

Documentación

Proyecto agenda

Iván Gallego

índice de contenidos

1. [Actividad principal](#)
2. [Actividad principal XML](#)
3. [Adaptador](#)
4. [Holder](#)
5. [Interfaz de comunicación de datos](#)
6. [Contacto XML](#)
7. [Acción contacto](#)
8. [Acción contacto XML](#)
9. [Perfil XML](#)
10. [Dato \(POJO\)](#)

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 View.OnClickListener, View.OnLongClickListener {}
```

Variables de clase

`datoTemp` lo utilizaremos para cuando le demos click a la imagen del perfil, ya que si le damos click a la imagen, no se actualiza la variable `indiceListaPulsado`

```
public ArrayList<Dato> datos;
private int indiceListaPulsado;
private RecyclerView recyclerView;
private Adaptador adaptador;
private SwipeDetector swipeDetector;
private Dato datoTemp;
private final int COD_ACTIVITY_EDITAR = 1;
private final int COD_ACTIVITY_ADD = 2;
private final int COD_ELEGIR_IMAGEN = 3;
private final int COD_TOMAR_FOTO = 4;
```

onCreate

1. Indicaremos cuál es el activity que cargaremos.
2. Inicializaremos la lista de datos
3. Cargaremos los datos desde el archivo `datos.xml`
4. Inicializaremos el adaptador pasándole los datos, el `SwipeDetector` que nos permitirá detectar el desplazamiento y el recycler.
5. Le asignaremos los listener al adaptador para detectar el `click`, el `swipe`, el `longClick` y el `imageClick` con un método propio
6. Inicializaremos el `recycler`
7. Finalmente inicializamos el `FAB` y le asignamos el evento que nos permitirá añadir un contacto

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    datos = new ArrayList<>();
    cargarDatos();
    swipeDetector = new SwipeDetector();
    adaptador = new Adaptador(datos);
    adaptador.setClickListener(this);
    adaptador.setOnLongClickListener(this);
    adaptador.setOnTouchListener(swipeDetector);
    adaptador.setImageClickListener(new OnImageClickListener() {
        @Override
        public void onImageClick(final Dato dato) {
            mostrarPerfil(dato);
        }
    });
    recyclerView = findViewById(R.id.recycler);
    recyclerView.setAdapter(adaptador);
    recyclerView.setLayoutManager(new LinearLayoutManager(this,
        RecyclerView.VERTICAL, false));

    FloatingActionButton fab = findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            addContacto();
        }
    });
}
```

`mostrarPerfil`

En éste método lo único que hacemos es mostrar un diálogo enseñando la imagen de perfil y dos iconos, uno para elegir la imagen de la galería y otro para tomar una foto

```
private void mostrarPerfil(final Dato dato) {
    final AlertDialog.Builder builder = new
```

```

AlertDialog.Builder(MainActivity.this);
    LayoutInflater inflater = MainActivity.this.getLayoutInflater();
    View vista = inflater.inflate(R.layout.perfil_contacto, null);
    TextView nombrePerfil = vista.findViewById(R.id.nombrePerfil);
    ImageView imagenPerfil = vista.findViewById(R.id.imagenPerfil);
    imagenPerfil.setImageBitmap(dato.getImagen());
    nombrePerfil.setText(dato.getNombre());
    builder.setView(vista);
    final AlertDialog dialog = builder.create();
    ImageView imagenCamara = vista.findViewById(R.id.botonCamara);
    imagenCamara.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tomarFoto(dato, dialog);
        }
    });
    ImageView imagenGaleria = vista.findViewById(R.id.botonGaleria);
    imagenGaleria.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            elegirImagen(dato, dialog);
        }
    });
    dialog.show();
}

