Documentación

Documentación del proyecto

Damián Ivanov Kotchev

2020

Contenido

[Parte de BBDD 2](#_Toc56250943)

[Lo que se nos pide: 2](#_Toc56250944)

[Lo que hemos hecho: 2](#_Toc56250945)

[El esquema de validación: 2](#_Toc56250946)

[El insertOne de Prueba: 3](#_Toc56250947)

[Mongodb Compass: 4](#_Toc56250948)

[Parte de SSII 6](#_Toc56250949)

# Parte de BBDD

## Lo que se nos pide:

Crear los usuarios y permisos correspondientes para poder interactuar con la base de datos de maneras diferentes (guest, admin).

Crear una base de datos Mongodb en el MongoDB Atlas

crear una colección y datos para poder guardar un esquema de validación de como se deberían guardar los datos…

Y conectarnos por Mongodb Compass para hacer algunas simples querys.

## Lo que hemos hecho:

### Problemas:

Hemos tenido un problema con los usuarios, pensábamos que se podían hacer desde el Shell, pero no, te obligan a hacerlo desde la página web de MongoDbAtlas.

### Usuarios:

Captura de pantalla de computadora

Descripción generada automáticamenteHemos creado dos usuarios (guest, admin\_amenities). El usuario “guest” solo podrá hacer búsquedas de la colección “packs” de la base de datos “amenities” mientras que el usuario “admin\_amenities” solo podrá hacer mantenimiento de CRUD sobre la base de datos de amenities, no podrá acceder a ninguna otra base de datos y ni siquiera saber si hay otras, su “scope” solo será para este proyecto.

### El esquema de validación:

El esquema de Validación lo hemos creado después de planificar por nosotros mismos los datos que queremos guardar en la base de datos, lo hemos probado en MongoDbAtlas y funciona, ya está puesto a producción y nos haría falta crear el JSON de la parte de la programación que se ajuste al esquema, introdujimos un insert en la colección packs como referencia a como debe estar guardado en futuras ocasiones.

Código:

db.createCollection("packs", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["PricePack","NamePack","ContentPack","HasCupon","HasParking"],

properties: {

"\_id":{},

"PricePack":{

bsonType: "int",

description: "The price of the Pack"

//debe ser 0 (Free) o x (buy)

},

"NamePack":{

bsonType: "string",

description: "The name of the Pack"

},

"ContentPack":{

bsonType: "array",

items: {

bsonType:"object",

required: ["name"],

properties:{

name:{

bsonType:"string",

description: "este es el nombre del elemento del pack"

},

description:{

bsonType:"string",

description: "esta es la descrición del elemento"

}

}

}

},

"HasCupon":{

type:"boolean",

description:"Para saber si tiene o no descuento el pack"

},

"HasParking":{

bsonType:"bool",

description:"Para saber si el pack es solo para los que tienen parking"

}

}

}

}

});

### El insertOne de Prueba:

db.packs.insertOne(

{

PricePack:NumberInt(0),

NamePack:"Pack de Prueba",

ContentPack:[

{

name:"Chuches",

description:"esto es una bolsa de chuches"

},

{

name:"golosinas",

description:"esto e suna bolsa de golosinas"

