

## Errores utilizando el servidor remoto

Un comentario a destacar antes de puntualizar en cada error es que, al pedir el archivo de configuración de BD remota, este era similar al que teníamos en nuestras BD locales, sólo difería en algunos puntos específicos como IDs de las BD, que entendemos, no serían un problema.

- **Cookies:** Varios navegadores solo guardan aquellas cookies que vienen de otros orígenes, si tienen como atributo Same-site=None y Secure. La cookie que devuelve el servidor, no tiene estos atributos, por lo tanto, no se guarda. Esto nos supuso un problema ya que nuestra idea era utilizar esta cookie en los pedidos al servidor. Al no tenerla, debimos implementar otro mecanismo que en nuestro caso fue utilizar el Basic Auth. Lo que hicimos fue en el constructor de cada BD agregar un parámetro fetch, el fetch intercepta cada solicitud que sale de PouchDB y agrega los datos del usuario que previamente fueron almacenados, y de esta manera, en cada solicitud se cumple lo necesario para el Basic Auth.
- **GET Error 401 (Unauthorized):** Este error está asociado al problema anterior, cuando se realizan peticiones se requiere de una identificación, en un primer momento al tener el error con la cookie de autenticación, no encontramos el origen de este error, hasta ver el fallo en guardar la cookie, pero al utilizar el fetch, este error desaparece. Incluso se actualizaron los archivos \_security de cada base de datos, con los roles de 'profesor' y 'profesor\_root' (ver archivo de aclaraciones). A pesar de estas soluciones, surgieron otros problemas.
- **POST Error 400 Índices (Bad Request):** En la aplicación, en el constructor del componente BD, se realiza una creación de los índices en la base de datos, por si no estaban creados. Lo que pudimos notar es que, a diferencia de cuando lo hacíamos contra nuestra base de datos local, estos índices no se generan en la base de datos remota. También intentamos crear los índices en la BD remota utilizando CURL y Postman, pero tampoco nos fue posible.
- **POST Error 400 Find (Bad Request):** De la mano con el problema anterior, al no estar creados los índices, si se realiza una solicitud de búsqueda para, por ejemplo, listar los jugadores, se da este error ya que los índices no están creados. Tampoco se pueden ver listados de usuarios de la aplicación, usuarios pendientes, etc.
- **Error al utilizar el método getSession() de PouchDB:** Por lo que pudimos entender, haciendo una comparación entre la aplicación cuando utiliza la BD local y cuando lo hace con la remota, es que este método utiliza la cookie de session para poder identificar al usuario. Como mencionamos antes, esta cookie no se guarda, por lo tanto, este método no puede funcionar. El principal efecto de este error lo podemos notar en el SideMenu de la aplicación, donde al desplegarlo se quiere mostrar el usuario actual y también el mismo se modifica para, en caso de ser un usuario administrador, brindar las opciones disponibles para este. Al no funcionar, no solo no se muestra el usuario, sino que tampoco se brindan las opciones especiales. También se usaba en la vista de cobros, que, al no obtener un usuario valido, no se renderiza.

Todos estos inconvenientes fueron discutidos con un profesor, y, aun así, no fue posible encontrar una solución. La documentación disponible online era escasa y de poca ayuda. Debido a que ninguno de los integrantes dispone de tiempo para continuar con el proyecto, y que los problemas son externos al desarrollo de la app, consideramos que lo mejor sería finalizar las prácticas profesionales supervisadas en este punto. Aun así, creemos que el proyecto podría ser continuado por otros estudiantes que deban realizar sus PPS en el futuro.

Es por ello que se recopiló una serie de deudas técnicas y errores que deberían trabajarse en el futuro. Además de lo explicado hasta ahora, debería tenerse en cuenta lo siguiente:

- **Actualización de categoría automático desde servidor:** El proyecto hasta el momento no tiene un servidor, y CouchDB, no ofrece stored procedures ni opciones similares. Hasta ahora, un profesor debería actualizar la categoría de todos los jugadores a mano cada año, pero esto debería poder realizarse automáticamente.
- **Referencia temporal desde servidor:** Nuevamente, al no tener un servidor disponible, para muchas operaciones se usa el timestamp local del dispositivo, el cual es fácilmente alterable. Por ello debería usarse el timestamp provisto por un servidor.