```

onActivityResult

Recogemos los datos de los intents que hemos hecho

```

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == COD_ACTIVITY_EDITAR &&
        resultCode == RESULT_OK && data != null) {
        Dato d = data.getParcelableExtra("contacto");
        datos.set(indiceListaPulsado, d);
    } else if (requestCode == COD_ACTIVITY_ADD &&
        resultCode == RESULT_OK && data != null) {
        Dato d = data.getParcelableExtra("contacto");
        datos.add(d);
    } else if (requestCode == COD_ELEGIR_IMAGEN && resultCode == RESULT_OK && data != null) {
        Uri rutaImagen = data.getData();
        datoTemp.setImagen(bitmapFromUri(rutaImagen));
    } else if (requestCode == COD_TOMAR_FOTO && resultCode == RESULT_OK && data != null) {
        datoTemp.setImagen((Bitmap) data.getExtras().get("data"));
    } else if (resultCode == RESULT_CANCELED) {
        Toast.makeText(this, "Se ha cancelado la operación",
            Toast.LENGTH_LONG).show();
    }
}

```

```

    }
    recyclerView.setAdapter(adaptador);
    if (resultCode != RESULT_CANCELED) {
        guardarDatos();
    }
}

```

tomarFoto()

Se llama a un Intent implícito que nos devolverá la foto tomada que luego capturaremos en el método `onActivityResult`

```

private void tomarFoto(Dato dato, AlertDialog dialog) {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, COD_TOMAR_FOTO);
        datoTemp = dato;
    }
    dialog.cancel();
}

```

elegirImagen

Se llama a un Intent implícito que nos devolverá la imagen elegida

```

private void elegirImagen(Dato dato, AlertDialog dialog) {
    Intent intent = new Intent(Intent.ACTION_OPEN_DOCUMENT);
    intent.setType("image/*");
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, COD_ELEGIR_IMAGEN);
        datoTemp = dato;
    }
    dialog.cancel();
}

```

onClick

Se implementa la gestión del `click` y del `swipe`

```

@Override
public void onClick(View v) {
    indiceListaPulsado = recyclerView.getChildAdapterPosition(v);
    Dato d = datos.get(recyclerView.getChildAdapterPosition(v));
    if (swipeDetector.swipeDetected()) {
        switch (swipeDetector.getAction()) {
            case LR:

```

```

        llamarContacto(d);
        break;
    case RL:
        enviarMensaje(d);
        break;
    }
} else {
    editarDatos(d);
}
}
}

```

llamarContacto

Utilizo un Intent implícito que abrirá la aplicación de teléfono con el teléfono del contacto preparado para llamar

```

private void llamarContacto(final Dato d) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("¿Llamar a " + d.getNombre() + "?");
    builder.setPositiveButton("LLAMAR", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            if (d.getTelefono().isEmpty()) {
                Toast.makeText(MainActivity.this, "El contacto no tiene teléfono",
                    Toast.LENGTH_LONG).show();
            } else {
                Intent intent = new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:" + d.getTelefono()));
                if (intent.resolveActivity(getPackageManager()) != null) {
                    startActivity(intent);
                }
            }
        }
    });
    builder.setNegativeButton("CANCELAR", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    builder.create().show();
}

```

enviarMensaje

Utilizo un Intent implícito que nos abrirá la aplicación del correo preparado para enviar un mensaje a un destinatario

```

private void enviarMensaje(final Dato d) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("¿Enviar mensaje a " + d.getNombre() + "?");
    builder.setPositiveButton("ENVIAR", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            if (d.getCorreo().isEmpty()) {
                Toast.makeText(MainActivity.this, "El contacto no tiene correo electrónico",
                    Toast.LENGTH_LONG).show();
            } else {
                Intent intent = new Intent(Intent.ACTION_SENDTO);
                intent.setData(Uri.fromParts("mailto", d.getCorreo(), null));
                Intent chooser = Intent.createChooser(intent, "Enviar mensaje...");
                startActivity(chooser);
            }
        }
    });
    builder.setNegativeButton("CANCELAR", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    builder.create().show();
}

```

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);
}

```

Métodos de utilidad

bitmapFromUri

Este método lo utilizo porque, al llamar al Intent de elegir imagen, me devuelve un Uri y no un Bitmap y utilizando este método, lo convierto a Bitmap.

