Nuestra clase anterior ha sido muy interesante, cargada de conceptos. Esta idea de buscar un registro, borrarlo en la base de datos sql, pero antes de hacerlo tomar el public ID y de la imagen para también borrar esa imagen en el servidor de imágenes, nos ha dejado pensando. Es un concepto muy fuerte. Es muy importante también porque nos ayuda a optimizar el espacio. Ahora, en… en uno de nuestros últimos encuentros, vamos a encarar el desafío de modificar un dato. También tendremos que buscarlo, pero eso ya sabemos hacerlo. Y luego, cambiar los datos, bueno, los datos que sea relevante, pertinente, cambiar. No borrar el registro y querer… volver a crearlo porque va a seguir teniendo el mismo id, ni tampoco sobreescribir todos los campos, sino sólo aquellos que necesitamos. Es momento de revisar nuestro código para encarar la última de nuestras operaciones relevantes aquí en el CRUD. Vamos a ver, o k, estamos corriendo nuestro proyecto. Vamos a ingresar

a través del login con un mail válido. Como siempre.

Todo esto funciona, vamos a nuestro listado, y tenemos que agregar aquí, la opción de modificar. Muy bien, vamos a hacer eso.

Perfecto.

Tenemos que dirigirnos ahora a la vista. A la vista del listado. Y acá vamos a agregar un botón, en la vista del listado, para que nos dirija a otro controlador, y así poder borrar. Vamos a hacerlo copiando exactamente esta misma sintaxis. Va a ser igual al botón que tenemos arriba. La única diferencia es que en vez de enviarnos a la ruta el 'listado/borrar/', nos va a llevar a la lista, a la ruta 'listado/modificar/' también con un id, ¿cierto? con el id que corresponda, que tiene que ver con aquel botón al que le damos clic. Recuerden: acá en la vista vamos a tener el botón y va a levantar el mismo id de la fila que corresponda. Vamos a ver si se agregó. Perfecto. Se agregó. Ya tenemos un paso dado. Ahora, lo que nos faltaría es, bueno, volver aquí a nuestro código. Una vez acá en nuestro código, vamos a crear una vista para modificar, para cambiar, nuestro registro. En esa vista es donde vamos a cargar todos los datos que vengan de nuestra base de datos. Vamos a hacerlo así, miren. Aquí vamos a crear la vista. La vista se llama modificar.hbs, y es simplemente una copia del formulario, éste, que tenemos aquí. Es el mismo formulario copiado con alguna modificación que ya les voy a explicar y que tiene que ver con la necesidad de realizar las ediciones. Lo voy a cambiar porque me ha quedado en otra carpeta. Wsto tiene que estar en la carpeta de vistas. Ahora sí, está donde… donde debe estar. Vamos a copiar nuestro texto. Noten ustedes que es exactamente el formulario que nos copiamos de la vista admin.

Lo único que vamos a cambiar acá es que cuando presiona en el botón submit los vamos a mandar a otro controlador, no lo vamos a mandar a listado. Eso lo tenemos que tener en cuenta. Bueno, sí, en realidad acá lo vamos a mandar al listado, en nuestro formulario está acá, pero no lo vamos a enviar a otro sitio que no sea el listado/modificar, con el método post. Perfecto, eso lo vamos a copiar a continuación. Listado/modificar, muy bien vamos a hacer esa ruta ahora mismo. Aquí tenemos nuestras rutas, y aquí vamos a definir nuestro modificar.

Modificar.js. ¿Qué haremos en este controlador? Bueno, este controlador solamente va a renderizar la vista.

Express va a ser igual a

require

Express. Aquí vamos a definir el router. Router igual express con la función router.

