**¿Por qué no es buena idea usar esta forma de solución?**

1. No es buena práctica abrir una Activity desde un contexto de ViewHolder (el método lanzarActivityVerEquipo).
2. No es buena idea usar campos estáticos si se puede evitar (y nosotros usamos la listaEquipos en el MainActivity).
3. **El VIEWHOLDER NO DEBE incluir lógica de la aplicación**, **y llamar a una segunda pantalla es lógica**. La FUNCIÓN DEL VIEWHOLDER debe ser sólo inflar las vistas (los TextView, ImageView,…) y darles valores (rellenarlos). Como mucho se le puede añadir alguna pequeña lógica relacionada con la UI de sus vistas, pero **NUNCA LÓGICA de la aplicación.**
4. La lógica debería situarse en el adaptador y no en el ViewHolder (la menor posible y relacionada con sus funciones) y el resto en las correspondientes activities o fragments de la aplicación.

**¿Cómo solucionamos los problemas anteriores?**

**RESPUESTA**: usando **LISTENERS**

Los Listeners ya los usasteis al programar el click de los botones. En concreto usasteis el OnClickListener. ¿Qué es? Es un escuchador que asignamos a una vista con el método “setOnClickListener” y que le dice a esa vista quién va a escuchar sus clicks y por tanto, a reaccionar a ellos. En ese ejemplo del click, quien reacciona es un método “onClick” (ahí es donde ponéis vuestro código).

Internamente ese OnClickListener es un INTERFAZ de Java que incluye un método “onClick”, que sólo se desarrolla cuando lo implementamos.

Bueno, pues la idea aquí, en nuestra solución ideal para los equipos de fútbol va a ser la misma; con la única diferencia de que en este caso, Android no nos proporciona un Listener, sino que tendremos que hacerlo nosotros. ¿Cómo se hace? Es un interfaz Java, que llevará un método dentro y que definiremos en el Adaptador. Luego, cuando lo queramos usar, debemos implementarlo y llamar a su método.

Vamos a verlo. Es la FORMA 2.

**FORMA 2: SOLUCIÓN IDEAL**: Usando un interface propio para informar de que se pulsó un click en un elemento de la lista: