

Bases de datos (DB)

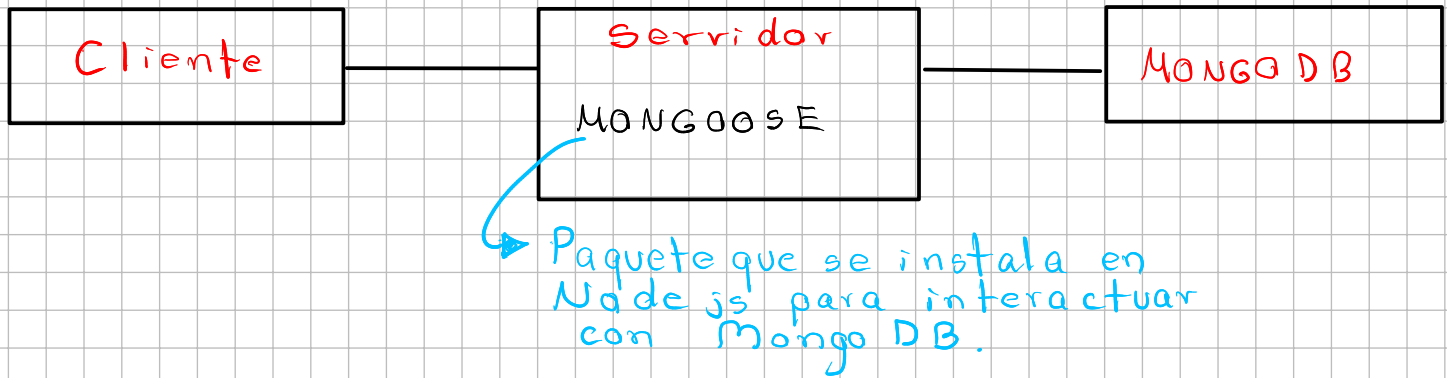
• Bases de datos Relacionales (SQL)

- Se empezaron a usar en los años 60's. Las no-relacionales surgieron recientemente.
- Colección de elementos de datos organizados en un conjunto de tablas formalmente descritas, desde donde se pueden acceder los datos.
- La interacción con los datos se hace mediante el **Lenguaje de Consultas Estructuradas (SQL)**.
 - ↳ Comandos SQL para manipular datos
- Organizan la información en partes pequeñas que se integran mediante identificadores **Keys**. A diferencia de las bases de datos no-relacionales que no tienen un identificador que sirva para relacionar dos o más conjuntos de datos.
- Tienen mayor capacidad de almacenamiento.
- Menos vulnerables ante fallas.

• Bases de datos No Relacionales (NoSQL)

- Se diseñan para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas.
- Fáciles de desarrollar.
- La información se organiza mediante documentos y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar.
- Tienen un esquema dinámico: no se requiere la estructura de los datos para su manipulación.

- MongoDB → Se recomienda cuando se está usando JS.
¿Cómo Funciona?



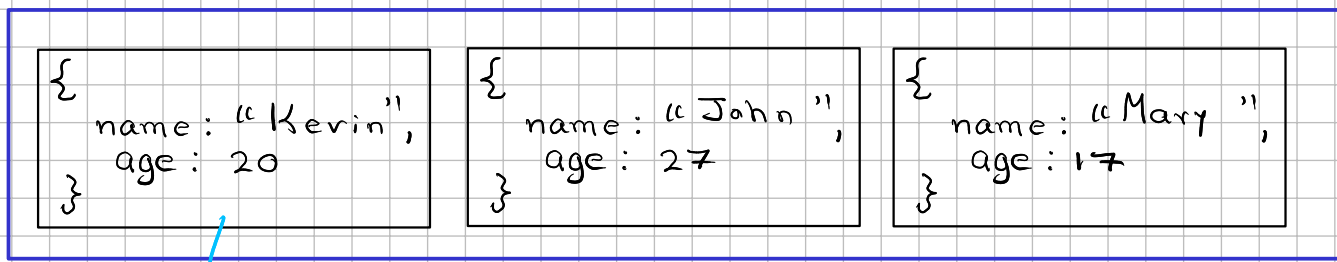
- Pasos para ejecutar MongoDB (Windows)

- ① Crear la carpeta "data/db"
mongo no sabe dónde alojar los datos.
- ② Agregar mongo al path

Teoría de MongoDB

- Collections → Es la manera en que se organizan los datos.

Ejemplo: Users Collection



Document (JSON Object)

• Comandos de Mongo

- db: mostrar la base de datos actual
- show dbs: listado de las bases de datos
- use <db name>: crear una base de datos ó usar una ya creada
- db.createCollection("name")
- db.<collection>.insert({JSON})
 .insert([{JSON}, {JSON}])
- db.<collection>.drop(): eliminar una colección
- db.<collection>.find().pretty()
 .find({Target})
 .find({Target}, {Filter})
- show collections
- db.<collection>.update({Key}, {new values JSON})
 .update({Key}, { \$set: { "att": new value } })