

A veces no alcanza con la información en tipo digamos de texto que se nos entrega a través de un formulario. A veces queremos, también, cargar archivos: PDF, archivos de música, de vídeo, imágenes. Bueno, para hacer esto tenemos que aprender algunas técnicas muy puntuales. Comenzaremos en esta clase.

Vamos a cortar nuestro flujo de trabajo aquí. El servidor lo vamos a cerrar, porque tenemos que instalar una serie de paquetes para lo que viene a continuación. Son, si mal no recuerdo, tres. Bueno, luego podemos agregar o quitar, si es que nos equivocamos pero, vamos a comenzar. Creo recordarlos bien, uno es para el "file upload", es una dependencia propietaria de "express". Así que vamos a instalar "express file upload" que nos va a permitir así, como ahora leemos el objeto "request", particularmente, el objeto "body" de la "request", vamos a poder leer también los archivos que vengan en una carga de formulario, por lo tanto, en algún campo de formulario, ¿verdad?

En un campo "input" del tipo "file" para hacer eso, necesitamos esta dependencia "express file upload" también vamos a necesitar "cloudinary". Esta dependencia nos permitirá conectarnos con un servidor que sirve para almacenar las imágenes en un sitio distinto al de nuestra base de datos. Para tener en la base de datos, solamente una ID que identifique a esa imagen, para hacer nuestra base de datos principal más liviana. Para hacer esa conexión entre las dos bases de datos, necesitamos a "cloudinary" y también, vamos a necesitar un paquete que se llama "dot env" o "punto ENV", sirve para leer las variables de entorno y en las variables de entorno vamos a almacenar una clave que nos permitirá conectarnos con "cloudinary". Ya verán todo esto, por supuesto, simplemente se los quiero ir comentando, para que no estén oscuras.

Ahora veremos en detalle cómo funcionan y cómo se configura.

Muy bien, tenemos nuestras dependencias instaladas, tenemos "express file upload", tenemos "dot env" y tenemos "cloudinary". Hablando de "cloudinary", vamos a ir al "cloudinary" y les voy a mostrar dónde está y haremos una cuenta allí, nos registraremos, es totalmente gratuito. Quiero, quiero advertirles esto fíjense aquí, tenemos el sitio de "cloudinary", es un servidor, ¿verdad?, en la nube, para almacenar información. Pueden ser imágenes, bueno, archivos, tiene un servicio muy bueno y tiene una versión gratuita que es la que vamos a utilizar. Ustedes pueden aquí registrarse gratis, yo tengo ya una cuenta, por lo tanto sólo voy a loguearme. Aquí estoy dentro de "cloudinary" y necesitamos estos datos que son los datos de seguridad para poder utilizar la aplicación, podemos copiarlos uno por uno e ir activando los en cada instancia o mucho mejor aún, copiar esta secuencia en nuestras variables de entorno. Vamos a copiar esto, click y nos los llevamos. ¿dónde los llevamos?, a un archivo que vamos a crear aquí. Ese archivo va a llamarse "punto ENV", igual que la dependencia que instalamos "dot env". "Dot" en inglés significa punto ¿verdad? Así que esto es todo en "dot env" o "punto ENV", el mismo nombre que la dependencia que instalamos recién, ahí está, ¿la ven? Bueno y aquí dentro solamente vamos a copiar nuestras credenciales para poder utilizar el servicio de "cloudinary". Con esto estamos bien, solamente falta un paso para usar correctamente "dot env" y es indicar en nuestro archivo de entrada, en nuestro "A, punto, js", bueno, que vamos a habilitar a nuestra aplicación para que lea las variables de entorno, por lo tanto, en este nivel vamos a requerir "dot env" y vamos a indicarle "config" solo eso, ya está, no hace falta más, nada que esto. Muy bien, tenemos resuelto el tema de los permisos ahora vamos a configurar nuestro "express file upload", es decir, el paquete que acabamos de instalar y que nos permite leer archivos que provengan de un formulario, vamos a venir, bueno, en cualquier sitio está bien, aquí, en la sección en

