API REST NodeJs

Unidad 1

Angela Maria Bastidas Vergara

Carlos Andrés Cardona Quintero

Ingeniería web II

Profesor: Julio Cesar Martínez

IU Digital de Antioquia

Medellín

Antioquia

2022

**Introducción**

En este documento se presenta una breve explicación de la realización del proyecto, su estructura y cómo se llevó a cabo su funcionamiento. Se pretende llevar a la práctica los conceptos aprendidos y recomendaciones vistas en clase.

Está la carpeta modelos, con cada uno de los modelos de la base de datos, la cual contiene el nombre, el serial y demás modelos que se habían sido indicados. En la estructura del proyecto casi todos son iguales, aunque los modelos de usuarios e inventario tienen algunas diferencias.

El servicio del modelo Inventarios se diferencia principalmente por el hecho de que todos los esquemas son anidados.

El servicio de usuario, se diferencia de los demás porque en éste se incluyen los campos de email y nombre.

Dentro de la carpeta rutas, explicaremos básicamente en qué consiste el inventario, debido a que, consideramos, es la carpeta primordial en este caso (sin restarle importancia a las demás, evidentemente), pero nos detendremos principalmente en ella.

En inventario, se ve que en el código se está declarando una constante llamada Router, dicho Router es de ‘express’, ya que tenemos la constante de Router; se procede, entonces, a hacer la petición de tipo get, en la petición get se está diciendo que con la url que proporciona, se hace un llamado async en la que se pone un requerimiento y una necesaria respuesta.

La función que desempeña el try, es decir, lo que se está haciendo con el método find, es que va y busca todos los inventarios que están insertados en la base de datos. El populate sirve para, aparte del id, dar unos esquemas anidados y muestra, también, los otros atributos de los otros atributos de servicio.

El método post, sirve en este caso para guardar el inventario nuevo y aquí cambia; ya no es por medio del método get sino post, que es para guardar. Se pasa la ruta en donde se va a guardar y salimos. Esta es una función async porque tiene que hacer una petición al servidor.

En el try declaramos una variable cuya intención es validar si existe el serial, puesto que no puede haber seriales iguales en la base de datos. Se valida mediante un if.

Después, se declara una instancia nueva de tipo inventario, y a ella se le pasan los parámetros que están recibiendo por la req y le decimos inventario = await, para que espere a que se guarde esa instancia de inventario. Posteriormente, se envía el inventario como respuesta.

El método put, lo que hace es buscar el id del inventario que se quiere actualizar y, de no existir, al usuario le aparece una alerta que dice ‘’no existe’’. Sin embargo, en caso de ya exista el serial, como en el paso anterior de guardar, le aparece al usuario que ‘’el usuario ya existe’’, y es de aclarar que, en este caso en particular, no se declara una nueva instancia de inventario debido a que esta ya existe.

**Conclusiones:**

En el presente trabajo, tuvimos un acercamiento a Node.js y a algunas de sus funcionalidades. Exploramos diversas utilidades que ofrece esta herramienta en el ámbito del desarrollo de software y llevamos a cabo un proyecto funcional, en el cual nos apropiamos de cierta manera de un sistema gestor de contenidos (CMS), el cual, sin duda, nos será de gran ayuda para proyectos futuros.

Entendimos, además, la manera en la cual los CMS se usan en proyectos con los cuales nos relacionamos, como por ejemplo el comercio electrónico y la manera en que entender y profundizar en estas herramientas nos convierte en desarrolladores más capaces y con una gran capacidad tanto transformadora como mercantil.