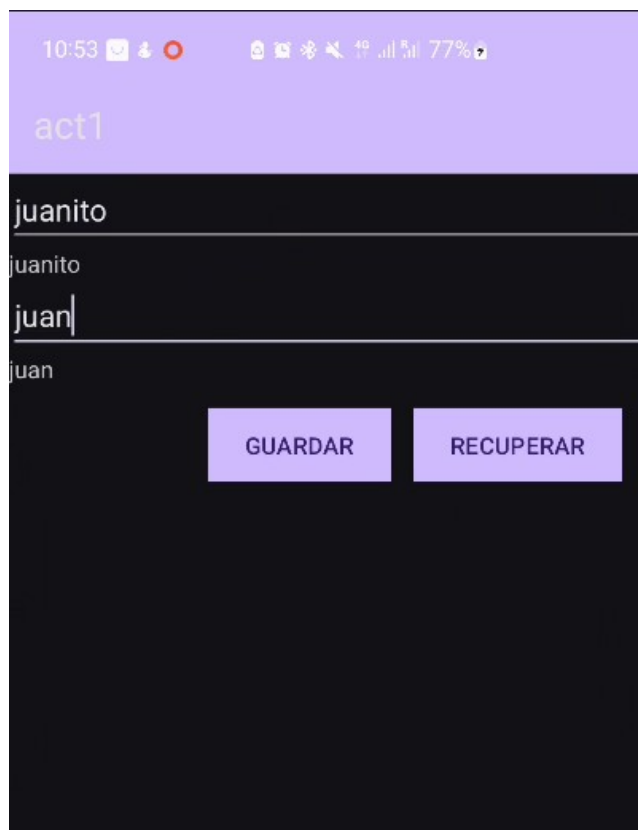
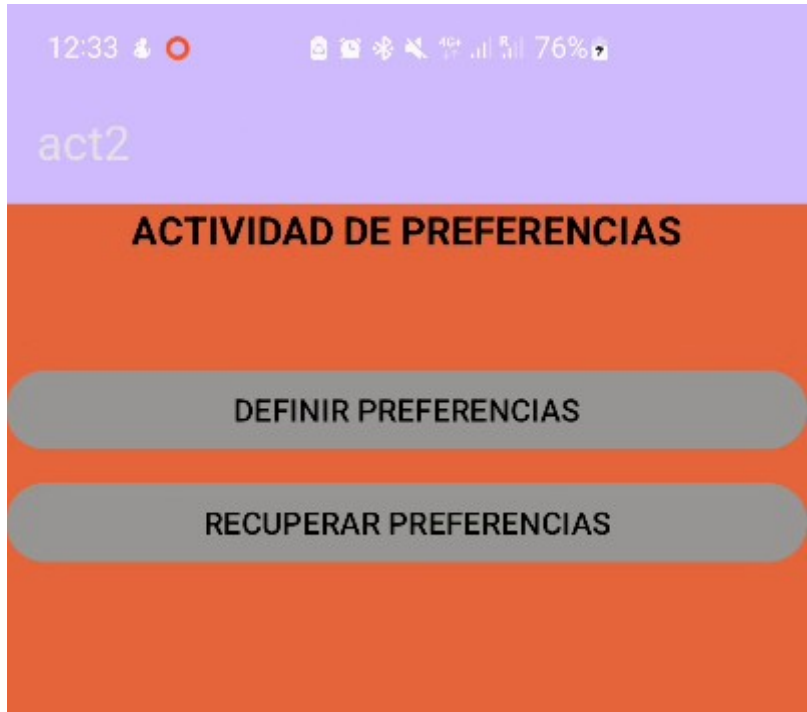


Act1



Act2



12:34

76%

Categoría 1

ÚNICO SISTEMA OPERATIVO

Marca si tienes preferencia por un solo SO



Categoría 2

Preferencia 3

Descripcion de la preferencia

SISTEMA OPERATIVO

Pulsa para elegir Sistema Operativo

SISTEMA OPERATIVO

Pulsa para elegir Sistema Operativo

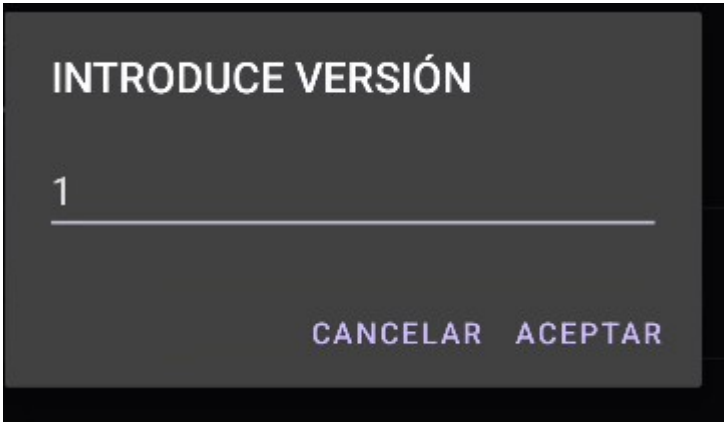
Indicar país

☒ WINDOWS

☐ LINUX

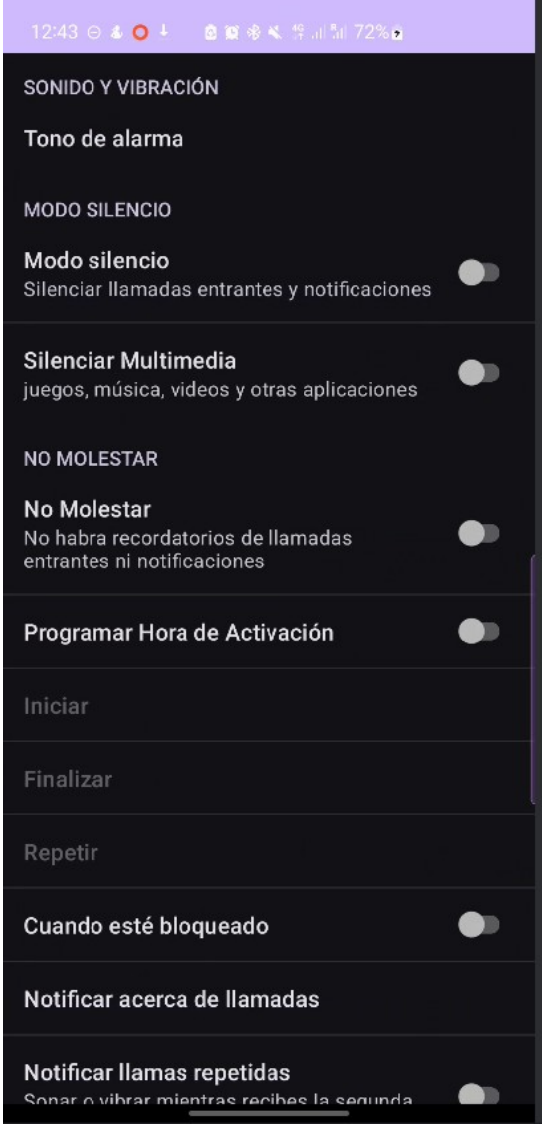
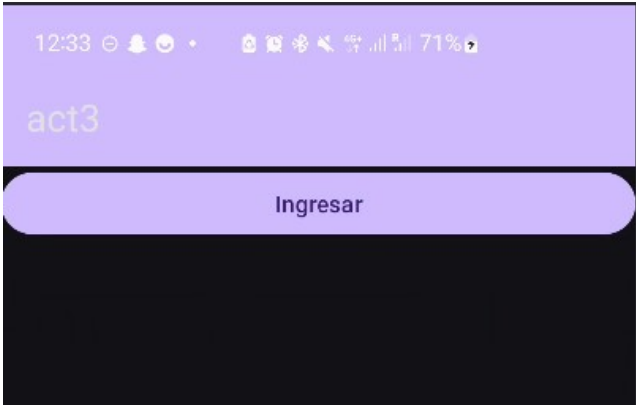
☐ IOS

CANCELAR



| | | | | | |
|-------------------------|-------------|-------------|------------------|---|----------------|
| 2025-02-06 12:31:58.790 | 14649-14649 | Preferences | com.example.act2 | I | S0 Único: true |
| 2025-02-06 12:31:58.790 | 14649-14649 | Preferences | com.example.act2 | I | S0: WIN |
| 2025-02-06 12:31:58.790 | 14649-14649 | Preferences | com.example.act2 | I | Versión: 1 |

