



Introducción a las bases de datos NoSQL

2024

CONTENIDO

1. ¿Qué es una base de datos?
2. Breve introducción a Json
3. Introducción a las bases de datos NoSQL
4. Tipos de datos en MongoDB
5. Cierre y conclusión



¿Qué es una base de datos?

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS).

```
1 {
2   "users": [
3     {
4       "userId": 1,
5       "firstName": "Chris",
6       "lastName": "Lee",
7       "phoneNumber": "555-555-5555",
8       "emailAddress": "clee@fileinfo.com"
9     },
10    {
11      "userId": 2,
12      "firstName": "Action",
13      "lastName": "Jackson",
14      "phoneNumber": "555-555-5556",
15      "emailAddress": "ajackson@fileinfo.com"
16    },
17    {
18      "userId": 3,
19      "firstName": "Ross",
20      "lastName": "Bing",
21      "phoneNumber": "555-555-5557",
22      "emailAddress": "rbing@fileinfo.com"
23    },
24    {
25      "userId": 4,
26      "firstName": "David",
27      "lastName": "Reeves",
28      "phoneNumber": "555-555-5558",
29      "emailAddress": "dreeves@fileinfo.com"
30    },
31    {
32      "userId": 5,
33      "firstName": "Josie",
34      "lastName": "Mac",
35      "phoneNumber": "555-555-5559",
36      "emailAddress": "jmac@fileinfo.com"
37    }
38  ]
39 }
```

Introducción a Json

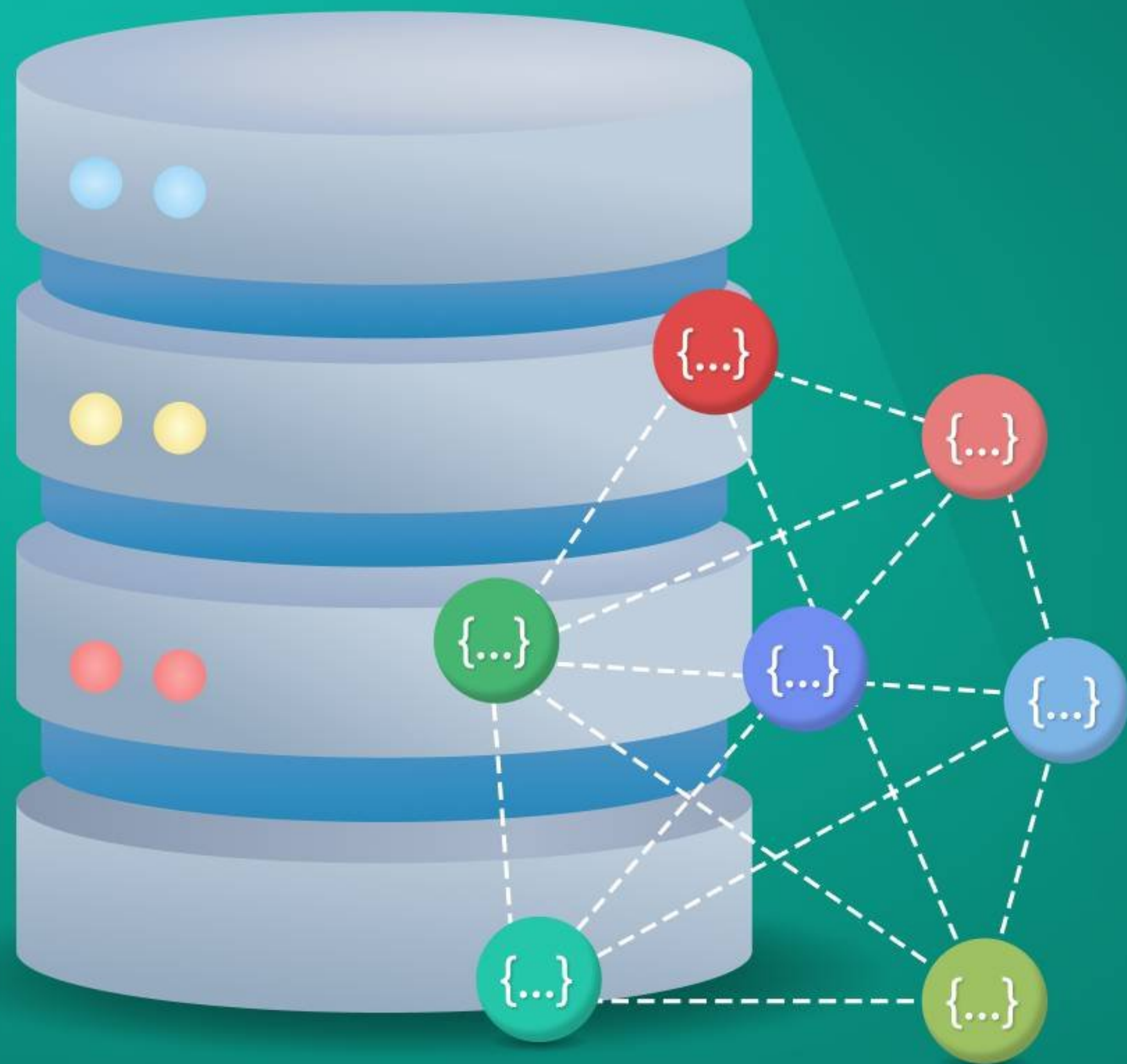
JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

```
{
  "libro": [
    {
      "id": "01",
      "lenguaje": "Java",
      "edición": "tercera",
      "autor": "Herbert Schildt"
    },
    {
      "id": "07",
      "lenguaje": "C++",
      "edición": "segunda",
      "autor": "E.Balagurusamy"
    }
  ]
}
```

Platzl

```
{
  "datos": {
    "title": "datos",
    "description": "Info SensorTemperatura",
    "type": "object",
    "properties": {
      "identificador": {
        "type": "string",
        "required": true
      },
      "timestamp": {
        "type": "integer",
        "minimum": 0,
        "required": true
      },
      "temperatura": {
        "type": "number",
        "required": true
      },
      "coordenadaGps": {
        "required": true,
        "$ref": "#/gps"
      }
    }
  },
  "Mannock": {
```

```
{
  "clients": [
    {
      "first_name": "Sigrid",
      "last_name": "Mannock",
      "age": 27,
      "amount": 7.17
    },
    {
      "first_name": "Joe",
      "last_name": "Hinnners",
      "age": 31,
      "amount": [
        1.9,
        5.5
      ]
    },
    {
      "first_name": "Theodoric",
      "last_name": "Rivers",
      "age": 36,
      "amount": 1.11
    }
  ]
}
```

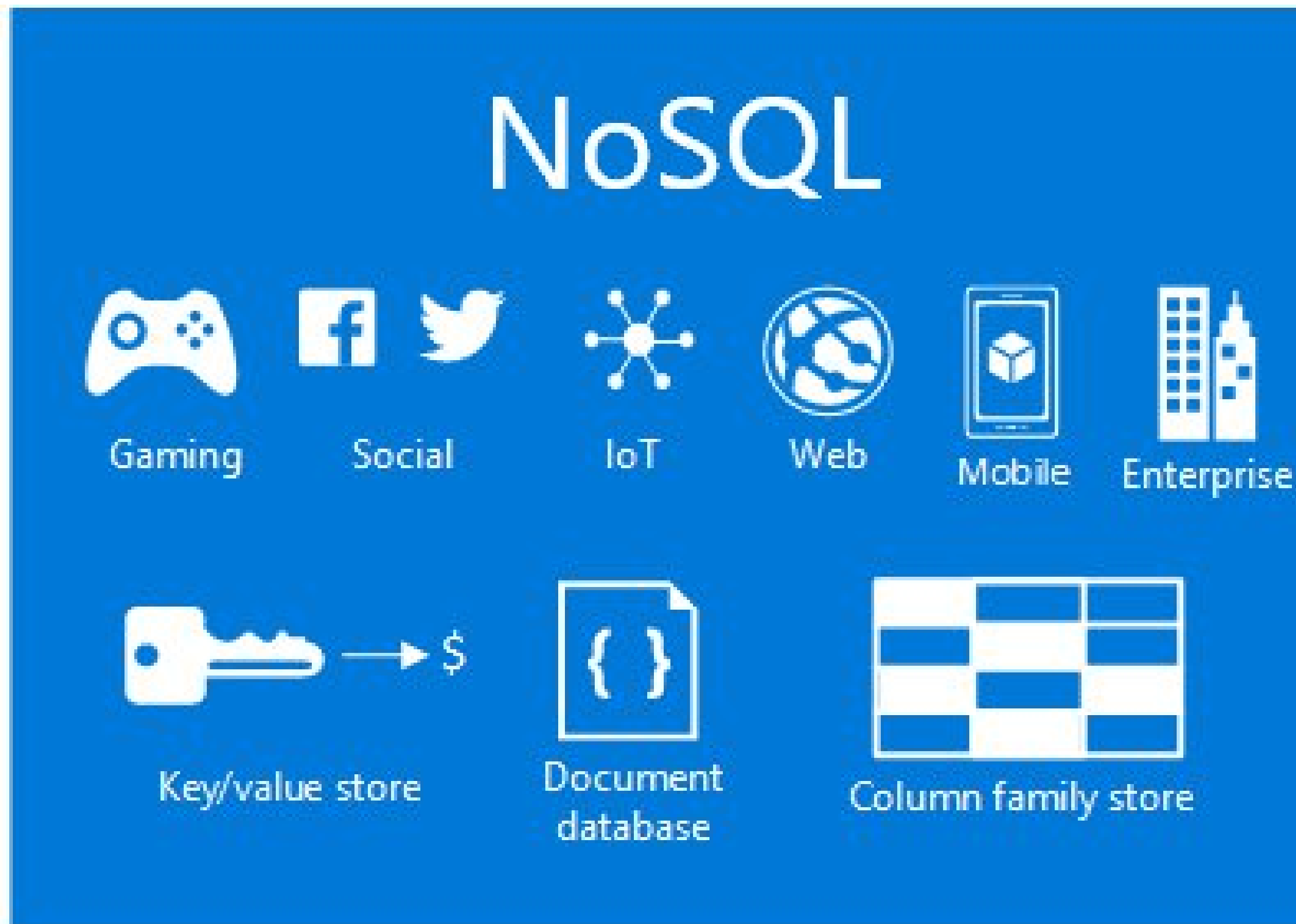



Introducción a las bases de datos NoSQL

¿Qué son las bases de datos NoSQL?

Las bases de datos NoSQL son sistemas de gestión de bases de datos que difieren significativamente de los sistemas tradicionales relacionales (SQL). Se caracterizan por su capacidad para manejar grandes volúmenes de datos no estructurados y semiestructurados.

Importancia de las bases de datos NoSQL



Las bases de datos NoSQL son utilizadas en una variedad de aplicaciones, como redes sociales, análisis de big data, IoT (Internet de las cosas), aplicaciones web y móviles, entre otros.



Ventajas y Desventajas de NoSQL

1. Escalabilidad Horizontal
2. Flexibilidad de esquema
3. Rendimiento optimizado para cargas específicas
4. Gestión de Big Data
5. Adaptabilidad a las tendencias actuales

1. Falta de estandarización
2. Madurez relativa
3. Consistencia vs Disponibilidad
4. Menor énfasis en la integridad de datos
5. Curva de aprendizaje

Colecciones de datos y Documentos

¿Cómo funciona?

MongoDB

SQL

Base de Datos

Base de Datos

Colecciones

Tablas

Documentos

Filas (registros)

Campos

Columnas

Colecciones de datos y Documentos

DATA TYPES



STRING

name: String

```
{  
  name: "John"  
}
```



NUMBER

likes: Number

```
{  
  likes: 5  
}
```



DATE

timeStamp: Date

```
{  
  timeStamp: ISODate("...")  
}
```



ARRAY

tags: Array

OR

```
tags: []  
{  
  tags: ["tag1", "tag2"]  
}
```



BOOLEAN

published: Boolean

```
{  
  published: true  
}
```

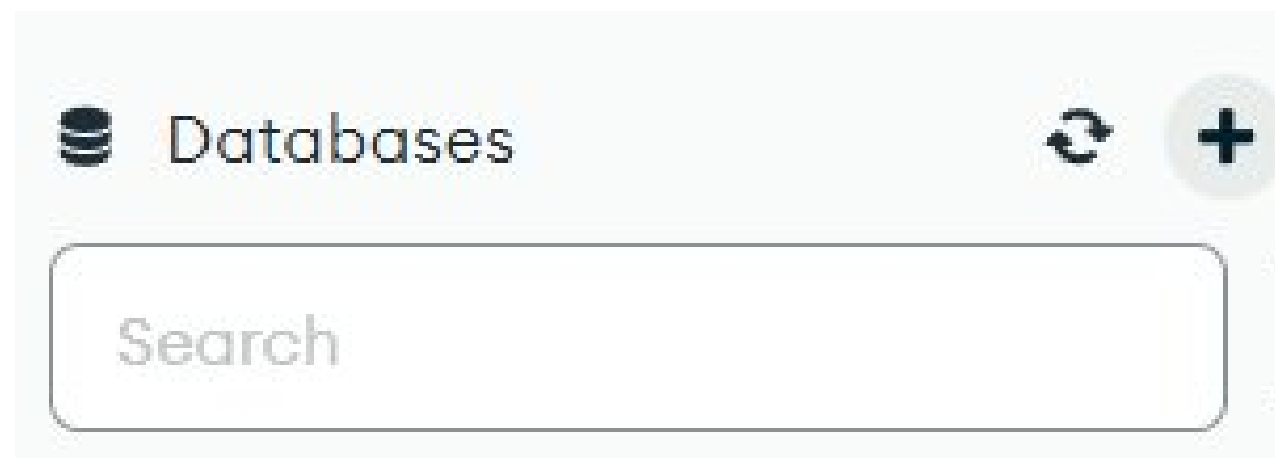


ObjectId

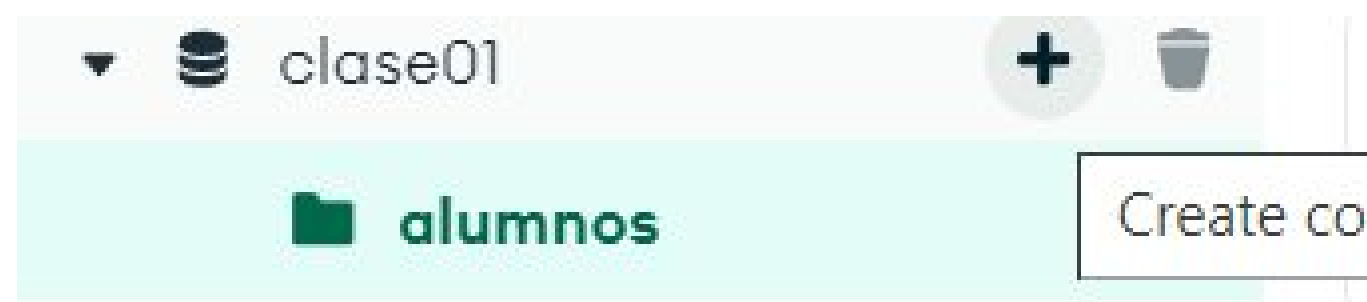
_creator: Schema.ObjectId

```
{  
  _creator: "41239878"  
}
```

- 1. Crear una Base de Datos y una Colección
- Abre MongoDB Compass y conéctate a tu instancia de MongoDB.
- Una vez conectado, puedes crear una nueva base de datos seleccionando "Create Database". Por ejemplo, puedes nombrar tu base de datos miPrimeraDB y tu colección usuarios.
- Haz clic en "Create Database" para finalizar.

A screenshot of the 'Create Database' dialog box. It has a title bar with a close button (x). The form contains two input fields: 'Database Name' and 'Collection Name'. Below these is a checkbox for 'Time-Series' with a description: 'Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)'. There is also an expandable section for 'Additional preferences (e.g. Custom collation, Capped, Clustered collections)'. A light blue information box at the bottom states: 'Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)'. At the bottom right are 'Cancel' and 'Create Database' buttons.

- Insertar Documentos en tu Colección
- Para insertar documentos en tu colección usuarios, sigue estos pasos:
- Navega a tu base de datos miPrimeraDB y luego a la colección usuarios.
- Selecciona la pestaña "Documents" y luego haz clic en "Insert Document".
- Aquí puedes insertar un documento en formato JSON. Por ejemplo:
- json
- {
- "nombre": "Juan",
- "edad": 30,
- "email": "juan@ejemplo.com"
- }



