

10. Swipe y onClick en los ítems del RecyclerView.

Cuando se ha implementado el *swipe* en los ítems del *layout*, se ha perdido la detección del *onClick* sobre estos.

Para mantener la detección de los *click* sobre los ítems del RecyclerView al mismo tiempo que se permite realizar *swipe* (por medio de animaciones MotionLayout) habría que establecer un `View.OnTouchListener` en el *view* contenido en el *view holder* (este *view* se obtendría a través de `dataBinding.root` o por medio del atributo `itemView` que tiene de todo *view holder* ya que en este atributo es donde se almacena la vista que se le pasa como parámetro al constructor de la superclase `ViewHolder`).

Se ha modificado la clase `FilmItemViewHolder` para que incluya un nuevo atributo `gestureDetector` y para que en sí el *view holder* implemente el interface `View.OnTouchListener`. En la implementación del método exigido por esta interface (`fun onTouch(view: View, event: MotionEvent): Boolean`) se utiliza el atributo `gestureDetector` para gestionar el `MotionEvent`. Es importante hacer notas que el método debe devolver `false` para que se puedan seguir detectando los gestos relativos a la animación *swipe*.

Si en el `gestureDetector` se ejecutase el método `fun onSinleTapUp()` implicaría que se ha detectado el gesto *onClick* y, por tanto sería ahí donde se debería ejecutar la acción correspondiente. Por este motivo, ahora es necesario que se le deba especificar al *adapter* qué acción debe ejecutarse cuando se detecte los *onClick*. Para representar esta acción se debe facilitar un valor al atributo `onItemClickListener` del *adaper* (a través del nuevo método `setOnClick`) El dato debe ser una función que reciba como parámetro un `Film.ItemFilm` y devuelva `Unit`. Este valor será fijado por la *activity* después de crear el *adapter*:

```
adapter.setOnItemClickListener { filmItem -> filmItem.edit() }
```