

Tenemos el círculo prácticamente cerrado. Podemos registrarnos. Si estamos, si nuestro usuario, nuestra contraseña son correctos, vamos a poder acceder a una sección, además tenemos rutas protegidas, por lo tanto, sólo podremos acceder si estamos registrados, independientemente de si conocemos o no la ruta interna, dónde están esos recursos, claro, pero no queremos quedarnos registrados por siempre, tal vez sería bueno poder cerrar la sesión como en la vida real.

En cualquier sistema ustedes se pueden dar de alta, pero también pueden decir "okey, quiero cerrar mi sesión". Completaremos ahora, entonces, el círculo implementando esta funcionalidad.

Vamos a dar el cierre con una función para darnos de baja de la sesión activa, es decir, para cerrar la sesión. Pero primero veamos una característica que quiero mostrarles a continuación. Esto lo vamos a poner en responsive, para poder verlo correctamente. Recuerden, ya tenemos protegida la ruta de administración, no podemos entrar directamente, nos dice que tenemos que loguearnos. Vamos a loguearnos, "A, arroba B, punto C" y podemos entrar y nos muestra cuál es el usuario actual. Luego, vamos, por supuesto, a darle una mejor vista a todo esto, por ahora sólo quiero que aprendan la lógica. Pero nos faltaría algo más. Si salimos de esta sección, aunque ya está activa una sesión de usuario y debiéramos poder volver, no podemos. Bueno, esto se debe a una configuración que tenemos aquí, que vino por defecto, es la del funcionamiento de las "cookies". Esto vamos a, miren, podemos ponerlo en "false" pero, por supuesto, es lo mismo que borrarlo. Okey, así que lo vamos a borrar. Bueno, aunque excede por bastante el alcance de este curso, de todas maneras, no quiero dejarlos a oscuras. Les voy a explicar aunque sea por arriba por qué ocurre esto.

Esta funcionalidad, que venía por defecto, sirve para cuando trabajamos con, fíjense que lo tenemos explicado por aquí arriba, esto se utiliza cuando trabajamos con servidores seguros, "https", pero nuestro mundo gatito se está corriendo en un servidor local, sin el protocolo de seguridad "https", por lo tanto, en estos casos, lo que ocurre con la con la "cookie", es el lugar, el sitio donde se almacena la información de nuestra sesión de usuario, ocurre que, lo tenemos, creo que por aquí explicado, lo que ocurre es que el usuario, luego de setearla, no la devuelve al servidor y se pierde esa verificación, ¿okey? Al perderse esa verificación, el servidor cree que somos un usuario que está fuera de registro, o sea, que no está registrado y nos pide que volvamos a registrarnos. La nota técnica está aquí, en "cookies secure", que es lo que acabamos de eliminar, ¿okey?. Aquí lo que indica es precisamente eso, cuando los seteamos a "true" tenemos que estar trabajando en un entorno de servidores "https". Muy bien. Como no estamos trabajando con un servidor seguro en este momento, simplemente podemos eliminarla. Ahora, lo que va a ocurrir es que una vez que nos logueemos, creo que hemos actualizado el código, espero que sí, con "A punto", perdón, "arroba B punto C" y nuestro e-mail registrado, vamos a seguir ingresando. Pero si salimos acá y queremos volver a ingresar, ahora podemos hacerlo, ¿por qué?, bueno, precisamente porque ha registrado, ha detectado que somos el mismo usuario que está logueado. Todavía esta es una sesión activa. Nos faltaría corregir que aquí nos muestre el nombre, eso lo podemos hacer indicando en nuestra vista de administración, acá podríamos indicar que, bueno, venga el dato, que no lo estamos renderizando. Si venimos a administración, fíjense que no estamos enviando el dato del usuario activo. Para hacer eso, recuerden que tenemos una variable, un objeto, que se llama "rec punto session" y en su propiedad "punto user" tenemos el mail del usuario actual, por lo tanto, lo que podemos hacer es, en esta ruta, pasar ese objeto a la