Create Collection

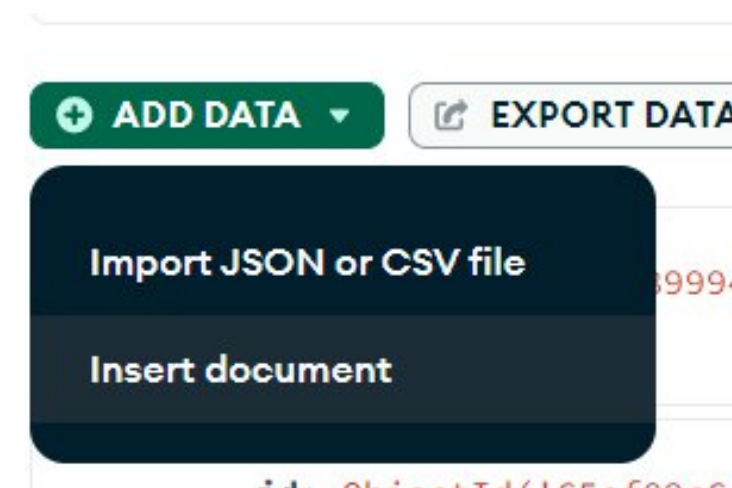
Collection Name

☐ Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Collection



Insert Document

To collection clase01.alumnos

VIEW

```


1 /**
2  * Paste one or more documents here
3  */
4  {
5    "_id": {
6      "$oid": "6601e3d4f7fb6c87fe4ef8f8"
7    }
8  }


```


Cancel


Insert


- 3. Leer Documentos
- Para ver los documentos en tu colección:
- Asegúrate de estar en la pestaña "Documents" de tu colección usuarios.
- Aquí puedes ver todos los documentos que has insertado. También puedes utilizar la barra de búsqueda para filtrar documentos por criterios específicos.

 { edad: 30 }

 ADD DATA ▾

 EXPORT DATA ▾

 UPDATE

 DELETE

```
_id: ObjectId('65ef89c6428b1168471bc91e')
nombre : "Juan"
edad : 30
```

```
_id: "mi_id_personalizado"
nombre : "Ejemplo"
apellido : "Apellido"
edad : 30
curso : "Programación"
```


- 4. Actualizar Documentos
- Para actualizar un documento existente:
 - Encuentra el documento que deseas actualizar en la lista de documentos.
 - Haz clic en el botón "Edit Document" de ese documento.
 - Modifica los campos que deseas actualizar. Por ejemplo, puedes cambiar la "edad" de Juan a 31.
 - Haz clic en "Update" para guardar los cambios.

```
nombre : "Juan"
edad : 30
```

[Edit document](#)

```
_id: "mi_id_personalizado"  
nombre : "Ejemplo"  
apellido : "Apellido"  
edad : 30  
curso : "Programación"
```

```
1  _id: "mi_id_personalizado"
2  nombre: "Ejemplo/"
3  +
4  apellido: "TEST/"
5  edad: 30
6  curso: "Programación/"
```

```
String
String
String
Int32
String
```

Document modified.

CANCEL

UPDATE

Operador	Descripción	Ejemplo de Uso
\$set	Modifica el valor de un campo o lo crea si no existe.	{ "\$set": { "correo": "nuevo.email@ejemplo.com" } }
\$unset	Elimina un campo de un documento.	{ "\$unset": { "telefono": "" } }
\$inc	Incrementa o decrementa el valor de un campo numérico.	{ "\$inc": { "edad": 1 } } (suma 1 a la edad)
\$mul	Multiplica el valor de un campo numérico.	{ "\$mul": { "precio": 1.1 } } (aumenta un 10%)
\$rename	Cambia el nombre de un campo.	{ "\$rename": { "direccion": "ubicacion" } }
\$min	Cambia el valor del campo solo si el nuevo valor es menor.	{ "\$min": { "precio": 500 } }
\$max	Cambia el valor del campo solo si el nuevo valor es mayor.	{ "\$max": { "nota": 7.0 } }
\$currentDate	Establece la fecha/hora actual en un campo tipo fecha.	{ "\$currentDate": { "ultima_actualizacion": true } }
\$push	Agrega un elemento a un array.	{ "\$push": { "asignaturas": "Big Data" } }
\$pull	Elimina un elemento específico de un array.	{ "\$pull": { "asignaturas": "Matemáticas" } }
\$addToSet	Agrega un elemento a un array solo si no existe .	{ "\$addToSet": { "asignaturas": "Python" } }
\$pop	Elimina el primer (-1) o el último (1) elemento de un array.	{ "\$pop": { "asignaturas": -1 } } (elimina el primero)
\$each	Agrega múltiples valores a un array con \$push.	{ "\$push": { "asignaturas": { "\$each": ["IA", "Machine Learning"] } } }
\$pullAll	Elimina múltiples valores específicos de un array.	{ "\$pullAll": { "asignaturas": ["Java", "C++"] } }

- 5. Eliminar Documentos
- Para eliminar un documento:
 - Encuentra el documento que deseas eliminar en la lista de documentos.
 - Haz clic en el botón "Delete Document" para ese documento.
 - Confirma la eliminación.

nombre : "Juan"
edad : 30

Remove document



_id: "mi_id_personalizado"
nombre : "Ejemplo"
apellido : "TEST"
edad : 30
curso : "Programación"



• Cargar archivo

Cómo Cargarlo en MongoDB Compass

- Abre MongoDB Compass y selecciona la base de datos donde quieres cargar los datos.
- Crea una nueva colección (por ejemplo, productos).
- Haz clic en "Import Data" y selecciona el archivo CSV.
- Configura los campos correctamente:
 - _id como Integer.
 - precio y stock como Integer.
 - ventas como Array (puede requerir ajuste manual si no se parsea automáticamente).
- Haz clic en "Import" y verifica los datos.

Import

To collection Incap_2025.Productos
Specify Fields and Types [Learn more about data types](#)

<input checked="" type="checkbox"/> _id	<input checked="" type="checkbox"/> nombre	<input checked="" type="checkbox"/> categoria	<input checked="" type="checkbox"/> precio	<input checked="" type="checkbox"/> stock	<input checked="" type="checkbox"/> ventas
Int32	String	String	Int32	Int32	EJSON
1	Laptop Dell XPS	Electrónica	1500	5	[{"fecha": "2023-01-01"}]
2	iPhone 13 Pro	Electrónica	1200	10	[{"fecha": "2023-01-01"}]
3	Cafetera Nespresso	Hogar	250	15	[{"fecha": "2023-01-01"}]
4	TV Samsung 55"	Electrónica	800	7	[{"fecha": "2023-01-01"}]
5	Zapatillas Nike Air	Moda	100	20	[{"fecha": "2023-01-01"}]
6	Teclado Mecánico	Accesorios	80	12	[{"fecha": "2023-01-01"}]
7	Mouse Logitech MX	Accesorios	70	18	[{"fecha": "2023-01-01"}]
8	Monitor LG 24"	Electrónica	300	8	[{"fecha": "2023-01-01"}]
9	Silla Ergonómica	Oficina	200	5	[{"fecha": "2023-01-01"}]
10	Escritorio de Madera	Hogar	350	3	[{"fecha": "2023-01-01"}]

Cancel Import

Ejercicios con la colección alumnos

● Ejercicio 1: Filtrar alumnos de una carrera específica

□ Encuentra todos los alumnos que estudian Ingeniería Informática.

Ejercicios con la colección alumnos

● Ejercicio 1: Filtrar alumnos de una carrera específica

□ Encuentra todos los alumnos que estudian Ingeniería Informática.

```
{ "carrera": "Ingeniería Informática" }
```

Ejercicios con la colección alumnos

● Ejercicio 2: Buscar alumnos mayores de 21 años

□ Encuentra a los alumnos cuya edad sea mayor a 21 años.

Ejercicios con la colección alumnos

● Ejercicio 2: Buscar alumnos mayores de 21 años

□ Encuentra a los alumnos cuya edad sea mayor a 21 años.

```
{ "edad": { "$gt": 21 } }
```

Ejercicios con la colección alumnos

- Ejercicio 3: Buscar alumnos que tengan una nota superior a 6.0 en alguna asignatura
- Encuentra alumnos que tengan al menos una asignatura con una nota mayor a 6.0.

Ejercicios con la colección alumnos

- Ejercicio 3: Buscar alumnos que tengan una nota superior a 6.0 en alguna asignatura
- Encuentra alumnos que tengan al menos una asignatura con una nota mayor a 6.0.

```
{ "notas.nota": { "$gt": 6.0 } }
```


Ejercicios con la colección alumnos

● Ejercicio 4: Actualizar el correo de un alumno

□ El alumno Juan Pérez ha cambiado su correo. Ahora su nuevo correo es `juan.perez2024@ejemplo.com`.

