

 READ

Operadores Lógicos

# Operadores lógicos

Los operadores lógicos son muy útiles y los podemos usar en diferentes tipos de consultas

AND



Inputs		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

OR



Inputs		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

NOT



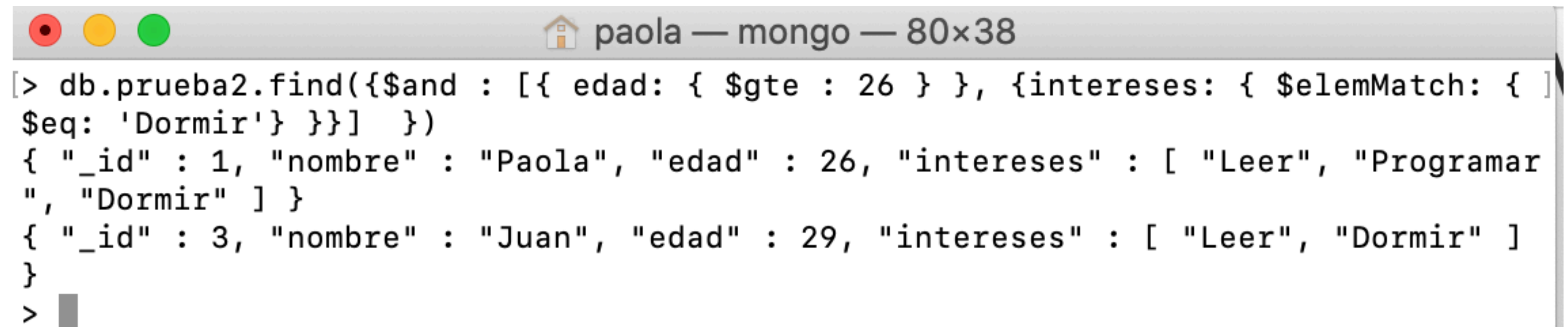
Input	Output
A	C
0	1
1	0

## Operadores lógico \$AND

Cuando necesitamos construir consultas que deban cumplir varias condiciones utilizaremos los operadores lógicos.

El operador \$and lo hemos utilizando en forma implícita, por ejemplo:

```
db.prueba2.find({$and : [{ edad: { $gte : 26 } }, {intereses: { $elemMatch: { $eq: 'Dormir'} } }] })
```



The screenshot shows a terminal window titled "paola — mongo — 80x38". The command entered is `db.prueba2.find({$and : [{ edad: { $gte : 26 } }, {intereses: { $elemMatch: { $eq: 'Dormir'} } }] })`. The output displays two documents: one for Paola (age 26, interests: Leer, Programar, Dormir) and one for Juan (age 29, interests: Leer, Dormir).