Acto seguido, vamos por nuestra ruta get.

other get, raíz, y acá le vamos a pasar nuestra función con la request y la response. Esta vez, no use funciones flecha. Recuerden, podemos ir cambiando. Vamos a hacer el render de la vista que creamos recién. La vista se llama modificar. Bueno, ¿qué nos falta ahora? pasarle como siempre el usuario con la sesión activa: 'req, session.user'. Muy bien, vamos a exportar este módulo: 'module.exports = router'. Antes de que lo olvidemos, porque eso puede ocurrir, ¿verdad? Suele ocurrir, bueno, acá tengo algún error, perdón, me falta 'req.session.user'. Lo había escrito mal. Ahora sí, está perfecto. Vamos a nuestro index, y acá tenemos que importar los enrutadores. Vamos a agregar el de modificar. Modificar, vamos a dejarle la misma nomenclatura, router. Acá el archivo se llama modificar, lo acabamos de crear, ¿verdad? y lo vamos a llamar también aquí abajo del todo, donde están nuestras rutas, y también le vamos a pasar el middleware de autorización para que no puedan ingresar allí si es que no están debidamente autorizados. ModificarRouter. Acá tenemos que ir a la lista la ruta modificar.

Pasamos el middleware que chequea si el usuario está logueado, si tiene verificación, si está autorizado, en cuyo caso lo deja pasar, sino no. Muy bien, vamos a ver ahora cómo podemos continuar con esto. Tendríamos que hacer una función dentro de razas, donde están nuestros modelos, para cambiar un registro. Podríamos ponerle, obviamente, 'cambiarRaza', ¿no? Vamos a definirlo como una función asincrónica, async function. Vamos a usar otra vez la palabra reservada function para ir mostrando alternativas, 'cambiarRaza'. Y acá vamos a recibir dos parámetros. Todos los datos que queremos cambiar porque es probable que digamos este… en el formulario vengan algunos datos que estén iguales y otros que se han sido cambiados- Los recibiremos todos, y vamos a indicar cuál es el id del registro al que nosotros queremos cambiar en la base de datos. Eso también es súper importante. Vamos a poner toda nuestra operación en un bloque try catch como hacemos habitualmente. Aquí vamos a recibir el error y vamos a consolearlo. Vamos a mostrarlo por consola. En caso de que haya uno, definimos nuestra query: query va a ser igual y acá tenemos que usar la sintaxis, no sé si ustedes lo recuerdan, update. Esto es para modificar un registro. La tabla se llama: razas, las razas de nuestros gatitos, y vamos a setear. Escribir un nuevo valor para cambiar en la base de datos. Acá vamos a poner los datos, y en el where, en la cláusula where

lo que vamos a insertar es el id, ¿verdad? Muy bien, lo vamos a hacer en la línea de abajo. Aquí haremos una fila. Vamos a decir que se va a llamar fila como hacemos habitualmente cuando hay registros, row, y esto va a guardar una operación asincrónica porque es una consulta a nuestra base de datos. O sea, es una pool query. Acá tenemos que pasar la consulta propiamente dicha y además entre estos corchetes, vamos a pasar los datos y el id. Ahí va la información que queremos cambiar y aquí, el lugar, el registro donde queremos que esos cambios surtan efectos. Y vamos a retornar, como hacemos siempre, el resultado en una fila. Tendríamos que importar, perdón, que exportar esta

función. Vamos a ver si no lo hemos hecho. No, no lo hicimos aún. Por lo tanto, acá vamos junto a las demás funciones a exportarla, se llama 'cambiarRaza', cambiar raza, y el autocompletado me está ayudando. Perfecto, ahora pensemos juntos, ¿dónde se va a ejecutar esta función? Bueno, podría ejecutarse en varios lugares. Nosotros estamos trabajando en la vista del listado, por lo tanto, voy a seguir ahí. Puede haber otras formas de hacerlo, perfectamente, pero vamos a… ya que estamos trabajando acá, vamos a seguir. Noten que acá tenemos nuestro… nuestra ruta para borrar. Ahora podríamos tener una ruta para modificar. Recuerden, desde este formulario nuestra información viaja con el método post, ¿ok? Entonces, tenemos que capturar esta ruta con el método post. Tenemos que referirnos a la ruta modificar, que está en el archivo listado, pero indicar que se trata del método post. Bueno, vamos a hacerlo, entonces, ahora mismo, sin perder tiempo. Acá vamos a indicar el router.post y la ruta se llama listado barra modificar. Ya estamos en el archivo listado, por lo tanto, solamente tengo que incluir acá modificar. Y acá, es una función asincrónica como todas las que venimos haciendo con base de datos. Nuestro objeto request, nuestro objeto response. Ahí están debidamente identificados. Y acá, vamos a hacer la lógica de nuestra función. Vamos a desestructurar para no escribir tanto. Vamos a recibir el id, la raza, las características

