

Microproyecto:

Inventario de piezas de
electrodomésticos
Versión MongoDB

Carlos Collado Calvo

24/05/2024

1. INTRODUCCIÓN	3
2. HERRAMIENTAS Y MÉTODOS	4
3. PERSPECTIVA ESTÁTICA	5
3.1. E/R (Entidad-Relación)	5
3.2. Pasos a tablas	6
3.3. DDL (Data Definition Language)	7
3.4. DML (Data Manipulation Language)	8
3.5. DQL (Data Query Language)	9
3.6. DCL (Data Control Language)	10
4. PERSPECTIVA DINÁMICA	11
5. CONCLUSIONES	12
6. ENLACE A REPOSITORIO	13

1. INTRODUCCIÓN

Tras haber creado una muestra de una base de datos a través de MySQL, que sería considerado el punto de almacenamiento y administración para la aplicación de inventario que se creó en el anterior proyecto, ahora deberemos intentar alimentarla de con una base de datos diferente, no relacional, es decir, que no esté basada en tablas.

Para ello se utilizará MongoDB un sistema de base de datos NoSQL, orientado a documentos, de código abierto y que utiliza un lenguaje de consulta propio (MQL). Las principales diferencias entre MySQL y MongoDB son las siguientes:

	MONGODB	MYSQL
MODELO DE DATOS	MongoDB almacena los datos en documentos JSON y, a continuación, los organiza en colecciones.	MySQL almacena los datos en columnas y filas. El almacenamiento de datos es tabular y relacional.
ESCALABILIDAD	MongoDB usa la replicación y la participación para escalar horizontalmente.	MySQL usa el escalado vertical y lee réplicas para mejorar el rendimiento a escala.
LENGUAJE DE CONSULTA	MongoDB usa el lenguaje de consulta MQL.	MySQL usa el lenguaje de consulta SQL.
RENDIMIENTO	MongoDB destaca en la inserción o actualización de una gran cantidad de registros.	MySQL es más rápido cuando se selecciona una gran cantidad de registros.
FLEXIBILIDAD	MongoDB no tiene un esquema, lo que proporciona más flexibilidad y le permite trabajar con datos estructurados, semiestructurados y no estructurados.	MySQL tiene un esquema rígido que funciona bien con datos estructurados.
SEGURIDAD	MongoDB usa los certificados Kerberos, X.509 y LDAP para autenticar a los usuarios.	MySQL usa métodos de autenticación integrados.

