**Mongo db**

Orienda a documentos (objeto JSON)

Se compone de colecciones (no es necesario que tenga esquema definido)

Mongoose(para usar squemas)

500mb mongo Atlas gratis

Acceder a mongo

<https://www.mongodb.com/try/download/shell>

Mongo atlas

Interfaz de usuario gráfica, Texto, Aplicación

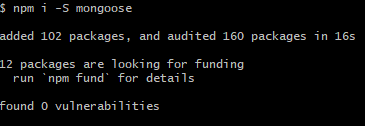
Descripción generada automáticamente

**Enlace para conectar con mongo dg**

mongodb+srv://miltonXD:<password>@cluster0.venxvyj.mongodb.net/?retryWrites=true&w=majority

libreria para conectarse con mongo db

npm i -S mongoose



Conexión

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Captura de pantalla de computadora

Descripción generada automáticamente

Se crea un nuevo archivo



Dentro del archivo se escribe

*//crear una constante mongoose*

*//importamos la dependecia de mongoose*

const mongoose= require('mongoose');

*//conexion a la base de datos*

*//se llama a la constante*

*//se llama al método connect*

*//se pega la url sacada de la web de mongodb*

mongoose.connect('mongodb+srv://miltonXD:Atlas1234@cluster0.venxvyj.mongodb.net/miApp?retryWrites=true&w=majority')

**Modelo de base de datos**

*//modelo de base dedatos*

*//un modelo tiene que tener un nombre 'User'*

*//objeto js con la forma de los documentos que se guardaran dentro de la coleccion User*

const User=mongoose.model('User',{

    nombreusuario:String,

    edad: Number,

})

**Operaciones**

**crear**

const crear= async ()=>{

*//constante user instancia de modelo User*

*//new nueva instancia del modelo User*

*//datos nombreusuario y edad*

    const user= new User({nombreusuario:'Milton',edad:21})

*//guardar en la base de datos*

*//await por que devuelve una promesa user.save()*

*//se guarda en una cosntante para imprimir en la consola (no es necesario)*

    const saveUser=*await* user.save()

    console.log(saveUser)

};

*//ejecutar la funcion*

crear();

**Texto

Descripción generada automáticamente**

**Buscar todo**

*//buscar todo*

const buscarTodo= async ()=>{

*//constante gurara los usuarios*

*//await*

*//User coleccion*

*//find() buscará todo lo que se encuentre dentod de la colección User*

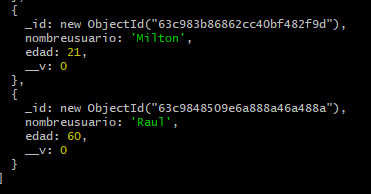
    const users= *await* User.find()

    console.log(users)

};

*//ejecutamos la funcion*

buscarTodo();

****

**Buscar todos los elementos que cumplan con la propiedad**

*//buscar los que cumplan con el parametro*

const buscarporParamtro= async ()=>{

*//se guarda el valor de la busqueda en la cosntante user*

*//.find({})dento del corchete se añade como se lo va a buscar*

*//traera un arreglo*

    const user=*await* User.find({nombreusuario:'Milton'})

*//imprime el usuario buscado*

    console.log(user)

};

*//ejecucion de la funcion*

 buscarporParamtro();

trae un arreglo

Texto

Descripción generada automáticamente

**Para que retorne solo 1 valor**

*//buscar uno*

const buscarUno =async ()=>{

*//se guarda el valor de la busqueda en user*

*//.findOne significa que traera un unico resultado*

*//traera un objeto*

    const user= *await* User.findOne({nombreusuario:'Raul'})

*//impresion*

    console.log(user)

};

*//ejecucion de la funcion*

buscarUno();

**Texto

Descripción generada automáticamente**

**Actualizar**

*//actualizar*

const actualizar = async ()=>{

*//primero de busca el resultado requerido*

    const user= *await* User.findOne({ nombreusuario:'Milton'});

*//se actualiza algun parametro*

    console.log(user);

    user.edad=22;

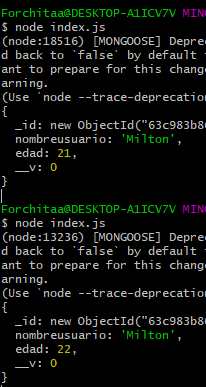
*//se guarda*

*await* user.save();

};

*//ejecucion de la funcion*

actualizar();

****

**Eliminar**

*//eliminar uno*

const eliminar = async () => {

*//se busca el valor*

    const user = *await* User.findOne({ nombreusuario: 'Raul' });

*//conprobacion si exite*

    console.log(user);

*//se usa una consicion para validar si el usuario existe*

*if* (user) {

*//eliminacion*

*//remove permite elminar el recurso de la base de datos*

*//unicamente si existe el valor*

*await* user.remove();

    }

};

*//ejecucion de la funcion*

eliminar();

**Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente**