

## Taller Frameworks del Lado del cliente

Se realizó la mayoría de funcionalidades a excepción del buscar por sedes y eliminar usuarios. El primero no se realizó debido a que en la arquitectura actual, en el maestro detalle de una sede ya se puede ver este filtro por defecto como se plantea en el taller, el eliminar usuarios no se realizó porque al estar usando Firebase authentication no es posible eliminar al usuario una vez después de registrado por temas de seguridad y de garantizar que la aplicación sea estable.

Algunas dificultades encontradas fueron el uso de componentes basados en clases en lugar de funciones. Al realizar los componentes basados en clases, se imposibilitó el uso de Hooks como herramienta primordial para la facilitación de estados en el ciclo de vida de la aplicación. Esto generó repercusiones en el manejo de props entre componentes, por lo que se ingeniaron otros mecanismos como la obtención de valores mediante el path del url. Se contempló el uso de Redux para el almacenamiento centralizado de estados, sin embargo, dado el avance del proyecto y el corto tiempo, se empleó la solución anteriormente mencionada.

Dada la longevidad de la librería, existen diversas maneras de atacar un problema, no obstante, algunas soluciones son incompatibles si se conectan. Por ejemplo, existe documentación de cómo resolver un problema mediante componentes base pero esta a su vez, si trata de solucionar otro problema del mismo componente, hay un gran margen de que la segunda solución no sea compatible (incluya componentes funcionales, Hooks, entre otros) con la primera.

Se concluye que al usar una librería como React, las posibilidades de escalar una aplicación de acuerdo a las necesidades del desarrollador son inimaginables pues hay dependencias que se pueden incluir sobre la marcha y que, a su vez, permiten manejar el peso de la aplicación teniendo en cuenta solo lo necesario. El manejo de estados y renderizado requiere de un conocimiento adicional en comparación con frameworks como Vue y Angular.

Respecto al enunciado, hay puntos que se desconocían si realmente eran factibles, dando margen a la pérdida de tiempo en búsqueda de soluciones. Por ejemplo, el borrado de una colección (caso de colección de usuarios dentro de un documento de una sede específica) ya que, según la documentación de firebase <sup>1</sup>, no es posible y no es buena práctica, impidiendo así, el borrado en cascada de sede y posteriormente de los usuarios.

No es claro si deben haber sedes previamente construidas para la asignación de nuevos usuarios o si estos pueden ser independientes y posteriormente indexarlos a una sede. También se desconoció el propósito de atributos de una sede/usuario `activ@`, pues no especificaba qué comportamiento debería poseer como, por ejemplo, ocultarlo de los listados de usuarios/sedes dependiendo del caso.

## **BIBLIOGRAFÍA**

1. <https://firebase.google.com/docs/firestore/manage-data/delete-data>