Práctica MVC

SOLUCIONARIO

En este documento iremos reflexionando sobre lo realizado en la práctica

# Estructura básica de la aplicación

**Copia y pega el código de app.js**

var express = require('express');

const hbs = require('express-handlebars');

const path = require('path');

var app = express();

app.engine('hbs', hbs({

    extname: 'hbs',

    defaultLayout: 'layout',

    layoutsDir: path.join(\_\_dirname, './views/layouts/')

   }));

 app.set('view engine', 'hbs');

app.get('/', function (req, res) {

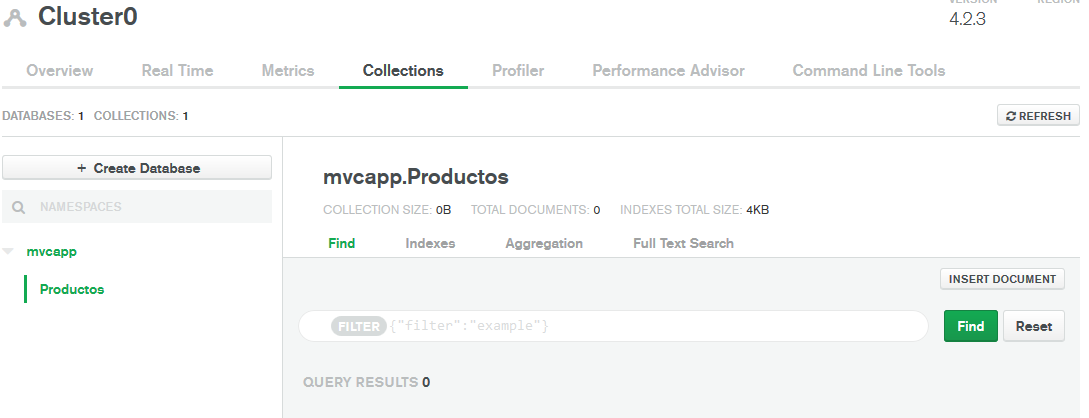
    res.render('home', { title: " aquí pon un mensaje "});

});

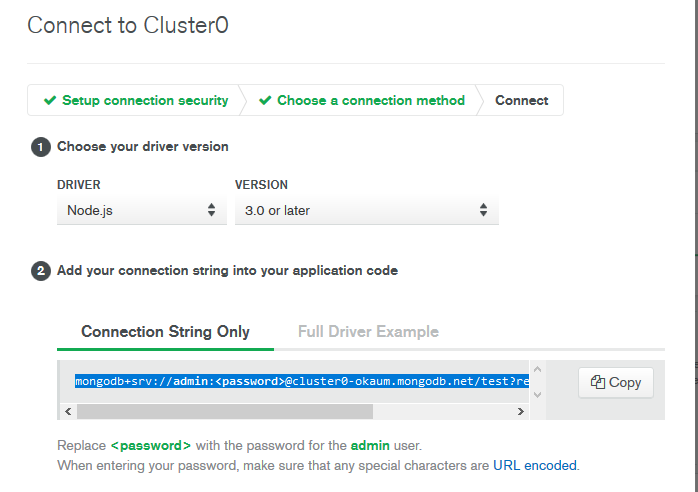
app.listen(3000);

# Conexión con la base de datos

**Muestra pantalla de tu base de datos en mongodb atlas**



**Muestra la ruta de conexión a tu base de datos**



# Configuración de variables de entorno

**Pegar código de package.json donde usas cross-env**

  "dev": "cross-env NODE\_ENV=development nodemon app.js"

**Pegar código de como usas config para conectarte a la base de datos**

if (process.env.NODE\_ENV !== 'production')

     require('dotenv').config();

module.exports = {

    port: process.env.PORT || 3000,

    env: process.env.ENV || 'development',

    dbUser: process.env.DB\_USER,

    dbPassword: process.env.DB\_PASSWORD,

    dbName: process.env.DB\_NAME,

    dbCluster: process.env.DB\_CLUSTER,

    get dbUrl() {

        return `mongodb+srv://${this.dbUser}:${this.dbPassword}@${this.dbCluster}.mongodb.net/${this.dbName}?retryWrites=true&w=majority`

    },

    //si se desea algo especifico

    production: {

        //aquí se pone la configuración de producción

    },

    development: {

        //aquí se pone la configuraciópn de development

    }

};

const mongoose = require('mongoose');

const URL =  require('../config/config.js').dbUrl;

console.log(URL);

mongoose.connect(URL, {

 useNewUrlParser: true,

 useCreateIndex: true,

 useUnifiedTopology: true

}).then(()=>{

 console.log("Connected to database");

}).catch((err)=>{

 console.log("Not connected to database", err);

});

module.exports = mongoose;

**Código de .env**

PORT=3000

ENV=development

DB\_USER=admin

DB\_PASSWORD=jBpsMtN009Wf7eDO

DB\_NAME=mvcapp

DB\_CLUSTER=cluster0

# Creación de rutas

**Muestra el código de tus rutas de ambos archivos**

const router = require('express').Router();

router.get('/', async (req,res)=>{

    res.send("favorites");

});

router.post('/:id/', async (req, res) => {

    res.send("favorite id");

   });

module.exports = router;

const router = require('express').Router();

router.get('/', async (req,res)=>{

    res.send("products");

});

router.get('/:id/', async (req, res) => {

    res.send("product id");

   });

module.exports = router;

**Muestra el código donde importas los routers en app.js**

const favoriteRouter = require('./routes/favorite.route');

const productRouter = require('./routes/product.route');

app.use('/favorite',favoriteRouter);

app.use('/product',productRouter);

# Creación de modelos

**Muestra el código de DB donde haces test de tu base de datos**

Respuesta:

**Pega el código de tu modelo de Producto**

Respuesta:

# Creación de controladores

**Muestra el código del controlador completo**

Respuesta:

**Muestra un screenshot del json que regresa**

Respuesta:

# Crear vista

**Muestra el archivo hbs**

Respuesta:

**Muestra screenshot de los productos**

Respuesta:

# Creación de middlewares

**Muestra código de middlewares que validen los query params o los route params (existencia del id)**

Respuesta

# Completar para favoritos

**Muestra los códigos relacionados**

Respuesta

# Conclusiones

Pon las conclusiones de la práctica