2. HERRAMIENTAS Y MÉTODOS

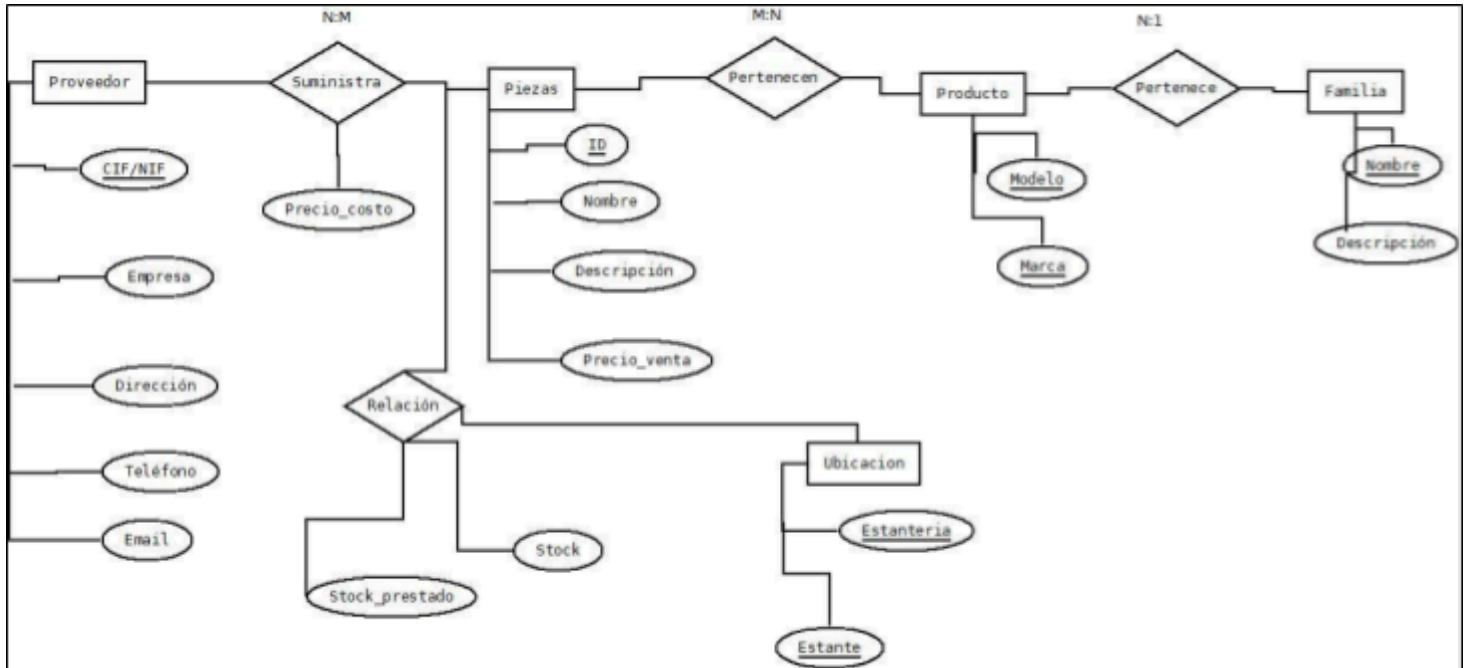
Breve descripción de las herramientas utilizadas en el proyecto:

https://cloud.mongodb.com/	Servicio Cloud gratuito host de BDD
MongoDB	BDD sobre el que gira el ejercicio
VSCode	Herramienta para gestionar la BDD MongoDB
Draw.IO	Creación de Esquemas
DIA	Creación Esquemas E-R
Chat GPT	Consultas
Apuntes varios	Consultas
Ayuda profesional	Consultas y Apoyo Emocional

3. PERSPECTIVA ESTÁTICA

3.1. E/R (Entidad-Relación)

Diagrama que representa la estructura de las entidades y sus relaciones:



Mediante este primer esquema Entidad/Relación de la base de datos a utilizar para nuestra aplicación, obtenemos la información necesaria para obtener un inventario completo donde situemos tanto proveedores, piezas, productos e incluso el propio electrodoméstico, relacionando entre sí éstos elementos.

Sin embargo, a la hora de aplicar la base de datos a efectos de la aplicación a desarrollar, simplificaremos mucho la misma a fin de evitar redundancias y duplicidades innecesarias en otros departamentos de gestión de datos del taller dónde se pondrá en funcionamiento.

Por ejemplo, a efectos de control del inventario en un taller tan pequeño dónde el proveedor siempre es el mismo, no nos es necesario añadir en nuestra aplicación la opción de relacionar proveedores

3.2. Pasos a tablas

Detalles sobre cómo se transformaron los diagramas E/R en tablas de base de datos:

Proveedor = (CIF_NIF, Empresa, Dirección, Teléfono, Email)

C.Alt: Empresa

VNN: Empresa

Piezas = (ID, Nombre, Descripción, Precio_venta)

VNN:Nombre

Producto = (Modelo, Marca)

Familia = (Nombre, Descripción)

VNN: Descripción

Ubicación = (Estantería, Estante)

R.I.: null(Estantería) => null(Estante)

3.3. DDL (Data Definition Language)

Ejemplos de código que definen la estructura de la base de datos:

Para crear las colecciones en MongoDB se han realizado las siguientes instrucciones:

```
use('colladoinventario')

db.createCollection("Proveedor")
db.createCollection("Pieza")
db.createCollection("Pieza_Costo")
db.createCollection("Ubicacion")
db.createCollection("Pieza_Ubicacion")
db.createCollection("Familia")
db.createCollection("Producto")
db.createCollection("Pieza_pertenece_Producto")
```

3.4. DML (Data Manipulation Language)

Ejemplos de código que manipulan los datos en la base de datos:

```
db.Proveedor.insertMany([
  { CIF: '9873490823', Empresa: 'CNA Group', Direccion: 'CALLE PATATA', Ciudad: 'BARCELONA',
    Telefono: '+34 93 3561182', Email: 'cacafuti@gmail.com' },
  { CIF: '2342398789', Empresa: 'DPLGroup', Direccion: 'CALLE HABICHUELA', Ciudad: 'VALENCIA',
    Telefono: '+34 96 3561187', Email: 'cacafuto@gmail.com' },
  { CIF: '7236483292', Empresa: 'FerSay', Direccion: 'CALLE GOMINOLA', Ciudad: 'VALENCIA', Telefono:
    '+34 93 3561184', Email: 'cacafutu@gmail.com' },
  { CIF: '7236412354', Empresa: 'BongoDb', Direccion: 'CALLE FALSA', Ciudad: 'VALENCIA', Telefono:
    '+34 93 3561234', Email: 'cacafute@gmail.com' }
]);
```

```
db.Pieza.insertMany([
  { ID: '15101000', Nombre: 'Motor', Descripcion: 'Grupo motor para campana extractora de cocina',
    Precio_venta: 95.59 },
  { ID: '15100033', Nombre: 'Botonera', Descripcion: 'Caja de mandos para campana extractora de
    cocina', Precio_venta: 63.69 },
  { ID: '485189911103', Nombre: 'Condensador', Descripcion: 'Condensador 35 mm - 450V para
    lavadora', Precio_venta: 6.85 },
  { ID: 'V991000H1', Nombre: 'Bomba', Descripcion: 'Bomba desagüe EBS2556-0707 para lavadora',
    Precio_venta: 28.89 },
  { ID: '481946818049', Nombre: 'Burlete', Descripcion: 'Burlete medidas 98.7x57.1 cm para frigorifico
    combi', Precio_venta: 33.50 },
  { ID: '83040745', Nombre: 'Resistencia', Descripcion: 'Resistencia INFERIOR 900/300w para horno',
    Precio_venta: 15.25 }
]);
```


3.5. DQL (Data Query Language)

Ejemplos de código que consultan la base de datos:

1ª Consulta sobre toda la información disponible sobre la pieza con ID:15101000

```
db.Pieza.find({ ID: '15101000' })
```

2ª Consulta sobre todos los proveedores que tenemos en Valencia

```
db.Proveedor.find({ Ciudad: 'VALENCIA' })
```

3.6. DCL (Data Control Language)

Ejemplos de código que controlan los permisos y la seguridad de la base de datos:

En MongoDB podemos controlar la creación de usuarios y roles

1º Creamos un usuario (JoanProfe) con un rol (lectura)

use 'colladoinventario'

```
db.createUser({
  user: "JoanProfe",
  pwd: "PasswordJoanProfe",
  roles: [
    { role: "read", db: "colladoinventario" }
  ]
})
```

