

Taller de **CRUD**: **DELETE** en MongoDB

Requisitos previos

Antes de comenzar, asegúrate de tener **uno de los siguientes entornos de ejecución**:

- MongoDB instalado localmente
- MongoDB Compass (interfaz gráfica)
- Docker instalado para contenedor MongoDB
- Mongo Shell (mongosh) o conexión mediante terminal

Crear la base de datos LabTestDB y la colección clientes

A) Si usas la terminal con mongosh:

1. Abre el terminal y ejecuta: `mongosh`
2. Crea o cambia a la base de datos: `use LabTestDB`
3. Verifica que estás en la base correcta: `db`

B) Con Docker (entorno desechable para pruebas):

1. Levanta un contenedor de MongoDB:

```
docker run -d -p 27017:27017 --name mongo-taller mongo
```

2. Accede al contenedor: `docker exec -it mongo-taller mongosh`
3. Luego crea la base: `use LabTestDB`

B) Importar los datos del archivo .json

Puedes utilizar el archivo `LabTestDB.clientes.json` para precargar datos de ejemplo. En el entorno local, haz lo siguiente:

1. Guarda el archivo como `clientes.json`
2. Ejecuta desde la terminal:

```
mongoimport --db LabTestDB --collection clientes --file clientes.json --
jsonArray
```

Asegúrate de estar en el mismo directorio donde está el archivo `LabTestDB.clientes.json`.

Nota: recuerda que `mongoimport` es un ejecutable que debes descargar de las herramientas de gestión de MongoDB. En el siguiente enlace encontrarás dicho recurso <https://www.mongodb.com/try/download/database-tools> (si descargas el zip, en la carpeta `bin/` podrás encontrar dicho ejecutable, copia y pega en la carpeta `bin` de tu instalación de MongoDB).

Verifica que la colección clientes no está vacía (para los ejercicios de UPDATE)

```
db.clientes.find()
```

DELETE en MongoDB

1. Eliminar un cliente por su `_id` específico

```
coleccion.deleteOne({ _id: "CL10" })  
// Borra el documento cuyo _id es exactamente "CL10"
```

2. Eliminar clientes con `estatus` igual a "inactivo"

```
coleccion.deleteMany({ estatus: "inactivo" })  
// Elimina todos los documentos con estatus "inactivo"
```

3. Eliminar el primer cliente ordenado por menor `puntos_fidelidad`

```
coleccion.deleteOne(  
  { estatus: "activo" },  
  { sort: { puntos_fidelidad: 1 } }  
)  
// Elimina el cliente activo con menos puntos
```

4. Usar `collation` para eliminar por nombre sin distinguir mayúsculas

```
coleccion.deleteOne(  
  { nombre: "maria garcia" },  
  { collation: { locale: "en", strength: 2 } }  
)  
// Elimina independientemente de mayúsculas/minúsculas
```

5. Eliminar clientes sin email registrado

```
coleccion.deleteMany({ email: { $exists: false } })  
// Borra documentos donde falta el campo "email"
```

6. Eliminar clientes que tienen más de 3 métodos de pago

```
coleccion.deleteMany({ "metodos_pago.3": { $exists: true } })  
// Borra clientes con al menos 4 métodos de pago
```

7. Eliminar clientes con nombre que empiece por "Test"

```
coleccion.deleteMany({ nombre: /^Test/ })  
// Elimina clientes cuyo nombre comienza por "Test"
```

8. Eliminar documentos con `telefono` nulo

```
coleccion.deleteMany({ telefono: null })  
// Borra los que tienen el campo explícitamente nulo
```

9. Eliminar clientes con fecha de registro anterior a 2023

```
coleccion.deleteMany({  
  fecha_registro: { $lt: ISODate("2023-01-01T00:00:00Z") }  
})  
// Limpieza de registros antiguos
```

10. Eliminar todos los clientes que tengan "fraude" en `comentarios`

```
coleccion.deleteMany({  
  comentarios: /fraude/i  
})  
// Búsqueda insensible a mayúsculas en comentarios
```

11. Eliminar todos los clientes que no tengan puntos de fidelidad

```
coleccion.deleteMany({ puntos_fidelidad: { $exists: false } })  
// Elimina clientes sin ese campo
```

12. Usar hint para forzar un índice durante el borrado

```
coleccion.deleteOne(  
  { email: "usuario@example.com" },  
  { hint: { email: 1 } }  
)  
// Fuerza uso del índice en el campo email
```

13. Eliminar clientes con historial vacío

```
coleccion.deleteMany({ historial_acceso: { $size: 0 } })  
// Usa $size para evaluar longitud de array
```

14. Eliminar cliente con preferencias.notificaciones en false

```
coleccion.deleteOne({ "preferencias.notificaciones": false })  
// Borra un cliente que tenga notificaciones desactivadas
```

15. Eliminar clientes cuyo email no contenga dominio

```
coleccion.deleteMany({ email: { $not: /@/ } })  
// Borra registros con emails mal formateados
```

16. Eliminar clientes con un campo estatus que contenga espacios

```
coleccion.deleteMany({ estatus: / / })  
// Elimina valores erróneos con espacios
```

17. Eliminar solo un cliente con más de 1000 puntos

```
coleccion.deleteOne({ puntos_fidelidad: { $gt: 1000 } })  
// Borra solo uno aunque haya más
```

18. Eliminar clientes de ciudades específicas

```
coleccion.deleteMany({  
  "direccion.ciudad": { $in: ["Puebla", "Toluca", "Mérida"] }  
})  
// Borra clientes en una lista de ciudades
```

19. Eliminar usando condiciones combinadas

```
coleccion.deleteMany({  
  $and: [  
    { estatus: "activo" },  
    { puntos_fidelidad: { $lt: 10 } }  
  ]  
})  
// Borra clientes activos con pocos puntos
```

20. Eliminar todos los documentos de la colección (con precaución)

```
coleccion.deleteMany({})  
// Borra todo el contenido, útil para reiniciar la colección
```

Evaluación: Ejercicios sobre DELETE en MongoDB

Instrucciones para el estudiante

1. Crea un archivo llamado `evaluacion_delete.js`.
2. Añade al inicio del archivo:

```
use("LabTestDB");  
const coleccion = db.getCollection("clientes_eval_delete");
```

3. Completa cada uno de los ejercicios escribiendo la operación de borrado adecuada.

1. Elimina un cliente que tenga el `estatus` "suspendido" y tenga menos de 50 puntos de fidelidad.

```
coleccion.deleteOne({  
  $and: [  
    { estatus: "suspendido" },  
    { /* completa aquí */ }  
  ]  
});
```

2. Borra todos los clientes cuyo nombre empiece con "Test" o termine con "Demo" (usa expresión regular).

```
coleccion.deleteMany({  
  nombre: { $regex: /* completa aquí */ }  
});
```

3. Elimina el primer cliente (usando `sort`) que tenga más de 3 métodos de pago registrados.

```
coleccion.deleteOne(  
  { "metodos_pago.3": { $exists: true } },  
  {  
    sort: { /* completa aquí */ }  
  }  
);
```

4. Elimina todos los clientes que no tengan el campo `fecha_registro`.

```
coleccion.deleteMany({  
  /* completa aquí */  
});
```

5. Elimina un cliente cuyo nombre sea "luis fernando", sin importar el uso de mayúsculas/minúsculas. Usa `collation`.

```
coleccion.deleteOne(  
  { nombre: "luis fernando" },  
  {  
    collation: { /* completa aquí */ }  
  }  
);
```

6. Elimina todos los clientes que hayan sido registrados antes del 1 de enero de 2022.

```
coleccion.deleteMany({  
  fecha_registro: { /* operador adecuado */ }  
});
```

7. Elimina clientes cuyo campo `comentarios` contenga la palabra "fraude", sin importar mayúsculas.

```
coleccion.deleteMany({  
  comentarios: { $regex: /fraude/i }  
});
```

8. Borra todos los documentos de clientes que no tengan correo electrónico válido (es decir, sin @ en el campo `email`).

```
coleccion.deleteMany({  
  email: { $not: /@/ }  
});
```

9. Elimina todos los documentos donde el array `historial_acceso` esté vacío.

```
coleccion.deleteMany({  
  historial_acceso: { $size: /* completa aquí */ }  
});
```

10. Usa `hint` para eliminar un cliente por su correo electrónico, forzando el uso del índice `{ email: 1 }`.

```
coleccion.deleteOne(  
  { email: "cliente123@example.com" },  
  {  
    hint: { /* completa aquí */ }  
  }  
);
```