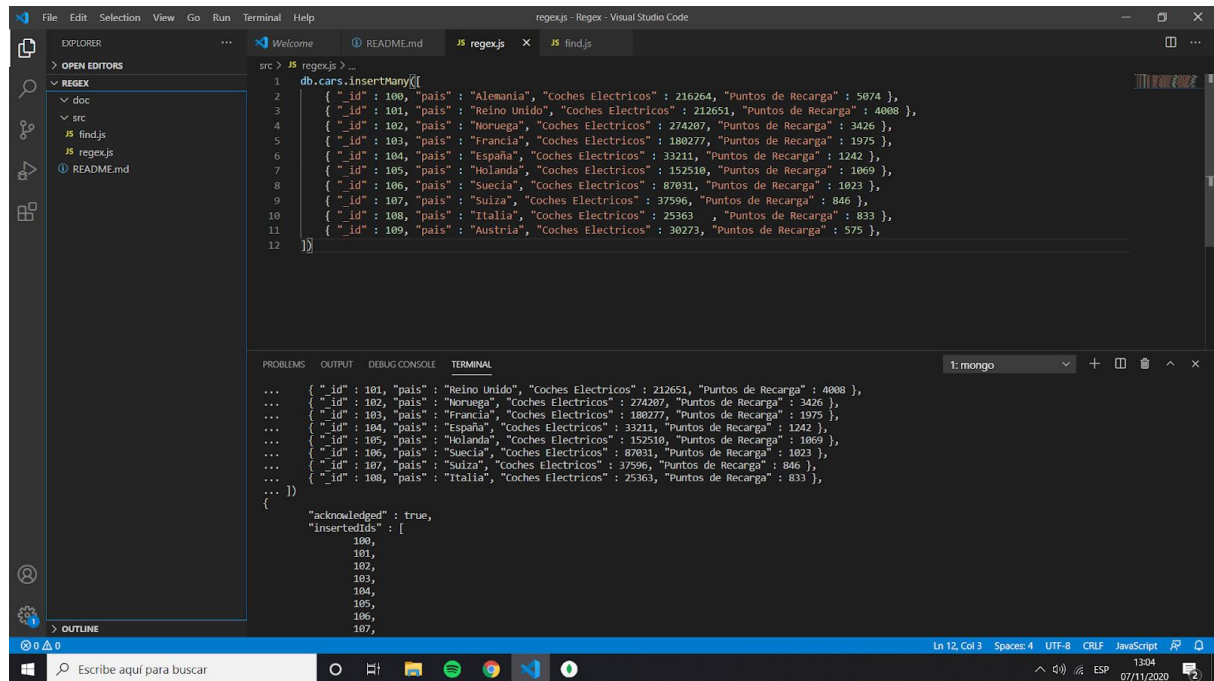


Documentación del proyecto con \$regex



```
src > JS regexjs > ...
1 db.cars.insertMany([
2   { "_id" : 100, "pais" : "Alemania", "Coches Electricos" : 216264, "Puntos de Recarga" : 5074 },
3   { "_id" : 101, "pais" : "Reino Unido", "Coches Electricos" : 212651, "Puntos de Recarga" : 4008 },
4   { "_id" : 102, "pais" : "Noruega", "Coches Electricos" : 274207, "Puntos de Recarga" : 3426 },
5   { "_id" : 103, "pais" : "Francia", "Coches Electricos" : 180277, "Puntos de Recarga" : 1975 },
6   { "_id" : 104, "pais" : "España", "Coches Electricos" : 33211, "Puntos de Recarga" : 1242 },
7   { "_id" : 105, "pais" : "Holanda", "Coches Electricos" : 152510, "Puntos de Recarga" : 1069 },
8   { "_id" : 106, "pais" : "Suecia", "Coches Electricos" : 87031, "Puntos de Recarga" : 1023 },
9   { "_id" : 107, "pais" : "Suiza", "Coches Electricos" : 37596, "Puntos de Recarga" : 846 },
10  { "_id" : 108, "pais" : "Italia", "Coches Electricos" : 25363, "Puntos de Recarga" : 833 },
11  { "_id" : 109, "pais" : "Austria", "Coches Electricos" : 30273, "Puntos de Recarga" : 575 },
12 ])
```

```
... { "_id" : 101, "pais" : "Reino Unido", "Coches Electricos" : 212651, "Puntos de Recarga" : 4008 },
... { "_id" : 102, "pais" : "Noruega", "Coches Electricos" : 274207, "Puntos de Recarga" : 3426 },
... { "_id" : 103, "pais" : "Francia", "Coches Electricos" : 180277, "Puntos de Recarga" : 1975 },
... { "_id" : 104, "pais" : "España", "Coches Electricos" : 33211, "Puntos de Recarga" : 1242 },
... { "_id" : 105, "pais" : "Holanda", "Coches Electricos" : 152510, "Puntos de Recarga" : 1069 },
... { "_id" : 106, "pais" : "Suecia", "Coches Electricos" : 87031, "Puntos de Recarga" : 1023 },
... { "_id" : 107, "pais" : "Suiza", "Coches Electricos" : 37596, "Puntos de Recarga" : 846 },
... { "_id" : 108, "pais" : "Italia", "Coches Electricos" : 25363, "Puntos de Recarga" : 833 },
... ]}
{
  "acknowledged" : true,
  "insertedIds" : [
    100,
    101,
    102,
    103,
    104,
    105,
    106,
    107,
    108,
    109
  ]
}
```

Con estos datos trabajaremos.

	DOCUMENTS	10	TOTAL SIZE	901B	AVG. SIZE	90B	INDEXES
Documents	Aggregations	Schema	Explain Plan	Indexes	Validation		
FILTER OPTIONS Fi							
ADD DATA VIEW {}							
Displaying documents 1 - 10 of 10							
	<pre>_id: 100 pais: "Alemania" Coches Electricos: 216264 Puntos de Recarga: 5074</pre>						
	<pre>_id: 101 pais: "Reino Unido" Coches Electricos: 212651 Puntos de Recarga: 4008</pre>						
	<pre>_id: 102 pais: "Noruega" Coches Electricos: 274207 Puntos de Recarga: 3426</pre>						
	<pre>> _id: 103 pais: "Francia" Coches Electricos: 180277 Puntos de Recarga: 1975</pre>						
	<pre>_id: 104 pais: "España" Coches Electricos: 33211 Puntos de Recarga: 1242</pre>						

Aquí se ve como aparece en el mongo db Compass.

El primer find que haremos será este:

```
💡 Para todos los que el número de coches electricos termine en 4
db.cars.find( { CochesElectricos: { $regex: /4$/ } } )
```

Y tal como dice el enunciado:

```
> db.cars.find( { CochesElectricos: { $regex: /4$/ } } )
{ "_id" : 100, "pais" : "Alemania", "CochesElectricos" : "216264", "Puntos de Recarga" : 5074 }
>
```

Nos sale el único que acaba en 4 en Coches Electricos.

```
//Para todos los paises que empiecen con A(mayúscula)
db.cars.find( {pais: {$regex: /^A/}})
```

```
> db.cars.find( {pais: {$regex: /^A/}})
{ "_id" : 100, "pais" : "Alemania", "CochesElectricos" : "216264", "Puntos de Recarga" : 5074 }
{ "_id" : 109, "pais" : "Austria", "CochesElectricos" : "30273", "PuntosdeRecarga" : 575 }
>
```

Y aquí vemos que nos salen los dos que empiezan con A mayúscula, si queremos que no distinga entre mayúscula y minúscula, haremos esto:

```
💡 Para todos los paises que empiecen con A, may o min.
db.cars.find( {pais: {$regex: /^A/i}})
```

```
}
> db.cars.find( {pais: {$regex: /^A/i}})
{ "_id" : 100, "pais" : "Alemania", "CochesElectricos" : "216264", "Puntos de Recarga" : 5074 }
{ "_id" : 109, "pais" : "Austria", "CochesElectricos" : "30273", "PuntosdeRecarga" : 575 }
{ "_id" : 110, "pais" : "angola", "CochesElectricos" : "25", "PuntosdeRecarga" : 2 }
>
```

He añadido un nuevo campo con la a en minúscula para que tenga sentido el find.

Ahora voy a utilizar \$not para hacer el contrario al anterior.

```
//Para todos los paises que NO empiecen con A, may o min.
db.cars.find( { pais: { $not: /^A/i } } )
```

```
> db.cars.find( { pais: { $not: /^A/i } } )
{ "_id" : 101, "pais" : "Reino Unido", "CochesElectricos" : "212651", "Puntos de Recarga" : 4008 }
{ "_id" : 102, "pais" : "Noruega", "CochesElectricos" : "274207", "Puntos de Recarga" : 3426 }
{ "_id" : 103, "pais" : "Francia", "CochesElectricos" : "180277", "Puntos de Recarga" : 1975 }
{ "_id" : 104, "pais" : "España", "CochesElectricos" : "33211", "PuntosdeRecarga" : 1242 }
{ "_id" : 105, "pais" : "Holanda", "CochesElectricos" : "152510", "PuntosdeRecarga" : 1069 }
{ "_id" : 106, "pais" : "Suecia", "CochesElectricos" : "87031", "PuntosdeRecarga" : 1023 }
{ "_id" : 107, "pais" : "Suiza", "CochesElectricos" : "37596", "PuntosdeRecarga" : 846 }
{ "_id" : 108, "pais" : "Italia", "CochesElectricos" : "25363", "PuntosdeRecarga" : 833 }
>
```

Ahora vamos a usar el \$eq

```
//Para que el pais sea España. Exactamente escrito así.
db.cars.find( { pais: { $eq: "España" } } )
```

```
> db.cars.find( { pais: { $eq: "España" } } )
{ "_id" : 104, "pais" : "España", "CochesElectricos" : "33211", "PuntosdeRecarga" : 1242 }
>
>
```

Y es igual a España.

```
db.cars2.insertMany([
  { item: "mercedes", qty: 50, size: { h: 200, w: 200, uom: "cm" } },
  { item: "Opel", qty: 100, size: { h: 180, w: 230, uom: "cm" } },
  { item: "BMW", qty: 75, size: { h: 200, w: 233, uom: "cm" } },
  { item: "Nissan", qty: 45, size: { h: 175, w: 300, uom: "cm" } },
])
```

Voy a añadir más datos a otra base para seguir probando demas comandos.

💡 Para encontrar los coches con más de 51 unidades.

```
db.cars2.find({ qty: { $gt: 51 } })
```

```
> db.cars2.find({ qty: { $gt: 51 } })
{ "_id" : ObjectId("5fa82579f1470cb3902f0910"), "item" : "Opel", "qty" : 100, "size" : { "h" : 180, "w" : 230, "uom" : "cm" } }
{ "_id" : ObjectId("5fa82579f1470cb3902f0911"), "item" : "BMW", "qty" : 75, "size" : { "h" : 200, "w" : 233, "uom" : "cm" } }
> []
```

Aqui vemos el operador gt que se usa para encontrar lo mayor a lo especificado.

Ahora usamos el gte, que se usa para encontrar lo mayor o igual a lo especificado.

//Para encontrar los coches con más o igual a 50 unidades.

```
db.cars2.find( { qty: { $gte: 50 } } )
```

```
> db.cars2.find( { qty: { $gte: 50 } } )
{ "_id" : ObjectId("5fa82579f1470cb3902f090f"), "item" : "mercedes", "qty" : 50, "size" : { "h" : 200, "w" : 200, "uom" : "cm" } }
{ "_id" : ObjectId("5fa82579f1470cb3902f0910"), "item" : "Opel", "qty" : 100, "size" : { "h" : 180, "w" : 230, "uom" : "cm" } }
{ "_id" : ObjectId("5fa82579f1470cb3902f0911"), "item" : "BMW", "qty" : 75, "size" : { "h" : 200, "w" : 233, "uom" : "cm" } }
> []
```

