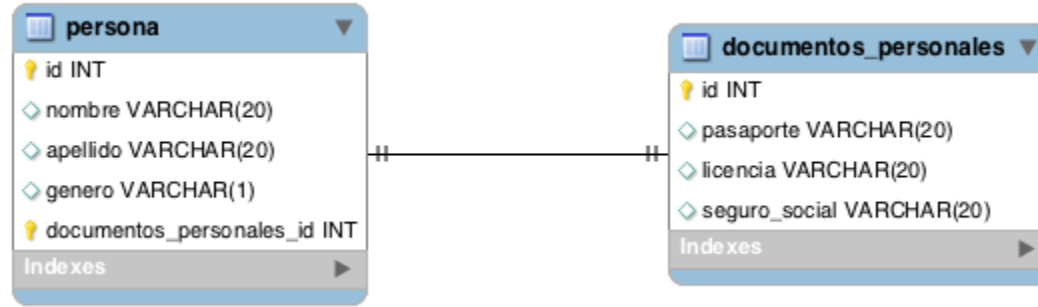
```
C:\mongodb\bin>mongo
MongoDB shell version: 2.2.0
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
```

```
switched to db universidad
> db.estudiante.insert({codigo:1,nombre:'mike',paterno:'stevenson',nropasaporte:70586});
> dv.estudiante.insert({codigo:2,nombre:'sandra',paterno:'monasterios',materno:'mamani'});
Tue Jun 17 16:08:51 ReferenceError: dv is not defined (shell):1
> db.estudiante.insert({codigo:2,nombre:'sandra',paterno:'monasterios',materno:'mamani'});
> db.estudiante.find();
{ "_id" : ObjectId("53a04bb382654e1bc1ffd56e"), "codigo" : 1, "nombre" : "mike", "paterno" : "stevenson", "nropasaporte" : 70586 }
{ "_id" : ObjectId("53a04c0f82654e1bc1ffd56f"), "codigo" : 2, "nombre" : "sandra", "paterno" : "monasterios", "materno" : "mamani" }
> db.estudiante.insert({codigo:3,nombre:'jose',estatura:178});
>
```

# Comandos básicos

Comandos	Significado
<code>use nombre_bd</code>	Crea una base de datos
<code>db.getCollectionNames()</code>	Muestra las colecciones de bd
<code>db.nombre_coleccion.insert({...})</code>	Inserta documentos a una colección
<code>db.nombre_coleccion.remove({...})</code>	
<code>db.nombre_coleccion.find({...})</code>	Elimina documentos de una colección
	Busca documentos en una colección

# Relaciones 1-1



```
Persona = {  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M'  
}
```

```
DocumentosPersonales = {  
  pasaporte: 'P123',  
  licencia: 'L456',  
  seguro_social: 'S789'  
}
```

↓

```
Persona = {  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M',  
  documentosPersonales: {  
    pasaporte: 'P123',  
    licencia: 'L456',  
    seguro_social: 'S789'  
  }  
}
```

```
db.persona.insert({  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M'  
})
```

```
db.persona.insert({  
  pasaporte: 'P123',  
  licencia: 'L456',  
  seguro_social: 'S789'  
})
```

```
db.persona.insert({  
  nombre: 'Mario',  
  apellido: 'Chavez',  
  genero: 'M',  
  documentosPersonales: {  
    pasaporte: 'P123',  
    licencia: 'L456',  
    seguro_social: 'S789'  
  }  
})
```

# Consultas Find

```
db.collection.find(query, projection)
```

## Selectores

Comandos	Significado
\$lt	Menor que
\$lte	Menor que o igual
\$gt	Mayor que
\$gte	Mayor que o igual
\$ne	Distinto
\$exists: false   true	Existe
\$or: [{}, {}]	ó

## Creacion de la base de datos

Creamos la base de datos **padron** con la coleccion **persona** y ejecutamos el script *persona.js*

```
mongo ruta_a_script.js
```

### Ejemplo 1

1. Mostrar aquellas personas de sexo masculino

```
db.persona.find({genero: 'M'});
```

2. Mostrar aquellas personas cuya edad es mayor a 25

```
db.persona.find( { edad: { $gt: 25 } } );
```

3. Mostrar a personas de genero femenino que tengan 22 o menos años

```
db.persona.find( { genero: 'F', edad: { $lte: 25 } } );
```

4. Mostrar a las personas hombres o mujeres que tengas mas de 25 años

```
db.persona.find( { $or: [{genero: 'F'}, {genero: 'M'}], edad: { $gt: 25 } } );
```

5. Mostrar las personas que tengan pasaporte y ademas sean varones

```
db.persona.find( {genero: 'M', pasaporte: { $exists: true } } );
```

## 6. Mostrar solo los nombres de los mujeres

```
db.persona.find( {genero: 'F'}, { nombre: 1} );
```

### Iteradores y modificadores

Comandos	Significado
cursor.forEach(fn)	Itera el cursor
cursor.order({})	Ordena la salida del cursor
cursor.limit(numero)	Limita la cantidad de salida del cursor

## Ejemplo 2

### 1. Mostrar las direcciones en latinoamerica de todos los varones



```
1  var conexion = new Mongo("localhost");
2
3  //base de datos a usar
4  var db = conexion.getDB("padron");
5
6  db.persona.find({genero: 'M', direcciones: {$exists: true}}).forEach(function(p) {
7      print("----" + p.nombre + "----");
8      var direccionesLatinoamerica= p.direcciones.filter(function(dir, ind) {
9          switch(dir.pais){
10             case 'Bolivia':
11             case 'Argentina':
12             case 'Peru':
13             case 'Chile':
14                 return true;
15             default:
16                 return false;
17         }
18     })
19
20     direccionesLatinoamerica.forEach(function(dirLat) {
21         print(dirLat.pais);
22     })
23 })
```