# INTRODUCCION:

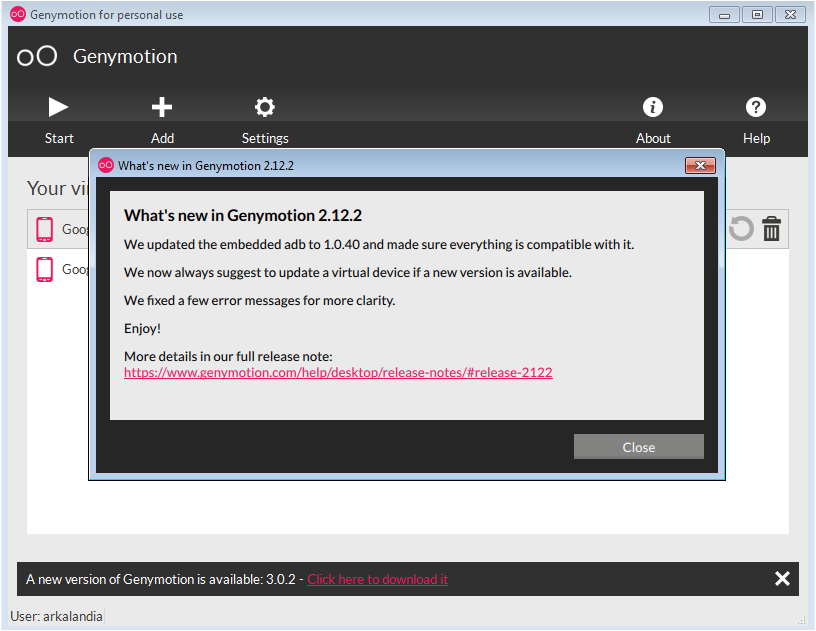
Tengo un procesador FX-8350 con una instalación de Windows Ultimate. Me he visto obligado a realizar las tareas con mucha paciencia ya que utilizaba las imágenes ARMEabi (de 10 a 20 veces más lentas bajo este Setup de lo normal).

Debido a eso este proyecto lo empiezo creándome un entorno de desarrollo decente.

# ENTORNO: GENYMOTION PARA EL PROYECTO

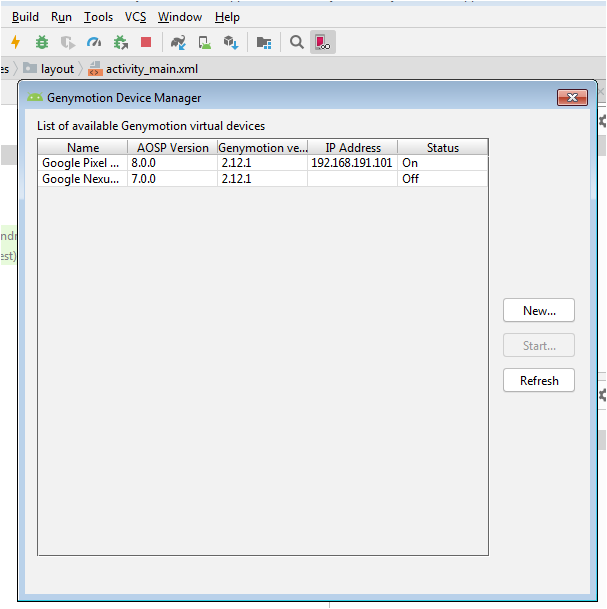
Creando un entorno para emular aplicaciones válido en mi equipo (AMD FX y win7).

1-Instalar genymotion Android emulator (Version Free y registrarse).



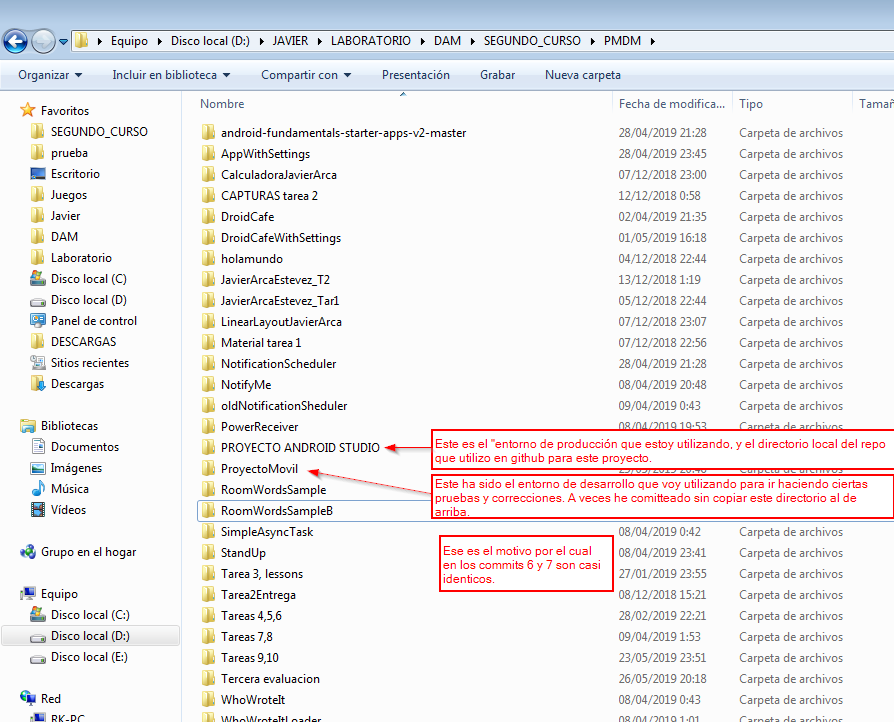
# ENTORNO: INSTALANDO PLUG-IN

Instalo el plugin Studio genymotion (Vía plugins en SDK).



# ENTORNO DE PRODUCCION Y DESARROLLO

Explicado de forma breve en la siguiente fotografía.

