# WS REST para Mi Todo App

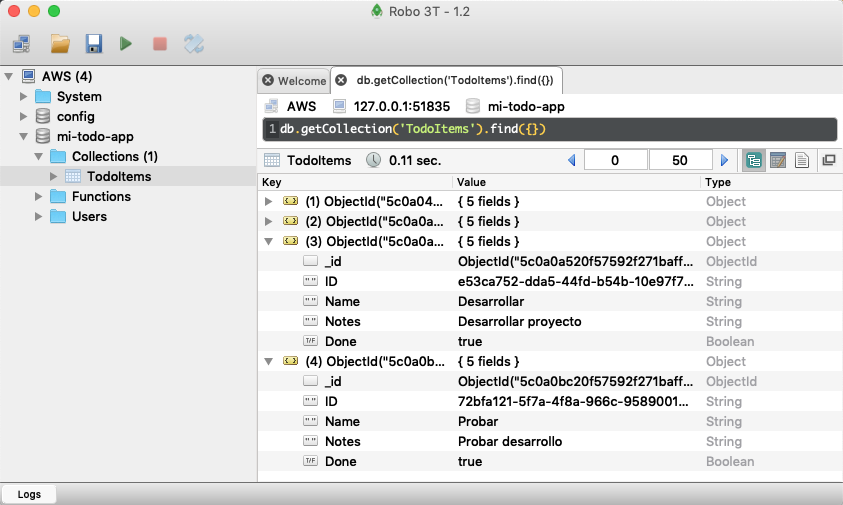
Construcción de Backend en NodeJS con MongoDB en una plataforma Linux Ubuntu 18.04 para el proyecto de Mi-todo-APP:

## Instalación de paquetería necesaria del sistema operativo

#sudo apt-get install npm nodejs mongodb

## Creación de la DB

Se recomienda instalar un cliente para MongoDB como RoboMongo, en el cual se deberá de construir la DB llamada “mi-todo-app” y dentro de la misma, una colección llamada “TodoItems”:



Dentro de la colección, se recomienda insertar algunos documentos para que pueda comenzarse con información sembrada para efectos de pruebas. El esquema de dichos documentos se muestra a continuación:



## Desarrollo del middleware con NodeJS

#mkdir ws\_mitodoapp

#cd ws\_mitodoapp

#npm init

#npm install express mongodb body-parser –save

#nano app.js

### Código de la aplicación en NodeJS

// DEFINICIÓN DE LIBRERIAS DE NODE JS

// http://expressjs.com/

const expressLib = require("express");

// REFERENCIA: https://zellwk.com/blog/crud-express-mongodb/

const mongoLib = require("mongodb").MongoClient;

// INSTANCIAMOS EXPRESS

var appExpress = expressLib();

// https://www.npmjs.com/package/body-parser

const bodyParserLib = require('body-parser');

// PARÁMETRO DE CONFIGURACIÓN DE BODY-PASER

var urlencodedParser = bodyParserLib.urlencoded({ extended: false });

// HABILITAMOS BODY-PARSER COMO PARSER JSON

appExpress.use(bodyParserLib.json());

appExpress.get("/api/todoitems", (req, res) => {

    // MOSTRAR INFORMACION DE LA URL EN LA CONSOLA

    console.log(req.originalUrl);

    // CÓDIGO PARA CONEXIÓN CON HOST CON MONGODB

    mongoLib.connect("mongodb://localhost:27017", (err, client) => {

        // CONEXIÓN CON LA BASE DE DATOS NOSQL

        const db = client.db("mi-todo-app");

        // REFERENCIA: https://www.w3schools.com/nodejs/nodejs\_mongodb\_find.asp

        db.collection("TodoItems").find({}, { projection: { \_id: 0, ID: 1, Name: 1, Notes: 1, Done: 1 } }).toArray(function (err, result) {

            if (err) throw err;

            console.log(result);

            res.send(result);

        });

        // CIERRE DE LA CONEXIÓN CON EL HOST DE MONGODB

        client.close();

    });

});

appExpress.post("/api/todoitems", (req, res) => {

    // MOSTRAR INFORMACION DE LA URL EN LA CONSOLA

    console.log(req.originalUrl);

    // CÓDIGO PARA CONEXIÓN CON HOST CON MONGODB

    mongoLib.connect("mongodb://localhost:27017", (err, client) => {

        if (err) throw err;

        var dbo = client.db("mi-todo-app");

        dbo.collection("TodoItems").insertOne(req.body, function (err, res) {

            if (err) throw err;

            console.log("1 document inserted");

            client.close();

        });

    });

});

appExpress.put("/api/todoitems", (req, res) => {

    // MOSTRAR INFORMACION DE LA URL EN LA CONSOLA

    console.log(req.originalUrl);

    // CÓDIGO PARA CONEXIÓN CON HOST CON MONGODB

    mongoLib.connect("mongodb://localhost:27017", (err, client) => {

        if (err) throw err;

        var dbo = client.db("mi-todo-app");

        var myquery = { "ID" : req.body.ID };

        var newvalues = { $set: { "Name" : req.body.Name, "Notes" : req.body.Notes, "Done" : req.body.Done } };

        dbo.collection("TodoItems").updateOne(myquery, newvalues, function (err, res) {

            if (err) throw err;

            console.log("1 document updated");

            client.close();

        });

    });

});

appExpress.delete("/api/todoitems/:ID", (req, res) => {

    // MOSTRAR INFORMACION DE LA URL EN LA CONSOLA

    console.log(req.originalUrl);

    // CÓDIGO PARA CONEXIÓN CON HOST CON MONGODB

    mongoLib.connect('mongodb://localhost:27017', (err, client) => {

        // CONEXIÓN CON LA BASE DE DATOS NOSQL

        const db = client.db('mi-todo-app');

        // OPERACIÓN DE INSERTAR CON LA COLECCIÓN

        db.collection('TodoItems').deleteOne({ 'ID' : req.params.ID }, function (err, result) {

            if (err) return console.log(err);

            console.log("BORRADO DE MONGO: ");

        });

        // CIERRE DE LA CONEXIÓN CON EL HOST DE MONGODB

        client.close();

    });

});

// POSICIONAMOS EL SERVICIO EN EL PUERTO 3000/TCP Y MOSTRAMOS UN MENSAJE DE SALIDA PARA MONITOREAR EL LANZAMIENTO DEL SERVICIO

appExpress.listen(3000, () => console.log('Ejemplo app listening on port 3000'));

## Ejecución del middleware en la terminal de Linux

#node app.js