

Ahora vamos a hacer algo muy interesante, también muy lindo, muy interesante. Tiene su complejidad pero lo vamos a superar, como hemos superado todos los desafíos hasta ahora. Vamos a aprender el concepto de rutas con parámetros, cómo buscar un registro cuyo ID, cuyo identificador, va a venir en un parámetro, en una ruta con parámetros. Lo vamos a recoger en otro objeto, ¿recuerdan que teníamos “rec punto body”, el cuerpo de la request o de la petición, “rec punto files”, los archivos que venían en la petición, todo esto es de un formulario, pero tenemos un objeto que no captura datos de un formulario, sino, directamente de la barra de navegación y se llama “rec params”, podríamos incluir ahí un número, el número de un registro que queremos buscar. Efectivamente, buscarlo a través de nuestro modelo de datos y después, por ejemplo, borrarlo. Estos desafíos son los que vamos a enfrentar a continuación.

Bueno, vamos a abrir nuevamente nuestra tabla de administración, para ver dónde estamos parados y para tratar de avanzar, recuerden, nuestro próximo paso será lograr el borrado de un registro de nuestra base de datos, pero también de la imagen que tiene asociada en la base de datos de Cloudinary, es decir, vamos a, como cuando agregamos, hacemos una doble operación, haremos lo propio para borrar. Ahora les voy a explicar por qué. Ven, veamos, si se ve el listado, bueno, aquí hemos emprolijado un poquito el listado, aquí podríamos eliminar este gatito, pero les quiero mostrar lo siguiente: cuando agregamos un registro, vamos a agregar cualquiera aquí, bueno, no importa la información, porque esto lo vamos a borrar, así que va a durar poco. Perdón, le puse enviar y hay que cargar sí o sí una imagen. Vamos a cargar este gatuno, le ponemos enviar y si ahora vamos al listado, veremos que tenemos dos cargados y en Cloudinary y van a ver que ahora si refresco, va a aparecer la imagen del gatito siamés, perfecto. Nosotros tenemos que borrar el registro, aquí con el botón eliminar, pero tenemos que lograr que se borre el registro de la base de datos, acá sí actualizo también va a aparecer, acá, miren, pero a su vez, queremos que se elimine la imagen de Cloudinary, del servidor, para no ir dejando migajas detrás nuestro, porque si bien este es un servicio gratuito, ahora que estamos usando poco espacio en un contexto real, profesional, necesitamos mucho más espacio y el espacio se cobra, por lo tanto, no es óptimo ir dejando imágenes que ya no necesitamos, ni imágenes, ni archivos de ningún tipo, ¿cierto?

Aclarado esto, vamos a ver si podemos hacer esta operación. Tendríamos que crear acá una función, en realidad dos funciones, vamos a hacer una función para leer una raza, independientemente una, una sola, a la que le pasemos un ID y otra para borrar, así que vamos a poner acá “async function borrar”, no, “leer raza”, cuidado con esto, es en singular, “no leer razas”, una sola, pero va a recibir un parámetro, va a recibir el parámetro ID. Bueno, noten que a veces hacemos las “arrow functions”, otras veces hacemos funciones tradicionales, es simplemente para darle un poco de variedad y que ustedes se vayan acostumbrando.

Aquí dentro de un bloque “try catch”, bueno, el error simplemente lo vamos a lanzar por consola, si es que existe y vamos a hacer nuestra función para traer un registro particular, puntual.

