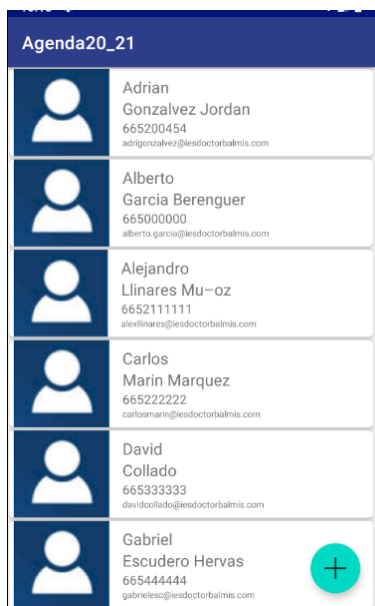


# Exercici Proposat Agenda

[Descarregar aquest exercici](#)

A partir de l'exercici del bloc 7 **Proposat Agenda**, afegirem més funcionalitat usant els nous coneixements adquirits en el bloc 8, Començarem afegint un nou fragments que contindrà un `recyclerView`. Al principi l'agenda no estarà ordenada i les dades es guardaran en un `ArrayList`, les insercions en la mateixa seran seqüencials (pel final).

L'aspecte serà com el següent, i a més podrem interactuar amb els elements del `recycler` (realitzar crida, eliminar, etc)



És convenient mantindre la llista de dades en l'activitat principal com a element estàtic, per a no perdre la informació en navegar entre els fragments.

## Funcionalitat del Clic en el Botó flotant

En prémer sobre el **FAB** es carregarà el `FragmentNuevoContacto`, que ja teníem construït en el bloc anterior. Ara es llevarà la funcionalitat de lliscament. Aquest fragment permetrà ficar les dades de l'usuari, avisarà amb un `SnackBar` en prémer sobre el **FAB** en cas que no s'haja etiquetat amb almenys un element o si no s'ha emplenat com a mínim nom i tlf.

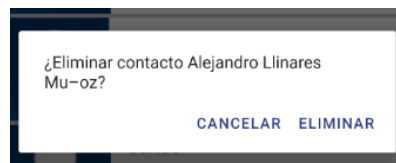
Si s'emplenen les dades mínimes i premem sobre FAB, es guardarà el nou contacte en el array , es tancarà aquest Fragment tornant al `FragmentRecycler`.

## Funcionalitat del Clic sobre un element del recycler

En cas de realitzar una pulsació curta **OnClick**, es carregarà el FragmentEditarContacto amb les dades del contacte premut. Per a això podem usar un **ViewMode**. També controlarem si es deixa alguns de les dades imprescindibles sense emplenar mitjançant Toast. Si és tot correcte en prémer el **FAB** es modificarà el contacte corresponent.

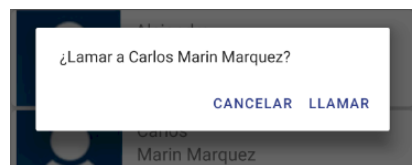
## Funcionalitat del Long Clic

En cas que l'usuari realitze una **pulsació llarga** sobre algun element del recycler, es mostrarà un dialogue preguntant si es vol eliminar el contacte. Només s'eliminarà si premem a eliminar, en cas contrari es tancarà el dialogue. Eliminarem el contacte del array i actualitzarem l'adaptador.

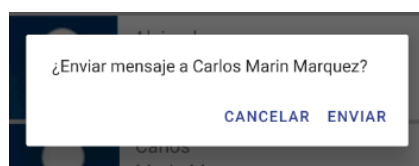


## Funcionalitat del desplaçament esquerra/dreta sobre algun element del recycler

Si en la finestra inicial faig un **desplaçament cap a la dreta** damunt d'un contacte, llavors llance el intent del dial amb el telèfon del contacte, mostrant un dialogue que permeti acceptar o rebutjar la crida. Si no té telèfon s'avisarà amb un missatge.



Si en la finestra principal faig un **desplaçament cap a l'esquerra**, es manarà un correu al contacte. Si no té adreça de correu es presentarà un missatge indicant-lo.



💡 Per a ajudar a la funcionalitat dels moviments dreta i esquerra, s'adjunta la classe **'swipeDetector'** que gestionarà el tipus de moviment. Aquesta classe implementa un listener per a detectar els moviments sobre la pantalla. En el fragment del recycler definirem, una variable d'aquesta classe:

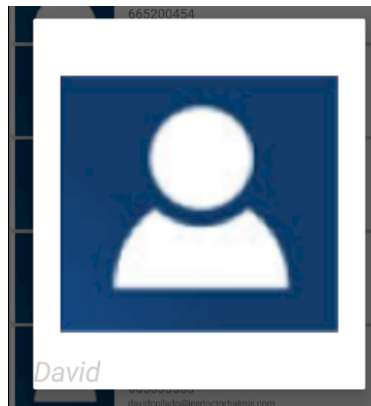
```
var swipeDetector: SwipeDetector= SwipeDetector()
```

I a continuació escriurem el `setOnItemClickListener()` del nostre adaptador usant els mètodes de la classe per a detectar els moviments. Important, haurem de passar l'objecte `swipeDetector` al listener corresponent de la classe adaptador.

```
adaptador.setOnTouchListener(swipeDetector)
```

## Funcionalitat del Clic sobre la imatge del contacte

En cas de prémer sobre la imatge es mostrarà un dialogue personalitzat en el qual la imatge es mostrarà d'una grandària major juntament amb el nom del contacte, com podem veure en la següent captura



## Funcionalitat del Lliscament esquerra/dreta sobre el Fragment d'edició

En cas de realitzar un lliscament sobre la pantalla edició carregarem el següent element de la llista, de manera que podrem recórrer tots els contactes des d'aquesta pantalla.

Lògicament el lliscament a la dreta carregarà l'element següent al mostrat en aqueix moment , mentre que cap a l'esquerra es carregarà l'anterior. Haurem de controlar que en arribar al primer o últim element, ja no ocórrega res per a evitar desbordaments del array.

Per a controlar el lliscament utilitzarem la classe que ja coneixem `SwipeDetector` , crearem un objecte d'aquest tipus que serà llançat amb el mètode `seOnTouchListener` de la vista sobre la qual vulguem controlar el moviment. Després haurem de controlar el moviment en el mètode **OnClick**.