

Practica 1. Mongo DB

Alumnos:

- Flores Garzofino Mauricio Joel (SiS)
- Gonzales Suyu Franz Reinaldo (SIS - CICO)

Parte 3.

Enunciados

1. Insertar un alumno mujer con dos materias y padre de apellido "Torres".

Comando:

```
db.alumnos.insertOne( { nombres: "Diana", apellidos: "Torres", sexo: "F", casado: false, fecha_nacimiento: new Date("2003-01-01"), materias: ["SIS101", "SIS252"], padre: { nombre: "Ernesto", apellido: "Torres"} })
```

Output:

```
bd_practica> db.alumnos.insertOne( { nombres: "Diana", apellidos: "Torres", sexo: "F", casado: false, fecha_nacimiento: new Date("2003-01-01"), materias: ["SIS101", "SIS252"], padre: { nombre: "Ernesto", apellido: "Torres"} })
{
  acknowledged: true,
  insertedId: ObjectId('689e3442e69c9e58f289b043')
}
```

2. Insertar cinco alumnos a la vez con datos variados.

Comando:

```
db.alumnos.insertMany([
  {
    "nombres": "Luis",
    "apellidos": "García",
    "sexo": "M",
    "casado": true,
    "fecha_nacimiento": new Date("1995-07-20"),
    "materias": ["MAT201", "FIS201", "QUI105"],
    "padre": {
      "nombre": "Carlos",
      "apellido": "García"
    }
  },
  {
    "nombres": "Sofía",
    "apellidos": "López",
    "sexo": "F",
    "casado": false,
    "fecha_nacimiento": new Date("1998-11-12"),
    "materias": ["HIS301", "ART101"],
    "padre": {
```

```
        "nombre": "José",
        "apellido": "López"
    },
    {
        "nombres": "Javier",
        "apellidos": "Martínez",
        "sexo": "M",
        "casado": false,
        "fecha_nacimiento": new Date("2001-03-05"),
        "materias": ["COM350", "SIS256", "SIS00"],
        "padre": {
            "nombre": "Ricardo",
            "apellido": "Martínez"
        }
    },
    {
        "nombres": "María",
        "apellidos": "Fernández",
        "sexo": "F",
        "casado": false,
        "fecha_nacimiento": new Date("2002-09-25"),
        "materias": ["SIS101", "SIS252"],
        "padre": {
            "nombre": "Alberto",
            "apellido": "Fernández"
        }
    },
    {
        "nombres": "Daniel",
        "apellidos": "Torres",
        "sexo": "M",
        "casado": true,
        "fecha_nacimiento": new Date("1996-04-18"),
        "materias": ["SOC101"],
        "padre": {
            "nombre": "Ernesto",
            "apellido": "Torres"
        }
    }
];
```

Output:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId( '689e348fe69c9e58f289b044' ),
    '1': ObjectId( '689e348fe69c9e58f289b045' ),
    '2': ObjectId( '689e348fe69c9e58f289b046' ),
    '3': ObjectId( '689e348fe69c9e58f289b047' ),
    '4': ObjectId( '689e348fe69c9e58f289b048' )
  }
}
```

3. Insertar un alumno sin campo padre.

Comando:

```
bd.alumnos.insertOne({ nombres: "David", apellidos: "Castro", sexo: "M", casado: false, fecha_nacimiento: new Date("2004-04-04"), materias: ["SIS252"] })
```

Output

```
bd_practica> bd.alumnos.insertOne({ nombres: "David", apellidos: "Castro", sexo: "M", casado: false, fecha_nacimiento: new Date("2004-04-04"), materias: ["SIS252"] })
ReferenceError: bd is not defined
bd_practica> db.alumnos.insertOne({ nombres: "David", apellidos: "Castro", sexo: "M", casado: false, fecha_nacimiento: new Date("2004-04-04"), materias: ["SIS252"] })
{
  acknowledged: true,
  insertedId: ObjectId( '689e352ae69c9e58f289b049' )
}
```

4. Insertar a un alumno hombre casado, nacido en 1990.

Comando:

```
db.alumnos.insertOne({ nombres: "Roger", apellidos: "Valdivieso", sexo: "M", casado: true, fecha_nacimiento: new Date("1990-09-09")})
```

Output:

```
bd_practica> db.alumnos.insertOne({ nombres: "Roger", apellidos: "Valdivieso", sexo: "M", casado: true, fecha_nacimiento: new Date("1990-09-09")})
{
  acknowledged: true,
  insertedId: ObjectId( '689e3eaae69c9e58f289b04e' )
}
```

5. Insertar un alumno con 4 materias y padre "Carlos Pérez".

Comando:

```
db.alumnos.insertOne({
  "nombres": "Juan",
  "apellidos": "Pérez",
  "sexo": "M",
  "casado": false,
  "fecha_nacimiento": new Date("2002-09-10"),
  "materias": [
    "HIS201",
```

```

    "LIT105",
    "SOC202",
    "ART110"
  ],
  "padre": {
    "nombre": "Carlos",
    "apellido": "Pérez"
  }
});

```

Output:

```

bd_practica> db.alumnos.insertOne({
...   "nombres": "Juan",
...   "apellidos": "Pérez",
...   "sexo": "M",
...   "casado": false,
...   "fecha_nacimiento": new Date("2002-09-10"),
...   "materias": [
...     "HIS201",
...     "LIT105",
...     "SOC202",
...     "ART110"
...   ],
...   "padre": {
...     "nombre": "Carlos",
...     "apellido": "Pérez"
...   }
... });
...
{
  acknowledged: true,
  insertedId: ObjectId('689e3e69e69c9e58f289b04d')
}

```

6. Insertar un alumno solo con "COM350" como materia.

Comando:

```

db.alumnos.insertOne({
  "nombres": "Erick",
  "apellidos": "Arciénega",
  "sexo": "M",
  "casado": false,
  "fecha_nacimiento": new Date("2005-02-10"),
  "materias": [
    "COM350",
  ],
  "padre": {
    "nombre": "David",

```

```
"apellido": "Arciénega"
}
});
```

Output:

```
bd_practica> db.alumnos.insertOne({
...   "nombres": "Erick",
...   "apellidos": "Arciénega",
...   "sexo": "M",
...   "casado": false,
...   "fecha_nacimiento": new Date("2005-02-10"),
...   "materias": [
...     "COM350",
...   ],
...   "padre": {
...     "nombre": "David",
...     "apellido": "Arciénega"
...   }
... });
...
{
  acknowledged: true,
  insertedId: ObjectId('689e3e37e69c9e58f289b04c')
}
```

7. Listar alumnos cuyo padre tenga nombre "Luis" o "Miguel".

Comando:

```
db.alumnos.find({ "padre.nombre": { $in: ["Luis", "Miguel"] } })
```

Output:

```

bd_practica> db.alumnos.find({ "padre.nombre": { $in: ["Luis", "Miguel"] } })
[
  {
    _id: ObjectId('689dee388be1607eabeec4cd'),
    nombres: 'Hugo',
    apellidos: 'Martínez Martínez',
    sexo: 'M',
    casado: true,
    fecha_nacimiento: ISODate('2003-06-02T12:31:51.785Z'),
    materias: [ 'SIS00' ],
    padre: { nombre: 'Miguel', apellido: 'Vargas' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4d3'),
    nombres: 'Valeria',
    apellidos: 'Pérez López',
    sexo: 'F',
    casado: true,
    fecha_nacimiento: ISODate('1990-08-28T05:52:41.253Z'),
    materias: [ 'SIS00', 'COM350', 'SIS258', 'SIS256' ],
    padre: { nombre: 'Luis', apellido: 'García' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4da'),
    nombres: 'Pedro',
    apellidos: 'García Rodríguez',
    sexo: 'M',
    casado: false,
    fecha_nacimiento: ISODate('2003-07-26T13:25:22.675Z'),
    materias: [ 'COM350', 'SIS256', 'SIS258', 'SIS00' ],
  }
]

```

8. Mostrar alumnos no casados que tengan "SIS256".

Comando:

```
db.alumnos.find({ casado: false, "materias": { $nin: ["SIS256"] } })
```

Output:

```

bd_practica> db.alumnos.find({ casado: false, "materias": { $nin: ["SIS256"] } })
[
  {
    _id: ObjectId('689dee388be1607eabeec4cc'),
    nombres: 'Carlos',
    apellidos: 'Flores Pérez',
    sexo: 'M',
    casado: false,
    fecha_nacimiento: ISODate('2005-07-24T22:52:05.735Z'),
    materias: [ 'SIS00', 'COM350' ],
    padre: { nombre: 'Carlos', apellido: 'Pérez' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4d0'),
    nombres: 'Camila',
    apellidos: 'Ramírez López',
    sexo: 'F',
    casado: false,
    fecha_nacimiento: ISODate('1995-07-04T08:41:36.603Z'),
    materias: [ 'SIS258' ],
    padre: { nombre: 'Hugo', apellido: 'Vargas' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4d9'),
    nombres: 'Gabriela',
    apellidos: 'Pérez Torres',
    sexo: 'F',
    casado: false,
    fecha_nacimiento: ISODate('1997-10-24T05:56:29.752Z'),
    materias: [ 'SIS258', 'COM350' ],
  }
]

```

9. Contar alumnos con más de 2 materias.

Comando:

```
db.alumnos.find({ "materias.1": { $exists: true } })
```

Output:

```

bd_practica> db.alumnos.find({ "materias.1": { $exists: true } })
[
  {
    _id: ObjectId('689dee388be1607eabeec4ca'),
    nombres: 'Pedro',
    apellidos: 'Vargas Sánchez',
    sexo: 'M',
    casado: true,
    fecha_nacimiento: ISODate('1997-02-02T12:54:32.988Z'),
    materias: [ 'COM350', 'SIS256', 'SIS00' ],
    padre: { nombre: 'Pedro', apellido: 'Rodríguez' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4cc'),
    nombres: 'Carlos',
    apellidos: 'Flores Pérez',
    sexo: 'M',
    casado: false,
    fecha_nacimiento: ISODate('2005-07-24T22:52:05.735Z'),
    materias: [ 'SIS00', 'COM350' ],
    padre: { nombre: 'Carlos', apellido: 'Pérez' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4cb'),
    nombres: 'Elena',
    apellidos: 'Martínez Torres',
    sexo: 'F',
    casado: false,
    fecha_nacimiento: ISODate('1995-02-27T18:22:40.689Z'),
    materias: [ 'SIS258', 'COM350', 'SIS256' ],
  }
]

```

10. Mostrar los 10 más viejos con nombres y fecha_nacimiento.

Comando:

```

db.alumnos.find({}, { nombres: 1, fecha_nacimiento: 1, _id: 0 }).sort({ fecha_nacimiento: 1 })
.limit(10);

```

Output:

```

bd_practica> db.alumnos.find({}, { nombres: 1, fecha_nacimiento: 1, _id: 0 }).sort({ fecha_nacimiento: 1 }).limit(10);
[
  {
    nombres: 'Diana',
    fecha_nacimiento: ISODate('1990-01-31T10:25:34.576Z')
  },
  {
    nombres: 'Paula',
    fecha_nacimiento: ISODate('1990-02-26T13:36:52.262Z')
  },
  {
    nombres: 'Camila',
    fecha_nacimiento: ISODate('1990-03-03T05:45:48.016Z')
  },
  {
    nombres: 'Carla',
    fecha_nacimiento: ISODate('1990-07-11T13:42:40.242Z')
  },
  {
    nombres: 'Andrés',
    fecha_nacimiento: ISODate('1990-08-12T06:41:48.967Z')
  },
  {
    nombres: 'Valeria',
  }
]

```

11. Buscar alumnos con "COM350" y "SIS00".

Comando:

```

db.alumnos.find({ "materias": { $all: ["COM350", "SIS00"] } })

```


Output:

```
bd_practica> db.alumnos.find({ "materias": { $all: ["COM350", "SIS00"] }})
[
  {
    _id: ObjectId('689dee388be1607eabeec4ca'),
    nombres: 'Pedro',
    apellidos: 'Vargas Sánchez',
    sexo: 'M',
    casado: true,
    fecha_nacimiento: ISODate('1997-02-02T12:54:32.988Z'),
    materias: [ 'COM350', 'SIS256', 'SIS00' ],
    padre: { nombre: 'Pedro', apellido: 'Rodríguez' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4cc'),
    nombres: 'Carlos',
    apellidos: 'Flores Pérez',
    sexo: 'M',
    casado: false,
    fecha_nacimiento: ISODate('2005-07-24T22:52:05.735Z'),
    materias: [ 'SIS00', 'COM350' ],
    padre: { nombre: 'Carlos', apellido: 'Pérez' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4ce'),
    nombres: 'Gabriela',
    apellidos: 'Pérez Torres',
    sexo: 'F',
    casado: true,
    fecha_nacimiento: ISODate('2005-10-16T00:05:57.917Z'),
    materias: [ 'COM350', 'SIS256', 'SIS00' ],
  }
]
```

12. Mostrar solo nombres y sexo de los alumnos mujeres.

Comando: `db.alumnos.find({ "sexo": "F" }, { nombres: 1, sexo: 1, _id: 0 })`

Output:

```
bd_practica> db.alumnos.find( { "sexo": "F" }, { nombres: 1, sexo: 1, _id: 0 })
[
  { nombres: 'Elena', sexo: 'F' },
  { nombres: 'Gabriela', sexo: 'F' },
  { nombres: 'Camila', sexo: 'F' },
  { nombres: 'Sofía', sexo: 'F' },
  { nombres: 'Carla', sexo: 'F' },
  { nombres: 'Valeria', sexo: 'F' },
  { nombres: 'Elena', sexo: 'F' },
  { nombres: 'Gabriela', sexo: 'F' },
  { nombres: 'Ana', sexo: 'F' },
  { nombres: 'Paula', sexo: 'F' },
  { nombres: 'Lucía', sexo: 'F' },
  { nombres: 'Carla', sexo: 'F' },
  { nombres: 'Elena', sexo: 'F' },
  { nombres: 'María', sexo: 'F' },
  { nombres: 'Carla', sexo: 'F' },
  { nombres: 'Sofía', sexo: 'F' },
  { nombres: 'Camila', sexo: 'F' },
  { nombres: 'Paula', sexo: 'F' },
  { nombres: 'Paula', sexo: 'F' },
  { nombres: 'Carla', sexo: 'F' }
]
```

13. Ordenar todos los alumnos por apellido ascendente.

Comando:

```
db.alumnos.find({}).sort({apellido: 1})
```

Output:

```
bd_practica> db.alumnos.find({}).sort({apellidos: 1})
[
  {
    _id: ObjectId('689e352ae69c9e58f289b049'),
    nombres: 'David',
    apellidos: 'Castro',
    sexo: 'M',
    casado: false,
    fecha_nacimiento: ISODate('2004-04-04T00:00:00.000Z'),
    materias: [ 'SIS252' ]
  },
  {
    _id: ObjectId('689e348fe69c9e58f289b047'),
    nombres: 'María',
    apellidos: 'Fernández',
    sexo: 'F',
    casado: false,
    fecha_nacimiento: ISODate('2002-09-25T00:00:00.000Z'),
    materias: [ 'SIS101', 'SIS252' ],
    padre: { nombre: 'Alberto', apellido: 'Fernández' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4e0'),
    nombres: 'Jorge',
    apellidos: 'Flores García',
    sexo: 'M',
    casado: true,
    fecha_nacimiento: ISODate('2000-09-25T23:58:16.771Z'),
```

14. Listar alumnos nacidos entre 1998 y 2003.

Comando:

```
db.alumnos.find({ "fecha_nacimiento": { $gte: new Date("1998-01-01"), $lt: new Date("2004-01-01") } })
```

Output:

```

bd_practica> db.alumnos.find({
...   "fecha_nacimiento": {
...     $gte: new Date("1998-01-01T00:00:00Z"),
...     $lt: new Date("2004-01-01T00:00:00Z")
...   }
... })
...
[
  {
    _id: ObjectId('689dee388be1607eabeec4c7'),
    nombres: 'Luis',
    apellidos: 'Pérez López',
    sexo: 'M',
    casado: true,
    fecha_nacimiento: ISODate('1999-05-20T23:42:05.958Z'),
    materias: [ 'COM350' ],
    padre: { nombre: 'Daniel', apellido: 'Martínez' }
  },
  {
    _id: ObjectId('689dee388be1607eabeec4cd'),
    nombres: 'Hugo',
    apellidos: 'Martínez Martínez',
    sexo: 'M',
    casado: true,
    fecha_nacimiento: ISODate('2003-06-02T12:31:51.785Z'),
    materias: [ 'SIS00' ],
    padre: { nombre: 'Miguel', apellido: 'Vargas' }
  },
]

```

15. Mostrar alumnos cuyo apellido termine en "ez".

- bd_practica> db.alumnos.find({apellidos: /ez\$/i})