vista "rec punto session punto user". Ahora, ¿qué va a ocurrir? Que vamos de vuelta. Supongamos que estamos en inicio. Si intentamos ingresar ahora no vamos a poder, pero, ¿por qué?, porque cada vez que guardamos y se reinicia el servidor, el sistema re-comienza, entonces ahora si es una nueva sesión, pero probemos de "A arroba B punto C", 1-2-3, entramos y aquí nos dice cuál es el usuario registrado. Vamos a salir. Quiero volver a entrar ,no solamente debe permitirme el acceso, sino que, además, como ahora agregamos que para los ingresos directos desde la ruta nos renderice en la vista el nombre del usuario que, en este caso, contiene el e-mail, bueno, debiera hacer esas dos cosas, dejarnos entrar nuevamente, porque estamos en una sesión activa y mostrarnos quiénes somos. ¡Boom! Ahí está. Hemos entrado y tenemos nuestro e-mail.

Sólo faltaría ahora un método para darnos de baja, es decir, ya tenemos el "login", podríamos, en esta vista de administración, tener un "log out", ¿no? Sí, vamos a hacerlo. Miren, esto es súper rápido, es un método de "express session" que corre de manera directa. Simplemente vamos a agregar acá abajo, después lo vamos a mejorar, ¿okey?, esto que hay la la visualización lo vamos a mejorar, por ahora sólo quiero la lógica. Vamos a crear un controlador dentro de "login", que se llame "log out" y a eso hago referencia aquí, "log out" o cerrar sesión, como ustedes prefieran, es lo mismo. Bueno, vamos a poner "cerrar sesión"; "cerrar sesión".

Bueno, acá nos va a aparecer ahora "cerrar sesión" que, por ahora, no hace nada. Queremos que haga algo. Dijimos que vamos a crear este controlador en la ruta de "login". Entonces, venimos a "login" acá a nuestro, nuestro "login" y vamos a crear simplemente aquí, en cualquier sitio, no puedo ponerlo abajo un router, con una petición de tipo "get" que para el ingreso a "log out", acá simplemente, bueno pues, se desconecte, que esto va a recibir una función, vamos a hacer función "flecha" que es más rápido, le indicaremos que cuando entre a esta ruta, lo único que tiene que hacer es "rec". Recuerden que tenemos el objeto "session" desde que instalamos nuestro "express session" podemos manejarlo, punto y un método 'destroyed' y esto va a destruir o cerrar la sesión actual. Luego de esto, bueno, podríamos volver a algún sitio, no podríamos ir a vamos a hacer un "re-direct", pero directo y vamos a re-dirigirnos a una ruta, a la ruta raíz. Podríamos hacer un render también, ¿no?, es para mostrarles una manera distinta, o sea, cuando ingresemos acá a través de este botón, se va a destruir la sesión, ya no podremos ingresar acá, porque no existirá el objeto "rec punto session punto user", tendrá un valor de "undefined" este objeto, ya no va a tener un valor, el valor de, del usuario registrado y no podremos pasar el middleware que nos deja entrar acá. Tendremos que volver a registrarnos. Vamos a ver si esto funciona.

Bueno, a ver, actualizamos acá, vamos a "admin", no podemos entrar de manera directa, por supuesto, recuerden al grabar el archivo, se reinicia el servidor y se reinicia la sesión. Entonces, vamos a registrarnos con un usuario que sabemos que existe en la base de datos, nuestro famoso "A arroba B punto C", con su contraseña súper segura, 1-2-3, es broma, por supuesto, no es súper segura. ¿Funciona? Sí puedo entrar. Sisalgo, ¿podré volver?, directamente "admin" debiera pasar el middleware, porque tengo un usuario registrado y lo paso y, ¿si escribo aquí o si doy click en "cerrar sesión"? Cerró sesión y me re-dirigió a la vista principal. ¿Podré volver a ingresar a "admin"? No debería y efectivamente, no puedo. Y así terminamos todo lo que tiene que ver con, no solamente el "login" y el "log out", sino también, con la protección de rutas a través de la autorización y la información de sesiones.