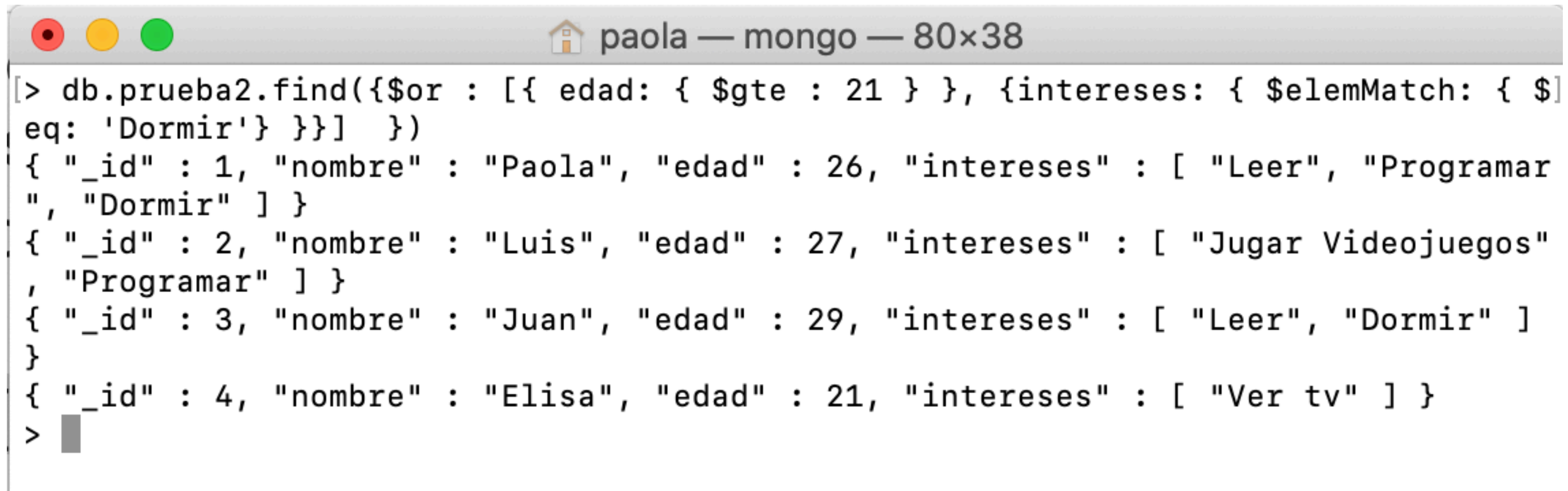
```
[> db.prueba2.find({$and : [{ edad: { $gte : 26 } }, {intereses: { $elemMatch: { $eq: 'Dormir'} } }] })
{ "_id" : 1, "nombre" : "Paola", "edad" : 26, "intereses" : [ "Leer", "Programar", "Dormir" ] }
{ "_id" : 3, "nombre" : "Juan", "edad" : 29, "intereses" : [ "Leer", "Dormir" ] }
>
```

## — Operadores lógico \$OR

Cuando necesitamos construir consultas que deban cumplir con UNA o VARIAS condiciones utilizaremos los operadores lógicos.

El operador \$or lo hemos utilizando en forma implícita, por ejemplo:

```
db.prueba2.find({$or : [{ edad: { $gte : 21 } }, {intereses: { $elemMatch: { $eq: 'Dormir'} } }] })
```

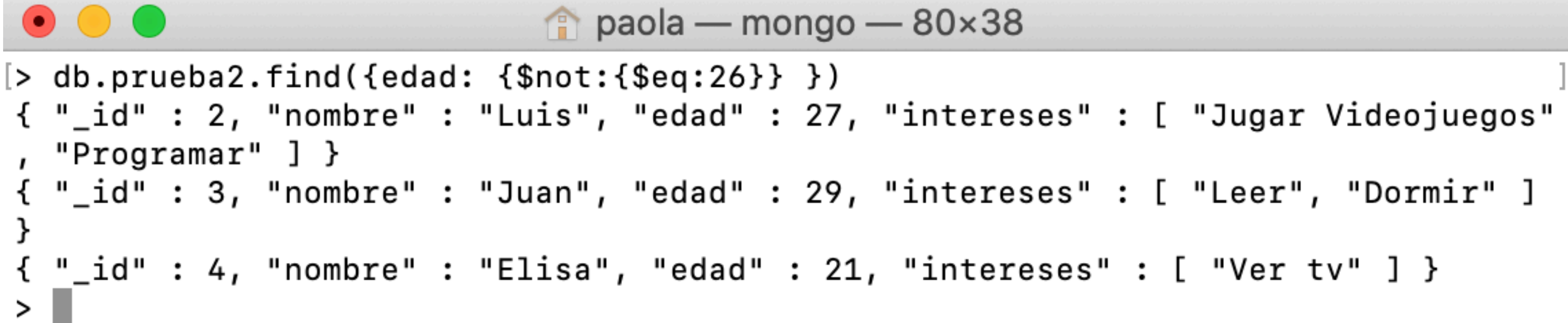


```
paola — mongo — 80x38
[> db.prueba2.find({$or : [{ edad: { $gte : 21 } }, {intereses: { $elemMatch: { $eq: 'Dormir'} } }] })
{ "_id" : 1, "nombre" : "Paola", "edad" : 26, "intereses" : [ "Leer", "Programar", "Dormir" ] }
{ "_id" : 2, "nombre" : "Luis", "edad" : 27, "intereses" : [ "Jugar Videojuegos", "Programar" ] }
{ "_id" : 3, "nombre" : "Juan", "edad" : 29, "intereses" : [ "Leer", "Dormir" ] }
{ "_id" : 4, "nombre" : "Elisa", "edad" : 21, "intereses" : [ "Ver tv" ] }
>
```