Ejercicios con la colección alumnos

● Ejercicio 4: Actualizar el correo de un alumno

□ El alumno Juan Pérez ha cambiado su correo. Ahora su nuevo correo es `juan.perez2024@ejemplo.com`.

```
{ "nombre": "Juan", "apellido": "Pérez" }
```

```
{ "$set": { "correo": "juan.perez2024@ejemplo.com" } }
```

Ejercicios con la colección alumnos

- Ejercicio 5: Eliminar a un alumno que se retiró
 - Elimina al alumno llamado Carlos Rojas de la base de datos.

Ejercicios con la colección alumnos

- Ejercicio 5: Eliminar a un alumno que se retiró
- Elimina al alumno llamado Carlos Rojas de la base de datos.

```
{ "nombre": "Carlos", "apellido": "Rojas" }
```

Luego, haz clic en "Delete".

Ejercicios con la colección docentes

- Ejercicio 6: Filtrar docentes del departamento de Informática
- Encuentra todos los docentes que pertenecen al departamento de Informática.

Ejercicios con la colección docentes

- Ejercicio 6: Filtrar docentes del departamento de Informática
- Encuentra todos los docentes que pertenecen al departamento de Informática.

```
{ "departamento": "Informática" }
```

Ejercicios con la colección docentes

- **Ejercicio 7: Buscar docentes que enseñan "Bases de Datos"**
- Encuentra todos los docentes que imparten la asignatura de **"Bases de Datos"**.

Ejercicios con la colección docentes

- **Ejercicio 7: Buscar docentes que enseñan "Bases de Datos"**
- Encuentra todos los docentes que imparten la asignatura de **"Bases de Datos"**.

```
{ "asignaturas": "Bases de Datos" }
```

Ejercicios con la colección docentes

- Ejercicio 8: Aumentar la edad de un docente
- El docente Roberto Morales ha cumplido años. Debemos aumentar su edad en 1.

Ejercicios con la colección docentes

- Ejercicio 8: Aumentar la edad de un docente
- El docente Roberto Morales ha cumplido años. Debemos aumentar su edad en 1.

```
{ "nombre": "Roberto", "apellido": "Morales" }
```

```
{ "$inc": { "edad": 1 } }
```

Ejercicios con la colección docentes

- Ejercicio 9: Agregar una nueva asignatura a un docente
- El docente Lucía Torres ahora impartirá la asignatura de "Cloud Computing". Debemos agregar esta nueva asignatura.

Ejercicios con la colección docentes

- Ejercicio 9: Agregar una nueva asignatura a un docente
- El docente Lucía Torres ahora impartirá la asignatura de "Cloud Computing". Debemos agregar esta nueva asignatura.

```
{ "nombre": "Lucía", "apellido": "Torres" }
```

```
{ "$push": { "asignaturas": "Cloud Computing" } }
```

Ejercicios con la colección docentes

- Ejercicio 10: Eliminar un docente que dejó de trabajar en la institución
- El docente Manuel Gutiérrez ha dejado la institución. Debemos eliminarlo de la base de datos.

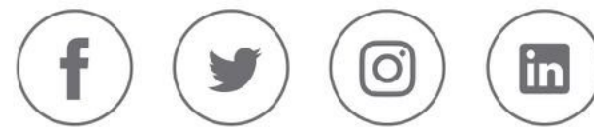
Ejercicios con la colección docentes

- Ejercicio 10: Eliminar un docente que dejó de trabajar en la institución
- El docente Manuel Gutiérrez ha dejado la institución. Debemos eliminarlo de la base de datos.

```
{ "nombre": "Manuel", "apellido": "Gutiérrez" }
```

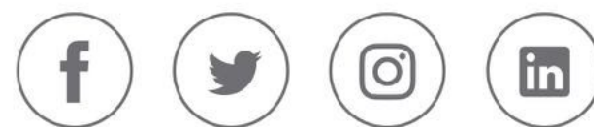
Luego, haz clic en "Delete".

MUCHAS GRACIAS!



inacap.cl

UNIVERSIDAD TECNOLÓGICA DE CHILE
INSTITUTO PROFESIONAL
CENTRO DE FORMACIÓN TÉCNICA



inacap.cl