

06. Definir qué hacer cuando el usuario haga click en los ítems del RecyclerView.

En esta iteración se quiere poder realizar una acción cuando el usuario realice un "click" en los elementos del RecyclerView.

En concreto, en el layout correspondiente a los ítems de tipo film (layout con identificaror R.layout.film_item) se ha definido la propiedad android:onclick directamente en el elemento raíz. El valor de esta propiedad es una expresión binding que define el listener View.OnClickListener que se asignaría a dicha vista. Este listener sería lo que se ejecutaría cuando el usuario hiciese click sobre el item.

En la clase FilmItem (la que representa los ítems de tipo film) se ha definido el método click(). Este método es el que se quiere que se ejecute cuando el usuario haga click en ítems de tipo FilmItem. Este método del FilmItem invoca a su vez a un método del view model. Se ha añadido un nuevo parámetro en el constructor del FilmItem que recibe una expresión lambda que indica qué se debe ejecutar en el método click().

En el view model se ha definido un Flow de tipo SharedFlow (atributo events). A este tipo de Flow también se puede suscribir otros objetos para recibir notificaciones cuando se modifica el valor del SharedFlow. Pero, a diferencia de lo que ocurre en un StateFlow, los nuevos suscriptores del SharedFlow no reciben notificación correspondiente al valor actual del SharedFlow. Es decir, cuando se cambia el valor del SharedFlow, éste notificará a todos los objetos que estén suscritos en ese momento. Si posteriormente se suscribiese un nuevo objeto, éste no recibiría la notificación.

El Activity será el objeto que se suscribe al SharedFlow. Lo hará cuando el ciclo de vida del Activity sea STARTED (visible) y cancelará su suscripción cuando la activity deje de estar visible. Si durante el tiempo en el que la Activity está oculta se produjesen cambios en el estado del SharedFlow, estos cambios no serán notificados a la Activity. Si la Activity regresase nuevamente al estado visible, volvería a suscribirse al SharedFlow pero tampoco recibiría notificación del estado (o valor) en el que se encuentra dicho SharedFlow. Sólo recibiría notificaciones de sus futuras modificaciones.

A través del SharedFlow, la Activity recibe eventos y tiene la capacidad de reaccionar ante ellos (operación collect). Sólo recibirá eventos a partir del momento en el que se suscribe por lo que se no podrá procesar los eventos producidos mientras no estuvo suscrita. De esta forma se evita la "repetición" del mismo.

Al igual que sucedía con el flow de tipo StateFlow, un SharedFlow también es un flow de sólo lectura. Es imposible modificar su estado a través de su interface (los métodos que contiene). Para ello, el SharedFlow se crea como una vista de sólo lectura de otro flow de tipo MutableStateFlow. El interface de

este último sí permite alterar el valor que contiene, por lo que automáticamente se “cambiará” el estado del SharedFlow (emitiéndose las notificaciones a sus correspondientes observadores). En ViewModel se define un atributo de tipo MutableSharedFlow (atributo _events) a partir del que se crea posteriormente el SharedFlow.

La acción que se realiza en el método click() del FilmItem, implica la emisión de un nuevo evento en el atributo _events (MutableStateFlow) que provocará el cambio en el StateFlow y la correspondiente notificación al Activity.

La binding expression definida en el atributo android:onclick del layout implica la invocación del método click() del FilmItem.

Los eventos se definen a través del interface Event. De momento se define una única implementación de este interface, la clase OnClickFilm (más adelante se definirán otro tipo de eventos). Esta clase tiene como payload un Film que representa el Film sobre el que se ha realizado la operación "click".

La activity recibirá notificaciones de distintos tipos de eventos. Cuando el tipo de evento es OnClickFilm muestra en pantalla (a través de un Snackbar) un mensaje que indica el título de la película pulsada por el usuario.