```

private Bitmap bitmapFromUri(Uri uri) {
    ImageView imageViewTemp = new ImageView(this);
    imageViewTemp.setImageURI(uri);
}

```

```

        BitmapDrawable d = (BitmapDrawable) imageViewTemp.getDrawable();
        return d.getBitmap();
    }

```

convertirStringBitmap

```

public Bitmap convertirStringBitmap(String encodedString) {
    try {
        byte[] encodeByte = Base64.decode(encodedString, Base64.DEFAULT);
        Bitmap bitmap = BitmapFactory.decodeByteArray(encodeByte, 0,
encodeByte.length);
        return bitmap;
    } catch (Exception e) {
        e.getMessage();
        return null;
    }
}

```

convertirImagenString

```

public String convertirImagenString(Bitmap bitmap) {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.PNG, 100, baos);
    byte[] b = baos.toByteArray();
    String temp = Base64.encodeToString(b, Base64.DEFAULT);
    return temp;
}

```

Main activity XML

En el layout principal, tenemos un **Recycler** y **FAB**

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

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

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="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"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:srcCompat="@drawable/ic_plus" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Adaptador

En el adaptador implementamos los métodos necesarios extender de `RecyclerView.Adapter` y creamos los nuestros para gestionar los `click`, `swipe`...

```

public class Adaptador extends RecyclerView.Adapter
    implements View.OnClickListener, View.OnLongClickListener,
    View.OnTouchListener {
    private ArrayList<Dato> datos;
    private View.OnClickListener clickListener;
    private OnImageClickListener imageClickListener;
    private View.OnLongClickListener longClickListener;
    private View.OnTouchListener touchListener;

    public Adaptador(ArrayList<Dato> datos) {
        this.datos = datos;
    }

    @NonNull
    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
    int viewType) {
        View 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(Dato dato) {
                imageClickListener.onImageClick(dato);
            }
        });
        return h;
    }
}

```



```
}

@Override
public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int
position) {
    ((Holder) holder).bind(datos.get(position));
}

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

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 {
    ImageView imagen;
    TextView nombre, apellido, telefono, correo;
    OnImageClickListener imageClickListener;
    Dato dato;

    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(Dato d) {
        nombre.setText(d.getNombre());
        apellido.setText(d.getApellido());
        telefono.setText(d.getTelefono());
        correo.setText(d.getCorreo());
        imagen.setImageBitmap(d.getImagen());
        dato = d;
    }

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

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

```

Interfaz de comunicación de datos

```
public interface OnImageClickListener {  
    void onImageClick(Dato dato);  
}
```

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"/>  
  
        <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>

```

Accion contacto

Este Java - XML lo utilizo tanto para editar un contacto, como para añadir uno nuevo

```

public class AccionContacto extends AppCompatActivity
    implements View.OnFocusChangeListener, View.OnClickListener {}

```

onCreate

1. Recibimos el dato del intent
 1. Si el dato es nulo, el propósito es para añadir uno nuevo
 2. Si no es nulo, el propósito es editar
2. Utilizamos el `resetTint` para reiniciar el color de los iconos en caso de que se quedaran de otro color al cancelar una edición o adición previa
3. Inicializamos los elementos del XML
4. Inicializamos el texto de los `EditText`, solo si el propósito es editar

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.accion_contacto);
    Intent intentRecibido = getIntent();
    datoRecibido = intentRecibido.getParcelableExtra("contacto");
    resetTint();
    proposito = datoRecibido == null ? ADD : EDITAR;

    editNombre = findViewById(R.id.editNombre);
    editApellido = findViewById(R.id.editApellido);
    editTelefono = findViewById(R.id.editTelefono);
    editCorreo = findViewById(R.id.editCorreo);
}

```

```

editNombreContenedor = findViewById(R.id.editNombreContenedor);
aceptarButton = findViewById(R.id.aceptar);
aceptarButton.setOnClickListener(this);
editNombre.setOnFocusChangeListener(this);
editApellido.setOnFocusChangeListener(this);
editTelefono.setOnFocusChangeListener(this);
editCorreo.setOnFocusChangeListener(this);

if (proposito == EDITAR) {
    editNombre.setText(datoRecibido.getNombre());
    editApellido.setText(datoRecibido.getApellido());
    editTelefono.setText(datoRecibido.getTelefono());
    editCorreo.setText(datoRecibido.getCorreo());
}
}

```

onFocusChange

Este método es para cambiar el color del icono al hacer **focus**

