

**UNIVERSIDAD MAYOR REAL Y PONTIFICIA DE SAN FRANCISCO
XAVIER DE CHUQUISACA**

**FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA DE SISTEMAS**



MICROSERVICIOS (COM600)

PRACTICA #1

DOCENTE: Ing. Montellano Barriga Carlos

ESTUDIANTES: Vidaurre Mejia Christian Paul

Miranda Veniz Marcela

SUCRE-BOLIVIA

Importando archivo json

```
C:\Program Files\MongoDB\Server\8.0\bin>mongoimport --db bd_practica --collection alumnos --file d:\alumnos.json --jsonArray
2025-08-14T14:51:49.289-0400    connected to: mongodb://localhost/
2025-08-14T14:51:49.319-0400    50 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\8.0\bin>
```

Parte 2. Ejercicios Guiados (Con Respuestas)

Creación

1. Insertar un alumno llamado "Carlos López"...

```
> db.alumnos.insertOne({
  nombres: "Carlos",
  apellidos: "López",
  sexo: "M",
  casado: false,
  fecha_nacimiento: new Date("2001-05-10"),
  materias: ["SIS256", "SIS258"],
  padre: { nombre: "Juan", apellido: "López" }
})
< {
  acknowledged: true,
  insertedId: ObjectId('689e310b7dd56e05cf089a4b')
}
bd_practica>
```

```
_id: ObjectId('689e310b7dd56e05cf089a4b')
nombres: "Carlos"
apellidos: "López"
sexo: "M"
casado: false
fecha_nacimiento: 2001-05-10T00:00:00.000+00:00
▶ materias: Array (2)
▶ padre: Object
```

2. Insertar tres alumnos de una sola vez usando insertMany.

```
bd_practica> db.alumnos.insertMany([
  { nombres: "Ana", apellidos: "Pérez", sexo: "F", casado: true, fecha_nacimiento: new Date("1999-03-20"), materias: ["COM350"], padre: { no
  { nombres: "Luis", apellidos: "Torres", sexo: "M", casado: false, fecha_nacimiento: new Date("2002-11-15"), materias: ["SIS256", "SIS00"],
  { nombres: "Elena", apellidos: "Ramírez", sexo: "F", casado: false, fecha_nacimiento: new Date("2000-07-05"), materias: ["SIS258", "COM350
  ]})
```

```

> db.alumnos.insertMany([
  { nombres: "Ana", apellidos: "Pérez", sexo: "F", casado: true, fecha_nacimiento: new Date("1999-03-20"), materias: ["COM350"], padre: { nombre: "Pedro", apellidos: "Pérez", sexo: "M", casado: true, fecha_nacimiento: new Date("1995-01-15") }, },
  { nombres: "Luis", apellidos: "Torres", sexo: "M", casado: false, fecha_nacimiento: new Date("2002-11-15"), materias: ["SIS256", "SIS00"], padre: { nombre: "Luis", apellidos: "Torres", sexo: "M", casado: false, fecha_nacimiento: new Date("2002-11-15") }, },
  { nombres: "Elena", apellidos: "Ramírez", sexo: "F", casado: false, fecha_nacimiento: new Date("2000-07-05"), materias: ["SIS258", "COM350"], padre: { nombre: "Luis", apellidos: "Torres", sexo: "M", casado: false, fecha_nacimiento: new Date("2002-11-15") }, }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('689e31e17dd56e05cf089a4c'),
    '1': ObjectId('689e31e17dd56e05cf089a4d'),
    '2': ObjectId('689e31e17dd56e05cf089a4e')
  }
}
bd_practica>

```

3. Insertar un alumno que no tenga materias.

```

> db.alumnos.insertOne({
  nombres: "Jorge",
  apellidos: "García",
  sexo: "M",
  casado: false,
  fecha_nacimiento: new Date("1998-08-12"),
  materias: [],
  padre: { nombre: "Luis", apellido: "García" }
})
< {
  acknowledged: true,
  insertedId: ObjectId('689e32077dd56e05cf089a4f')
}
bd_practica> |

