Documentación

Proyecto agenda

Iván Gallego

Índice de contenidos

Main activity

La clase principal debe implementar los listener para gestionar el click sobre los contactos y el click largo

```
public class MainActivity extends AppCompatActivity implements
        OnClickItemListener, OnEditContact, OnAddContact, OnFabClicked
```

Variables de clase

```
public ArrayList<Contacto> contactos;
    private int indiceListaPulsado;
    private DrawerLayout drawerLayout;
    private NavigationView navigationView;
    private Layout layout;
    // El fragment que contiene el nombre
    private VistaContactos vistaContactos;
    private OnRecyclerUpdated onRecyclerUpdated;
```

`onCreate`

1. Asignamos la vista, cargamos los datos
2. Creamos la toolbar

3. Le asignamos a la toolbar un botón de home
4. Detectamos la orientación de la pantalla, si está en horizontal, el layout será `GRID` y si no será `LINEAR`
    - `LINEAR` o `GRID` es un enumerado de tipo `LAYOUT`
5. Inicializamos el fragment, con el fragment que contiene el `RecyclerView` utilizando el método `onReplaceFragment`
6. Asignamos el listener que nos permitirá detectar cuál es el botón pulsado
    - El listener filtrará los elementos del `recycler` dependiendo del elemento pulsado

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    cargarDatos();

    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    ActionBar actionBar = getSupportActionBar();
    actionBar.setHomeAsUpIndicator(R.drawable.ic_menu);
    actionBar.setDisplayHomeAsUpEnabled(true);

    drawerLayout = findViewById(R.id.drawer_layout);

    if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_PORTRAIT) {
        layout = Layout.GRID;
    } else {
        layout = Layout.LINEAR;
    }

    replaceFragment();

    navigationView = findViewById(R.id.navigation_view);
    navigationView.setNavigationItemSelectedListener(
            new NavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
            switch (menuItem.getItemId()) {
                case R.id.familia_menu_option:
                    vistaContactos.adaptador.getFilter().filter("Familia");
                    break;
                case R.id.amigos_menu_option:
                    vistaContactos.adaptador.getFilter().filter("Amigo");
                    break;
                case R.id.trabajo_menu_option:
                    vistaContactos.adaptador.getFilter().filter("Trabajo");
                    break;
                case R.id.todos_menu_option:
                    vistaContactos.adaptador.getFilter().filter("Todo");
                    break;
            }
            onRecyclerUpdated.onRecyclerUpdated(layout);
```

```
                menuItem.setChecked(true);
                drawerLayout.closeDrawers();
                return true;
            }
        });
    }
```

`replaceFragment`

Este método nos cambiará el fragment que esté actualmente activo y colocará en su lugar el fragment que contiene el `recycler`

```
private void replaceFragment() {
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    vistaContactos = new VistaContactos(contactos, layout);
    onRecyclerUpdated = vistaContactos;
    transaction.add(R.id.fragment_container, vistaContactos);
    transaction.addToBackStack(null);
    transaction.commit();
}
```

`onClickItemListener`

Al hacer click sobre un elemento del recycler, reemplazaremos el fragment del `recycler` por el fragment de editar datos

```
@Override
public void onClickItemListener(Contacto contacto, int i) {
    indiceListaPulsado = i;
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    AccionContacto fragment = new AccionContacto();
    Bundle args = new Bundle();
    args.putParcelable("contacto", contacto);
    fragment.setArguments(args);
    transaction.replace(R.id.fragment_container, fragment);
    transaction.addToBackStack(null);
    transaction.commit();
}
```

`onEditContact`

A través de este método, recogemos el contacto que se ha editado al modificar un contacto a travé del fragment apropiado, después, cambiamos el fragment por el que tiene el `recycler`

```
@Override
public void onEditContact(Contacto contacto) {
    contactos.set(indiceListaPulsado, contacto);
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    VistaContactos fragment = new VistaContactos(contactos, layout);
    onRecyclerUpdated = fragment;
    transaction.replace(R.id.fragment_container, fragment);
    transaction.addToBackStack(null);
    transaction.commit();
}
```

`onAddContact`

Parecido al método anterior, recogemos el dato que se ha generado y lo añadimos al array de datos, después, reemplazamos el fragment de añadir, con el del `recycler`

