

Desarrollo Backend con NodeJS

Mongo

Bases de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso

Relacionales

Almacena la información en algo llamado tablas y luego crea relaciones entre ellas en base a identificadores únicos



No relacionales

Las bases de datos no relacionales guardan toda la información en un único registro y no en varias tablas separadas



Instalando mongo

<https://docs.mongodb.com/manual/installation/>

Comandos básicos

Nos permite interactuar con Mongo desde la línea de comando

- <https://docs.mongodb.com/manual/mongo/>
- <https://docs.mongodb.com/manual/reference/mongo-shell/>

Comandos básicos

- `show dbs`
 - Muestra todas las bases de datos creadas

Comandos básicos

- `use DATABASE_NAME`
 - Crea una nueva base de datos con el nombre `DATABASE_NAME` y nos posiciona en ella
 - Si la base de datos ya existe solo nos posiciona en ella

Comandos básicos

- db
 - nos indica que base de datos está seleccionada actualmente

Comandos básicos

- `db.dropDatabase()`
 - Elimina la base de datos seleccionada

Comandos básicos

- `db.createCollection("name")`
 - Crea una collection con el nombre "name" dentro de la base de datos seleccionada

Comandos básicos

- `db.name.drop()`
 - Elimina la collection con nombre “name”

Comandos básicos

- `db.name.insert(JSON)`
 - Agrega un registro dentro de nuestra collection “name”

Queries

- `db.name.find()`
 - Nos devuelve un listado con todos los registros de la collection NAME

Queries

- `db.michis.find(name: "Alyx")`
 - `{ "_id" : ObjectId("5fd7c28da1a7bbc2353ca8b8"), "name" : "Alyx" }`
- <https://docs.mongodb.com/manual/tutorial/query-documents/>

Si prefieren una GUI: Robo 3T

Robo 3T es una GUI (interfaz gráfica) para poder manejar nuestra información de mongo

- <https://robomongo.org/download>

Integrando mongo con node

- Mongoose
 - Mongoose es la librería que vamos a utilizar para conectar Mongo con nuestra aplicación Node
 - `npm install —save mongoose`

Conectando node a la base de datos con mongoose

```
const mongoose = require("mongoose");
```

```
mongoose.connect('mongodb://localhost:27017/basedegatos', {  
  useNewUrlParser: true  
});
```

Models

Un modelo de datos es donde vamos a escribir el plan de nuestra collection. Esto quiere decir que vamos a decir que propiedades van a existir y que tipo de dato van a contener. En mongoose se escriben por medio de un **schema**

Utilizando schemas en node:

```
const mongoose = require("mongoose");  
const testSchema = mongoose.Schema(  
  {  
    testProperty: {  
      type: String,  
    },  
    anotherProperty: {  
      type: Number,  
      required: true  
    },  
    yetAnotherProperty: []  
  }  
);  
module.exports = mongoose.model("Test", testSchema);
```

El nombre del modelo:

- Singular
- Capitalizado
- de tipo string

```
module.exports = mongoose.model("Test", testSchema);
```