```
bd_practica> N1TIRfXcDjfl8XTIOS6Q4Nl
bd_practica> db.alumnos.find({apellidos: /ez$/i})
{
  _id: ObjectId('689dee388be1607eabeec4c8'),
  nombres: 'Carla',
  apellidos: 'Vargas Martínez',
  sexo: 'F',
  casado: false,
  fecha_nacimiento: ISODate('1992-05-16T02:47:46.804Z'),
  materias: ['COM350', 'SIS258', 'SIS256'],
  padre: { nombre: 'Andrés', apellido: 'García' }
}
{
  _id: ObjectId('689dee388be1607eabeec4cf'),
  nombres: 'Sofía',
  apellidos: 'Ramírez Ramírez',
  sexo: 'F',
  casado: false,
  fecha_nacimiento: ISODate('1999-02-20T09:02:10.434Z'),
  materias: ['SIS256', 'COM350', 'SIS00', 'SIS258'],
  padre: { nombre: 'Juan', apellido: 'Ramírez' }
}
{
  _id: ObjectId('689dee388be1607eabeec4d0'),
  nombres: 'Camila',
  apellidos: 'Ramírez-López',
  sexo: 'F',
  casado: false,
  fecha_nacimiento: ISODate('1995-07-04T08:41:36.603Z'),
  materias: ['SIS258'],
  padre: { nombre: 'Hugo', apellido: 'Vargas' }
},
```

1. use bd_practica
2.
3. db.alumnos.find({
4. apellidos: /ez\$/i
5. }, {
6. nombres: 1,
7. apellidos: 1,
8. _id: 0
9. })
10. Explicación:
• use bd_practica : selecciona la base de datos.
• db.alumnos.find(...) : consulta en la colección alumnos
• { apellidos: /ez\$/i } :
• /ez\$/ → expresión regular que busca cadenas que terminen en 'ez'.
• i = insensible a mayúsculas/minúsculas.
• { nombres: 1, apellidos: 1, _id: 0 } :
• Proyección para mostrar solo nombres y apellidos.

Ejemplo de resultados (parcial):
desarrollo web Generación de imágenes Generar

¿Cómo puedo ayudarte hoy?

16. Contar alumnos casados y con "SIS258".

- bd_practica> db.alumnos.find({casado: true, materias: "SIS258"})

```
25. Eliminar a los alumnos cuyo padre tenga apellido "Ramírez".
26. Eliminar a los que nacieron después de 2004.

bd_practica> db.alumnos.find({casado: true, materias: "SIS258"}).count()
12
bd_practica>
```

17. Cambiar el nombre "Luis" por "Luis Alberto".

- bd_practica> db.alumnos.updateMany({nombres:"Luis"}, {\$set:{nombres:"Luis Alberto"}})

```
bd_practica> db.alumnos.updateMany({nombres:"Luis"}, {$set:{nombres:"Luis Alberto"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
```

18. Agregar "SIS777" a todos los alumnos con "COM350".

- bd_practica> db.alumnos.updateMany({materias:"COM350"}, {\$addToSet: {materias:"SIS777"}})

```
bd_practica> db.alumnos.updateMany({materias:"COM350"}, {$addToSet: {materias:"SIS777"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 34,
  modifiedCount: 34,
  upsertedCount: 0
}
```

17. Cambiar el nombre "Luis" por "Luis Alberto".

```
bd_practica> db.alumnos.updateMany({nombres:"Luis"}, {$set:{nombres:"Luis Alberto"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
```

18. Agregar "SIS777" a todos los alumnos con "COM350".

