Actividad: Diseño de API con base de datos MongoDB

Objetivos

A través de esta actividad vamos a poder establecer los principios básicos de una arquitectura de aplicación compleja para una aplicación generada a partir del framework Express. Además, nos va a ayudar a entender cómo podemos trabajar con los diferentes métodos y acciones disponibles dentro de una base de datos generada a partir de MongoDB.

Es importante conocer qué ficheros y directorios nos van a servir para afrontar de la manera óptima el desarrollo de aplicaciones complejas en las cuales incluyamos interacciones dinámicas. Cuando definimos diferentes urls y diferentes manejadores para dichas urls debemos establecer patrones de trabajo para todos ellos y así poder centrarnos más en qué hace cada manejador y no en cómo lo hace.

Pautas de elaboración

En esta actividad vamos a generar una aplicación con ExpressJS que nos permita realizar todas las acciones básicas (CRUD) sobre el grupo de datos que conforman los inmuebles de una finca.

Para ello debemos contar con la conexión a una base de datos MongoDB dentro de nuestro ordenador para poder acceder a ella de manera local.

El proyecto debe cumplir con las siguientes pautas:

* Debemos generar un **modelo** para cada uno de los inmuebles, con los siguientes datos: **piso, letra, extensión (m2), número de habitaciones, alquilado (sí/no), nombre propietario, mail de contacto.**
* Dentro de dicho modelo se debe establecer el nombre que tendrá la colección dentro de la base de datos. Para las acciones relacionadas con el modelo debemos usar la librería **Mongoose**.
* Debemos generar las rutas necesarias para poder crear, borrar, listar y actualizar elementos del modelo anterior tomando como base la **url/api/inmuebles.**
* Nuestro api debe seguir una serie de buenas prácticas que nos permitan en un futuro poder ampliar sin excesivos problemas la funcionalidad de nuestra aplicación.
* Se debe además comprobar cualquier tipo de error que suceda relativo a la interacción con la base de datos en cualquiera de las acciones llevadas a cabo.

Extensión y formato

Hay que recordar generar el fichero **.gitignore** en la creación del proyecto para así evitar la subida de directorios innecesarios al repositorio con el que vamos a trabajar.

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Diseño de API con base de datos MongoDB | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Se ha generado correctamente el fichero con la estructura de carpetas necesarias para llevar a cabo el desarrollo del API. | 0,5 | 5% |
| Criterio 2 | Se ha generado un fichero **inmueble.js** dentro de la carpeta models con la estructura que define cada uno de los documentos a crear dentro de la base de datos. | 1 | 10% |
| Criterio 3 | Se ha generado un fichero inmuebles.js dentro de la carpeta routes y además se han realizado todas las tareas que lo unen con **app.js.** Las rutas deberían funcionar. | 1 | 10% |
| Criterio 4 | Se dispone de una ruta que nos devuelve en formato JSON todos los inmuebles contenidos dentro de la base de datos. | 1 | 10% |
| Criterio 5 | Se dispone de una ruta que nos permite crear documentos dentro de la colección Inmuebles de nuestra base de datos. | 2 | 20% |
| Criterio 6 | Se dispone de una ruta que nos permite actualizar cualquier tipo de documento, recibiendo a través de la misma el documento a actualizar. | 2 | 20% |
| Criterio 7 | Se dispone de una ruta que nos permite eliminar cualquier tipo de documento recibido a través de dicha ruta. | 1,5 | 15% |
| Criterio 8 | Todas las peticiones tienen sus errores recogidos y se informa de manera correcta al usuario de lo sucedido. | 1 | 10% |
|  |  | **10** | **100 %** |