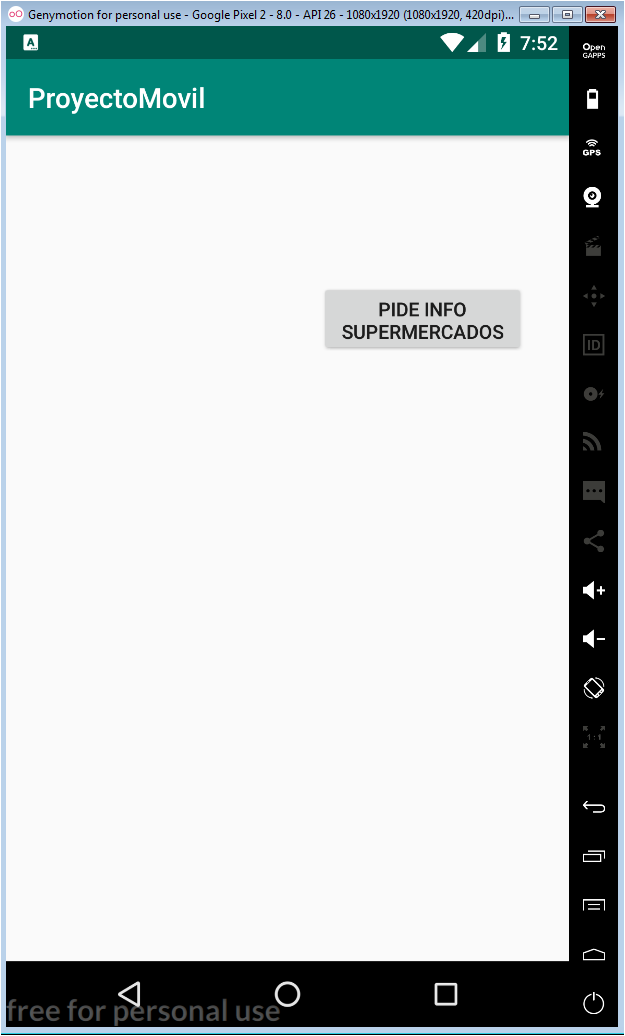
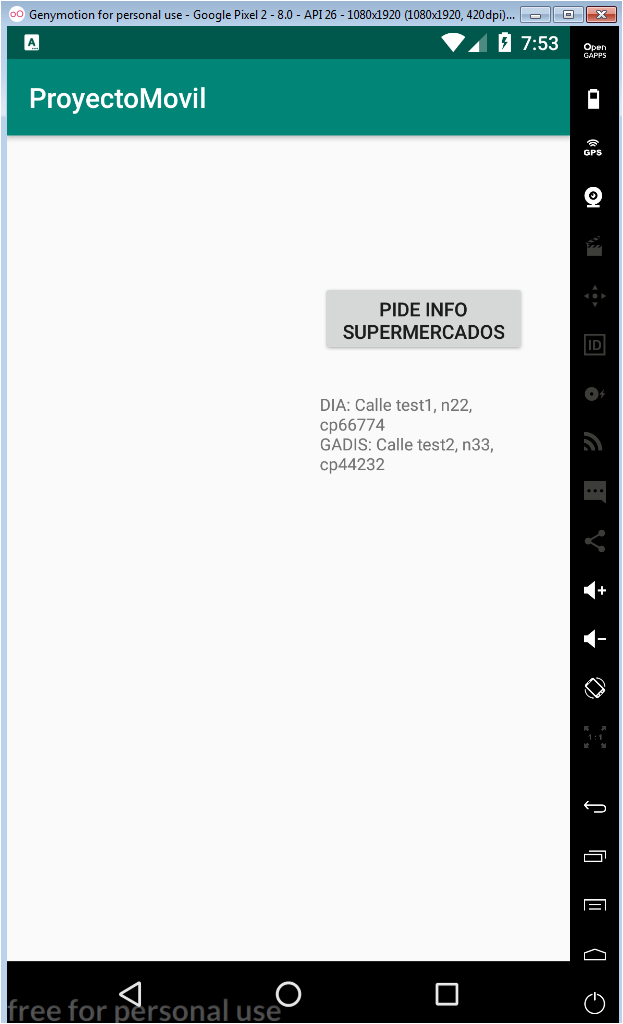


El proyecto final es ProyectoMóvil.

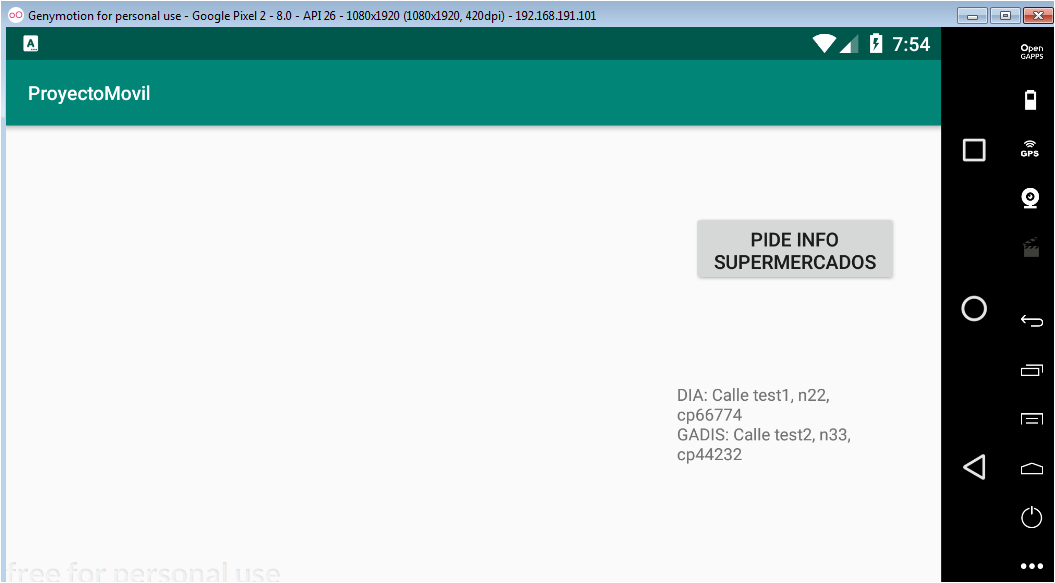
# GENERANDO LA PRIMERA ASYNCTASK DEL PROYECTO

NOTA: Hay más de una AsyncTask (Hay también la de la BD).

Implementando **AsyncTask** para mostrar información supermercados: **SI**

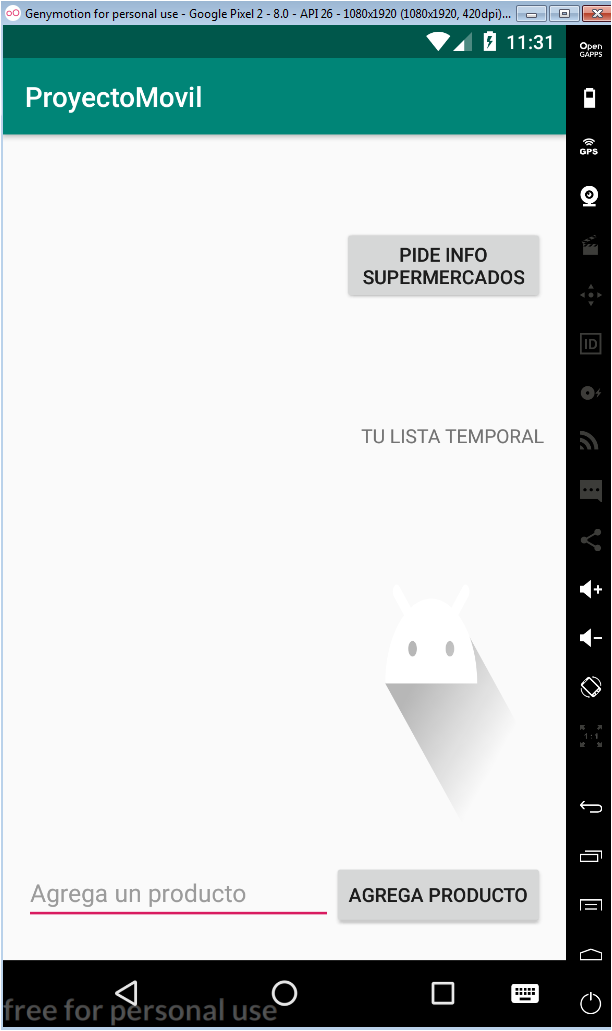
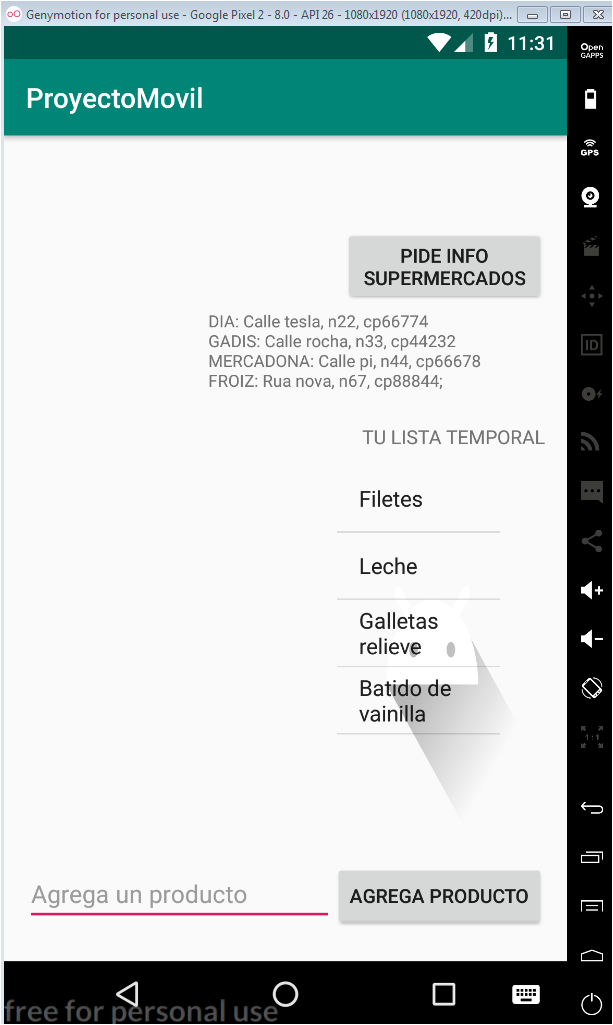
 