características y el detalle. Todo esto, ¿dónde está? en el cuerpo de nuestra request. Recuerden, ahí es donde viajan los datos desde nuestro formulario. Cuando hay un archivo viajan en req.files, cuando hay parámetros en req.params. Ya lo hemos visto. Ahora solamente capturaremos esto. Y vamos a crear el, bueno, un objeto data que va a tener la raza, las características… características y el detalle. Todos estos son campos que capturamos en nuestro formulario. Después también, en nuestro formulario, acá en modificar, tenemos el dato de la id porque lo necesito, pero lo tenemos oculto porque no queremos que el usuario toque este campo. Este campo no debe ser tocado, pero si lo necesitamos en el paso posterior. Por eso, lo cargamos aquí, pero lo dejamos oculto. Bueno, es una manera, hay otras de hacer la de hacer esto también, pero esta es una manera bastante rápida. Entonces, regresamos acá y les dije que tenemos el campo req.body.id. No se ve en el formulario pero lo tenemos, por eso, acá lo desestructuramos. ¿Para qué? Recuerdan que nuestra función de cambiar raza recibe un objeto data, o bueno, un objeto cualquiera con la información, y además el id para ver dónde tiene que hacer esos cambios. Vamos a llamar a esa función entonces: await razas.cambiarRaza, data y el id. Todos estos datos los tenemos acá. Y una vez que termine de hacer esto, le vamos a decir, bueno, que redireccione, que vuelva con un redirect a… al listado de administración. Muy bien, vamos a ver que nos falta para que esto corra efectivamente. Estamos acá, ahora es momento de probar los errores que siempre pueden ocurrir. Vamos a loguearnos

y acá estamos, vamos a ver si podemos modificar al siamés. Ajá, bueno, acá tenemos un problema. No puede llevarnos a esta… no puede llevarnos a esta sección, a modificar 9. Bueno, vamos a tratar de depurarlo a ver qué está ocurriendo acá. Perfecto. Nos está pasando un parámetro, no necesitamos un parámetro. Creo que en algún lado en algún sitio tenemos una ruta con parámetros que nos está… nos está dando un problema. ¿Cómo llegamos a la lista 'modificar'? Llegamos desde el listado, vamos a ver cómo estamos llamándola en listado. Claro, esto no me gusta, esto no, esto no es correcto, esto es un enfoque que hicimos antes, pero que no necesariamente vamos a aplicar acá. Vamos a ver si lo tenemos bien. Venimos al listado, modificar con el id. Bueno, venimos aquí a modificar el… perdón, al listado. Tenemos una lista modificar, ajá, tenemos una lista modificar, pero no está recibiendo el id. Vamos a ver, pero efectivamente tenemos la ruta del post, pero si ustedes se

fijan nos dice que no encuentra la ruta con el id. Y claro, es que vamos demasiado rápido me parece. Nos falta efectivamente la ruta para recibir ese id y mostrarlo en la pantalla a través de la vista modificar porque este es el método post. Este es el que va a recibir la información cuando venga de allí, pero nos falta decir: “bien, acá te muestro la información”. Eso es lo que nos está faltando acá. Vamos a ponerlo: router punto… Este sería un método get porque en realidad vamos a mostrar el formulario, y vamos a recibir el parámetro. Y sobre ese parámetro, el id que le vamos a pasar, vamos a renderizar los datos en la pantalla para que el usuario pueda modificarlo si así lo desea.

Bueno, bueno, entonces acá dentro tenemos que hacer dos cosas. Primero, tenemos que traer con el id que nos ha pasado el usuario, el registro del cual se trata. Por eso, vamos a hacer que acá, en la variable que se llame, bueno, se puede llamar data, tranquilamente. Recuerden que aquí el ámbito de cada una de estas funciones está delimitado por la llave, por lo tanto, pueden tener varios objetos que se llaman igual, siempre y cuando, no se vean entre sí. Eso no genera ningún problema. Y acá vamos a esperar la función que se llama ‘leerRaza’: razas.leerRaza, y ésta recibe un parámetro, ¿dónde está el parámetro con el id que nos están enviando? Bueno, está aquí, en la request, en el objeto params, en el objeto request.