```
@Override
public void onAddContact(Contacto contacto) {
    contactos.add(contacto);
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    VistaContactos fragment = new VistaContactos(contactos, layout);
    onRecyclerUpdated = fragment;
    transaction.replace(R.id.fragment_container, fragment);
    transaction.addToBackStack(null);
    transaction.commit();
}
```

`onFabClicked`

Con este método, recibimos cuando se ha pulsado el `FAB` desde el fragment principal del `recycler` y cambiamos el fragment actual por el de añadir contacto

```
@Override
public void onFabClicked() {
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    transaction.replace(R.id.fragment_container, new AccionContacto());
    transaction.addToBackStack(null);
    transaction.commit();
}
```

`onOptionsItemSelected`

Método para detectar las pulsaciones sobre los botones de la `Toolbar`

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    int id = item.getItemId();

    switch (id) {
        case android.R.id.home:
            drawerLayout.openDrawer(GravityCompat.START);
            break;
        case R.id.linear_option_menu:
            layout = Layout.LINEAR;
            onRecyclerUpdated.onRecyclerUpdated(Layout.LINEAR);
            break;
        case R.id.grid_option_menu:
            layout = Layout.GRID;
            onRecyclerUpdated.onRecyclerUpdated(Layout.GRID);
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

`cargarDatos`

```
private void cargarDatos() {
    contactos = new ArrayList<>();
    contactos.add(new Contacto("Iván", "Gallego", "601245789", "yo@yo.com"));
    contactos.add(new Contacto("Gallego", "Iván", "658984512", "yo@yo.com"));
    contactos.add(new Contacto("Daniel", "Acabado", "", ""));
    contactos.add(new Contacto("Estodorne", "Ideas", "658497415", "yo@yo.com"));
    contactos.add(new Contacto("Carlos", "Apellido", "684974523", ""));
    contactos.add(new Contacto("Juan", "Mastodonte", "", "yo@yo.com"));
    contactos.add(new Contacto("Marcos", "Calatraba", "784569815", ""));
    contactos.add(new Contacto("David", "Muñoz", "745123698", "yo@yo.com"));
    contactos.add(new Contacto("Sandra", "López", "696952356", "yo@yo.com"));
    contactos.add(new Contacto("Andrea", "García", "787878787", "yo@yo.com"));
    contactos.add(new Contacto("Ainhoa", "García", "", "yo@yo.com"));
}
```

`addContacto`

Llamo al activity `AccionContacto` que me devolverá un contacto nuevo

```
private void addContacto() {
    Intent i = new Intent(this, AccionContacto.class);
    startActivityForResult(i, COD_ACTIVITY_ADD);
}
```

`onCreateOptionsMenu`

Crear el menú de la `Toolbar`

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_layout, menu);
    return true;
}
```

Main activity XML

En el layout principal, el `DrawerLayout`, `Toolbar`, `FrameLayout` que contandrá el fragment y el `NavigationView`

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:theme="@style/ThemeOverlay.AppCompat.Dark">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="?attr/colorPrimary" />

        </com.google.android.material.appbar.AppBarLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <FrameLayout
```

```
                android:id="@+id/fragment_container"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_alignParentStart="true"
                android:layout_alignParentTop="true"
                android:layout_alignParentEnd="true"
                android:layout_alignParentBottom="true" />

        </RelativeLayout>

    </LinearLayout>

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/navigation_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/drawer_header"
        app:menu="@menu/menu_drawer" />

</androidx.drawerlayout.widget.DrawerLayout>
```

Adaptador

En el adaptador implementamos los métodos necesarios extender de `RecyclerView.Adapter` y creamos los nuestros para gestionar los `click`, `swipe`; además, implementamos la interfaz `Filterable` para filtrar los elementos del `recycler`...