```
bd_practica> db.alumnos.find({materias: "COM350"})
[
  {
    _id: ObjectId('689dee388be1607eabeec4ce'),
    nombres: 'Gabriela',
    apellidos: 'Pérez Torres',
    sexo: 'F',
    casado: true,
    fecha_nacimiento: ISODate('2005-10-16T00:05:57.917Z'),
    materias: [ 'COM350', 'SIS256', 'SIS00', 'SIS777' ],
    padre: { nombre: 'Hugo', apellido: 'Torres' }
  }
]
```

19. Poner casado: false a todos los alumnos con menos de 2 materias.

- db.alumnos.updateMany({\$expr:{\$lt:[\$size:"\$materias"], 2}}, {\$set:{casado:false}})

```
bd_practica> db.alumnos.updateMany({$expr:{$lt:[$size:"$materias"], 2}}, {$set:{casado:false}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 10,
  modifiedCount: 5,
  upsertedCount: 0
}
```

bd_practica> |

• { \$lt: [tamaño, 2] } : verifica si ese tamaño es menor que 2

• Así puedes hacer comparaciones numéricas con el tamaño del arreglo

desarrollo web Generación de imágenes Generación de Video Artefactos

¿Cómo puedo ayudarte hoy?

20. Quitar "SIS256" de todos los que la tengan.

- db.alumnos.updateMany({materias:"SIS256"}, {\$pull: {materias:"SIS256"}})

```

apellidos: "Vargas Martínez"
bd_practica> db.alumnos.updateMany({materias: "SIS256"}, {$pull: {materias: "SIS256"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 29,
  modifiedCount: 29,
  upsertedCount: 0
}
bd_practica> db.alumnos.find({materias: "SIS256"}).count()
0
casado: false
bd_practica> db.alumnos.updateMany({casado: false}, {$set: {casado: true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 29,
  modifiedCount: 29,
  upsertedCount: 0
}

```

21. Cambiar el nombre del padre "Juan" a "Juan José".

- db.alumnos.updateMany({"padre.nombre": "Juan"}, {\$set: {"padre.nombre": "Juan José"}})

```

bd_practica> db.alumnos.updateMany({"padre.nombre": "Juan"}, {$set: {"padre.nombre": "Juan José"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}

```

22. Agregar un campo activo: true a todos los documentos.

- db.alumnos.updateMany({}, {\$set: {activo: true}})

```

bd_practica> db.alumnos.updateMany({}, {$set: {activo: true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 55,
  modifiedCount: 55,
  upsertedCount: 0
}

```

23. Eliminar todos los alumnos hombres.

- db.alumnos.deleteMany({sexo: "M"})

```

bd_practica> db.alumnos.deleteMany({sexo: "M"})
{
  acknowledged: true,
  deletedCount: 31
}

```

24. Eliminar a quien tenga "SIS777" como materia.

- db.alumnos.deleteMany({materias: "SIS77"})

```
acknowledged: true,  
bd_practica> db.alumnos.deleteMany({materias: "SIS77"})  
{ acknowledged: true, deletedCount: 18 }  
bd_practica>
```

25. Eliminar a los alumnos cuyo padre tenga apellido "Ramírez".

- db.alumnos.deleteMany({"padre.apellido": "Ramirez"})

```
bd_practica> db.alumnos.deleteMany({"padre.apellido": "Ramirez"})  
{ acknowledged: true, deletedCount: 6 }
```

26. Eliminar a los que nacieron después de 2004.

- bd_practica> db.alumnos.deleteMany({ "fecha_nacimiento": { \$gt: ISODate("2004-12-31T23:59:59Z") } })

```
bd_practica> db.alumnos.deleteMany({ "fecha_nacimiento": { $gt: ISODate("2004-12-31T23:59:59Z") } })  
{ acknowledged: true, deletedCount: 0 }  
bd_practica>
```

27. Eliminar todos los casados.

- db.alumnos.deleteMany({ casado: true })

```
bd_practica> db.alumnos.deleteMany({ casado: true })  
{ acknowledged: true, deletedCount: 1 }  
bd_practica>
```

28. Eliminar al primer alumno que tenga "SIS258".

- db.alumnos.deleteOne({ materias: "SIS258" })

```
bd_practica> db.alumnos.deleteOne({ materias: "SIS258" })  
{ acknowledged: true, deletedCount: 1 }  
bd_practica>
```