Objeto request punto params y se llama id. Esta función, entonces, nos va a traer el registro, y ese registro, bueno, que es un gato, así que vamos a ponerle así. Lo vamos a mandar a nuestro formulario para que el usuario pueda editarlo, pero recuerden que primero tenemos que mapear. Tenemos que hacer un map, método de array, de nuestra fila. ¿Y qué vamos a hacer con cada fila? Bueno, simplemente la vamos a retornar. Vamos a retornarla

pero, ¿cómo la vamos a retornar? desestructurada, como un objeto. ¿Por qué? porque dentro de esa fila están todos los campos de la base de datos. Está id, nombre, raza, descripción, detalle, etcétera. Bueno, al pasarlo de esta manera nos ahorramos la necesidad de tener que extraer los datos uno por uno. Muy bien, ahora una vez que tenemos esto retornado, nos faltaría… Acá me está faltando o me está sobrando algo. Déjame ver la sintaxis. Acá me falta, por ejemplo, esta primera línea. Y acá me sobra esto. Ah, perfecto, muy bien, lo descubrí rápido aparece… aparentemente. Vamos a revisarlo por las dudas. Tenemos la función, acá, y acá tenemos que cerrar nuestro return. Y acá tenemos que cerrar la función, pero antes de que todo termine, Perdón, esto es al revés. Así es, llave, antes de que todo termine tenemos que llamar efectivamente la lista, la vista de modificar con este dato que hemos obtenido con nuestro gato. Entonces, será res render. Llamamos a la vista modificar

y vamos a pasarle acá como un objeto, nuestro gato. Claro, ahora, luego de esto se va a mostrar la vista modificar. Acá tenemos un for each de nuestro gato con el helper de handlebars each, y recuerden que acá tenemos helpers. Y vamos a hacer lo siguiente, en cada uno de los campos vamos a pintar la información que ingresa junto con el objeto gato. Gato.id, por eso acá ponemos el id aunque lo tenemos oculto, gato.raza por eso acá tenemos la raza, gato.características, bueno, y así sucesivamente. Ahora sí, creo que este era el paso que me estaba faltando, déjenme revisar. Para revisar esto tendríamos que reiniciar como hacemos siempre. Recuerden, esto es un poco tedioso pero lo que ocurre es que cuando reinicia el servidor, se cierra la sesión por un tema de seguridad. Por lo tanto el usuario, bueno, tiene que registrarse de nuevo. Si ahora mostramos la vista listado y ahora vamos a modificar… perfecto. Se ve feo pero esto es un tema que lo vamos a resolver fácil

cambiando las rutas del css. Noten que ya funciona. Acá tenemos el nombre del gato y acá tenemos los detalles. Ahora la cuestión del css que aquí no se ha cargado tiene que ver seguramente con la manera en que estamos indicando en nuestro archivo main, estoy seguro que es eso, la carga de nuestro css. ¿Por qué?, bueno, porque ya estamos trabajando con rutas anidadas, y cuando eso ocurre tenemos que tener en cuenta que la las carpetas se pueden ir anidando también. Por lo tanto, acá, voy a agregar una barra antes de css, y creo que con eso va a funcionar. Buenísimo, perfecto, al final era poca cosa, era bastante fácil. ¿Podemos volver? Sí, podemos volver. ¿Podemos agregar un registro? bueno, ya hemos visto que sí. También que podemos borrar. ¿Podemos modificar? podemos modificar, bueno, pero ustedes quieren chequear si de verdad se puede modificar. Bueno, vamos a cambiar algo, vamos a decir “el gato Tom”.

A ver si esto cambia. Ponemos ‘enviar’. Vamos al listado, y bueno, solamente nos está faltando que efectivamente esto impacte en la base de datos, pero tampoco se ha roto nuestro… nuestro programa. Por lo tanto, vamos a revisar esta cuestión y dejaremos terminada nuestra aplicación.