```
public class Adaptador extends RecyclerView.Adapter
        implements View.OnClickListener, View.OnLongClickListener,
View.OnTouchListener, Filterable {
    private ArrayList<Contacto> contactos, contactosCompletos;
    private View.OnClickListener clickListener;
    private OnImageClickListener imageClickListener;
    private View.OnLongClickListener longClickListener;
    private View.OnTouchListener touchListener;
    private Layout layout;

    private Filter filter = new Filter() {
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {
            String seleccion = constraint.toString();
            ArrayList<Contacto> datosFiltrados = new ArrayList<>();
            if (constraint == null || seleccion.length() == 0) {
                datosFiltrados.addAll(contactosCompletos);
            } else if (seleccion.equals("Todo")) {
                datosFiltrados.addAll(contactosCompletos);
            } else {
                for (Contacto c : contactos) {
                    if (seleccion.equals("Amigo") && c.isAmigo())
```

```
datosFiltrados.add(c);
                    else if (seleccion.equals("Trabajo") && c.isTrabajo())
datosFiltrados.add(c);
                    else if (seleccion.equals("Familia") && c.isFamilia())
datosFiltrados.add(c);
                }
            }
            FilterResults results = new FilterResults();
            results.values = datosFiltrados;
            results.count = datosFiltrados.size();
            return results;
        }

        @Override
        protected void publishResults(CharSequence constraint, FilterResults
results) {
            contactos.clear();
            contactos.addAll((List) results.values);
            notifyDataSetChanged();
        }
    };

    public Adaptador(ArrayList<Contacto> contactos, Layout layout) {
        this.contactos = contactos;
        contactosCompletos = new ArrayList<>(contactos);
        this.layout = layout;
    }

    @NonNull
    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        final View v;
        if (layout == Layout.GRID) {
            v = LayoutInflater.from(parent.getContext()).inflate(
                R.layout.entrada_agenda_compact, parent, false);
            v.setOnLongClickListener(this);
            v.setOnClickListener(this);
            v.setOnTouchListener(this);
            HolderCompact h = new HolderCompact(v);
            h.setImageClickListener(new OnImageClickListener() {
                @Override
                public void onImageClick(Contacto contacto, View view) {
                    imageClickListener.onImageClick(contacto, v);
                }
            });
            return h;
        } else {
            v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.entrada_agenda, parent,
false);
            v.setOnLongClickListener(this);
            v.setOnClickListener(this);
            v.setOnTouchListener(this);
```

```
            Holder h = new Holder(v);
            h.setImageClickListener(new OnImageClickListener() {
                @Override
                public void onImageClick(Contacto contacto, View view) {
                    imageClickListener.onImageClick(contacto, v);
                }
            });
            return h;
        }
    }

    @Override
    public void onBindViewHolder(
            @NonNull RecyclerView.ViewHolder holder, int position) {
        if (layout == Layout.GRID) {
            ((HolderCompact) holder).bind(contactos.get(position));
        } else {
            ((Holder) holder).bind(contactos.get(position));
        }
    }

    @Override
    public int getItemCount() {
        return contactos.size();
    }

    @Override
    public Filter getFilter() {
        return filter;
    }

    public void setOnClickListener(View.OnClickListener listener) {
        if (listener != null) {
            this.clickListener = listener;
        }
    }

    @Override
    public void onClick(View v) {
        if (clickListener != null) {
            clickListener.onClick(v);
        }
    }

    public void setOnLongClickListener(View.OnLongClickListener listener) {
        if (listener != null) {
            this.longClickListener = listener;
        }
    }

    @Override
    public boolean onLongClick(View v) {
        if (longClickListener != null) {
            longClickListener.onLongClick(v);
```

```
        }
        return false;
    }

    public void setOnTouchListener(View.OnTouchListener listener) {
        if (listener != null) {
            this.touchListener = listener;
        }
    }

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (touchListener != null) {
            touchListener.onTouch(v, event);
        }
        return false;
    }

    public void setImageClickListener(OnImageClickListener listener) {
        if (listener != null) {
            imageClickListener = listener;
        }
    }
}
```

Holder

Este es el `Holder` que utilizamos en el Adaptador, enlazamos los datos y los eventos...

```
class Holder extends RecyclerView.ViewHolder
        implements View.OnClickListener {
    private ImageView imagen;
    private TextView nombre, apellido, telefono, correo;
    private OnImageClickListener imageClickListener;
    private Contacto contacto;

    public Holder(View v) {
        super(v);
        imagen = v.findViewById(R.id.imageview);
        imagen.setOnClickListener(this);
        nombre = v.findViewById(R.id.nombre);
        apellido = v.findViewById(R.id.apellido);
        telefono = v.findViewById(R.id.telefono);
        correo = v.findViewById(R.id.correo);
    }

    public void bind(Contacto d) {
        nombre.setText(d.getNombre());
        apellido.setText(d.getApellido());
        telefono.setText(d.getTelefono());
        correo.setText(d.getCorreo());
```

```
            if (d.getImagen() != null) {
                imagen.setImageBitmap(d.getImagen());
            }
            contacto = d;
        }

        public void setImageClickListener(OnImageClickListener listener) {
            if (listener != null) {
                imageClickListener = listener;
            }
        }

        @Override
        public void onClick(View v) {
            if (imageClickListener != null) {
                imageClickListener.onImageClick(contacto, v);
            }
        }
    }
```

Contacto XML

El layout de cada entrada de la lista

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="5dp"
    app:cardElevation="5dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <ImageView
            android:id="@+id/imageview"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_default"/>

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:paddingStart="8dp"
```

```
            android:paddingTop="8dp"
            android:paddingBottom="8dp"
            android:layout_weight="3"
            android:orientation="vertical">

            <TextView
                android:id="@+id/nombre"
                style="@android:style/TextAppearance.Material.Medium"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/apellido"
                style="@android:style/TextAppearance.Material.Medium"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/telefono"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
                style="@android:style/TextAppearance.Material.Small"
                android:id="@+id/correo"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

        </LinearLayout>

    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Holder compacto

Este holder lo utilizo cuando la vista es de `GRID` en vez de `LINEAR`, solo cargamos el nombre y la imagen de perfil

```
class HolderCompact extends RecyclerView.ViewHolder implements
View.OnClickListener {
    private ImageView imagen;
    private TextView nombre;
    private OnImageClickListener imageClickListener;
    private Contacto contacto;

    public HolderCompact(View v) {
        super(v);
        imagen = v.findViewById(R.id.imageview);
        imagen.setOnClickListener(this);
        nombre = v.findViewById(R.id.nombre);
    }
```

```
    public void bind(Contacto d) {
        nombre.setText(d.getNombre());
        if (d.getImagen() != null) {
            imagen.setImageBitmap(d.getImagen());
        }
        contacto = d;
    }

    public void setImageClickListener(OnImageClickListener listener) {
        if (listener != null) {
            imageClickListener = listener;
        }
    }

    @Override
    public void onClick(View v) {
        if (imageClickListener != null) {
            imageClickListener.onImageClick(contacto, v);
        }
    }
}
```

Contacto XML compacto

Entrada de la agenda de un contacto compacta

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:cardCornerRadius="5dp"
    app:cardElevation="5dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
        android:layout_margin="16dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageview"
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_default" />
```

```
        <TextView
            android:id="@+id/nombre"
            style="@android:style/TextAppearance.Material.Medium"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Interfaces de comunicación de datos

```
public interface OnAddContact {
    public void onAddContact(Contacto contacto);
}

public interface OnClickItemListener {
    public void onClickItemListener(Contacto contacto, int i);
}

public interface OnEditContact {
    public void onEditContact(Contacto contacto);
}

public interface OnRecyclerUpdated {
    public void onRecyclerUpdated(Util.Layout layout);
}

public interface OnFabClicked {
    public void onFabClicked();
}

public interface OnImageClickListener {
    void onImageClick(Contacto contacto, View v);
}
```

Accion contacto

Este código lo utilizo tanto para editar un contacto, como para añadir uno nuevo

```
public class AccionContacto extends Fragment implements View.OnClickListener
```

Variables de clase

```
private OnEditContact editContact;
private OnAddContact addContact;
private EditText editNombre, editApellido, editTelefono, editCorreo;
```