## — Operadores lógico \$NOT

Cuando necesitamos construir consultas que deban cumplir varias condiciones utilizaremos los operadores lógicos.

```
db.prueba2.find({edad: {$not:{$eq:26}} })
```



The screenshot shows a terminal window titled "paola — mongo — 80x38". The command entered is `[> db.prueba2.find({edad: {$not:{$eq:26}} })]`. The output displays three documents from the `prueba2` collection, all of which have an age greater than 26. The first document is for Luis (age 27, interests: "Jugar Videojuegos", "Programar"). The second is for Juan (age 29, interests: "Leer", "Dormir"). The third is for Elisa (age 21, interest: "Ver tv").

```
[> db.prueba2.find({edad: {$not:{$eq:26}} })]  
{ "_id" : 2, "nombre" : "Luis", "edad" : 27, "intereses" : [ "Jugar Videojuegos"  
, "Programar" ] }  
{ "_id" : 3, "nombre" : "Juan", "edad" : 29, "intereses" : [ "Leer", "Dormir" ]  
}  
{ "_id" : 4, "nombre" : "Elisa", "edad" : 21, "intereses" : [ "Ver tv" ] }  
>
```

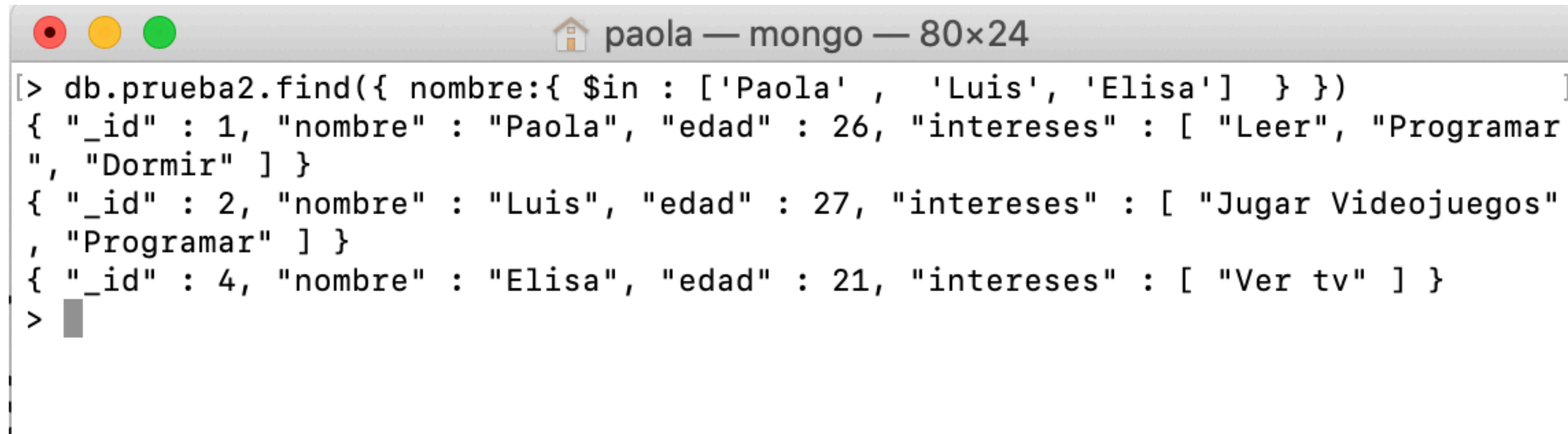
 READ

Otros Operadores

## — Operadores relacionales \$in

Para mostrar todos los documento dentro de una LISTA de valores, tenemos que utilizar el operador \$in:

```
db.prueba2.find({ nombre: { $in : ['Paola', 'Luis', 'Elisa']} })
```