```

Consultas

1. Mostrar los alumnos cuyo sexo sea "M" o "F" y que tengan "SIS256" o "COM350"...

```
> db.alumnos.find({ materias: { $in: ["SIS256", "COM350"] } }, { _id: 0, nombres: 1, apellidos: 1, materias: 1 })
< {
  nombres: 'Pedro',
  apellidos: 'Vargas Sánchez',
  materias: [
    'COM350',
    'SIS256',
    'SIS00'
  ]
}
{
  nombres: 'Valeria',
  apellidos: 'Pérez López',
  materias: [
    'SIS00',
    'COM350',
    'SIS258',
    'SIS256'
  ]
}
{
  nombres: 'Luis',
  apellidos: 'Torres Sánchez',
  materias: [
    'SIS00',
    'SIS256'
  ]
}
```

2. Listar los alumnos nacidos después del año 2000 y casados.

```

> db.alumnos.find({ fecha_nacimiento: { $gt: new Date("2000-01-01") }, casado: true })
< {
  _id: ObjectId('689dee388be1607eabeec4cd'),
  nombres: 'Hugo',
  apellidos: 'Martínez Martínez',
  sexo: 'M',
  casado: true,
  fecha_nacimiento: 2003-06-02T12:31:51.785Z,
  materias: [
    'SIS00'
  ],
  padre: {
    nombre: 'Miguel',
    apellido: 'Vargas'
  }
}
{
  _id: ObjectId('689dee388be1607eabeec4d8'),
  nombres: 'Elena',
  apellidos: 'Martínez Ramírez',
  sexo: 'F',
  casado: true,
  fecha_nacimiento: 2002-08-14T13:42:34.786Z,
  materias: [
    'SIS258',
    'SIS256',
    'SIS00',
    'COM350'
  ],

```

3. Buscar alumnas o cuyo padre tenga apellido "Pérez".

```

> db.alumnos.find({ $or: [{ sexo: "F" }, { "padre.apellido": "Pérez" }] })
< {
  _id: ObjectId('689dee388be1607eabeec4d0'),
  nombres: 'Camila',
  apellidos: 'Ramírez López',
  sexo: 'F',
  casado: false,
  fecha_nacimiento: 1995-07-04T08:41:36.603Z,
  materias: [
    'SIS258'
  ],
  padre: {
    nombre: 'Hugo',
    apellido: 'Vargas'
  }
}
{
  _id: ObjectId('689dee388be1607eabeec4d3'),
  nombres: 'Valeria',
  apellidos: 'Pérez López',
  sexo: 'F',
  casado: true,
  fecha_nacimiento: 1990-08-28T05:52:41.253Z,
  materias: [
    'SIS00',
    'COM350',
    'SIS258',
    'SIS256'
  ]
}

```

4. Contar cuántos alumnos llevan "SIS258".

```

> db.alumnos.countDocuments({ materias: "SIS258" })
< 31
bd_practica>

```

5. Mostrar los 3 alumnos más jóvenes con nombres, apellidos y fecha_nacimiento.

```
> db.alumnos.find({}, { _id: 0, nombres: 1, apellidos: 1, fecha_nacimiento: 1 }).sort({ fecha_nacimiento: -1 }).limit(3)
< {
  nombres: 'Hugo',
  apellidos: 'Vargas Rodríguez',
  fecha_nacimiento: 2005-11-25T11:52:24.432Z
}
{
  nombres: 'Gabriela',
  apellidos: 'Pérez Torres',
  fecha_nacimiento: 2005-10-16T00:05:57.917Z
}
{
  nombres: 'Carlos',
  apellidos: 'Flores Pérez',
  fecha_nacimiento: 2005-07-24T22:52:05.735Z
}
bd_practica>
```

