

**MAESTRIA EN SISTEMAS INTERACTIVOS CENTRADOS  
EN EL USUARIO**



---

**SARA MADELAINE GUTIÉRREZ DE FRÍAS**

**LAURA TERESA VAZQUEZ CORDOBA**

**JOSÉ ANTONIO DOMINGUEZ VELA**

**GESTIÓN DE DATOS**

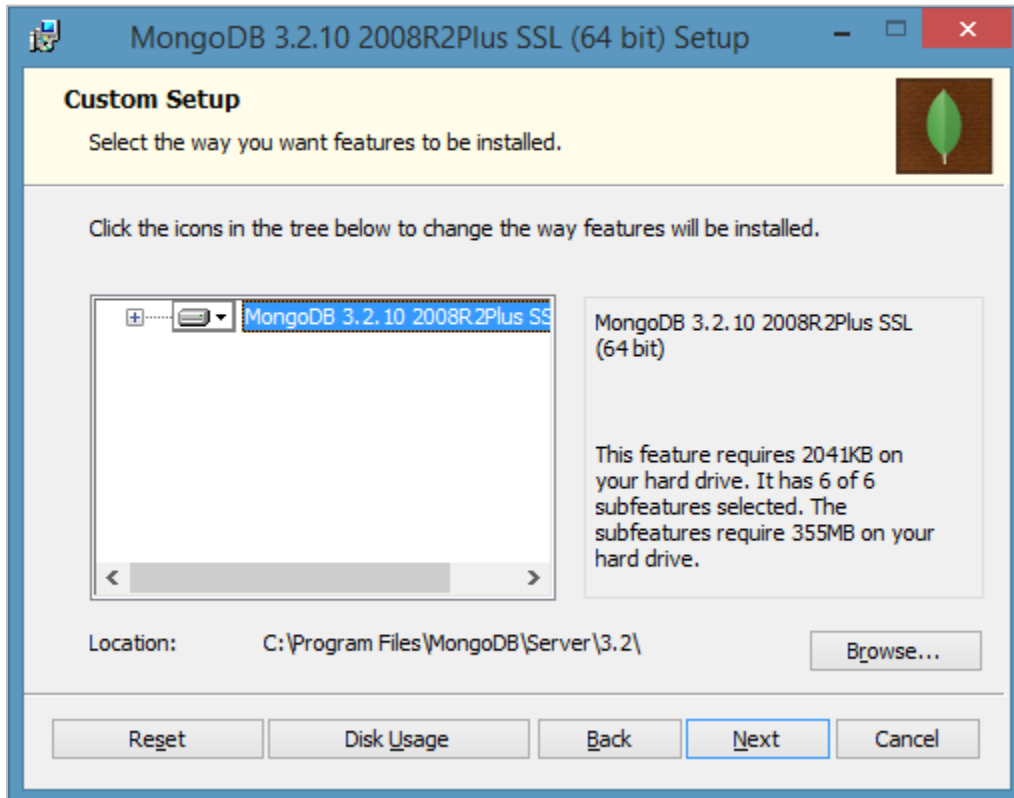


**mongoDB®**

**PROFESORA: LORENA ALONSO RÁMIREZ**

## Instalación de Mongo DB

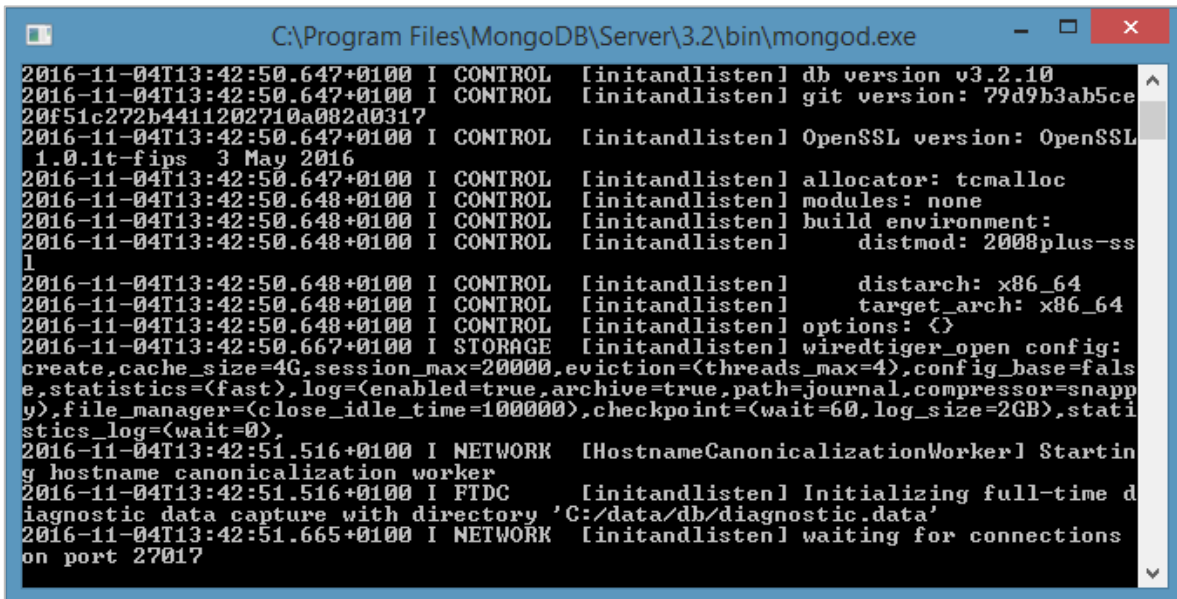
Para realizar la instalación de Mongo DB debemos descargar el instalador del software y ejecutarlo. Una vez ejecutado solo debemos seguir el paso a paso del asistente.



Una vez Mongo DB este instalado debemos crear las carpetas data y db, en el directorio C:\data\de, primero creando la carpeta data y dentro de data db. Aquí se van a guardar configuraciones de las bases de datos, etc.

Nombre	Fecha de modifica...	Tipo	Tamaño
Appparkins	10/11/17 10:27 a. m.	Carpeta de archivos	
Archivos de programa	2/12/17 10:37 a. m.	Carpeta de archivos	
Archivos de programa (x86)	13/11/17 1:55 p. m.	Carpeta de archivos	
CabLogs	14/12/14 4:25 a. m.	Carpeta de archivos	
data	2/12/17 10:38 a. m.	Carpeta de archivos	
Dev-Cpp	13/7/17 8:58 p. m.	Carpeta de archivos	
Fraps	18/6/16 12:37 p. m.	Carpeta de archivos	
Imagenes_Blog	23/10/17 6:52 a. m.	Carpeta de archivos	
inetpub	23/4/17 9:34 a. m.	Carpeta de archivos	

Luego debemos ejecutar MongoDB, para esto nos dirigimos al destino en el cual se instaló la aplicación: C:\Program Files\MongoDB\Server\3.2\bin abrimos una ventana de comando y ejecutamos mongod.exe



```
C:\Program Files\MongoDB\Server\3.2\bin\mongod.exe
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] db version v3.2.10
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] git version: 79d9b3ab5ce
20f51c272b4411202710a082d0317
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.1t-fips 3 May 2016
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] allocator: tcmalloc
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] modules: none
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] build environment:
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] distmod: 2008plus-ssl
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] distarch: x86_64
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] target_arch: x86_64
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] options: {}
2016-11-04T13:42:50.667+0100 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=4G,session_max=20000,eviction=(threads_max=4),config_base=false,
statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),
file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2016-11-04T13:42:51.516+0100 I NETWORK [HostnameCanonicalizationWorker] Starting
hostname canonicalization worker
2016-11-04T13:42:51.516+0100 I FTDC [initandlisten] Initializing full-time diagnostic
data capture with directory 'C:/data/db/diagnostic.data'
2016-11-04T13:42:51.665+0100 I NETWORK [initandlisten] waiting for connections
on port 27017
```

## Configuración de Mongo DB

A continuación, se muestra la función de cada uno de los comandos de MongoDB sobre la base de datos de la práctica.

- Crear / acceder a la base de datos: Con esto la base de datos ha sido generada.

```
C:\Program Files\MongoDB\Server\3.0\bin> mongo.exe
> use blogbasedatos
switched to db blogbasedatos
>
```

- Base de datos existentes: para visualizar las bases de datos existentes se utiliza el comando 'show dbs', la base de datos local se encuentra creada por defecto, para esta práctica se utilizará la base de datos 'blogbasedatos'.

```
C:\Program Files\MongoDB\Server\3.0\bin> mongo.exe
> show dbs
blogbasedatos    0.078GB
local            0.078GB
>
```

- Documentos: En la siguiente imagen se muestra el listado de los documentos existentes en la base de datos blogbasedatos.

```
C:\Program Files\MongoDB\Server\3.0\bin> mongo.exe
> use blogbasedatos
switched to db blogbasedatos
> show collections
comentarios
publicacions
system.indexes
usuarios
>
```

- Estructura de las colecciones:

Publicacions		
Nombre campo	Tipo de dato	Descripción
Título	String	Título de la publicación

<b>Contenido</b>	String	Descripción de la publicación
<b>Fecha</b>	Date	Fecha de la publicación

Comentarios		
Nombre campo	Tipo de dato	Descripción
<b>comentarioid</b>	Array[Comentarios Ref]	Referencia a un documento comentario
<b>publicacionId</b>	Objeto Publicacions	Referencia a documentos publicacions
<b>usuarioid</b>	Objeto Usuarios	Referencia a documento usuarios
<b>Comentario</b>	String	Texto del comentario
<b>Principal</b>	Number	Indica si es padre o hijo puede tener valor 0 si es hijo o 1 si es padre
<b>Fecha</b>	Date	Fecha de publicación del comentario

Usuarios		
Nombre campo	Tipo de dato	Descripción
<b>Nombre</b>	String	Nombre de usuario
<b>Email</b>	String	Correo electrónico
<b>Password</b>	String	Contraseña de acceso

- Usuarios: insertando nuevos usuarios: a continuación, se muestra un ejemplo de inserción de un nuevo usuario a la colección Usuarios.

```

C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.usuarios.find()
{ "_id" : ObjectId("5a25c3ee2fb3da3c22c58380"), "nombre" : "Laura Teresa Vázquez Córdoba", "password" : "12345", "email" : "laura@correo.com", "__v" : 0 }
{ "_id" : ObjectId("5a25c4562fb3da3c22c58381"), "nombre" : "Laura Vázquez Cordoba", "password" : "12345", "email" : "teresa@correo.com", "__v" : 0 }
>

```

- Sintaxis :db.usuarios.insert({nombre:"Sara Madelaine" email:"sara@correo.com" password: "12345"});

La siguiente imagen muestra el resultado de la ejecución del comando.

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.usuarios.insert({nombre:"Sara Madelaine", correo:"sara@correo.com", password:"12345"});
WriteResult({ "nInserted" : 1 })
```

- Listado de usuarios con los nuevos usuarios ingresados: La siguiente imagen muestra la lista de documentos usuarios, se han agregado 3 usuarios cada uno con las propiedades definidas anteriormente.

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.usuarios.insert({nombre:"Sara Madelaine", correo:"sara@correo.com", password:"12345"});
WriteResult({ "nInserted" : 1 })
> db.usuarios.find()
{ "_id" : ObjectId("5a25c3ee2fb3da3c22c58380"), "nombre" : "Laura Teresa Vázquez Córdoba", "password" : "12345", "email" : "laura@correo.com", "__v" : 0 }
{ "_id" : ObjectId("5a25c4562fb3da3c22c58381"), "nombre" : "Laura Vázquez Cordoba", "password" : "12345", "email" : "teresa@correo.com", "__v" : 0 }
{ "_id" : ObjectId("5a287c35d2bfdc98fb4acd46"), "nombre" : "Sara Madelaine", "correo" : "sara@correo.com", "password" : "12345" }
```

- Seleccionando una colección: La siguiente imagen muestra los comentarios existentes en la colección comentarios.

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.comentarios.find()
{ "_id" : ObjectId("5a25c7272fb3da3c22c58382"), "comentario" : "Hablando de MongoDB", "usuarioId" : ObjectId("5a25c3ee2fb3da3c22c58380"), "principal" : 1, "fecha" : ISODate("2017-12-04T22:07:35.280Z"), "comentarioId" : [ ], "__v" : 0 }
{ "_id" : ObjectId("5a25cde52fb3da3c22c58384"), "comentario" : "Primer comentario sobre MongoDB", "usuarioId" : ObjectId("5a25c3ee2fb3da3c22c58380"), "publicacionId" : ObjectId("5a25cc3d2fb3da3c22c58383"), "principal" : 1, "fecha" : ISODate("2017-12-04T22:36:21.070Z"), "comentarioId" : [ ObjectId("5a281a51431225001ef236e8") ], "__v" : 1 }
{ "_id" : ObjectId("5a25ce322fb3da3c22c58385"), "comentario" : "Haciendo otro comentario de Mongo", "usuarioId" : ObjectId("5a25c3ee2fb3da3c22c58380"), "publicacionId" : ObjectId("5a25cc3d2fb3da3c22c58383"), "principal" : 1, "fecha" : ISODate("2017-12-04T22:37:38.692Z"), "comentarioId" : [ ], "__v" : 0 }
{ "_id" : ObjectId("5a281a51431225001ef236e8"), "comentario" : "Respuesta de mongo", "usuarioId" : ObjectId("5a25c3ee2fb3da3c22c58380"), "publicacionId" : ObjectId("5a25cc3d2fb3da3c22c58383"), "principal" : 0, "fecha" : ISODate("2017-12-06T16:26:57.550Z"), "comentarioId" : [ ], "__v" : 0 }
```

- Seleccionando un documento: la siguiente sintaxis busca un comentario de la colección comentarios específico indicando un atributo de la colección comentarios.
  - Sintaxis: `db.comentarios.find("comentario":"Primer comentario sobre MongoDB");`

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.comentarios.find("comentario":"Primer comentario sobre MongoDB");
{ "_id" : ObjectId("5a25cde52fb3da3c22c58384"), "comentario" : "Primer comentario sobre MongoDB", "usuarioId" : ObjectId("5a25c3ee2fb3da3c22c58380"), "publicacionId" : ObjectId("5a25cc3d2fb3da3c22c58383"), "principal" : 1, "fecha" : ISODate("2017-12-04T22:36:21.070Z"), "comentarioId" : [ ObjectId("5a281a51431225001ef236e8") ], "__v" : 1 }
```

- Actualizando información de un documento
  - Actualizamos el título del documento publicación en la colección publicaciones.

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.publicacions.find();
{ "_id" : ObjectId("5a25cc3d2fb3da3c22c58383"), "titulo" : "MongoDB", "contenido" : "Hablando de MongoDB", "slug" : "MongoDB", "fecha" : ISODate("2017-12-04T22:29:17.889Z"), "__v" : 0 }
> db.publicacions.update({"_id":ObjectId("5a25cc3d2fb3da3c22c58383")},{"$set":{"titulo":"Hablemos de MongoDB"}});
```

- Sintaxis:
 

```
db.publicacions.update({"_id":ObjectId("5a25cc3d2fb3da3c22c58383")},{"$set":{"titulo":"Hablemos de MongoDB"}});
```

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.publicacions.update({"_id":ObjectId("5a25cc3d2fb3da3c22c58383")},{"$set":{"titulo":"Hablemos de MongoDB"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.publicacions.find()
{ "_id" : ObjectId("5a25cc3d2fb3da3c22c58383"), "titulo" : "Hablemos de MongoDB", "contenido" : "Hablando de MongoDB", "slug" : "MongoDB", "fecha" : ISODate("2017-12-04T22:29:17.889Z"), "__v" : 0 }
>
```

- Eliminando un documento
  - Eliminamos un documento Usuario de la colección Usuarios
  - Sintaxis: `db.usuarios.remove({"nombre":"Carla Morrison"});`

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.usuarios.find({"nombre":"Carla Morrison"});
{ "_id" : ObjectId("5a288221d2bfdc98fb4acd47"), "nombre" : "Carla Morrison", "correo" : "carla@correo.com", "password" : "12345" }
> db.usuarios.remove({"nombre":"Carla Morrison"});
WriteResult({ "nRemoved" : 1 })
> db.usuarios.find();
{ "_id" : ObjectId("5a25c3ee2fb3da3c22c58380"), "nombre" : "Laura Teresa Vázquez Córdoba", "password" : "12345", "email" : "laura@correo.com", "__v" : 0 }
{ "_id" : ObjectId("5a25c4562fb3da3c22c58381"), "nombre" : "Laura Vázquez Cordoba", "password" : "12345", "email" : "teresa@correo.com", "__v" : 0 }
{ "_id" : ObjectId("5a287c35d2bfdc98fb4acd46"), "nombre" : "Sara Madelaine", "correo" : "sara@correo.com", "password" : "12345" }
>
```

## Conexión con la aplicación

- Configuración de la conexión a la base de datos desde angular a MongoDB: En la siguiente imagen se muestran las librerías requeridas por Angular para realizar la configuración del servidor.

```
// CONFIGURACIÓN DEL SERVIDOR
// =====

var express    = require('express');
var app        = express();
var port       = process.env.PORT || 8080;
var bodyParser = require('body-parser');
var slug       = require('slug');
var mongoose   = require('mongoose');
var morgan     = require("morgan");
```

La siguiente imagen muestra la configuración necesaria para el uso de las funciones REST y el uso de archivos JSON, además la sintaxis de conexión a la base de datos y la asignación de los diferentes documentos con los que interactúa la base de datos.

```
//Configura el app para que acepte json
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(morgan("dev"));
app.use(function(req, res, next) {
  res.setHeader('Access-Control-Allow-Origin', '*');
  res.setHeader('Access-Control-Allow-Methods', 'GET, POST');
  res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type, Authorization');
  next();
});

// Configuración de la base de datos
mongoose.connect('mongodb://localhost/blogbasedatos');
var Publicacion = require('./mongodb/publicacion');
var Usuario = require('./mongodb/usuario');
var Comentario = require('./mongodb/comentario');
```

- Definición de la estructura de los documentos MongoDB desde Angular para interacción con base de datos.
  - Usuarios



```
JS Usuario.js x JS Publicacion.js JS Comentario.js JS controller.js ...\adm
1 | var mongoose = require('mongoose');
2 | var Schema = mongoose.Schema;
3
4 | var UsuarioSchema = new Schema({
5 |   nombre: String,
6 |   email: String,
7 |   password: String
8 | });
9
10 | module.exports = mongoose.model('Usuario', UsuarioSchema);
```

- Publicaciones

```
JS Usuario.js x JS Publicacion.js JS Comentario.js JS controller.js ...\adm
1 | var mongoose = require('mongoose');
2 | var Schema = mongoose.Schema;
3
4 | var UsuarioSchema = new Schema({
5 |   nombre: String,
6 |   email: String,
7 |   password: String
8 | });
9
10 | module.exports = mongoose.model('Usuario', UsuarioSchema);
```

- Comentarios de una publicación

```
JS Usuario.js JS Publicacion.js JS Comentario.js x JS controller.js ...\admin\... JS app.js
1 var mongoose = require('mongoose');
2 var Schema = mongoose.Schema;
3
4 var ComentarioSchema = new Schema({
5   comentarioId: [{
6     type: mongoose.Schema.Types.ObjectId,
7     ref: 'Comentario'
8   }],
9   publicacionId: {type: mongoose.Schema.Types.ObjectId, ref: 'Publicacion'},
10  usuarioId: {type: mongoose.Schema.Types.ObjectId, ref: 'Usuario'},
11  comentario: String,
12  principal: Number,
13  fecha: Date
14 });
15
16 module.exports = mongoose.model('Comentario', ComentarioSchema);
17
```

## Estructura de comentarios

En la siguiente imagen se muestra la estructura de un comentario en MongoDB, en este se señala que los comentarios hijos se anidan en un arreglo de comentarios tal como se definió anteriormente en la estructura del documento 'comentarios'.

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe
> db.comentarios.find({"comentario":"practica"});
{"_id": ObjectId("5a2adbc159983d0c0c3c6b33"), "comentario": "practica", "usuarioId": ObjectId("5a2ad26715fc116818250b44"), "publicacionId": ObjectId("5a2ad68159983d0c0c3c6b243"), "principal": 1, "fecha": ISODate("2017-12-08T18:36:49.816Z"), "comentarioId": [ ObjectId("5a2adb959983d0c0c3c6b35"), ObjectId("5a2adbe359983d0c0c3c6b36"), ObjectId("5a2adbec59983d0c0c3c6b37") ], "v": 3 }
```

La siguiente imagen muestra la actividad en la consola de npm del servidor node, en esta cada vez que se agrega un comentario a la colección imprime como está estructurado el documento JSON, los comentarios hijos se encuentran en el arreglo de referencias de tipo 'comentario', en cada posición del arreglo se encuentra el objeto comentario hijo.

```

ca: npm
DELETE /comentario/5a2adbc659983d0c0c3c6b34/5a2ad68159983d0c0c3c6b24 200 23.113 ms - 3111
POST /comentario/hijo/ 200 28.458 ms - 3409
POST /comentario/hijo/ 200 37.629 ms - 3708
POST /comentario/hijo/ 200 27.829 ms - 4007
GET /publicaciones 304 14.381 ms - -
GET /publicaciones 304 1.901 ms - -
GET /publicacion/Proyecto-Final 304 2.035 ms - -
[ { comentarioId:
  [ { comentarioId: [],
    __v: 0,
    fecha: 2017-12-08T18:37:13.275Z,
    principal: 0,
    publicacionId: 5a2ad68159983d0c0c3c6b24,
    usuarioId: [Object],
    comentario: ' practica2',
    _id: 5a2adbd959983d0c0c3c6b35 },
    { comentarioId: [],
      __v: 0,
      fecha: 2017-12-08T18:37:23.737Z,
      principal: 0,
      publicacionId: 5a2ad68159983d0c0c3c6b24,
      usuarioId: [Object],
      comentario: ' practica3',
      _id: 5a2adbe359983d0c0c3c6b36 },
      { comentarioId: [],
        __v: 0,
        fecha: 2017-12-08T18:37:32.529Z,
        principal: 0,
        publicacionId: 5a2ad68159983d0c0c3c6b24,
        usuarioId: [Object],
        comentario: ' practica4',
        _id: 5a2adbec59983d0c0c3c6b37 } ],
    __v: 3,
    fecha: 2017-12-08T18:36:49.816Z,
    principal: 1,
    publicacionId: 5a2ad68159983d0c0c3c6b24,
    usuarioId:
      { __v: 0,
        email: 'jvela@jvela.com',
        password: 'lakkck123',
        nombre: 'jVela',
        _id: 5a2ad26715fc116818250b44 },
    comentario: ' practica',
    _id: 5a2adbc159983d0c0c3c6b33 },

```

En la siguiente imagen muestra los comentarios padre e hijos tal y como se presentan en la interfaz.



- Definición de funciones REST de conexión a MongoDB desde Angular para interacción con base de datos:
  - Obtiene las publicaciones existente en las colecciones publicaciones

```
router.route('/publicaciones')
  .get(function(req, res) {
    //Obtienes todas las publicaciones con el modelo Publicacion
    Publicacion.find({}).sort('-fecha').exec(function(err, publicaciones) {
      if(err) {
        res.send(err);
      }
      res.json(publicaciones);
    });
  })
  .post(function(req, res) {
    Publicacion.create({
      titulo : req.body.titulo,
      contenido : req.body.contenido,
      slug : slug(req.body.titulo),
      fecha : new Date()
    }, function(err, publicacion) {
      if (err)
        res.send(err);
      Publicacion.find(function(err, publicaciones) {
        if (err)
          res.send(err)
        res.json(publicaciones);
      });
    });
  });
});
```

- Registro de un nuevo usuario

```

router.route('/registro')
  .post(function(req, res) {
    Usuario.findOne({email: req.body.email}, function(err, usuario) {
      if(err) {
        res.json({type: false, data: "Ha ocurrido un error!: " + err});
      }
      if (usuario) {
        res.json({type: false, data: "El email ya se encuentra registrado!"});
      } else {
        var usuarioModel = new Usuario();
        usuarioModel.email = req.body.email;
        usuarioModel.password = req.body.password;
        usuarioModel.nombre = req.body.nombre;
        usuarioModel.save(function(err, usuario) {
          if(err) {
            res.json({type: false, data: "Ha ocurrido un error!: " + err});
          }
          res.json({type: true, data: usuario});
        })
      }
    });
  });
});

```

- Acceso a usuario registrado

```

router.route('/usuarios')
  .get(function(req, res) {
    Usuario.find(function(err, usuarios) {
      if(err) {
        res.json({type: false, data: "Ha ocurrido un error!: " + err});
      }
      res.json(usuarios);
    });
  });
});

```

- Consulta publicaciones existentes

```

router.route('/publicacion/:publicacion_slug')
  .get(function(req, res) {
    Publicacion.findOne({slug: req.params.publicacion_slug}, function(err, publicacion) {
      if(err) {
        res.send(err);
      }
      res.json(publicacion);
    });
  });
});

```

- Crea / obtiene comentarios y comentarios hijos

```

router.route('/comentario')
  .post(function(req, res) {
    Comentario.create({
      comentario : req.body.texto,
      usuarioId : req.body.usuarioId,
      publicacionId : req.body.publicacionId,
      principal : 1,
      fecha : new Date()
    }, function(err, comentario) {
      if (err)
        res.send(err);
      //Obtienes los comentarios de la publicacion y sus comentarios hijos
      Comentario.find({publicacionId: req.body.publicacionId, principal: 1})
        .populate('comentarioId')
        .populate('usuarioId')
        .sort('-fecha')
        .exec(function(err, docs) {
          if(err) res.send(err);
          Comentario.populate(docs, {
            path: 'comentarioId.usuarioId',
            model: 'Usuario'
          },
          function(err, comentarios) {
            if(err) res.send(err);
            res.json(comentarios);
          });
        });
    });
  });
});

```

## Interfaz de Angular con Mongo DB

La tecnología utilizada para realizar la interfaz de usuario es Angular, utilizando como servidor web Node js.

- Pantalla de registro de nuevo usuario

¿Aún no eres usuario?

Regístrate ahora mismo

### Datos a llenar

NOMBRE

CORREO

CONTRASEÑA

Registrarse



- Mensaje de registro correcto

Se ha registrado correctamente

- Inicio de sesión

¿Ya eres usuario?

Inicia sesión

### Login

Iniciar Sesión

- Generar una nueva publicación

MÓDULO DE ADMINISTRACIÓN DE PUBLICACIONES

### GENERAR UNA NUEVA PUBLICACIÓN

Título

Contenido

PUBLICAR

### EDITAR PUBLICACIONES EXISTENTES

TÍTULO	FECHA	CONFIGURACIÓN
--------	-------	---------------

- Editar publicaciones existentes

### EDITAR PUBLICACIONES EXISTENTES

TÍTULO	FECHA	CONFIGURACIÓN
MongoDB	2017-12-04	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

- Edición de publicación existente

MÓDULO DE ADMINISTRACIÓN DE PUBLICACIONES

### EDITAR PUBLICACIÓN

MongoDB

Hablando de MongoDB

GUARDAR CAMBIOS




- Pantalla principal del blog

MongoDB - 04-12-2017

# MongoDB

Hablando de MongoDB

MONGODB



- Agregar un nuevo comentario

04-12-2017


# MONGODB

Hablando de MongoDB

Haciendo otro comentario de Mongo

Enviar Comentario

MONGODB



NODEJS

- Comentario añadido

¿Desesas desconectarte?

Cerrar Sesión


Comentario por: Laura Teresa Vázquez Córdoba 04-12-2017

Hablando de MongoDB

Responder Editar Eliminar

node JS

# ANGULARJS



- Respondiendo un comentario

Comentario por: Laura Teresa Vázquez Córdoba 04-12-2017

Haciendo otro comentario de Mongo

Responder Editar Eliminar

---

Comentario por: Laura Teresa Vázquez Córdoba 04-12-2017

Primer comentario sobre MongoDB

Responder Editar Eliminar

Respuesta de mongo

Enviar Comentario

Comentario por: Laura Teresa Vázquez Córdoba 04-12-2017

Haciendo otro comentario de Mongo

Responder Editar Eliminar

---

Comentario por: Laura Teresa Vázquez Córdoba 04-12-2017

Primer comentario sobre MongoDB

Responder Editar Eliminar

---

Comentario por: Laura Teresa Vázquez Córdoba 06-12-2017

Respuesta de mongo

Edit Eliminar