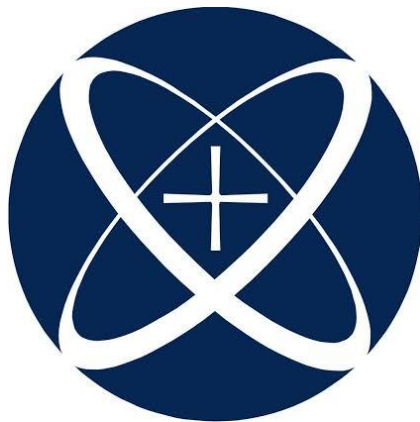


Instituto Tecnológico y de Estudios Superiores de Occidente.



ITESO

Universidad Jesuita
de Guadalajara

Métodos para Manejo de Estados y Filtrado de datos con bloc

Materia: Programación de dispositivos móviles

Profesor: Francisco Javier Camacho Gil

Fecha: 21/09/2020

Autor(es): Francisco Andres De La Torre Valle

Introducción

Esta actividad consiste en investigar qué es el manejo de estados, su importancia y algunos de los diferentes métodos de manejo que se emplean, y preparar información suficiente para responder preguntas al azar que se harán en la clase siguiente.

Añadido a esto se dejó el terminar un ejercicio en clase donde se utiliza una estructura bloc para manejar los datos en una lista de usuarios, donde se tiene que completar el filtrado de datos.

Desarrollo

Para comenzar, es importante responder a la pregunta central, ¿qué es el manejo de estados? Esto sería el administrar los datos dentro de una estructura de nuestra aplicación a manera que muestren información en pantalla, la actualicen y trabajen de forma eficiente, como se puede ver en la definición, la importancia de tener un buen manejo de estas es vital para que una aplicación sea responsiva y tenga un flujo natural de la información, de manera que cumpla con su objetivo sin representar un problema o inconveniente al usuario.

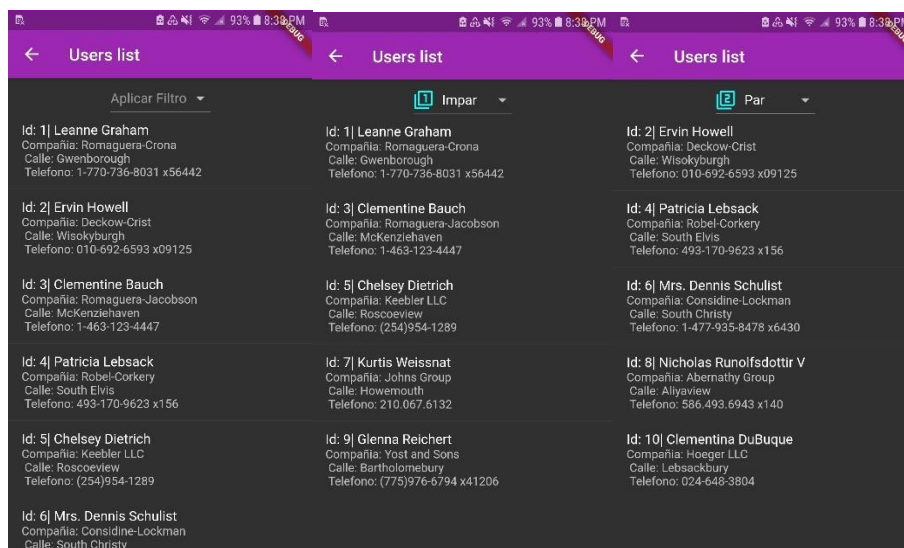
Los métodos que se revisaran en esta tarea son:

- **Provider:**
Este es un método de programación para poder realizar cambios de estructura significativos de manera más sencilla, como cambios de bases de datos, esta provee de funcionalidad específica y especial al código, sin embargo, la clase provider específica no es definida sino hasta empezar el run time. Dicho de manera simplificada, este método le permite a la app utilizar una de distintas implementaciones de la configuración, como se mencionó anteriormente se podría utilizar para emplear una arquitectura de bases de datos y/o cambiar a otra después.
- **BLoC:**
El componente de lógica de negocios o bien Business Logic Component trabaja a manera de intermediario entre la fuente de información y el widget que la necesita, este recibe la información y logística para aplicar para poder presentar la información resultante necesaria.
Un ejemplo de su funcionamiento es al pedir información de una base de datos o del resultado de una serie de operaciones lógicas que normalmente regresa una lista poco a poco, en lugar de esperar a tener una lista completa, esta arquitectura ayuda a ir entregando datos y encargarse de la logística de la recepción de estos.

Para el desarrollo práctico, lo primero que se hizo fue la parte fácil de agregar un subtítulo que contenga toda la información extra del usuario que se pidió, así como un poco de personalización. Una vez terminado esto, la lógica de filtrado no fue difícil, puesto que simplemente fue repetir el conseguir la información del método GetAllUsers y antes de regresar la lista hacer un if con el bool del método para saber si será par o

impar, y con base en esto acortar la lista igualándola a si misma siendo filtrada con where donde el modulo de 2 del id sea o no 0. A continuación se desarrolló un drop menú para el appbar que va en el apartado de hijos tipo widget, sin embargo para que la siguiente parte funcionara si se tuvo que sacar, de esta manera entra en el contexto del provider y puede reaccionar mandando las operaciones necesarias de add al Bloc para actualizar la pagina dependiendo de la opción seleccionada.

Capturas:



Conclusiones

Uno de los comentarios decía que, aunque parece ser mucho trabajo este es trabajo muy importante y útil pues asegura una mejor estructura para las aplicaciones, y la verdad es que después de ver de lo que son capaces me llama mucho la atención todo lo que se puede llegar a hacer con estos métodos, pues realmente creo que si mejoran mucho la calidad de una aplicación y extienden su ciclo de vida.

La necesidad del uso de Bloc en la programación es muy notoria si se busca trabajar con apis de manera efectiva, aunque es cierto que para ciertas partes de las pantallas puede ser muy engañoso, como el hacer que una parte de la aplicación influya a otra solo con esta estructura.

Observaciones

Realmente nunca había escuchado hablar de estos métodos, ni sabía que se pudiera trabajar de esta forma con comandos así, siendo este mi primer semestre trabajando con aplicaciones móviles realmente me falta mucho por aprender, pero me agrada la idea general detrás de estos métodos.

Me es impresionante lo mucho que facilita el poder estar haciendo operaciones con un api, pues realmente solía ser una tanto molesto al trabajar con ellas en web.

Bibliografía

Opia, C. (30 de Enero de 2019). How to handle state in Flutter using the BLoC pattern. Obtenido de <https://www.freecodecamp.org/news/how-to-handle-state-in-flutter-using-the-bloc-pattern-8ed2f1e49a13/>

Sheriff, P. D. (s.f.). The Provider Model. Obtenido de Code Magazine: <https://www.codemag.com/Article/0711081/The-Provider-Model>

Software, Q. (5 de Junio de 2020). Youtube. Obtenido de <https://www.youtube.com/watch?v=OEMHbGji5YE>

The Provider Model Pattern, Really? (5 de Septiembre de 2009). Obtenido de Simple Thread: <https://www.simplethread.com/the-provider-model-pattern-really/>

Dropdown menu:

"<https://www.developerlibs.com/2019/09/flutter-drop-down-menu-list-example.html>"

List con dos hijos para drop menu:

"<https://stackoverflow.com/questions/56442336/how-can-fix-listview-in-column-and-add-refreshindicator-to-the-listview>"

GitHub:

<https://github.com/FrankieDLT/Tarea3-Moviles-oto2020>

Video:

Se encuentra en el repositorio de Git

Criterio de evaluación ponderado.

- Agregar un botón al AppBar de la Home Page donde nos despliegue un menú con dos opciones **pares e impares** al elegir alguna de ellas, aplique el filtro desde el bloc y muestre en pantalla los elementos en la lista. **(55 pts)**
- Actualmente se muestra el nombre del user, mostrar además del **nombre del usuario, company name, street y phone** así como dividers entre cada elemento de la lista (tip: cambiar **ListView.builder** por **ListView.separated**) **(10 pts)**
- Video de app funcionando **(5 pts)**
- Link a repo de Git **(5 pts)**
- Reporte en formato de tarea **(25 pts)**
- En el reporte responder a la pregunta:

Responder a la pregunta:

¿Qué es el manejo de estados y para que nos sirve?

Investigar de manera individual **métodos para manejo de estados**. (Explicación teórica y ejemplo práctico)

1. **Provider**
2. **BLoC**
3. **MobX (opcional)**
4. **Redux (opcional)**
5. **Inherited widget (opcional recomendado)**