

Es muy importante el "debugging" o rastreo y solución de errores, se le llama "bug" o "ticho", con "v", "b-u-g", "bug", digamos, a un error, se le llama así, digamos, por una historia muy antigua, no viene al caso ahora, no tenemos tanto tiempo, se le llama así, "bug", "ticho", un error en el código y de "debuggear", o el "debugging", es, precisamente, ir rastreando dónde están los errores, es un procedimiento muy importante porque aprendemos mucho de nuestros errores. Vamos precisamente para terminar esta clase, y para darle un cierre a nuestro servicio, vamos a hacer un poco de "debugging". Luego, obviamente nos quedaría refactorización, que no entra en el alcance de estas clases y sería limpiar nuestro código y lo que ya hace bien, hacerlo mejor, eliminando cosas innecesarias o duplicadas que, efectivamente a nuestro código las tiene, pero por ahora, eso lo vamos a dejar fuera de alcance, porque lo importante es construirlo y que funcione, y ahora lo que haremos es quitar los errores para que, efectivamente, haga todo lo que nosotros.

Vamos a revisar, qué es lo que está ocurriendo y con esto creo que ya podremos terminar. Estamos muy cerca, de hecho, nuestra aplicación no se está completando, creo que tenemos algún problema en la construcción de alguna consulta de SQL. Vamos a ver qué ocurre cuando queremos modificar. Podemos modificar, cambiemos algo rápido para ver si, rápidamente, esto se refleja a campos, vamos a poner cualquier cosa, podemos enviar, podemos volver al listado y ahora sí se está cambiando.

Vamos a tratar de descubrir qué es lo que está impidiendo que cambiemos el registro, que es lo único que nos falta. Vamos a modificar así, a ver qué pasa, vamos a revisar nuevamente para entender bien dónde estamos fallando. Podemos ir al listado, cambiar algo, vamos a cambiar aquí el *id* de *Albinis*, por cualquier cosa, bien y esto no funciona y ¿dónde está el problema? ¡Vámonos al código que nos dice. Suele darnos el código, muchas veces, ¡ajá! Vamos a ver claro, no, no se rompe el código, pero tenemos un error en el SQL. Así nos está diciendo que estamos tratando de duplicar una entrada, es decir, nuestro problema está en la consulta de SQL. Así que, tiene que estar así.

Vamos a ir a esas y acá vamos a ver el método para cambiar una raza, recibe una una data, un objeto, una *ID* y construye el, la "query", dice, "actualizar razas con la data where", ¿eh, *where* *where*? ¿Dónde? ¿Dónde qué? No lo estamos indicando dónde queremos efectuar el cambio, bueno, dónde el *ID* sea igual al *ID* que está entrando acá. Por eso es bueno ir revisando y depurando línea a línea el código. Una vez que sabemos, por la interpretación de los errores, dónde, en qué sección puede estar el problema, entonces aquí nos está fallando sí, si más ni menos, que el *ID*, igual al *ID* que entra por parámetro, ya está, es esto, éste es el problema. De hecho, ahora deberíamos poder, bueno, tenemos que loguemos otra vez, creo que va a ser la última vez, si está funciona. Y ahora deberíamos venir al listado y poder modificar, a ver, vamos a ponerle "haca jajahehe", una raza muy puerca de gatos, podemos enviar. ¿habrá cambiado? Perfecto, cambio.

Vamos a hacer un doble check, vamos a chequear otra vez, por las dudas.

Vámonos a poner *Albinis*, podemos enviar y si venimos al listado, aquí están los cambios. También vimos que podemos eliminar, que las imágenes se eliminan de Cloudinary y el registro se borra de nuestra base de datos.

Hemos visto que podemos seguir navegando y también, podríamos tratar de volver, por ejemplo, a nuestro "admin" y debería dejarnos, porque seguimos con la sesión activa, no hemos hecho el "log out". Podríamos irnos de acá y tratar de entrar a la sección "listado", así también corre el middleware y nos deja entrar perfectamente, porque seguimos loguados,