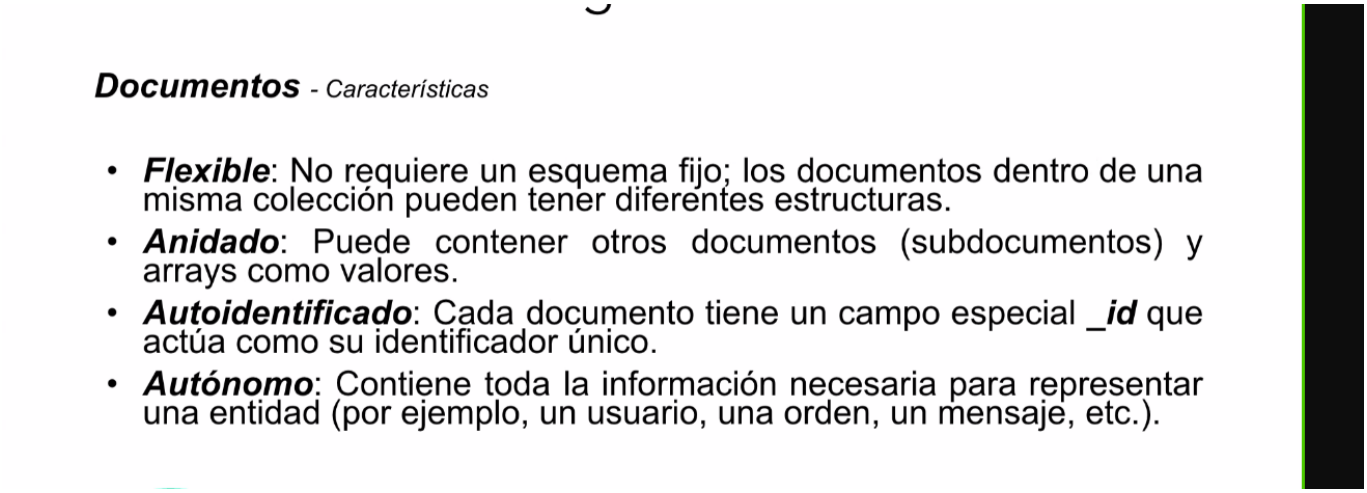
Vamos a estar trabajando con la persistencia de datos con mongo DB

Persistencia

- La persistencia se refiere a que el estado d eun Sistema sobrevive más allá de una única ejecución
- Esto se logra, en la práctica, almacenando dicho estado en archivos o en una Base de datos

Documentos de MONGODB



Operaciones CRUD básicas

Insert	`db.usuarios.insertOne({ nombre: "Carlos", edad: 28 });`

Find	`db.usuarios.findOne({ _id: new ObjectId("507f1f77bcf86cd799439011") });`
	`db.usuarios.find();`
	`db.usuarios.find({ edad: { $gte: 25 } });`
	`db.usuarios.find({ edad: { $gte: 18 }, pais: "AR" });`
	`db.usuarios.find({ $and: [ { edad: { $gte: 18 } },{ pais: "AR" } ] });`
	`db.usuarios.find({ $or: [ { edad: { $lt: 18 } }, { pais: "AR" } ] });`
	`db.usuarios.find().sort({ edad: -1 }).limit(5);`

Update	`db.usuarios.updateOne({ nombre: "Carlos" }, { $set: { edad: 29 } });`
Delete	`db.usuarios.deleteOne({ nombre: "Carlos" });`

$\color{lightblue} \text {eq}$: Igual a un valor. { edad: { $eq: 25}}

$\color{lightblue} \text {ne}$: Distinto de un valor. { edad: {$ne: 18}}

$\color{lightblue} \text {gt}$: Mayor que. { edad: {$gt: 18}}

$\color{lightblue} \text {gte}$: Mayor o igual que. { edad: { $gte: 18}}

$\color{lightblue} \text {lt}$: Menor que. {edad: {$lt: 65}}

$\color{lightblue} \text {lte}$: Menor o igual que. {edad: { $lte: 30}}

$\color{lightblue} \text {in}$: Está en una lista de valores. {pais: {$in: ["AR", "BR", "UY"] }}

$\color{lightblue} \text {nin}$: No está en una lista de valores. {pais: { $nin: ["US", "UK"] } }

$\color{lightblue} \text {exists}$: Verifica si el campo existe o no. { email: { $exists: true } }

$\color{lightblue} \text {type}$: Filtra por tipo de dato BSON. { edad: { $type: "int"}}

$\color{lightblue} \text {regex}$: Coincidencia con expresión regular. { nombre: { $regex: "^Lu", $options: "i" } }

ODM - Mongoose

- Un ODM (Object Document Mapper) es una herramienta que permite :
  - Mapear objetos JavaScript a documentos MongoDB.
  - Usar una interfaz orientada a objetos para interactuar con la base
  - Validar, estructurar y manipular los datos más fácilmente

Por qué usar ODM ?

- Define esquema de datos
- Maneja relaciones yu validaciones
- Evita escribir queries Mongo "a mano" todo el tiempo
- Permite usar métodos personalizados, middleware, etc.

Mongoose

```
npm install mongoose --save
```