# TESTEANDO SCREEN ROTATION INICIAL



# MADAPTER PARA LISTA DE RECADOS

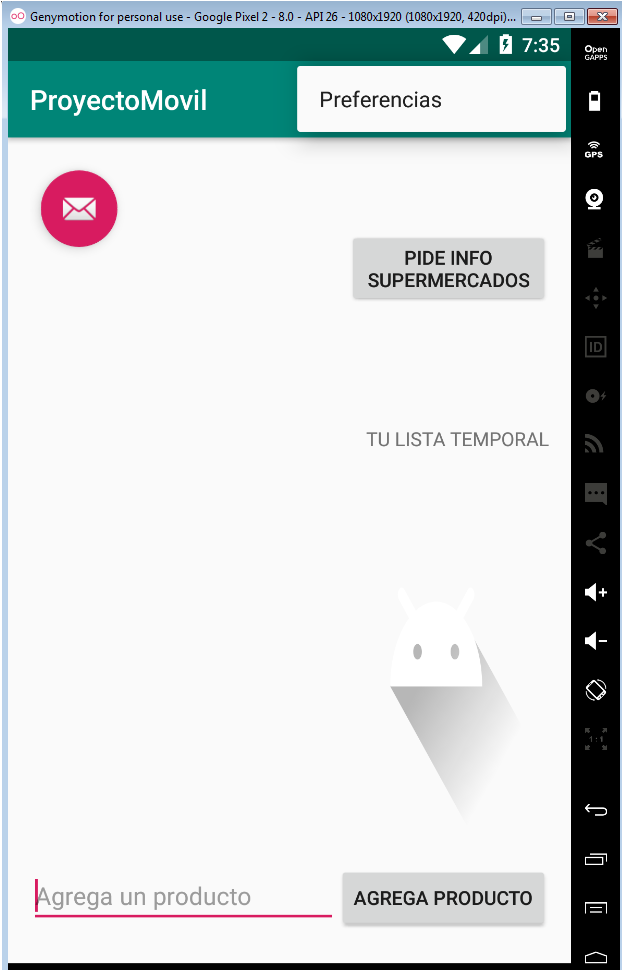
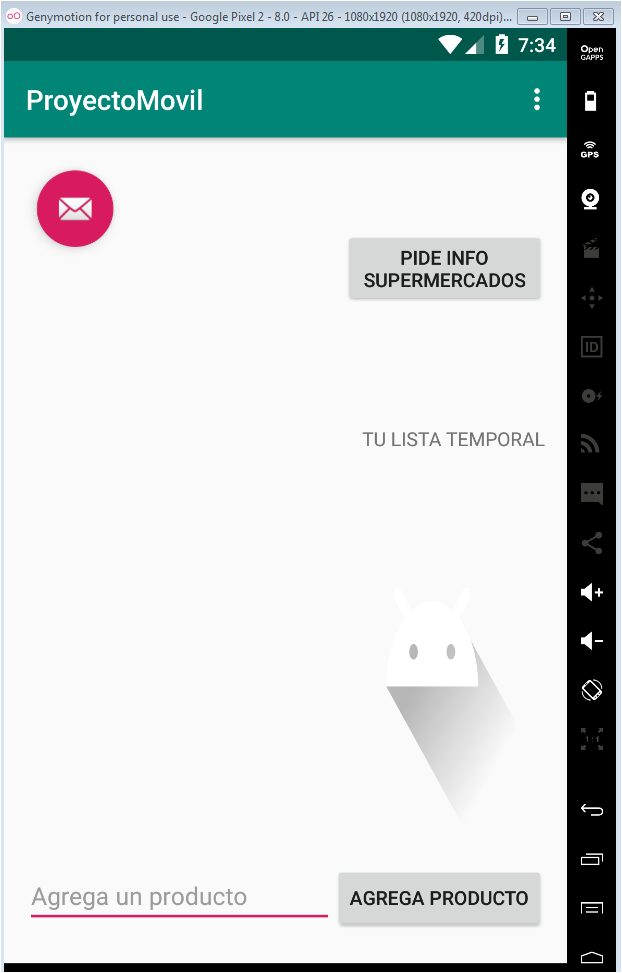
Para mostrar lista temporal (Será la lista de los recados/imprevistos).

Este adapter lo hago para generar listados temporales de artículos, no está en el contenido de esta evaluación pero es interesante para un día ir al super por si te acuerda alguna cosa por el camino. Me pareció que el m adapter quedaba bien.

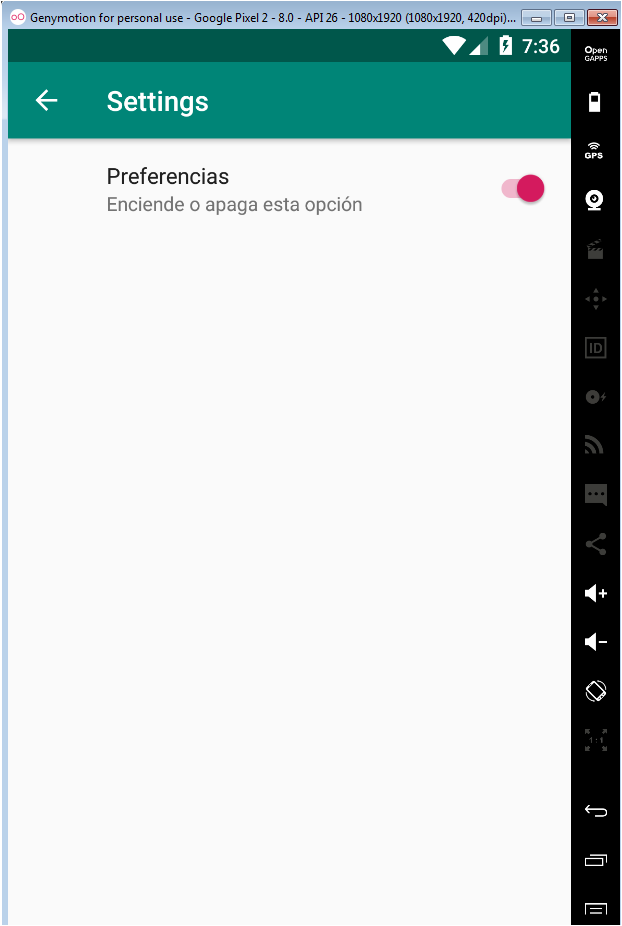
***NOTA:*** *Pendiente crear una forma de limpiar este listado.*

# IMPLEMENTANDO SETTINGS

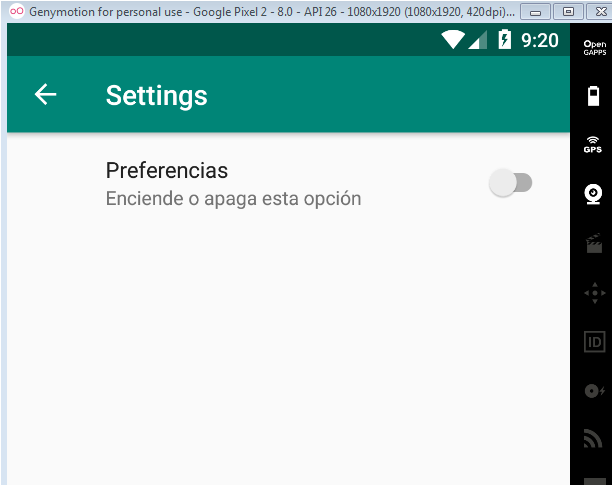


# SHARED PREFERENCES CON TOAST

Se implementa opción en Settings y un mensaje al volver a la actividad principal que informa si está activa o no dicha opción.



# TESTEANDO TOAST DE PREFERENCES

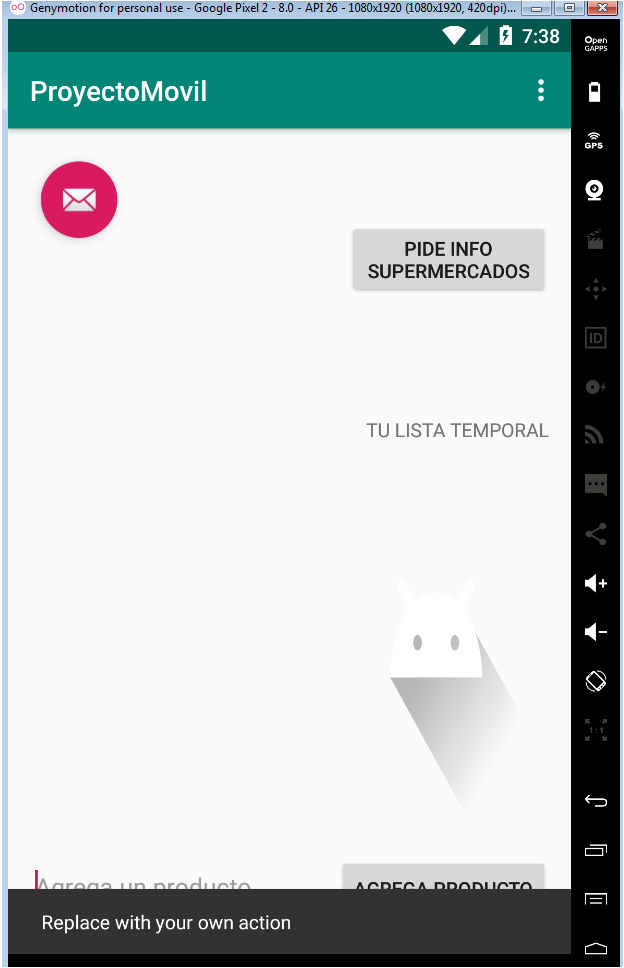
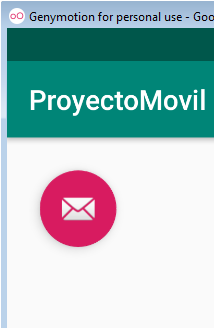


Se despliega el “mensaje” de false en opciones.

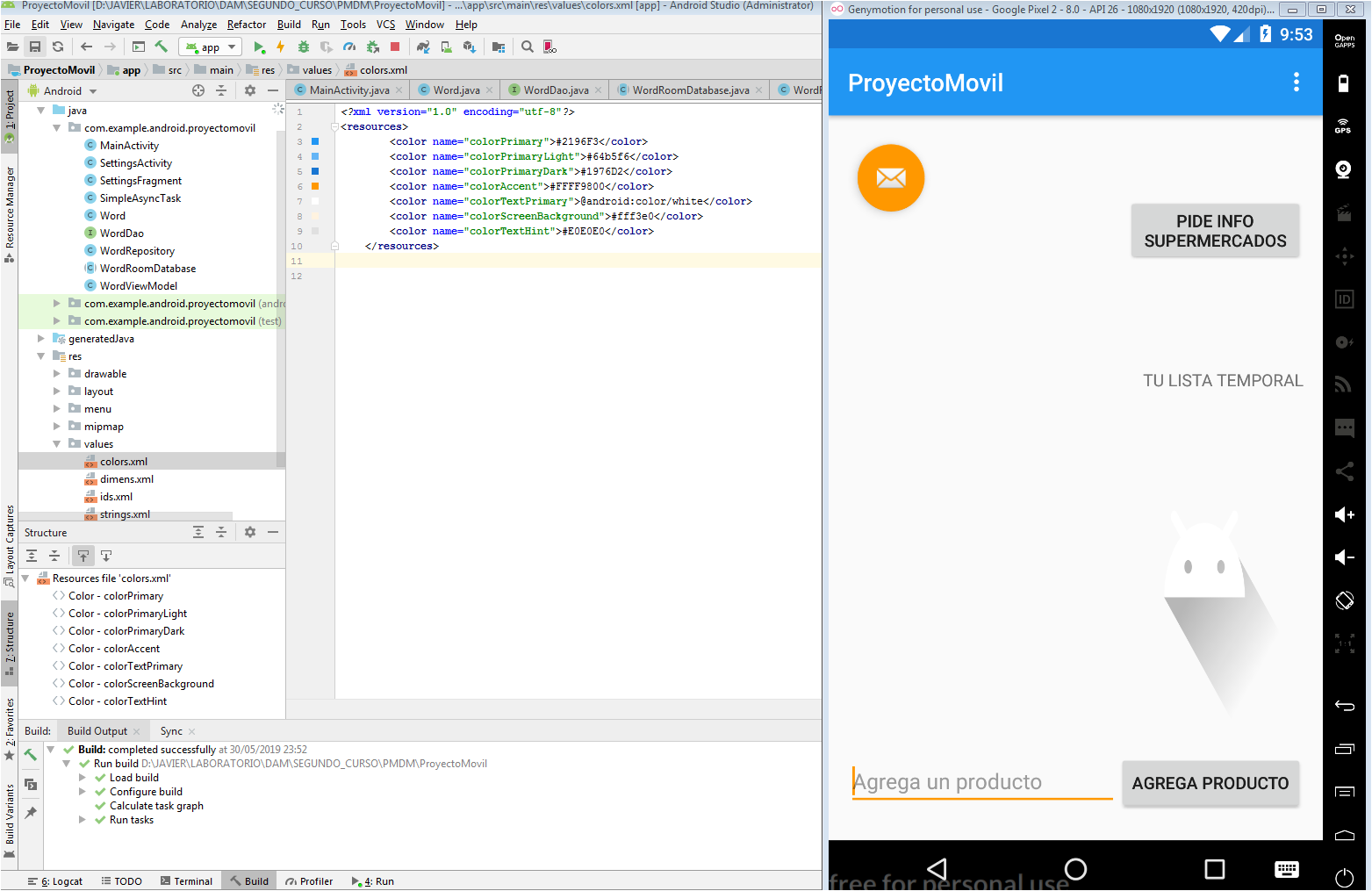
# AGREGANDO FLOATING BUTTON

FAV: POSTERIORMENTE SERÁ PARA AGREGAR ARTICULOS EN LA LISTA DEFINITIVA.

(Me gustaría engancharle una opción para enviar la lista a otro dispositivo…..)



# IMPLEMENTANDO ROOM



# REDISEÑO GRAFICO DE LA APLICACIÓN.

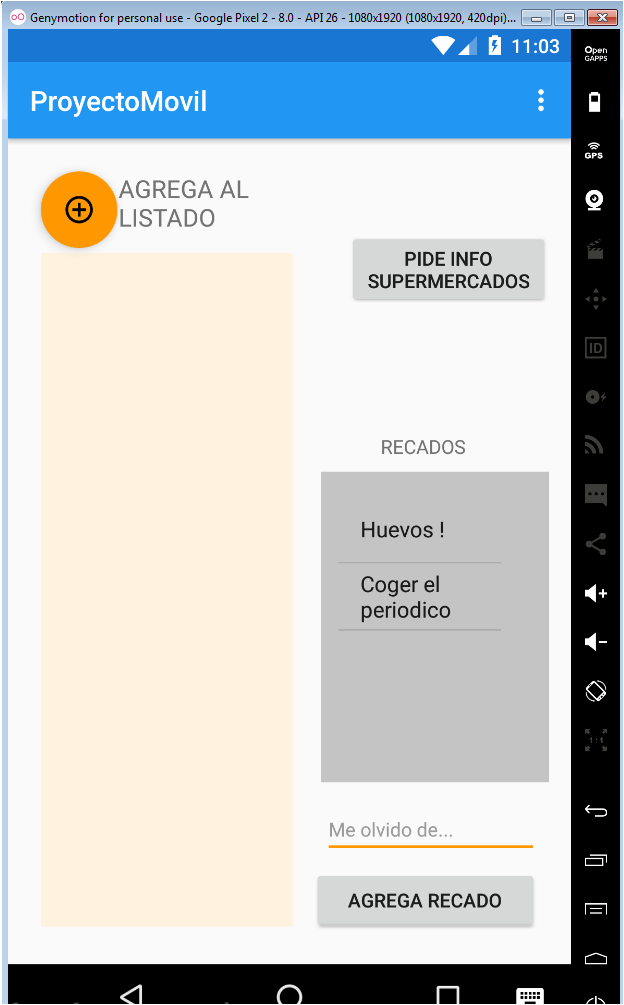
- Styles

- Paleta de colores

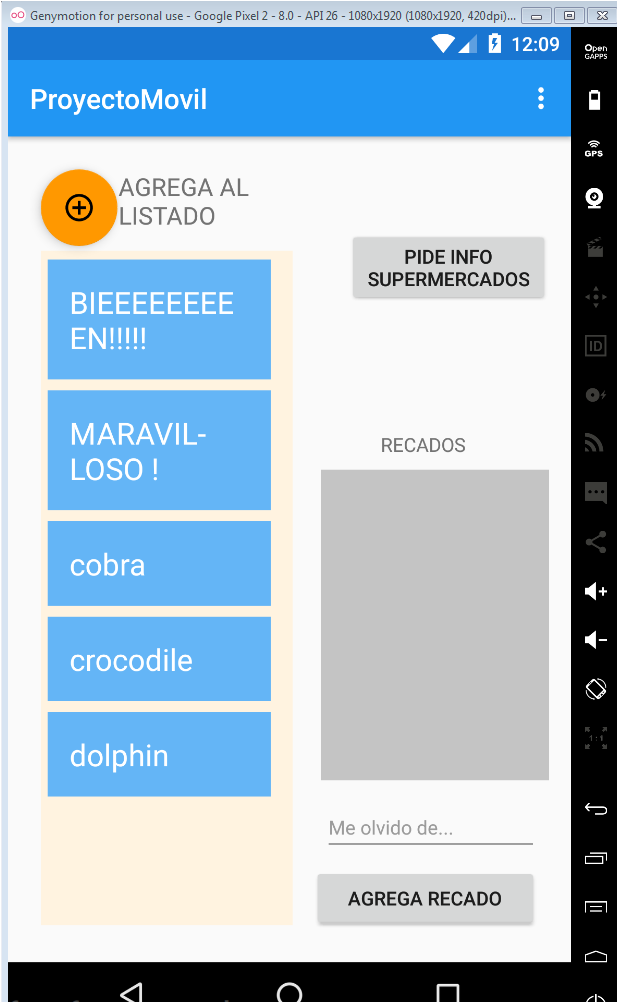
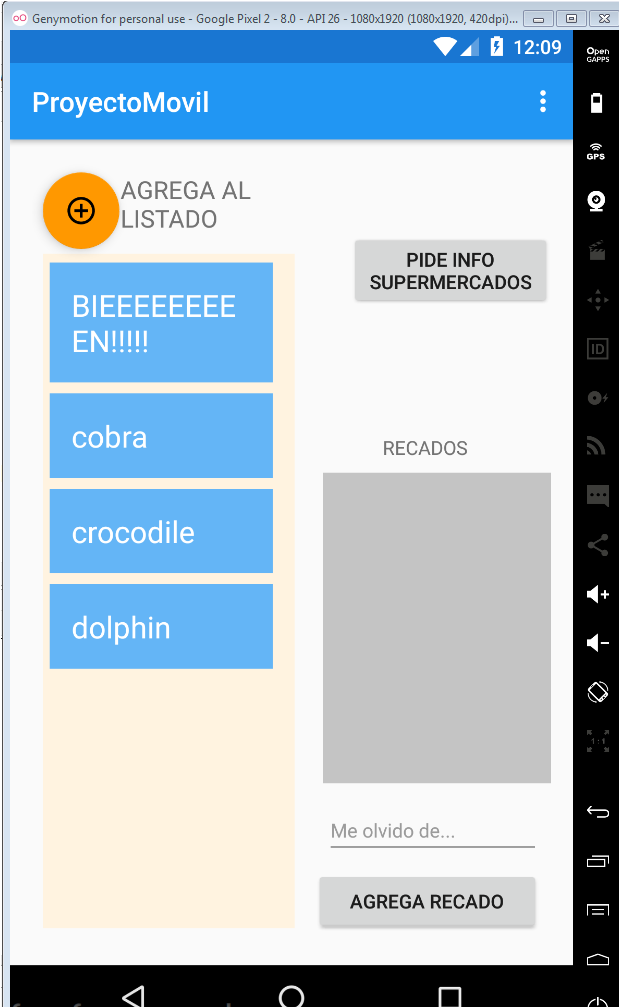
- layouts personalizados (Incluyendo Layout-land para vista panorámica)

…Y alguna cosa más (Drawables).

A continuación un vistazo.

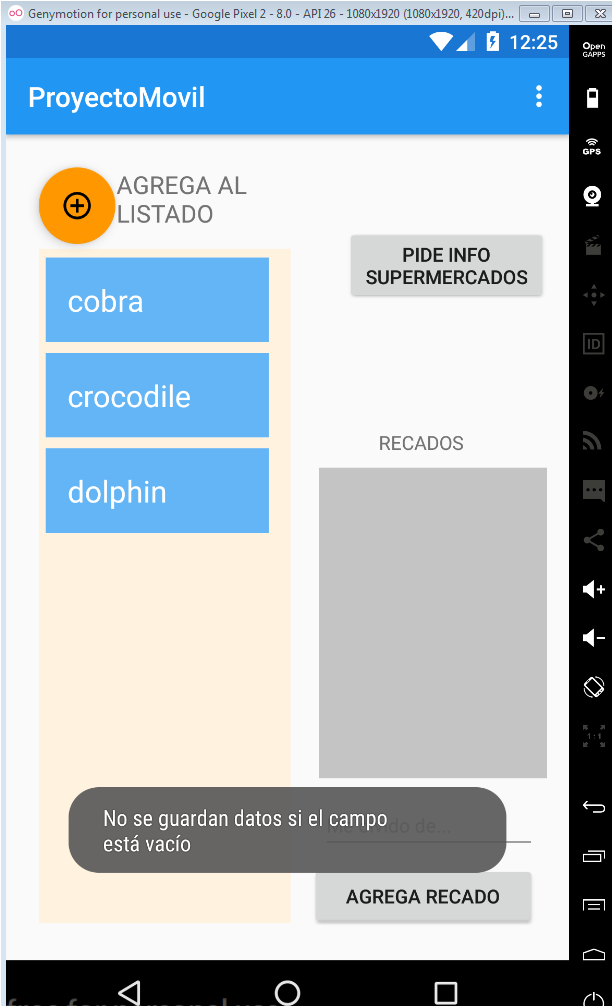
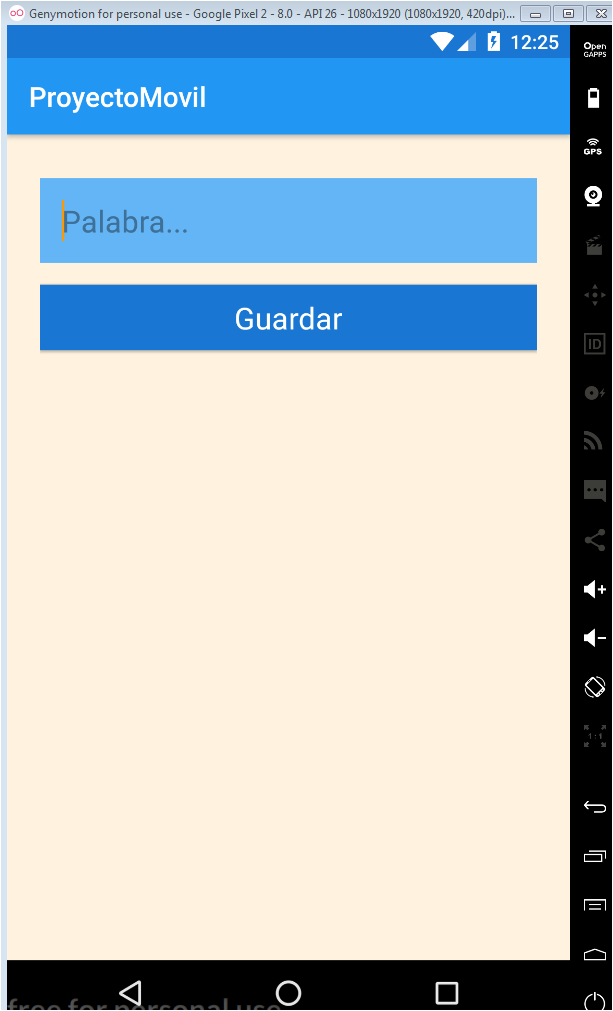


# IMPLEMENTANDO DAO, WORDLISTADAPTER, WORDREPOSITORY, WORDROOMDATABASE…



# FUNCIONAMIENTO DE @INSERT ON COLLISION

Aquí vemos que funciona correctamente.

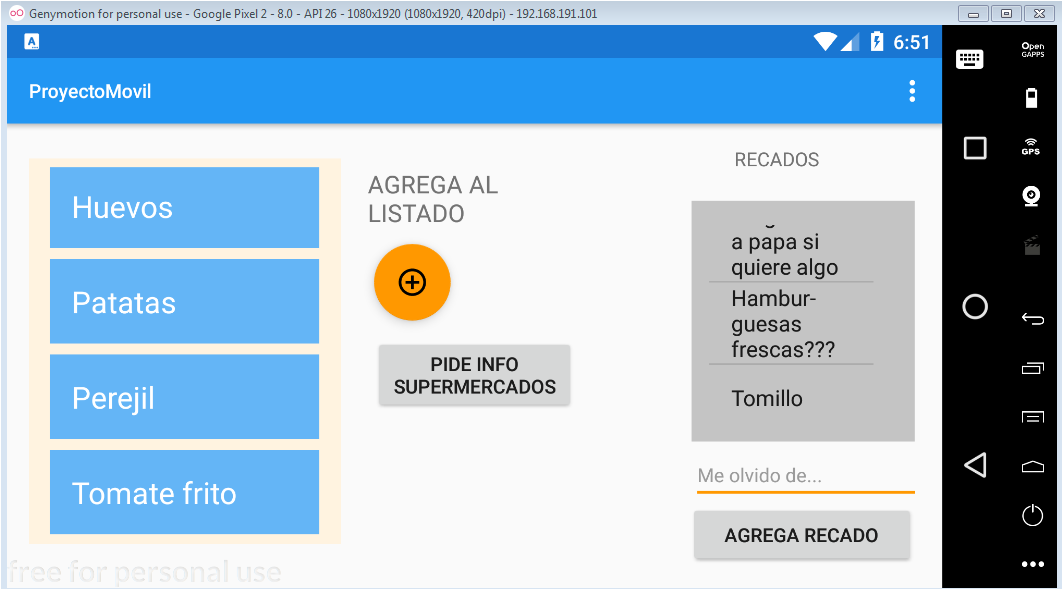


# LAYOUT-LAND PARA UNA VISTA PANORAMICA

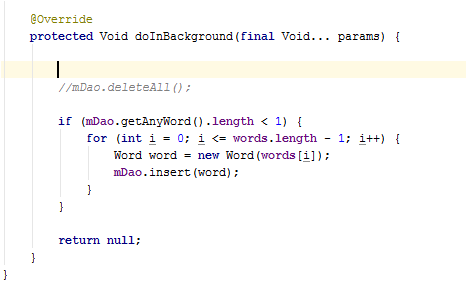


# COMPROBANDO DESLIZAMIENTO DE LISTAS

Compruebo que ambas listas se pueden deslizar para visualizar el resto de artículos.



# PERSISTENCIA ETERNA PARA LISTA PRINCIPAL (EXISTA LO QUE EXISTA)

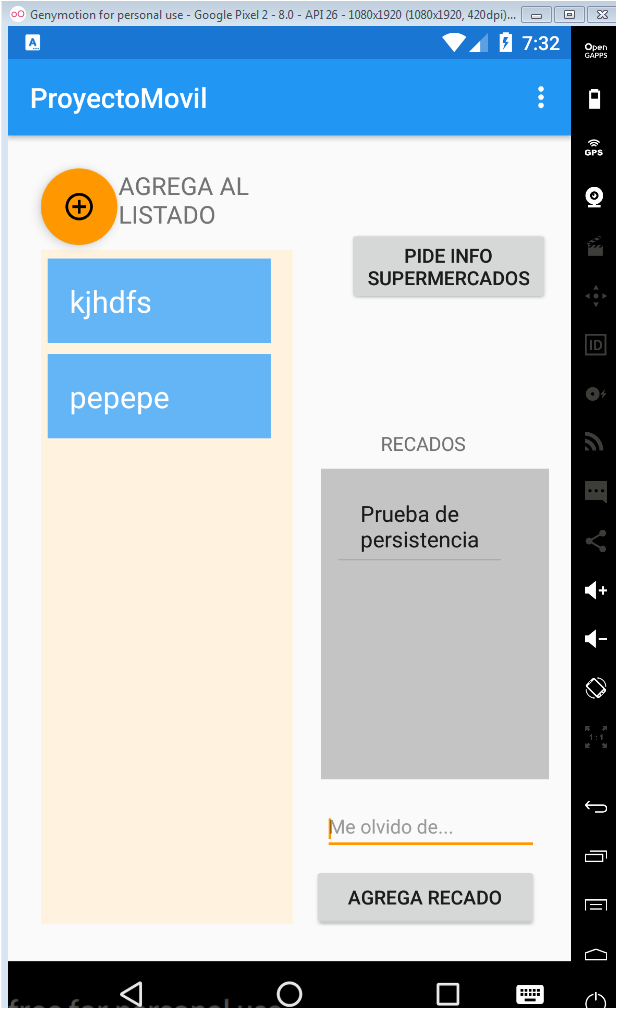
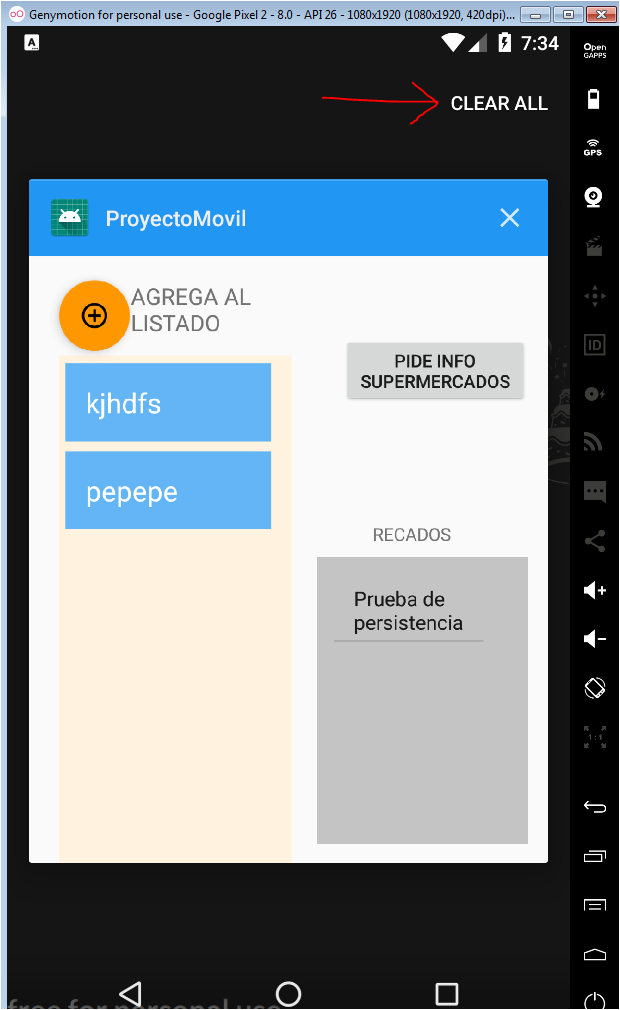


IMPORTANTE: Posteriormente cuando hagamos ese método utilizable vía mWordViewModel agregaré un botón para limpiar la lista persistente.

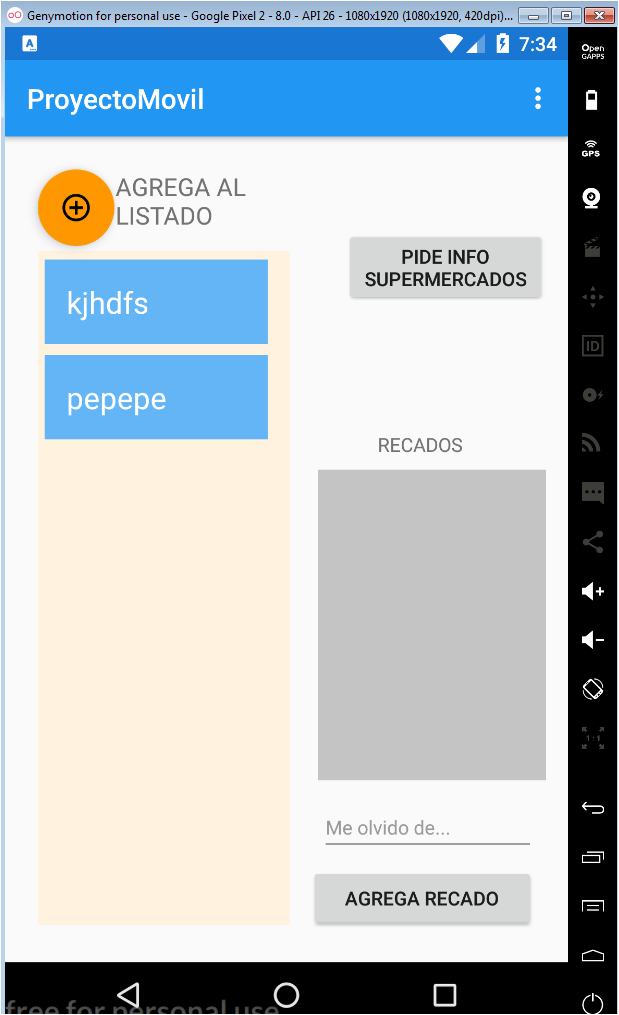
# COMPROBANDO PERSISTENCIA DE LISTA PRINCIPAL

Agregamos unos artículos a la lista principal y uno a la temporal como prueba.

Cerramos la aplicación (Usando clear all).

Y tras abrir de nuevo la aplicación vemos que ya tenemos persistencia en la lista principal.



# BOTON PARA LIMPIAR TODA LA LISTA PERSISTENTE

Agregamos unos artículos en la lista persistente…



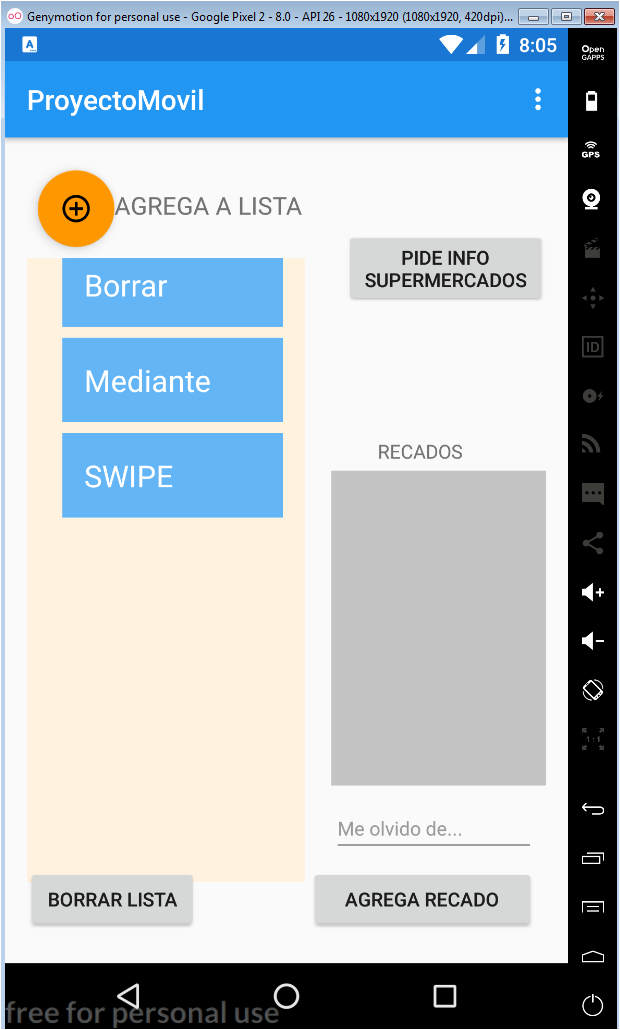
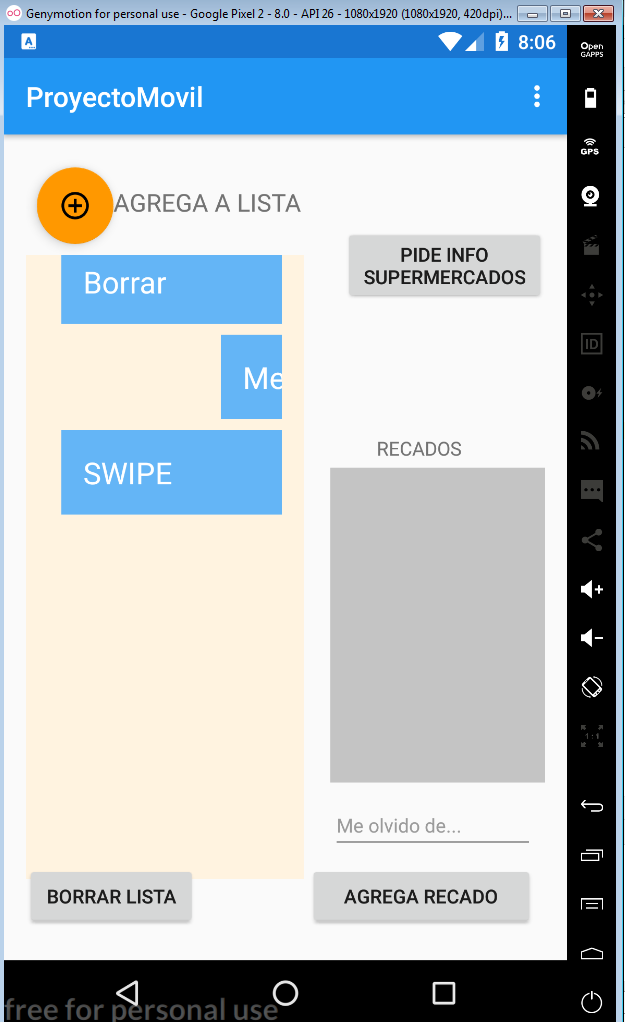
Y pulsamos el botón de borrar lista, todo correcto



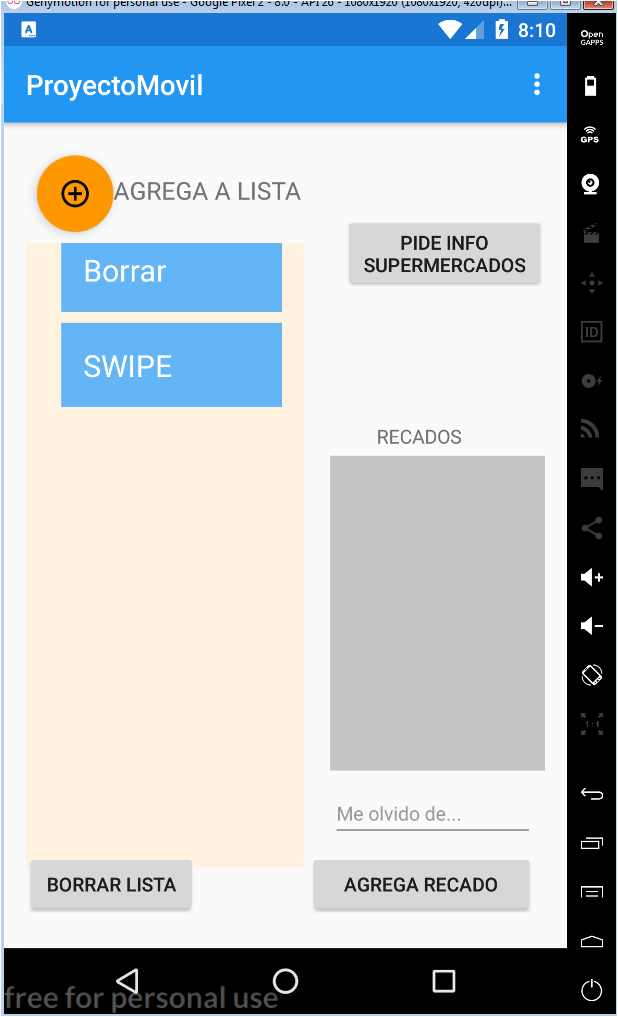
# BORRADO VIA SWIPE DE UN ARTÍCULO DE LA LISTA

Ya tenemos el swipe implementado para borrar artículos arrastrándolos en horizontal.

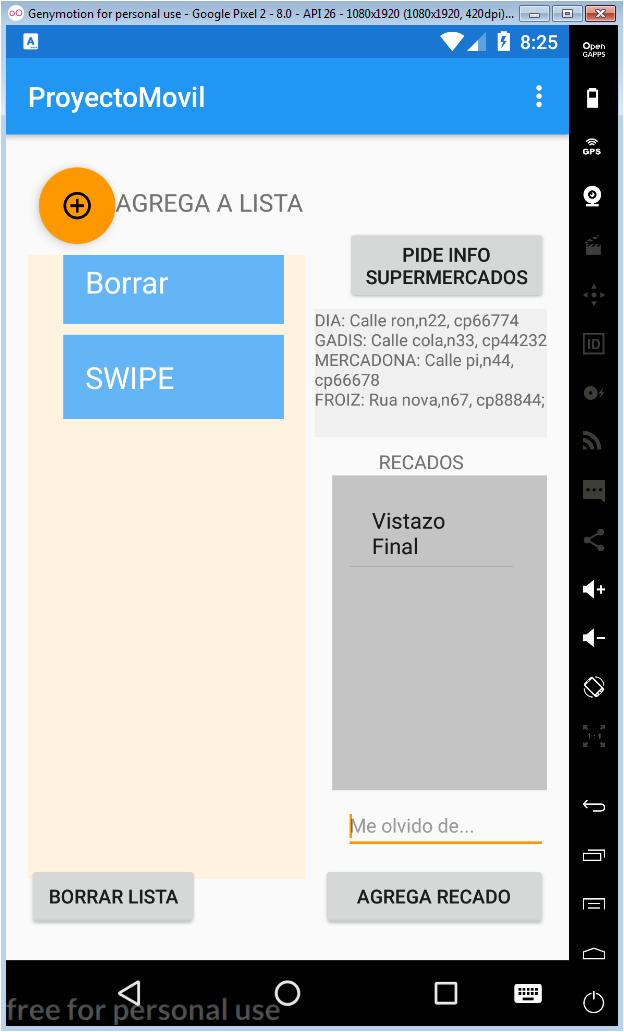
Creamos algunos artículos y probamos a “swipear” uno de ellos

Y vemos que funciona correctamente.



# RETOCANDO DISEÑOS HORIZONTAL Y VERTICAL + VISTAZO FINAL





DEPURANDO Y OPTIMIZANDO CODIGO:

Lo último que haremos será un remove unused resources, seguro que hay ficheros que no utilizamos.