Vamos a construir nuestra query, que va a ser igual a, bueno, vamos a tirar a cada sintaxis de SQL, vamos a hacer un “select asterisco from razas”. No hay que olvidarnos del “where”, “where ID” de igual y aquí vamos a capturar el ID que entra por parámetros. Tan sencillo como eso. ID, perfecto. Luego solamente nos falta lanzar hacia nuestra base de datos nuestra query. Vamos a almacenar el resultado en esta fila. ya que solamente hay un

resultado posible. porque en nuestra base de datos tenemos el campo ID y como clave única, auto-incremental, por lo tanto, sólo vendrá un resultado. De todas maneras, si ustedes quisieran asegurarse, no es necesario, pero sólo para tener presente que esto puede hacerse, bueno, limitamos los resultados a uno solo, al primero que encuentre, reitero, esto es innecesario, pero podemos mostrarlo y aquí vamos a aguardar el resultado de la consulta "pool query" y hagamos la consulta. Oka. Acá esto nos va a retornar, aquí abajo, vamos a indicar el return, un resultado puntual, con esto, esta función sirve para buscar una raza puntual, es decir, un registro, en realidad. No tenemos que olvidarnos de exportarlo y acordarnos que es en singular y luego vamos a construir la función, esta la vamos a hacer como "arrow function", es indistinto, borrar raza y esta función va a ser también de tipo asíncronico, porque estamos hablando de consultas a bases de datos y va a recibir el ID, porque hay que borrar una puntual, la que nosotros le indiquemos, okey, en ese sentido, funciona bastante parecido a la función de arriba, de más arriba, solamente que la otra busca un registro, en tanto que ésta lo borra. Otra vez vamos a declarar un bloque "try catch", en el que vamos a lanzar nuestro error a la consola, en caso de que exista, para tener un mínimo control de él y aquí dentro del Drive, bueno, vamos a hacer otra "query". Miren voy a copiar voy a copiar esto, porque es muy parecido lo copio, lo ponemos acá, solamente que acá no vamos a hacer un "select", sino que vamos a hacer un "delete" y que vamos a borrar. Vamos a borrar, de razas, "where ID", también, igual al ID, esto no es necesario tampoco, bueno, esta vez, lo quitamos y luego vamos a retornar el resultado de la fila borra, aquí. Vamos a retornar. Perfecto, vamos también a exportarlo para poder utilizarla luego. Entonces esta se llama "borrar raza", oka, estan exportadas. Ahora nos falta poner esto en funcionamiento. Recuerden que este procedimiento lo hacemos en nuestro controlador "listado".

Vamos a ir al listado acá y vamos a, bueno, vamos a definir. Vamos a necesitar "util", así que me lo voy a traer, me voy a traer "util", "const util, igual, require util", recordamos que no es necesario instalar este paquete, es una dependencia de núcleo de "Node", por lo tanto, no hace falta traerlo. Y además, nos vamos a traer, miren, yo recuerdo que aquí, en algún lado, donde era en "admin", acá está cuando definimos las funciones de Cloudinary, definimos la función de cargar y de borrar, pero la de borrar no la hemos usado nunca y como no la vamos a usar acá, nos la vamos a llevar para ser más eficientes, no dejemos esto aquí, que nadie lo usa. Lo vamos a usar, acá sí la vamos a necesitar, por lo tanto, vamos a implementar acá el método "destroy" de Cloudinary, donde lo vamos a usar. Bueno, miren, vengamos acá. Vamos a venir acá debajo de nuestro "get", vamos a traer una, acá vamos a recibir un parámetro. Les quiero mostrar cómo. Desde nuestro listado acá, vamos a implementar el botón para borrar acá. Este botón vamos a decir que vaya al controlador "listado", "borrar" y va a pasar el ID, éste ID, el ID, de cada una de las filas, ¿okey? Eso es lo que queremos hacer. Esto se llama "ruta con parámetros", cuando nosotros enviamos a una ruta de esta manera, luego de la barra un algún dato puntual, estamos enviando un parámetro que luego podrá ser consultado, manejado por esa ruta como parámetro, donde lo recibimos, bueno, tenemos que indicar, primero, el método. Recuerden, tenemos que ver de qué método se trata, acá estamos ante un método "get routers punto net", pero, ¿en qué ruta? En la ruta "borrar" de nuestro archivo listado, recuerden que hoy nosotros acá, el listado, indicamos que vamos al archivo listado, borrar, con el parámetro de la ruta. Muy bien, entonces, ahora acá, en el listado, no es necesario poner el listado, porque ya estamos en el archivo, simplemente indicamos "borrar" y para indicar que esta ruta recibe un parámetro, ponemos dos puntos y el nombre del parámetro que recibe. Va a recibir el

