

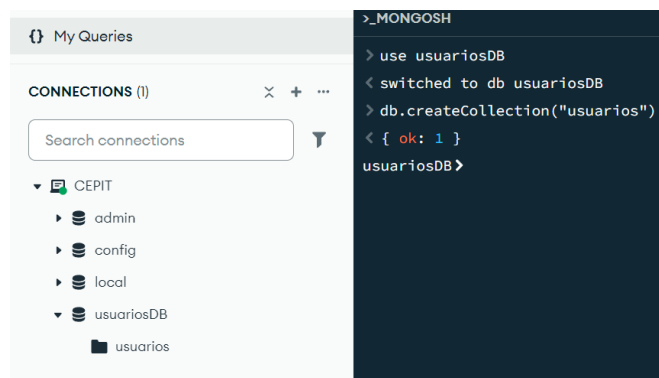
Practica mongoDB

1. crear una db de usuarios.

Preparo la base de datos usuariosDB, y luego creo la colección usuarios dentro de dicha base de datos.

```
> use usuariosDB
< switched to db usuariosDB
> db.createCollection("usuarios")
< { ok: 1 }
usuariosDB>
```

Recargo el entorno, y aparece:



2. crear 5 usuarios con 5 campos a eleccion (que no sean e-mail).

Me posiciono en la base de datos usuariosDB, y agrego 5 usuarios con sus respectivos campos a la colección usuarios (insertMany).

```
> db.usuarios.insertMany([
  { nombre: "Juan", apellido: "Pérez", edad: 30, dni: "12345678", ciudad: "Tandil" },
  { nombre: "Ana", apellido: "López", edad: 25, dni: "23456789", ciudad: "Mar del Plata" },
  { nombre: "Carlos", apellido: "Gómez", edad: 40, dni: "34567890", ciudad: "Bahía Blanca" },
  { nombre: "Lucía", apellido: "Martínez", edad: 35, dni: "45678901", ciudad: "Olavarría" },
  { nombre: "Miguel", apellido: "Ramírez", edad: 28, dni: "56789012", ciudad: "La Plata" }
])
```

Si voy a dicha colección, veo algunos nuevos datos:



3. Agregar un nuevo campo e-mail a 2 de los 5 registros.

Tomo 2 de los usuarios creados anteriormente, y les agrego un correo (updateOne para 2 campos).

```
> db.usuarios.updateOne({ nombre: "Juan" }, { $set: { email: "juan@correo.com" } })
db.usuarios.updateOne({ nombre: "Ana" }, { $set: { email: "ana@correo.com" } })
```

Ahora, si voy a dichos registros (luego de actualizar), veo lo siguiente:

```
_id: ObjectId('68220216a98aee888e817fdd')
nombre : "Juan"
apellido : "Pérez"
edad : 30
dni : "12345678"
ciudad : "Tandil"
email : "juan@correo.com"
```

```
_id: ObjectId('68220216a98aee888e817fde')
nombre : "Ana"
apellido : "López"
edad : 25
dni : "23456789"
ciudad : "Mar del Plata"
email : "ana@correo.com"
```

4. Actualizar el e-mail a uno de esos 2 registros para que no queden iguales.

No habían quedado iguales, pero para el ejercicio, tome uno de los usuarios anteriormente mencionados, y les modifique el correo que antes les cree.

```
> db.usuarios.updateOne({ nombre: "Ana" }, { $set: { email: "ana123@correo.com" } })
```

Si ahora reviso dicho registro, veo lo siguiente:

```
_id: ObjectId('68220216a98aee888e817fde')
nombre : "Ana"
apellido : "López"
edad : 25
dni : "23456789"
ciudad : "Mar del Plata"
email : "ana123@correo.com"
```

5. Eliminar 2 registros

Para eliminar 2 registros, use deleteOne (para cada uno)

```
> db.usuarios.deleteOne({ nombre: "Carlos" })
db.usuarios.deleteOne({ nombre: "Lucía" })
```

Por lo que ahora, de los 5 campos iniciales, quedan 3.

```
_id: ObjectId('68220216a98aee888e817fdd')
nombre : "Juan"
apellido : "Pérez"
edad : 30
dni : "12345678"
ciudad : "Tandil"
email : "juan@correo.com"
```

```
_id: ObjectId('68220216a98aee888e817fde')
nombre : "Ana"
apellido : "López"
edad : 25
dni : "23456789"
ciudad : "Mar del Plata"
email : "ana123@correo.com"
```

```
_id: ObjectId('68220216a98aee888e817fe1')
nombre : "Miguel"
apellido : "Ramírez"
edad : 28
dni : "56789012"
ciudad : "La Plata"
```

6. Agregar el campo EstadoUsuario a todos los registros y setearlo en "Activo"

Agrego a todos los campos (updateMany), el campo EstadoUsuario, y lo dejo seteado en Activo.

```
> db.usuarios.updateMany({}, { $set: { EstadoUsuario: "Activo" } })
```

Si veo los registros:

```
_id: ObjectId('68220216a98aee888e817fdd')
nombre : "Juan"
apellido : "Pérez"
edad : 30
dni : "12345678"
ciudad : "Tandil"
email : "juan@correo.com"
EstadoUsuario : "Activo"
```

```
_id: ObjectId('68220216a98aee888e817fde')
nombre : "Ana"
apellido : "López"
edad : 25
dni : "23456789"
ciudad : "Mar del Plata"
email : "ana123@correo.com"
EstadoUsuario : "Activo"
```

```
_id: ObjectId('68220216a98aee888e817fe1')
nombre : "Miguel"
apellido : "Ramírez"
edad : 28
dni : "56789012"
ciudad : "La Plata"
EstadoUsuario : "Activo"
```

7. Buscar los usuarios con e-mail.

Si veo la imagen del ejercicio anterior, Ana y Juan tienen un correo asignado (es lo que debería devolver la consulta).

Para ello, voy a utilizar find, con la condición de que exista el campo email.

```

> db.usuarios.find({ email: { $exists: true } })
< {
  _id: ObjectId('68220216a98aee888e817fdd'),
  nombre: 'Juan',
  apellido: 'Pérez',
  edad: 30,
  dni: '12345678',
  ciudad: 'Tandil',
  email: 'juan@correo.com',
  EstadoUsuario: 'Activo'
}
{
  _id: ObjectId('68220216a98aee888e817fde'),
  nombre: 'Ana',
  apellido: 'López',
  edad: 25,
  dni: '23456789',
  ciudad: 'Mar del Plata',
  email: 'ana123@correo.com',
  EstadoUsuario: 'Activo'
}

```

8. Hacer algun cambio que se te ocurra a la db.

Voy a tomar un usuario (Miguel) de la base de datos, y voy a modificar su nombre.

```

> db.usuarios.updateOne({ nombre: "Miguel" }, { $set: { nombre: "Miguel Ángel" } })

```

Si lo veo en los registros:

```

_id: ObjectId('68220216a98aee888e817fe1')
nombre : "Miguel Ángel"
apellido : "Ramírez"
edad : 28
dni : "56789012"
ciudad : "La Plata"
EstadoUsuario : "Activo"

```