Actualizaciones – 3 preguntas

1. Agregar "SIS999" a un alumno "Carlos".

```
> db.alumnos.updateOne({ nombres: "Carlos" }, { $addToSet: { materias: "SIS999" } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
bd_practica>
```

```

> db.alumnos.find({nombres:"Carlos"})
< {
  _id: ObjectId('689dee388be1607eabeec4e7'),
  nombres: 'Carlos',
  apellidos: 'López Vargas',
  sexo: 'M',
  casado: false,
  fecha_nacimiento: 2003-05-23T21:19:00.048Z,
  materias: [
    'COM350',
    'SIS999'
  ],
  padre: {
    nombre: 'Luis',
    apellido: 'Torres'
  }
}

```

2. Marcar como casado a todos con "SIS258".

```

> db.alumnos.updateMany({ materias: "SIS258" }, { $set: { casado: true } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 31,
  modifiedCount: 19,
  upsertedCount: 0
}
bd_practica>

```

3. Quitar "SIS00" de todos los alumnos que la tengan.


```
> db.alumnos.updateMany({ materias: "SIS00" }, { $pull: { materias: "SIS00" } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 34,
  modifiedCount: 34,
  upsertedCount: 0
}
bd_practica> |
```

Eliminaciones – 3 preguntas

1. Eliminar un alumno "Ana" con apellido que contenga "Pérez".

```
> db.alumnos.deleteOne({ nombres: "Ana", apellidos: /Pérez/ })
< {
  acknowledged: true,
  deletedCount: 1
}
bd_practica> |
```

2. Eliminar todos los alumnos sin materias.

```
> db.alumnos.deleteMany({ materias: { $size: 0 } })
< {
  acknowledged: true,
  deletedCount: 4
}
bd_practica> |
```

3. Eliminar todos los alumnos nacidos antes de 1995.

```
> db.alumnos.deleteMany({ fecha_nacimiento: { $lt: new Date("1995-01-01") } })
< {
  acknowledged: true,
  deletedCount: 13
}
bd_practica> |
```

Parte 3. Practica

Enunciados

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

```
bd_practica> db.alumnos.insertOne({
  nombres: "Marcela",
  apellidos: "Torres",
  sexo: "F",
  casado: false,
  fecha_nacimiento: new Date("2003-09-15"),
  materias: ["SIS101", "COM200"],
  padre: { nombre: "Pedro", apellido: "Torres" }
})
```

```

> db.alumnos.insertOne({
  nombres: "Marcela",
  apellidos: "Torres",
  sexo: "F",
  casado: false,
  fecha_nacimiento: new Date("2003-09-15"),
  materias: ["SIS101", "COM200"],
  padre: { nombre: "Pedro", apellido: "Torres" }
})
< {
  acknowledged: true,
  insertedId: ObjectId('689e34a17dd56e05cf089a50')
}
bd_practica> |

```

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

```

bd_practica> db.alumnos.insertMany([
  { nombres: "Pedro", apellidos: "Gutiérrez", sexo: "M", casado: true, fecha_nacimiento: new Date("1999-04-12"), materias: ["SIS256", "COM35"], padre: { nombre: "Pedro", apellido: "Gutiérrez" } },
  { nombres: "Lucía", apellidos: "Martínez", sexo: "F", casado: false, fecha_nacimiento: new Date("2001-07-25"), materias: ["MAT101", "SIS25"], padre: { nombre: "Lucía", apellido: "Martínez" } },
  { nombres: "Andrés", apellidos: "Pérez", sexo: "M", casado: false, fecha_nacimiento: new Date("2002-12-05"), materias: ["COM200"], padre: { nombre: "Andrés", apellido: "Pérez" } },
  { nombres: "Valeria", apellidos: "Torres", sexo: "F", casado: true, fecha_nacimiento: new Date("1998-03-18"), materias: ["SIS999", "MAT101"], padre: { nombre: "Valeria", apellido: "Torres" } },
  { nombres: "Santiago", apellidos: "Ramírez", sexo: "M", casado: false, fecha_nacimiento: new Date("2004-10-11"), materias: ["SIS256", "SIS258"], padre: { nombre: "Santiago", apellido: "Ramírez" } }
])
|

```

```

db.alumnos.insertMany([
  { nombres: "Pedro", apellidos: "Gutiérrez", sexo: "M", casado: true, fecha_nacimiento: new Date("1999-04-12"), materias: ["SIS256", "COM350"], padre: { nombre: "Pedro", apellido: "Gutiérrez" } },
  { nombres: "Lucía", apellidos: "Martínez", sexo: "F", casado: false, fecha_nacimiento: new Date("2001-07-25"), materias: ["MAT101", "SIS258"], padre: { nombre: "Lucía", apellido: "Martínez" } },
  { nombres: "Andrés", apellidos: "Pérez", sexo: "M", casado: false, fecha_nacimiento: new Date("2002-12-05"), materias: ["COM200"], padre: { nombre: "Andrés", apellido: "Pérez" } },
  { nombres: "Valeria", apellidos: "Torres", sexo: "F", casado: true, fecha_nacimiento: new Date("1998-03-18"), materias: ["SIS999", "MAT101"], padre: { nombre: "Valeria", apellido: "Torres" } },
  { nombres: "Santiago", apellidos: "Ramírez", sexo: "M", casado: false, fecha_nacimiento: new Date("2004-10-11"), materias: ["SIS256", "SIS258"], padre: { nombre: "Santiago", apellido: "Ramírez" } }
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('689e35017dd56e05cf089a51'),
    '1': ObjectId('689e35017dd56e05cf089a52'),
    '2': ObjectId('689e35017dd56e05cf089a53'),
    '3': ObjectId('689e35017dd56e05cf089a54'),
    '4': ObjectId('689e35017dd56e05cf089a55')
  }
}
bd_practica>

```

3. Insertar un alumno sin campo padre.

```
bd_practica> db.alumnos.insertOne({ nombres: "Juan", apellidos: "Fernández", sexo: "M",
    casado: false, fecha_nacimiento: new Date("2005-06-22"),
    materias: ["MAT101", "SIS256"] })
```

```
> db.alumnos.insertOne({ nombres: "Juan", apellidos: "Fernández", sexo: "M",
    casado: false, fecha_nacimiento: new Date("2005-06-22"),
    materias: ["MAT101", "SIS256"] })
< {
  acknowledged: true,
  insertedId: ObjectId('689e35837dd56e05cf089a57')
}
bd_practica>
```

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

```
bd_practica> db.alumnos.insertOne({ nombres: "Roberto", apellidos: "Mendoza", sexo: "M", casado: true, fecha_nacimiento: new Date("1990-04-10"), materias:
```

```
> db.alumnos.insertOne({ nombres: "Roberto", apellidos: "Mendoza", sexo: "M", casado: true, fecha_nacimiento: new Date("1990-04-10"), materias: ["SIS101", '
< {
  acknowledged: true,
  insertedId: ObjectId('689e35ac7dd56e05cf089a58')
}
bd_practica>
```

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

```
bd_practica> db.alumnos.insertOne({ nombres: "Javier", apellidos: "Pérez", sexo: "M", casado: false,
    fecha_nacimiento: new Date("2002-01-15"),
    materias: ["SIS101", "COM200", "MAT101", "SIS258"], padre: { nombre: "Carlos", apellido: "Pérez" } })
```

```
> db.alumnos.insertOne({ nombres: "Javier", apellidos: "Pérez", sexo: "M", casado: false,
    fecha_nacimiento: new Date("2002-01-15"),
    materias: ["SIS101", "COM200", "MAT101", "SIS258"], padre: { nombre: "Carlos", apellido: "Pérez" } })
< {
  acknowledged: true,
  insertedId: ObjectId('689e35db7dd56e05cf089a59')
}
bd_practica>
```

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

```
> db.alumnos.insertOne({ nombres: "Felipe", apellidos: "Gómez", sexo: "M", casado: false,
  fecha_nacimiento: new Date("2001-08-30"),
  materias: ["COM350"], padre: { nombre: "Andrés", apellido: "Gómez" } })
< {
  acknowledged: true,
  insertedId: ObjectId('689e36017dd56e05cf089a5a')
}
bd_practica>
```