```
private Button aceptarButton;
private Bitmap bitmap;
private Contacto contacto;
private ImageView imageView;
private RadioButton radioButtonFamilia, radioButtonTrabajo, radioButtonAmigos;
private final int EDITAR = 0;
private final int ADD = 1;
private final int COD_ELEGIR_IMAGEN = 2;
private final int COD_TOMAR_FOTO = 3;
private int proposito;
```

onCreate

1. Hacemos un inflate de la vista que queremos mostrar
2. Obtemos el contacto si existe, si lo hace, el propósito de el fragment es editar, si no, el propósito es añadir un contacto nuevo
3. Tomamos las acciones necesarias dependiendo del propósito
4. Asignamos los listener para la imagen para poder editarla o cambiarla, al hacer click, sale un Popup menú

```
@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
        @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
    super.onCreateView(inflater, container, savedInstanceState);
    View v = inflater.inflate(R.layout.accion_contacto, container, false);
    Bundle args = getArguments();

    if (args != null) {
        contacto = args.getParcelable("contacto");
    }
    proposito = contacto == null ? ADD : EDITAR;

    imageView = v.findViewById(R.id.profile_image_view);
    imageView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            PopupMenu pop = new PopupMenu(getContext(), v);
            pop.getMenuInflater().inflate(R.menu.menu_foto, pop.getMenu());
            pop.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()
{
                @Override
                public boolean onMenuItemClick(MenuItem item) {
                    switch (item.getItemId()) {
                        case R.id.camera:
                            tomarFoto();
                            break;
                        case R.id.galeria:
                            elegirImagen();
                            break;
```

```
                        case R.id.borrar:
                            contacto.setImagen(null);
                            break;
                    }
                    return true;
                }
            });
            pop.show();
        }
    });

    radioButtonAmigos = v.findViewById(R.id.radio_amigo);
    radioButtonFamilia = v.findViewById(R.id.radio_familia);
    radioButtonTrabajo = v.findViewById(R.id.radio_trabajo);
    editNombre = v.findViewById(R.id.editNombre);
    editApellido = v.findViewById(R.id.editApellido);
    editTelefono = v.findViewById(R.id.editTelefono);
    editCorreo = v.findViewById(R.id.editCorreo);
    aceptarButton = v.findViewById(R.id.aceptar);
    aceptarButton.setOnClickListener(this);

    if (proposito == EDITAR) {
        if (contacto.getImagen() != null){
            imageView.setImageBitmap(contacto.getImagen());
        }
        editNombre.setText(contacto.getNombre());
        editApellido.setText(contacto.getApellido());
        editTelefono.setText(contacto.getTelefono());
        editCorreo.setText(contacto.getCorreo());
        if (contacto.isTrabajo()) radioButtonTrabajo.setChecked(true);
        else if (contacto.isAmigo()) radioButtonAmigos.setChecked(true);
        else if (contacto.isFamilia()) radioButtonFamilia.setChecked(true);
    }

    return v;
}
```

El resto de métodos son autodescriptivos y son similares a la agenda anterior

Acción contacto XML

Una serie de `EditText` y un botón que nos permitirá editar o crear un nuevo contacto, además un conjunto de `radioButton` que nos permitirá clasificar los contactos

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp"
    android:orientation="vertical">
```

```
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1.5">

        <ImageView
            android:id="@+id/profile_image_view"
            android:layout_width="125dp"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_alignParentBottom="true"
            android:src="@drawable/ic_default" />

        <EditText
            android:id="@+id/editNombre"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="4dp"
            android:layout_centerInParent="true"
            android:layout_toEndOf="@id/profile_image_view"
            android:hint="Nombre"
            android:inputType="textPersonName" />

        <EditText
            android:layout_below="@id/editNombre"
            android:id="@+id/editApellido"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_margin="4dp"
            android:layout_toEndOf="@id/profile_image_view"
            android:hint="Apellido"
            android:inputType="textPersonName" />

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center_vertical">

        <ImageView
            android:id="@+id/imagenTelefono"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignTop="@id/editTelefono"
            android:layout_alignBottom="@id/editTelefono"
            android:layout_alignParentStart="true"
            android:padding="10dp"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_telefono" />
```

```
        <EditText
            android:id="@+id/editTelefono"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="8dp"
            android:layout_toEndOf="@id/imagenTelefono"
            android:hint="Teléfono"
            android:inputType="phone" />
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center_vertical">

        <ImageView
            android:id="@+id/imagenCorreo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignTop="@id/editCorreo"
            android:layout_alignBottom="@id/editCorreo"
            android:layout_alignParentStart="true"
            android:padding="10dp"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_mail" />

        <EditText
            android:id="@+id/editCorreo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="8dp"
            android:layout_toEndOf="@id/imagenCorreo"
            android:hint="Correo"
            android:inputType="textEmailAddress" />
    </RelativeLayout>

    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_weight="1"
        android:gravity="center_vertical">
        <RadioButton
            android:id="@+id/radio_familia"
            android:text="Famillia"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>
        <RadioButton
            android:id="@+id/radio_trabajo"
```

```
            android:text="Trabajo"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>
        <RadioButton
            android:id="@+id/radio_amigo"
            android:text="Amigos"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>
    </RadioGroup>

    <Button
        android:id="@+id/aceptar"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="16dp"
        android:layout_weight="0.5"
        android:text="Aceptar" />

</LinearLayout>
```

Vista contactos

El fragment que contiene el recycler

Variables de clase

```
private final int COD_ELEGIR_IMAGEN = 1;
private final int COD_TOMAR_FOTO = 2;
public Adaptador adaptador;
public RecyclerView recyclerView;
private OnClickItemListener clickItemListener;
private OnFabClicked fabClickListener;
private ArrayList<Contacto> contactos;
private SwipeDetector swipeDetector;
private int indiceListaPulsado;
private FloatingActionButton fab;
private Layout layout;
```

`onCreate`

1. Hacemos un inflate de la vista del `recycler`
2. Le asignamos el `onClickListener` al `FAB`
3. Finalmente, actualizamos el recycler con el adaptador

```
@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
        @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
    super.onCreateView(inflater, container, savedInstanceState);
    View rootView = inflater.inflate(R.layout.vista_contactos, container, false);

    fab = rootView.findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            fabClickListener.onFabClicked();
        }
    });

    swipeDetector = new SwipeDetector();
    recyclerView = rootView.findViewById(R.id.recycler);
    updateRecycler();
    return rootView;
}
```

`updateRecycler`

Le asignamos al recycler el adaptador y los listener correspondientes, el `LayoutManager` será el seleccionado mediante el menú de la `Toolbar`

```
private void updateRecycler() {
    adaptador = new Adaptador(contactos, layout);
    adaptador.setOnTouchListener(swipeDetector);
    adaptador.setOnClickListener(this);
    adaptador.setOnLongClickListener(new View.OnLongClickListener() {
        @Override
        public boolean onLongClick(View v) {
            eliminarContacto(v);
            return false;
        }
    });
    adaptador.setImageClickListener(new OnImageClickListener() {
        @Override
        public void onImageClick(final Contacto contacto, View v) {
            mostrarPopupMenu(contacto, v);
        }
    });
    if (layout == Layout.GRID) {
        recyclerView.setLayoutManager(new GridLayoutManager(getContext(), 3));
    } else {
        recyclerView.setLayoutManager(new LinearLayoutManager(getContext(),
RecyclerView.VERTICAL, false));
    }
```

```
        recyclerView.setAdapter(adaptador);
    }
```

`onRecyclerUpdated`

Se utiliza cada vez que se selecciona una opción de layout

```
@Override
public void onRecyclerUpdated(Layout layout) {
    this.layout = layout;
    updateRecycler();
}
```

`onAttach`

Para la comunicación de datos entre Activity y Fragment

```
@Override
public void onAttach(@NonNull Context context) {
    super.onAttach(context);
    try {
        clickItemListener = (OnClickItemListener) context;
        fabClickListener = (OnFabClicked) context;
    } catch (ClassCastException e) {
    }
}
```

Vista contactos XML

Contiene el `RecyclerView` y el `FAB`

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/fab_margin"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        app:srcCompat="@drawable/ic_plus" />

</RelativeLayout>
```