

Luego de utilizar a nuestros modelos y MySQL, a través de "Now" para conectarnos con una base de datos y comprobar si un usuario está o no está en la base de datos, ya podemos registrarnos, si poseemos apropiadamente los datos de conexión, los datos, por ejemplo, de un usuario y una contraseña o un e-mail y una contraseña y esos datos existen en la base de datos, seremos un usuario registrado, es decir, nos dejarán pasar y podremos hacer algunas cosas, pero tenemos que proteger las rutas. No alcanza con registrarnos. Una persona que conozca la dirección de una sesión protegida, secreta, digamos, si ésta no estuviera efectivamente protegida, podría pasar igual, aunque no, aunque no se hubiera registrado correctamente. ¿verdad? Bueno, eso es lo que queremos aprender a hacer ahora, proteger rutas.

May bien, vamos a recapitular. ¿Qué ocurre cuando venimos aquí debajo a "login"? Se nos abre este formulario, consultamos la base de datos y ponemos un dato mal, algo que no existe en nuestra base de datos, nos da un error y entonces el ponemos aquí, "A, arriba B, punto, C", así se llama el usuario, el único que tenemos registrado, su contraseña es 1 - 2 - 3. Perfecto, entramos a la sección de administración. Hecho aquí todo bien. Ahora, supongamos que avanzamos el proyecto desde cero y decidimos escribir acá, vamos a hacerlo más grande para que se note, dicen que nosotros no tenemos manera de acceder directamente a la administración, eso sólo ocurre si podemos loguearnos con las credenciales correctas. Pero, ¿qué ocurre si alguien descubre que nuestra ruta de acceso es "base admin", se loguearse? Va a ingresar igual, por lo tanto, esta ruta, es una ruta insegura y nuestro proceso de autenticación, bueno, es un tanto precario y funciona correctamente, pero para que cumple su cometido de proteger esta ruta hay que, precisamente, proteger la ruta para que solamente un usuario que se haya registrado pueda acceder, para esto vamos a usar un paquete que se llama "express session", nos va a servir para mantener almacenado en un sitio de la memoria, pongámonos en una suerte de variable temporal, los datos del usuario una vez que éste se ha registrado, es decir, que sea el logueado que ha comprobado sus credenciales, poniendo de manera correcta es un e-mail, en este caso, y su contraseña. Eso se verifica en la base de datos y si ese usuario existe, actualmente lo estamos enviando a esta sección, vamos a hacer algún paso intermedio. Para eso tenemos que instalar, primero, el paquete "express session", "npm" y "express session". Ahora vamos a poder trabajar con los, con un objeto que se llama "res punto session", que almacena todos los datos que nosotros necesitamos para comprobar que el usuario efectivamente está logueado y de esta forma, también, a través de un "middleware" de protección, qué vamos a hacer a continuación, bueno, poder verificar antes de garantizar el acceso a una ruta, si está o no está registrado, para que esto no sea posible si el usuario "admin" aquí, ahora puede ingresar, bueno, no puede ingresar, porque contamos el servicio para hacer la instalación ¿verdad? Pero entonces que va a poder ingresar. Si escribo aquí "admin", ingresa. Bueno, esto ya no va a ser posible, si logueamos lo que queremos a continuación.

Venimos al archivo "A punto js", aquí vamos a requerir, tenemos que poner en uso nuestro, nuestro session, lo que acabamos, el paquete que acabamos de importar recién, vamos a decir "const session" y esta dependencia, vamos a requerir de, bueno, "express session" así se llama la dependencia que instalamos. A continuación, tenemos que configurarlo. Bien la configuración. Está aquí en la documentación, por aquí nos dice cómo tenemos que configurar para su uso nuestro, nuestro middleware de session. Bien, entonces podemos