parámetro "ID" y ese es el que va a enviar, primero para buscar un registro y después para borrarlo. Esto, por supuesto, será asíncrono, ya que vamos a trabajar con bases de datos, o a recibir. Como siempre, aquí el objeto "request" y "response" y vamos a hacer aquí adentro tres cosas. Lo primero es ir, vamos a declarar una variable, una constante que se llame "row" y que va a recibir el registro de nuestra base de datos MySQL. Vamos a ir a buscarlo con la función, vamos a indicar "await", por supuesto, porque es un método asíncrono, recuerden, el nombre del archivo donde están todos los modelos y luego, el nombre de la función, "leer raza", recuerden, una sola y ¿dónde está el ID que viene desde la vista?, bueno, esos parámetros viajan en la request, puntualmente en un objeto que se llama "rec punto params" y dentro de "rec punto params" nos tenemos que preguntar por el ID, porque ese es el nombre que le hemos puesto cuando llegamos a esta ruta. Por lo tanto, vamos a poner "rec punto params punto ID". ¿Qué nos va a traer esto? Esto nos va a traer de nuestra base de datos MySQL, el registro que corresponde con este ID y ahora, con ese dato, queremos hacer dos cosas, primero, borrar la imagen de Cloudinary, para no dejar detrás nuestros recursos que ya no estamos utilizando.

Así que como es un proceso asíncrono, vamos a guardar que destruya con el método "destroy" y aquí recuerden, que nosotros estamos recibiendo la consulta de nuestra base de datos, en esto que termina siendo un arreglo, es "row", por lo tanto, si es un arreglo, vamos a ir a la posición cero, al índice cero, ahí está el único registro que trae y, ¿cómo se llama el campo que contiene la ID pública de la imagen que está almacenada en Cloudinary? Se llama "imagen". Recuerden, aquí estamos leyendo nuestra base de datos MySQL, aquí y con esta información, con el ID público, vamos a Cloudinary y borramos la imagen de esa base de datos. Una vez que esto está terminado, borramos efectivamente el registro completo de la base de datos, ahora sí, MySQL. Por lo tanto, vamos a llamar a la función "razas punto borrar raza", que también recibe un parámetro, el mismo parámetro aquí arriba, que está aquí arriba, "rec params punto ID", "rec punto params punto ID", ahí es donde está almacenada la información de la ID de la imagen. Y suponemos que todo va a salir bien, aunque puede fallar, por supuesto. Entonces, luego de esto, nos redirigimos, bueno, al listado. Queremos nuevamente ir al listado para que cargue la información y nos muestre la nueva disposición, evidentemente, si funciona tiene que haber un registro menos.

Bueno, vamos a probar esto. Vamos a nuestra aplicación y bueno, recuerden que habíamos cargado aquí a nuestro gatito siamés, en Cloudinary y lo tenemos y también lo tenemos en la base de datos. Ahora, si todo está bien, deberíamos poder eliminarlo y tendría que desaparecer de ambos sitios. Por las dudas, vamos a volver a cargar esto y nos vamos a volver a registrar. Si venimos al listado, deberían estar los dos y si ahora lo borramos, ¡oops!, se borró, vamos a ver si el trabajo está completo, debería haber desaparecido de Cloudinary la imagen del gato siamés. Vamos a refrescar. Perfecto. Y aquí ya no debería estar el registro del gato siamés. Bueno, ustedes lo están viendo, pero hay que refrescar la pantalla. Perfecto, hemos borrado, hemos borrado el registro de la base de datos MySQL y también de Cloudinary, es decir, de la base de datos en la nube de nuestro servidor dedicado para imágenes. Hemos borrado la imagen correspondiente. Ahora sí, podemos decir que el procedimiento es eficiente.