2º También podemos crear roles personalizados para ciertas partes de la base de datos, como el rol de oficinista

use colladoinventario

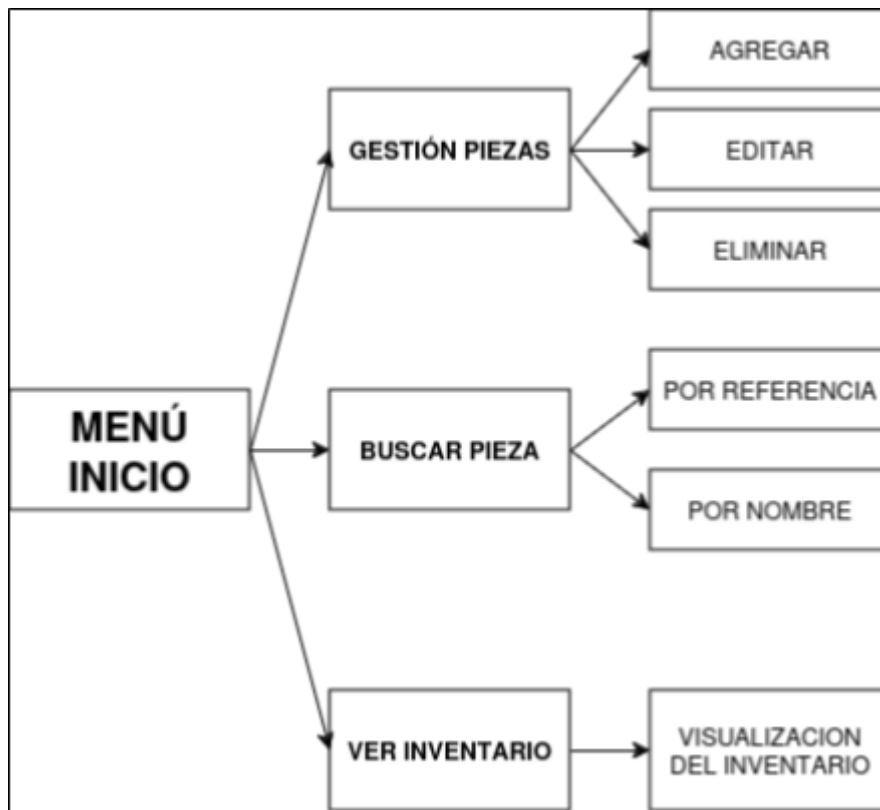
```
db.createRole({
  role: "oficinista",
  privileges: [
    {
      resource: { db: "colladoinventario", collection: "Pieza" },
      actions: [ "find", "insert", "update", "remove" ]
    }
  ],
  roles: []
})
```

3º Y añadimos ahora el trabajador de la oficina al rol de oficinista

use colladoinventario

```
db.createUser({
  user: "Anacleto",
  pwd: "244466666",
  roles: [
    { role: "oficinista", db: "colladoinventario" } ] })
```

4. PERSPECTIVA DINÁMICA



5.CONCLUSIONES

Consideramos este proyecto un buen inicio de lo que querríamos que fuera una aplicación completa y con mucha más funcionalidad a la hora de ponerlo en práctica en el taller de reparación de electrodomésticos. La base de datos generada da para mucha más utilidad que la que en principio puede funcionar con el esquema de la interfaz, ya que no sólo se puede consultar, añadir, modificar o eliminar productos, sino que se podría insertar hasta la ubicación en físico de los productos y/o diferentes proveedores.

Además, queda pendiente la opción de poder dar de alta a usuarios (futuros empleados) y darles permisos para poder manipular y operar con el inventario sin necesidad de que sea una sola persona la que lo utilice o directamente carecer de seguridad.

Aun así, consideramos que esta aplicación, aunque sencilla, proporciona una organización y ayuda extra, futuramente mejorable.

Ya a efectos más relacionados con preferencias personales puedo decir que a futuros espero estar más relacionado con bases de datos noSQL como MongoDB que al contrario. A priori parece más sencillo su uso y visualmente más fácil de leer. El uso de herramientas adicionales como Mongoose nos permiten además ampliar la facilidad de uso de MongoDB y evitar errores comunes que podríamos tener más fácilmente en MongoDB comparado con la rigidez de MySQL.

6.ENLACE A REPOSITORIO

A continuación se enlaza el repositorio en GitHub del Microproyecto completo:

https://github.com/CCC-Programoeba/BDD_ProyectoInventarioMongo