que hacemos la carga, ¿no?, vamos a ponerlo acá, abajo debajo de “expreses sesión”, entonces “const file upload” será igual a, bueno esto, vamos a requerir el paquete y aquí ya nos aparece el nombre, ¿oka?. Muy bien, ahora tenemos que configurar este servicio para utilizarlo, al igual que hicimos con “session”, por ejemplo, también tenemos la documentación. Fíjense que acá están, tenemos abiertos acá, miren, acá está "cloudinary", si quieren ver la documentación, la pueden buscar en “npm” "cloudinary". Después tenemos "express file upload", aquí están las configuraciones y también tenemos “dot env”.

Recuerden, siempre está muy bien ir a buscar la documentación de los paquetes que usamos, aunque sea para investigar, tómenlo como costumbre. Una vez que lo vamos a necesitar ahora, ¿oka?, yo los voy a guiar, pero sepan que hay que hacer esto, siempre está muy bien hacerlo. Bueno, entonces vamos a configurar nuestro middleware. Vamos a ponerlo por acá abajo, vamos a instalar acá el middleware "express file upload". Recuerden, este sirve para leer nuestros archivos, los archivos que, bueno, vamos a subir en un formulario, en algún formulario. Vamos a usar entonces acá “file upload” y esto recibe un objeto de configuración con estos valores. Estos son valores que están en la documentación, ¿cierto? Muy bien y mientras escribo esto, puedes leer arriba, aquí, el comentario y qué es lo que hace esta, qué hace, qué hacen estas dos propiedades básicamente bueno hacen que sea más eficiente el uso de nuestro sistema, ya que para optimizar los recursos, trabaja la carga de archivos en archivos temporales y no utiliza tanto la memoria ram de nuestro sistema ¿verdad? Por lo general, siempre tenemos mucha más memoria física en disco que memoria volátil en ram, "file dir" y bueno, acá le vamos a indicar el nombre que viene por defecto. Acá hay que indicar nada más que una carpeta, se puede poner cualquier nombre, este es el nombre que viene por defecto para crear la carpeta temporal. Bueno, ya está, esto está listo. Tenemos configurado entonces nuestro "file upload" de "express". Venimos a nuestro formulario de administración, donde cargamos los datos. Aquí me ha adelantado un poco y creé el formulario fuera de clase. Es un formulario como cualquier otro ¿verdad? Recuerdan, ya hemos hecho unos cuantos, pero tiene una diferencia con otros formularios, miren el de contacto, por ejemplo, no tiene tipo de codificación, no tiene un “enctype”, fíjense que aquí hay una un atributo que se llama en “enctype” y que puede tener estos datos, estos tres, estos tres tipos de codificación, texto simple, este es el valor que viene por defecto, si no ponemos nada, este será el valor, pero cuando trabajamos con archivos, estamos obligados a introducir este valor “multipart form data”, esto permitirá que nosotros trabajemos con carga de archivos en un formulario y que ese archivo pueda ser enviado al back-end para su proceso. Bien, entonces, recuerden, aquí donde no poder, no ponemos un tipo de codificación un “enctype” o un “encoding type”, bueno, el valor, es el valor por defecto, pero acá, vamos a tener que cargar un archivo de imagen, no importa qué tipo de archivo, sea o que vamos a tener que cargar un archivo, en este caso será la imagen, por lo tanto, hemos puesto el tipo de codificación en “multipart form data” y está muy bien esto, ya está preparado para trabajar. Simplemente, quería mostrarles este atributo para que lo tengan en cuenta y aquí cargaremos, lo que vamos a cargar acá son razas felinas, tenemos un formulario para cargar razas de gatitos o de gatos, tenemos un campo que se llama “raza”, otro “características”, uno que se llama “detalle” y luego, uno para la imagen. Bien, esto va a ir todo a una base de datos, ¡ajá!, dije bases de datos, pero no la hemos creado todavía, así que, ¡vamos, manos a la obra, a crear nuestra base de datos!. Vengamos a aquí, a nuestro, ¿dónde podemos venir?, acá está nuestra base de datos, se llama “kittens”, ya la estamos usando, ¿recuerdan? Tenemos que crear una nueva tabla, la tabla se llamará “razas”, pronto, muy bien, continuar. Bueno, voy a crearla directamente, me la voy a venir acá, vamos a