```

public void onFocusChange(View v, boolean hasFocus) {
    Drawable d = null;
    ImageView icono = null;
    switch (v.getId()) {
        case R.id.editNombre:
            icono = findViewById(R.id.imagenNombre);
            d = icono.getDrawable();
            break;
        case R.id.editApellido:
            icono = findViewById(R.id.imagenApellido);
            d = icono.getDrawable();
            break;
        case R.id.editTelefono:
            icono = findViewById(R.id.imagenTelefono);
            d = icono.getDrawable();
            break;
        case R.id.editCorreo:
            icono = findViewById(R.id.imagenCorreo);
            d = icono.getDrawable();
            break;
    }
    if (d != null) {
        DrawableCompat.wrap(d);
        if (hasFocus) {
            DrawableCompat.setTint(d,
ContextCompat.getColor(getApplicationContext(), R.color.colorAccent));
        } else {
            DrawableCompat.setTint(d,
ContextCompat.getColor(getApplicationContext(), R.color.colorBaseIconos));
        }
    }
}

```

```
    }
}
```

generarNuevoContacto

Utilizamos los datos de los `EditText` para crear uno nuevo

```
public Dato generarNuevoContacto() {
    Dato d = new Dato();
    d.setNombre(editNombre.getText().toString());
    d.setApellido(editApellido.getText().toString());
    d.setTelefono(editTelefono.getText().toString());
    d.setCorreo(editCorreo.getText().toString());
    return d;
}
```

onClick

Gestión del `onClick` que nos permitira enviar un dato nuevo al activity principal o devolver un dato editado

```
public void onClick(View v) {
    if (nombreCorrecto(editNombre.getText().toString())) {
        Intent intent = new Intent();
        if (proposito == EDITAR) {
            editarContacto();
            intent.putExtra("contacto", datoRecibido);
        } else {
            Dato nuevoDato = generarNuevoContacto();
            intent.putExtra("contacto", nuevoDato);
        }
        setResult(RESULT_OK, intent);
        resetTint();
        finish();
    }
}
```

nombreCorrecto

Comprobar que el nombre no este vacío para no devolver un dato vacío

```
private boolean nombreCorrecto(String nombre) {
    if (nombre.isEmpty()) {
        editNombreContenedor.setError("El nombre no debe estar vacío.");
        return false;
    }
}
```

```
        return true;
    }
}
```

resetTint

Volver a poner los icono del color original

```
private void resetTint() {
    ImageView iconoNombre = findViewById(R.id.imagenNombre);
    ImageView iconoApellido = findViewById(R.id.imagenApellido);
    ImageView iconoTelefono = findViewById(R.id.imagenTelefono);
    ImageView iconoCorreo = findViewById(R.id.imagenCorreo);
    ArrayList<Drawable> drawables = new ArrayList<>();
    drawables.add(iconoNombre.getDrawable());
    drawables.add(iconoApellido.getDrawable());
    drawables.add(iconoTelefono.getDrawable());
    drawables.add(iconoCorreo.getDrawable());
    for (Drawable d : drawables) {
        DrawableCompat.wrap(d);
        DrawableCompat.setTint(d, ContextCompat.getColor(getApplicationContext(),
R.color.colorBaseIconos));
    }
}
```

editarContacto

Cambiar los datos del contacto con los de los `EditText`

```
public void editarContacto() {
    datoRecibido.setNombre(editNombre.getText().toString());
    datoRecibido.setApellido(editApellido.getText().toString());
    datoRecibido.setTelefono(editTelefono.getText().toString());
    datoRecibido.setCorreo(editCorreo.getText().toString());
}
```

Acción contacto XML

Una serie de `EditText` y un botón que nos permitirá editar o crear un nuevo contacto Realmente son una serie `TextInputLayout` y `TextInputEditText` para poder hacer la animación de movimiento del `Hint` del `EditText` hacia arriba y el cambio de color de los iconos al hacer `focus`

```
<?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_marginStart="16dp"
```

```
android:layout_marginEnd="16dp"
android:orientation="vertical">

<TextView
    style="@android:style/TextAppearance.Material.Large"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="24dp"
    android:text="Contacto"
    android:textAlignment="center" />

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

    <ImageView
        android:id="@+id/imagenNombre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@id/editNombreContenedor"
        android:layout_alignBottom="@id/editNombreContenedor"
        android:layout_alignParentStart="true"
        android:padding="10dp"
        android:scaleType="fitCenter"
        android:src="@drawable/ic_contacto" />

