

■ ENTREGABLE 2

Primera toma de contacto con Crear y Leer documentos, conexión con la cuenta en

MongoDB Atlas

Fecha: 05/03/2021

1

■ TEST ENDPOINTS

Testear los primeros endpoints:

- Leer documentos (GET)
- Crear documentos (POST)

Estimación **4 puntos**

2

■ CONEXIÓN ATLAS

- Registrarse en Atlas Mongo DB
- Crear un cluster
- Conectarlo a la app a través de Mongoose

Estimación **2 puntos**

■ ENTREGABLE 3

Creación de los modelos de la base de datos y respectivos endpoints para su manipulación

Fecha: 11/03/2021

module.exports

require()

1

■ ESQUEMAS Y MODELOS

Dentro de la carpeta de modelos:

- Crear un archivo .js por cada modelo
- Montar el esquema con mongoose.schema
- Exportar el modelo

Estimación **4 puntos**

2

■ ENDPOINTS

Dentro de la carpeta de rutas:

- Crear un archivo .js por cada conjunto de rutas de la misma "familia"
- Utilizar express.Router() para montar las rutas get, post, update y delete que convengan
- Exportar el Router

Estimación **6 puntos**

index.js

module.exports

require()

EJEMPLOS MODELO

Acceder al *Schema* de
Mongoose

Construir un Schema

```
models > JS cityModel.js > ...  
1  const mongoose = require("mongoose");  
2  const Schema = mongoose.Schema;  
3  
4  const citySchema = new Schema({  
5    name: { type: String },  
6    country: { type: String }  
7  });  
8  module.exports = City = mongoose.model("City", citySchema);  
9
```

Exportar el modelo
para utilizarlo en otros
archivos

Conectar el Schema a su
respectiva collection en la DB
mediante el *model*

EJEMPLOS ENDPOINT

Crear la ruta con
express.Router()

Requerir el modelo
previamente creado y
exportado

```
api > JS cityRouter.js > ...
1  const express = require("express");
2  const cityRouter = express.Router();
3  const City = require("../models/cityModel");
4
5  cityRouter
6    .route("/")
7    .get((req, res) => {
8      //it finds all the cities
9      City.find({}, (err, cities) => {
10        if (err) {
11          res.status(400).send(err.response.data);
12        }
13        res.json(cities);
14      });
15    })
16
17  module.exports = cityRouter;
18
```

Lógica de la ruta

Exportar la ruta para
utilizarlo en otros
archivos