

Estamos en un punto en el que podemos registrarnos, damos de baja, cerrar la sesión, mejor dicho, podemos agregar recursos a nuestra base de datos, bueno, nos gustaría poder leer la información que tiene nuestra tabla, la tabla, bueno, cualquiera de las tablas que podamos tener en una base de datos.

Vamos a verlo en esta clase, en el código, ahí, en ese código.

Bueno, ahora nos gustaría que cuando ingresamos al administrador, tener la posibilidad de ir a un estado. Yo he creado aquí un formulario, en realidad es una tabla en la que vamos a leer la información que venga de nuestra base de datos, desde nuestra base de datos. ¿Ustedes recuerdan en la última clase pudimos agregar información a la base de datos? Aquí tenemos una función para agregar más, ahora nos gustaría también poder leerlas y mostrarlas en nuestro estado. Para esta vamos a crear una nueva función, aquí en nuestra carpeta de modelos, que se llama, por ejemplo, bueno, "Leer nazas".

Vamos a agregarla, creamos una "arrow function", vamos a hacerla con "function" también, para que ustedes vean que se puede hacer con cualquiera de las dos, de los dos métodos, "Leer nazas". Esto no recibe parámetros, porque va a leer toda nuestra base de datos, en realidad, toda la tabla de nazas y nos va a mostrar la información que contenga. Vamos a ponerlo en un "try catch" por que, reiteramos, ésta es una función asíncrona, ya que consulta una base de datos, por eso utilizamos una estructura "try catch", para poder atrapar eventuales errores. Si es que los hubiere vamos a construir nuestra "query", esto va a ser igual. Bien, aquí hay que seleccionar todo lo que haya en la tabla, así que será fácil es "select todo from nazas". Bueno, ahora vamos a traernos todas las filas que encontraremos. Recuerden que ahora tenemos un `state data`, que es el del gallo abasco, que cargamos en la clase pasada.

Para bueno, vamos a ir haciendo más. Eso espero, al menos. Y aquí a guardaremos, el resultado de nuestra consulta a la base de datos que, `data`, aquí hay algo que no está bien, me ha, automáticamente me ha importado algo. Tengan cuidado con esto, a veces cuando les da una sugerencia de ayuda, si ustedes presionan "enter", sin querer, o sin mirar, es probable que terminen cometiendo un error y aquí vamos a pasar en nuestro query, esta es muy sencilla, no recibe, refiere, ningún parámetro, vamos a buscar todo lo que haya en nuestra tabla y vamos a retornar el resultado. Lo hemos almacenado en la constante "rows". ¿verdad? En caso de que hubiere aquí un error, simplemente queremos verlo por pantalla, para bueno, poder depurarlo, claro, la función también hay que exportarla para poder utilizarla en otro sitio en nuestro controlador "nazar". Ya tenemos creado nuestra función. Ahora, ¿dónde vamos a usar esta función? Esta función deberíamos usarla en nuestro estado, en realidad, nuestro estado tendría que leer todas las nazas y cargarlas en la lista correspondiente, vamos a agregar un poco aquí para que nos deje trabajar mejor. Bueno, vamos a agrandar totalmente y esto lo pondremos cuando lo necesitemos. Bueno, carabita, tengo problemas para manejarlo, bueno, ahora sí, ahí va quedando un poco mejor. Entonces aquí, nosotros ya tenemos una lista de lo que haya en la base de datos, aquí hemos construido simplemente una tabla, bien, esto es una tabla un elemento semántico "table" y un producto de CSS, para mostrarlo ampliamente, pero aquí lo importante, como siempre está en los controladores, vamos a crear un controlador que se llama "el estado", "estado.js" y ¿qué tendremos en este controlador? Bueno, como en todos los demás vamos a registrar "express", también vamos a necesitar "Cloudinary", "Cloudinary", esto va a ser igual a