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/editNombreContenedor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="8dp"
        android:layout_toEndOf="@id/imagenNombre">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editNombre"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Nombre"
            android:inputType="textPersonName" />
    </com.google.android.material.textfield.TextInputLayout>
</RelativeLayout>

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

    <ImageView
        android:id="@+id/imagenApellido"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@id/editApellidoContenedor"
```



```
        android:layout_alignBottom="@id/editApellidoContenedor"
        android:layout_alignParentStart="true"
        android:padding="10dp"
        android:scaleType="centerCrop"
        android:src="@drawable/ic_apellidos"/>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/editApellidoContenedor"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="8dp"
    android:layout_toEndOf="@id/imagenApellido">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/editApellido"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Apellido"
        android:inputType="textPersonName" />
</com.google.android.material.textfield.TextInputLayout>
</RelativeLayout>

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

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

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/editTelefonoContenedor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="8dp"
        android:layout_toEndOf="@id/imagenTelefono">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editTelefono"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Teléfono"
            android:inputType="phone" />
    </com.google.android.material.textfield.TextInputLayout>
```

```

</RelativeLayout>

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

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

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/editCorreoContenedor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="8dp"
        android:layout_toEndOf="@id/imagenCorreo">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editCorreo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Correo"
            android:inputType="textEmailAddress" />
    </com.google.android.material.textfield.TextInputLayout>
</RelativeLayout>

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

</LinearLayout>

```

Perfil XML

El diálogo que sale cada vez que pulsas sobre una imagen de un contacto

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

    <ImageView
        android:id="@+id/botonCamara"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@id/imagenPerfil"
        android:layout_marginEnd="8dp"
        android:layout_toStartOf="@id/imagenPerfil"
        android:src="@drawable/ic_camera" />

    <ImageView
        android:id="@+id/imagenPerfil"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:layout_centerHorizontal="true"
        android:layout_margin="8dp"
        android:scaleType="centerCrop" />

    <ImageView
        android:id="@+id/botonGaleria"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@id/imagenPerfil"
        android:layout_marginStart="8dp"
        android:layout_toEndOf="@id/imagenPerfil"
        android:src="@drawable/ic_galeria" />

    <TextView
        android:id="@+id/nombrePerfil"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/imagenPerfil"
        android:layout_centerHorizontal="true"
        android:textAlignment="center" />

</RelativeLayout>
```

Dato (POJO)

```
public class Dato implements Parcelable {
    private String nombre;
    private String apellido;
    private String telefono;
    private String correo;
    private Bitmap imagen;
```

```
public Dato() {
    imagen = Bitmap.createBitmap(30,30, Bitmap.Config.RGB_565);
}

public Dato(String nombre, String apellido, String telefono, String correo,
Bitmap imagen) {
    this.nombre = nombre;
    this.apellido = apellido;
    this.telefono = telefono;
    this.correo = correo;
    this.imagen = imagen;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getApellido() {
    return apellido;
}

public void setApellido(String apellido) {
    this.apellido = apellido;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public String getCorreo() {
    return correo;
}

public void setCorreo(String correo) {
    this.correo = correo;
}

public Bitmap getImagen() {
    return imagen;
}

public void setImagen(Bitmap imagen) {
    this.imagen = imagen;
}

@Override
```

```
public int describeContents() {
    return 0;
}

@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(this.nombre);
    dest.writeString(this.apellido);
    dest.writeString(this.telefono);
    dest.writeString(this.correo);
    dest.writeParcelable(this.imagen, flags);
}

protected Dato(Parcel in) {
    this.nombre = in.readString();
    this.apellido = in.readString();
    this.telefono = in.readString();
    this.correo = in.readString();
    this.imagen = in.readParcelable(Bitmap.class.getClassLoader());
}

public static final Parcelable.Creator<Dato> CREATOR = new
Parcelable.Creator<Dato>() {
    @Override
    public Dato createFromParcel(Parcel source) {
        return new Dato(source);
    }

    @Override
    public Dato[] newArray(int size) {
        return new Dato[size];
    }
};
}
```