}

],

HasCupon:false,

HasParking:true

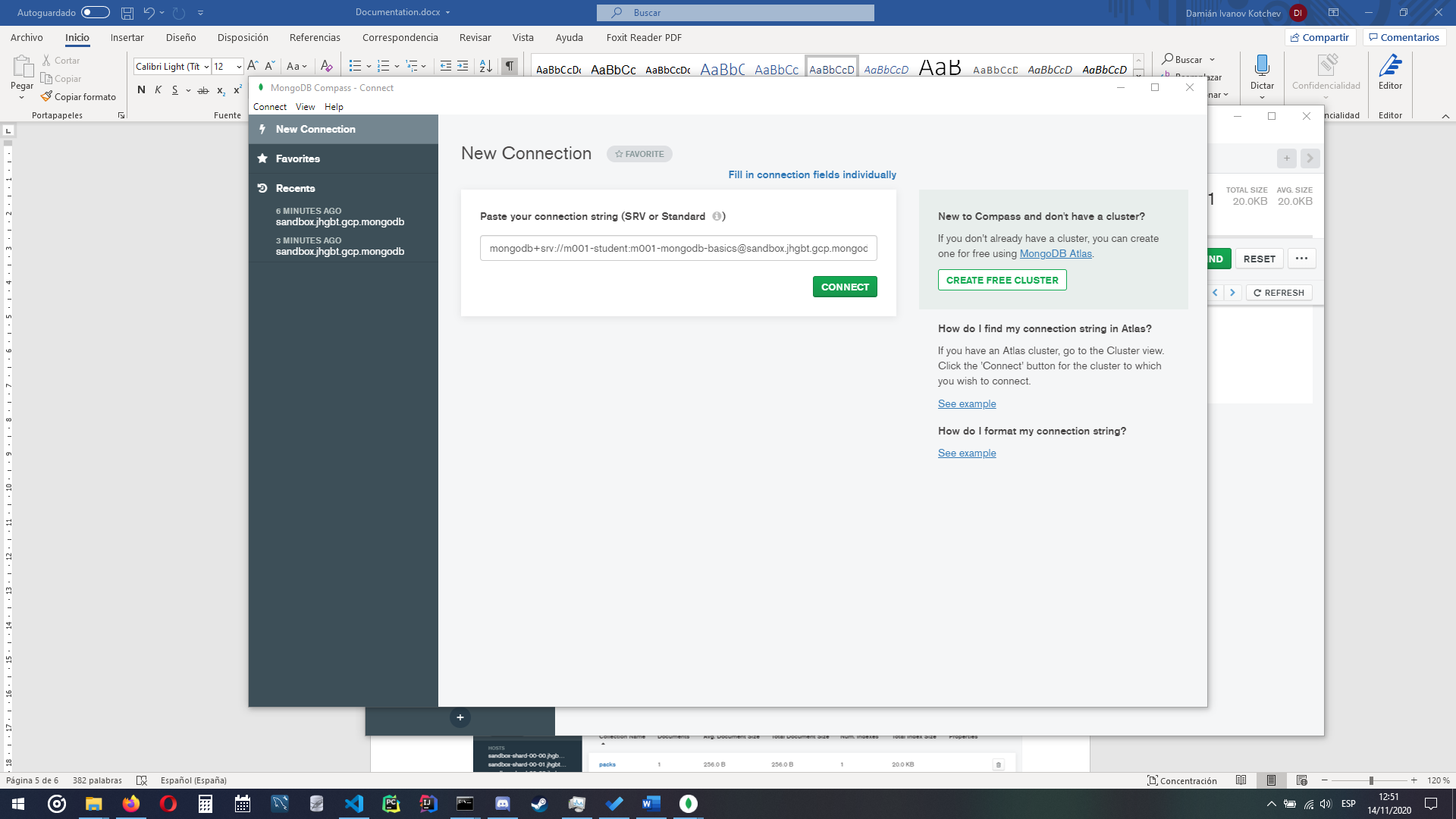
}

)

### Mongodb Compass:

Introducimos esta cadnea de conexión al omngodb compass y le damos a connect

mongodb+srv://m001-student:m001-mongodb-basics@sandbox.jhgbt.gcp.mongodb.net/test



Una vez dentro podemos ver que con el admin del atlas podemos acceder a todas las DBs

Una captura de pantalla de una computadora

Descripción generada automáticamente

Entramos en la DB de “Amenities” y dentro vemos que solo hay una olección de documentos que se llama “packs”

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Si entramos dentro podemos tener una vista rápida de lo que hay dentro guardado

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Si hacemos una simple query viendo que packs tiene cupones nos devuelve ninguno

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Pero si hacemos la query pidiendo lo contrario, es decir, lo que no tienen cupones, nos devuelve el único registro que hay en la base de datos.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

# Parte de SSII:

## Pequeño script de cuentas de GitHub:

Hemos añadido un pequeño script que nos permite cambiar de cuenta de GitHub de Personal a la educativa y viceversa, lo hemos añadido al proyecto como parte de la rama de Sistemas informáticos.

## Aislamiento de código y dependencias:

Hemos aislado el código y sus dependencias para que sean independientes de otros proyecto pasados y futuros para garantizar la modularidad de estos.

Lo hemos hecho con una librería de Python llamada “pipenv” y hemos creado el entorno virtualizado de Python para poder trabajar con él desde la CLI.

## Árbol de directorios:

Este es nuestro árbol de directorios de momento se irá actualizando según veamos conveniente.

Proyecto.

├───apuntes

│ └───BBDD

├───docs

│ └───CapturasDePantalla

├───PaginaWeb

│ ├───audio

│ ├───css

│ ├───html

│ ├───img

│ └───video

├───Scripts

├───src

├───test

└───venv