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

```
bd_practica> db.alumnos.find(
  { "padre.nombre": { $in: ["Luis", "Miguel"] } },
  { _id: 0, nombres: 1, apellidos: 1, padre: 1 }
)
```

```

> db.alumnos.find(
  { "padre.nombre": { $in: ["Luis", "Miguel"] } },
  { _id: 0, nombres: 1, apellidos: 1, padre: 1 }
)
< {
  nombres: 'Pedro',
  apellidos: 'García Rodríguez',
  padre: {
    nombre: 'Miguel',
    apellido: 'Ramírez'
  }
}
{
  nombres: 'Hugo',
  apellidos: 'Ramírez Flores',
  padre: {
    nombre: 'Miguel',
    apellido: 'Rodríguez'
  }
}
{
  nombres: 'Carlos',
  apellidos: 'López Vargas',
  padre: {
    nombre: 'Luis',
    apellido: 'Torres'
  }
}
{

```

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

```

> db.alumnos.find(
  { casado: false, materias: "SIS256" },
  { _id: 0, nombres: 1, apellidos: 1, materias: 1 }
)
< {
  nombres: 'Jorge',
  apellidos: 'Torres García',
  materias: [
    'SIS256',
    'COM350'
  ]
}
{
  nombres: 'Ana',
  apellidos: 'Rodríguez Flores',
  materias: [
    'SIS256'
  ]
}
{
  nombres: 'Andrés',
  apellidos: 'López Flores',
  materias: [
    'SIS256'
  ]
}
{
  nombres: 'Luis',
  apellidos: 'Torres',

```

9. Contar alumnos con más de 2 materias.

```
> db.alumnos.countDocuments({ "materias.2": { $exists: true } })  
< 10  
bd_practica> |
```

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


```
> db.alumnos.find(
  {},
  { _id: 0, nombres: 1, fecha_nacimiento: 1 }
).sort({ fecha_nacimiento: 1 }).limit(10)
< {
  nombres: 'Roberto',
  fecha_nacimiento: 1990-04-10T00:00:00.000Z
}
{
  nombres: 'Elena',
  fecha_nacimiento: 1995-02-27T18:22:40.689Z
}
{
  nombres: 'Luis',
  fecha_nacimiento: 1995-06-08T19:42:39.130Z
}
{
  nombres: 'Camila',
  fecha_nacimiento: 1995-07-04T08:41:36.603Z
}
{
  nombres: 'María',
  fecha_nacimiento: 1995-09-13T03:46:46.316Z
}
{
  nombres: 'Mario',
  fecha_nacimiento: 1996-03-18T03:55:59.471Z
}
```

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

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

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

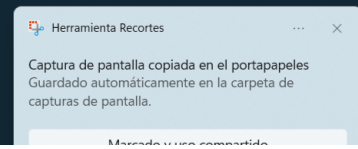
```
> db.alumnos.find(
  { sexo: "F" },
  { _id: 0, nombres: 1, sexo: 1 }
)
< {
  nombres: 'Camila',
  sexo: 'F'
}
{
  nombres: 'Elena',
  sexo: 'F'
}
{
  nombres: 'Gabriela',
  sexo: 'F'
}
{
  nombres: 'Ana',
  sexo: 'F'
}
{
  nombres: 'Elena',
  sexo: 'F'
}
{
  nombres: 'Sofía',
  sexo: 'F'
}
```

13. Ordenar todos los alumnos por apellido ascendente.

```
>_MONGOSH
> db.alumnos.find({}, { _id: 0, nombres: 1, apellidos: 1 }).sort({ apellidos: 1 })
< {
  nombres: 'Diego',
  apellidos: 'Fernández'
}
{
  nombres: 'Juan',
  apellidos: 'Fernández'
}
{
  nombres: 'Jorge',
  apellidos: 'Flores García'
}
{
  nombres: 'Paula',
  apellidos: 'Flores Pérez'
}
{
  nombres: 'Carlos',
  apellidos: 'Flores Pérez'
}
{
  nombres: 'Pedro',
  apellidos: 'García Rodríguez'
}
{
  nombres: 'Pedro',
  apellidos: 'Gutiérrez'
}
```