\$in

//Para encontrar los coches que tienen 45 y 75 unidades.

```
db.cars2.find( { qty: { $in: [ 45, 75 ] } } )
```

```
> db.cars2.find( { qty: { $in: [ 45, 75 ] } } )
{ "_id" : ObjectId("5fa82579f1470cb3902f0911"), "item" : "BMW", "qty" : 75, "size" : { "h" : 200, "w" : 233, "uom" : "cm" } }
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }
> []
```


\$lt , Menor que.

```
//Para encontrar los coches con menos unidades que 75.  
db.cars2.find( { qty: { $lt: 75 } } )
```

```
> db.cars2.find( { qty: { $lt: 75 } } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f090f"), "item" : "mercedes", "qty" : 50, "size" : { "h" : 200, "w" : 200, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }  
>
```

\$lte, Menor o igual que.

```
//Para encontrar los coches con menos o iguales unidades que 75.  
db.cars2.find( { qty: { $lte: 75 } } )
```

```
> db.cars2.find( { qty: { $lte: 75 } } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0911"), "item" : "BMW", "qty" : 75, "size" : { "h" : 200, "w" : 233, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }  
>
```

\$ne, Todos los que no sean igual al valor especificado.

```
💡Para encontrar los coches que no tengan 75 unidades.  
db.cars2.find( { qty: { $ne: 75 } } )
```

```
TERMINAL  
cars2.find( { qty: { $ne: 75 } } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f090f"), "item" : "mercedes", "qty" : 50, "size" : { "h" : 200, "w" : 200, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0910"), "item" : "Opel", "qty" : 100, "size" : { "h" : 180, "w" : 230, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }
```

\$nin

```
37  
38 //Para encontrar los coches que no tengan 75 ni 45 unidades.  
39 db.cars2.find( { qty: { $nin: [ 45, 75 ] } } )  
40  
41
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
> db.cars2.find( { qty: { $nin: [ 45, 75 ] } } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f090f"), "item" : "mercedes", "qty" : 50, "size" : { "h" : 200, "w" : 200, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0910"), "item" : "Opel", "qty" : 100, "size" : { "h" : 180, "w" : 230, "uom" : "cm" } }  
>
```

\$and , concatenar .

```
//Para encontrar los que tengan más o igual a 50 unidades y el campo qty exista.  
db.cars2.find( { $and: [ { qty: { $gte: 50 } }, { qty: { $exists: true } } ] } )
```

```
> db.cars2.find( { $and: [ { qty: { $gte: 50 } }, { qty: { $exists: true } } ] } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f090f"), "item" : "mercedes", "qty" : 50, "size" : { "h" : 200, "w" : 200, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0910"), "item" : "Opel", "qty" : 100, "size" : { "h" : 180, "w" : 230, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0911"), "item" : "BMW", "qty" : 75, "size" : { "h" : 200, "w" : 233, "uom" : "cm" } }  
> []
```

\$nor , este es el and para el not.

```
//Para encontrar los que NO tengan más o igual a 50 unidades y el campo qty exista.  
db.cars2.find( { $nor: [ { qty: { $gte: 50 } }, { qty: { $exists: false } } ] } )
```

```
> db.cars2.find( { $nor: [ { qty: { $gte: 50 } }, { qty: { $exists: false } } ] } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }  
> []
```

\$or, “o” , si se cumple una saldrá.

```
//Para encontrar los coches que tengan menos de 75 unidades y los que su tamaño sea el especificado(exactamente)  
db.cars2.find( { $or: [ { qty: { $lt: 75 } }, { size: { h: 180, w: 230, uom: "cm" } } ] } )
```

```
> db.cars2.find( { $or: [ { qty: { $lt: 75 } }, { size: { h: 180, w: 230, uom: "cm" } } ] } )  
{ "_id" : ObjectId("5fa82579f1470cb3902f090f"), "item" : "mercedes", "qty" : 50, "size" : { "h" : 200, "w" : 200, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0910"), "item" : "Opel", "qty" : 100, "size" : { "h" : 180, "w" : 230, "uom" : "cm" } }  
{ "_id" : ObjectId("5fa82579f1470cb3902f0912"), "item" : "Nissan", "qty" : 45, "size" : { "h" : 175, "w" : 300, "uom" : "cm" } }  
> []
```