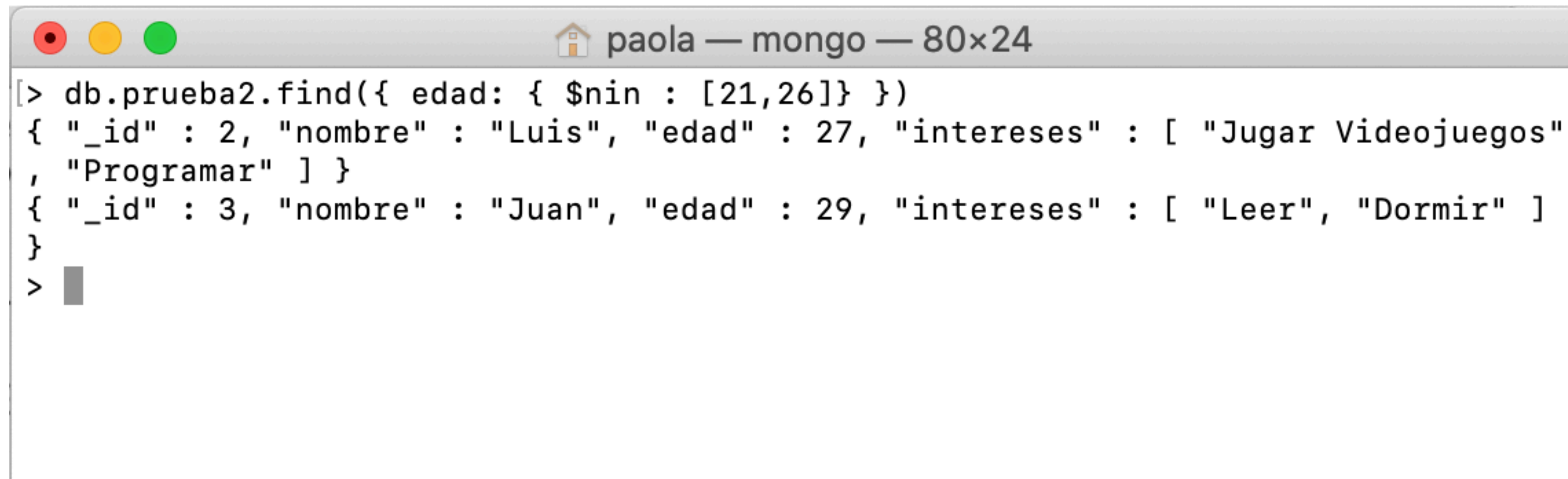
```
paola — mongo — 80x24
[> db.prueba2.find({ nombre:{ $in : ['Paola' , 'Luis', 'Elisa'] } }) ]
{ "_id" : 1, "nombre" : "Paola", "edad" : 26, "intereses" : [ "Leer", "Programar", "Dormir" ] }
{ "_id" : 2, "nombre" : "Luis", "edad" : 27, "intereses" : [ "Jugar Videojuegos", "Programar" ] }
{ "_id" : 4, "nombre" : "Elisa", "edad" : 21, "intereses" : [ "Ver tv" ] }
>
```



## — Operadores relacionales \$nin

Para mostrar todos los documento que NO están dentro de una LISTA de valores, tenemos que utilizar el operador \$nin:

```
db.prueba2.find({ edad: { $nin : [21,26]} })
```



```
paola — mongo — 80x24
[> db.prueba2.find({ edad: { $nin : [21,26]} })
{ "_id" : 2, "nombre" : "Luis", "edad" : 27, "intereses" : [ "Jugar Videojuegos"
, "Programar" ] }
{ "_id" : 3, "nombre" : "Juan", "edad" : 29, "intereses" : [ "Leer", "Dormir" ]
}
>
```



## — Operadores relacionales por patrón

Para mostrar todos los documentos que sigan cierto patrón se puede hacer de la siguiente manera :

`db.users.find({name: /a/})` —> Like '%a%'

`db.users.find({name: /^pa/})` —> Like 'pa%'

`db.users.find({name: /ro$/})` —> Like '%ro'

## Ejercicio 3. Con la BD Northwind

1. Mostrar los productos de las categorías 1, 3 y 7
2. Mostrar las ventas cuyo id sea el 10248, 10250 y 10253.
3. Mostrar a los empleados que NO provengan de la ciudad de Seattle, Tacoma y Kirkland
4. Mostrar a los empleados que NO sean representantes de ventas
5. Mostrar los productos que incluyan el texto Chef Anton's