14. Listar alumnos nacidos entre 1998 y 2003.

```
> db.alumnos.find({ fecha_nacimiento: { $gte: new Date("1998-01-01"), $lte: new Date("2003-12-31") } }, { _id: 0, nombres: 1, apellidos: 1, fecha_nacimiento: 1 })
< {
  nombres: 'Elena',
  apellidos: 'Martínez Ramírez',
  fecha_nacimiento: 2002-08-14T13:42:34.786Z
}
{
  nombres: 'Pedro',
  apellidos: 'García Rodríguez',
  fecha_nacimiento: 2003-07-26T13:25:22.675Z
}
{
  nombres: 'Juan',
  apellidos: 'Ramírez Vargas',
  fecha_nacimiento: 1998-11-05T17:08:20.145Z
}
{
  nombres: 'Sofía',
  apellidos: 'Ramírez Ramírez',
  fecha_nacimiento: 1999-02-20T09:02:10.434Z
}
{
  nombres: 'Jorge',
  apellidos: 'Flores García',
  fecha_nacimiento: 2000-09-25T23:58:16.771Z
}
```



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

```

> db.alumnos.find({ apellidos: /ez$/ }, { _id: 0, nombres: 1, apellidos: 1 })
< {
  nombres: 'Pedro',
  apellidos: 'Vargas Sánchez'
}
{
  nombres: 'Camila',
  apellidos: 'Ramírez López'
}
{
  nombres: 'Luis',
  apellidos: 'Torres Sánchez'
}
{
  nombres: 'Andrés',
  apellidos: 'Martínez Pérez'
}
{
  nombres: 'Elena',
  apellidos: 'Martínez Ramírez'
}
{
  nombres: 'Pedro',
  apellidos: 'García Rodríguez'
}
{
  nombres: 'Elena',
  apellidos: 'Vargas Martínez'
}

```

16. Contar alumnos casados y con "SIS258".

```

> db.alumnos.countDocuments({ casado: true, materias: "SIS258" })
< 19
bd_practica> |

```

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

```
> db.alumnos.updateMany({ nombres: "Luis" }, { $set: { nombres: "Luis Alberto" } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
bd_practica>
```

```
> db.alumnos.find({ nombres: "Luis Alberto" }, { _id: 0, nombres: 1, apellidos: 1 })
< {
  nombres: 'Luis Alberto',
  apellidos: 'Torres Sánchez'
}
{
  nombres: 'Luis Alberto',
  apellidos: 'Pérez López'
}
{
  nombres: 'Luis Alberto',
  apellidos: 'Torres'
}
bd_practica>
```

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

```
> db.alumnos.updateMany({ materias: "COM350" }, { $addToSet: { materias: "SIS777" } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 27,
  modifiedCount: 27,
  upsertedCount: 0
}
bd_practica>
```

```

> db.alumnos.find({ materias: "SIS777" }, { _id: 0, nombres: 1, apellidos: 1, materias: 1 })
< {
  nombres: 'Pedro',
  apellidos: 'Vargas Sánchez',
  materias: [
    'COM350',
    'SIS256',
    'SIS777'
  ]
}
{
  nombres: 'Andrés',
  apellidos: 'Martínez Pérez',
  materias: [
    'COM350',
    'SIS258',
    'SIS777'
  ]
}
{
  nombres: 'Jorge',
  apellidos: 'Torres García',
  materias: [
    'SIS256',
    'COM350',
    'SIS777'
  ]
}

```

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

```

> db.alumnos.updateMany({ "materias.1": { $exists: false } }, { $set: { casado: false } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 13,
  modifiedCount: 9,
  upsertedCount: 0
}
bd_practica>

```



```

> db.alumnos.find(
  { "materias.1": { $exists: false } },
  { _id: 0, nombres: 1, apellidos: 1, materias: 1, casado: 1 }
)
< {
  nombres: 'Camila',
  apellidos: 'Ramírez López',
  casado: false,
  materias: [
    'SIS258'
  ]
}
{
  nombres: 'Luis Alberto',
  apellidos: 'Torres Sánchez',
  casado: false,
  materias: [
    'SIS256'
  ]
}
{
  nombres: 'Ana',
  apellidos: 'Rodríguez Flores',
  casado: false,
  materias: [
    'SIS256'
  ]
}