hacerlo a mano. Mejor ¿verdad?, así practicamos un poco de SQL y no nos olvidamos. "Create table", "razas" y acá vamos a indicar, tendrá un ID que será de tipo entero, y vamos a indicarle "auto-increment", por lo tanto será nuestra clave primaria. Podríamos indicarle aquí la coma, luego tendremos un campo "raza" será de tipo "varchar", ¿cuánto le ponemos? 20 es demasiado, pero bueno, un valor y vamos a indicar un "constraint" que sea de tipo "unique", que no puede haber dos razas que se llamen igual, ¿cierto? No tiene ningún sentido. Bueno, y también tendremos características, sin acentos, obviamente aquí, "características", será de tipo "texto" porque bueno, puede ser un poco más largo que un texto "varchar" común y, también, tendremos "detalle", también será de tipo "texto".

Finalizaremos con imagen. En imagen vamos a almacenar una cadena de caracteres que será el ID que nos va a devolver "cloudinary" y vamos a almacenar la imagen en "cloudinary". De esa imagen vamos a obtener una suerte de clave de acceso, a una ID y ese es el que vamos a poner acá. Así nuestra base de datos principal se mantendrá muy liviana, muy rápida, al no tener una imagen cargada ¿verdad? Bueno y acá vamos a poner que esto es de tipo "varchar", vamos a darle, vamos a darle 255, nos va a sobrar, pero mejor así. Bueno, aquí tengo un problema y es que creo que aquí tengo que indicar, para que un campo sea auto-incremental, debo indicar que va a ser un campo, del campo de tipo "primary key", efectivamente, hay que ponerlo, no es automático, bueno, ya tenemos lista nuestra tabla. Vamos a describirla a ver si es como nosotros queríamos. "Describe razas" y tenemos el campo ID y auto-incremental, clave primaria, el campo "raza" que es de tipo "unique" y bueno, y acá tenemos además características detalle, imagen, muy bien. Podríamos modificarlos para que no sean, para no permitir que estos campos sean nulos, que no tengan valores nulos, pero bueno, vamos a hacerlo, pero vamos a hacerlo de acá, así cambiamos un poco también. Vamos a razas, estructura y acá vamos a indicar, vamos a editar nuestra estructura de la tabla, por ejemplo, "raza no puede ser nula", aquí está, listo. Tampoco puede ser nulo características, bueno, ningún campo en realidad. No lo olvides, sinceramente, si no lo habríamos hecho cuando definimos la tabla, hubiera sido más rápido, pero bueno, son poquitos campos, así que no va haber problema. Media está perfecto, tenemos nuestra, y a cada imagen, bueno, tampoco puede ser nula, ¿o sí?, no, vamos a dejarlo así, que no puede hacerlo. Bueno, ningún campo puede ser nulo. Me olvidé de cambiar detalles, miren y acá en detalle, vamos a indicar que tampoco puede ser un campo nulo.

Bueno, tenemos preparada nuestra base de datos para interactuar con el sistema. Bueno, ¿qué nos faltaría ahora? Ahora nos faltaría directamente crear una query, una consulta para poder cargar datos en nuestra base de datos. Imágenes en la base de datos de "cloudinary" y luego la información completa con el ID para acceder a esa imagen en nuestra base de datos MySQL. Eso, lo haremos a continuación.