```

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

```
> db.alumnos.updateMany({ materias: "SIS256" }, { $pull: { materias: "SIS256" } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 26,
  modifiedCount: 26,
  upsertedCount: 0
}
bd_practica>|
```

```
> db.alumnos.find({ materias: "SIS256" }, { _id: 0, nombres: 1, apellidos: 1, materias: 1 })
<
bd_practica>|
```

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

```
> db.alumnos.updateMany({ "padre.nombre": "Juan" }, { $set: { "padre.nombre": "Juan José" } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
bd_practica>
```

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

```
> db.alumnos.updateMany({}, { $set: { activo: true } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 48,
  modifiedCount: 48,
  upsertedCount: 0
}
```

```

> db.alumnos.find()
< {
  _id: ObjectId('689dee388be1607eabeec4ca'),
  nombres: 'Pedro',
  apellidos: 'Vargas Sánchez',
  sexo: 'M',
  casado: true,
  fecha_nacimiento: 1997-02-02T12:54:32.988Z,
  materias: [
    'COM350',
    'SIS777'
  ],
  padre: {
    nombre: 'Pedro',
    apellido: 'Rodríguez'
  },
  activo: true
}
{
  _id: ObjectId('689dee388be1607eabeec4d0'),
  nombres: 'Camila',
  apellidos: 'Ramírez López',
  sexo: 'F',
  casado: false,
  fecha_nacimiento: 1995-07-04T08:41:36.603Z,
  materias: [
    'SIS258'
  ],

```

23. Eliminar todos los alumnos hombres.

```

> db.alumnos.deleteMany({ sexo: "M" })
< {
  acknowledged: true,
  deletedCount: 32
}
> db.alumnos.find({ sexo: "M" }, { _id: 0, nombres: 1, apellidos: 1, sexo: 1 })
<
bd_practica> |

```

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

```

> db.alumnos.deleteMany({ materias: "SIS777" })
< {
  acknowledged: true,
  deletedCount: 10
}
> db.alumnos.find({ materias: "SIS777" }, { _id: 0, nombres: 1, apellidos: 1, materias: 1 })
<
bd_practica> |

```

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

```

> db.alumnos.deleteMany({ "padre.apellido": "Ramírez" })
< {
  acknowledged: true,
  deletedCount: 0
}
> db.alumnos.find({ "padre.apellido": "Ramírez" }, { _id: 0, nombres: 1, apellidos: 1, padre: 1 })
<
bd_practica> |

```

26. Eliminar a los que nacieron después de 2004.

```

> db.alumnos.deleteMany({ fecha_nacimiento: { $gt: new Date("2004-12-31") } })
< {
  acknowledged: true,
  deletedCount: 0
}
> db.alumnos.find({ fecha_nacimiento: { $gt: new Date("2004-12-31") } }, { _id: 0, nombres: 1, apellidos: 1, fecha_nacimiento: 1 })
<
bd_practica> |

```

27. Eliminar todos los casados.

```

> db.alumnos.deleteMany({ casado: true })
< {
  acknowledged: true,
  deletedCount: 1
}
> db.alumnos.find({ casado: true }, { _id: 0, nombres: 1, apellidos: 1, casado: 1 })
<
bd_practica>

```

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

```

> db.alumnos.find({ casado: true }, { _id: 0, nombres: 1, apellidos: 1, casado: 1 })
<
> db.alumnos.deleteOne({ materias: "SIS258" })
< {
  acknowledged: true,
  deletedCount: 1
}
> db.alumnos.find({ materias: "SIS258" }, { _id: 0, nombres: 1, apellidos: 1, materias: 1 })
< {
  nombres: 'María',
  apellidos: 'Torres Flores',
  materias: [
    'SIS258'
  ]
}
{
  nombres: 'Lucía',
  apellidos: 'Martínez',
  materias: [
    'MAT101',
    'SIS258'
  ]
}
bd_practica>

```