Act3



Programar Hora de Activación ☐

Iniciar

Finalizar

Repetir

Programar Hora de Activación ☒

Iniciar

Finalizar

Repetir

Introduce la Hora de inicio

CANCELAR ACEPTAR

Repetir

☐ Lunes-Viernes

☐ Sabado-Domingo

CANCELAR

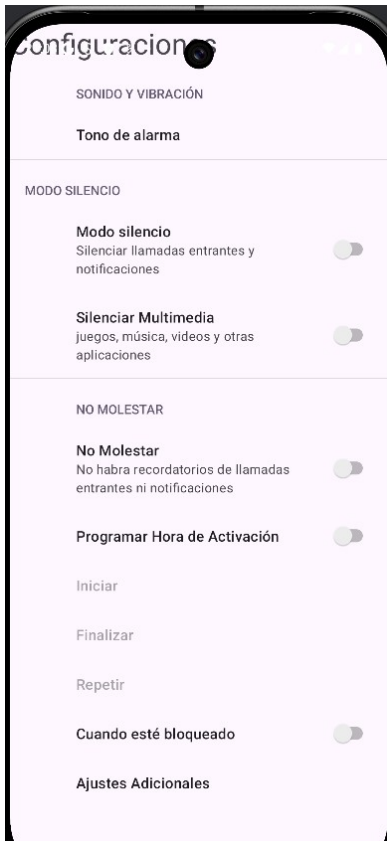
12:47 73%

Tonos de teclado al presionar ☐

sonidos de Toques ☐

Reproducir sonido al tomar capturas d.. ☐

Act 4)



```
public class OpcionesActivity extends PreferenceFragmentCompat{

    @Override // 1 usage
    public void onCreatePreferences(@Nullable Bundle savedInstanceState, @Nullable String rootKey) {

        setPreferencesFromResource(R.xml.opciones, rootKey);
    }

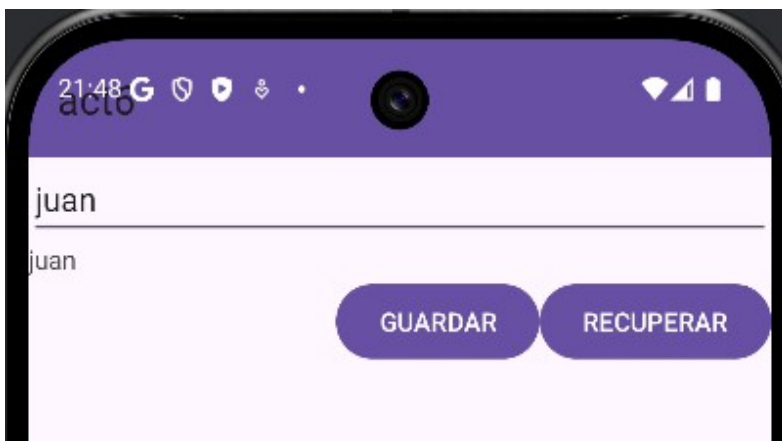
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

Act5)

Se debe de poner el atributo dentro de la etiqueta

```
<PreferenceCategory android:title="MODO SILENCIO"
    app:iconSpaceReserved="false">
```

Act6)



```

61         btnGuardar.setOnClickListener(new View.OnClickListener() {
62             @Override
63             public void onClick(View v) {
64
65                 try{
66
67                     OutputStreamWriter miFichero = new OutputStreamWriter(openFileOutput("fichero.txt", MODE_PRIVATE));
68                     miFichero.write(edittext.getText().toString());
69                     miFichero.close();
70                 }catch(Exception e){
71
72                     Log.e("Fichero", msg: "Error al escribir en la memoria interna");
73                 }
74             }
75         });
76
77         btnRecuperar.setOnClickListener(new View.OnClickListener() {
78             @Override
79             public void onClick(View v) {
80
81                 try{
82                     BufferedReader miFichero = new BufferedReader(new InputStreamReader(openFileInput("fichero.txt")));
83                     txt.setText(miFichero.readLine());
84                     miFichero.close();
85                 }catch(Exception e){
86
87                     Log.e("Fichero", msg: "Error al leer de la memoria interna");
88                 }
89             }
90         });

```

Act7)



```

btnResuperar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        try{

            File ruta_sd = getExternalFilesDir("prueba_sd.txt");

            File f = new File(ruta_sd.getAbsolutePath(), "prueba_sd.txt");
            BufferedReader in = new BufferedReader(new InputStreamReader(new FileInputStream(f)));

            String linea = in.readLine();
            txt.setText(linea);
            in.close();

        }catch(Exception e){

            Log.e("Ficheros", msg: "Error al leer desde la tarjeta SD");
        }
    }
});

```

```

btnGuardar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        boolean sdDisponible = false;
        boolean sdAccesoEscrtura = false;
        String estado = Environment.getExternalStorageState();

        if(estado.equals(Environment.MEDIA_MOUNTED)){

            sdDisponible = true;
            sdAccesoEscrtura =true;

        }else if(estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)){

            sdDisponible = true;
        }

        if(sdDisponible && sdAccesoEscrtura){

            try{

                File ruta_sd = getExternalFilesDir( type: null);

                File f = new File(ruta_sd.getAbsolutePath(), child: "prueba_sd.txt");

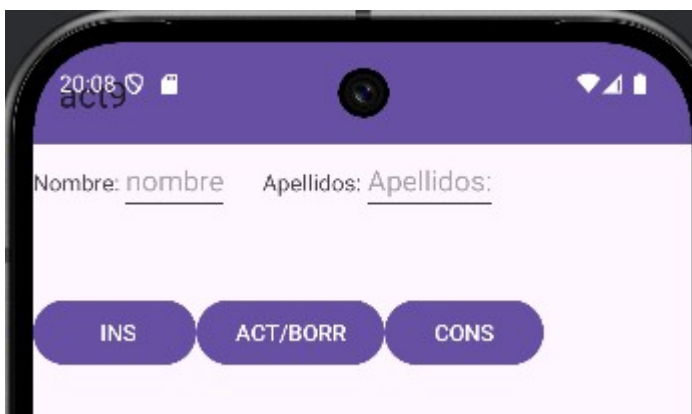
                OutputStreamWriter fout = new OutputStreamWriter(new FileOutputStream(f));

                fout.write(editext.getText().toString());
                fout.close();
            }catch (Exception e){

                Log.e( tag: "Ficheros", msg: "Error al escribir en la tarjeta SD");
            }
        }
    }
});

```

Act9)



```

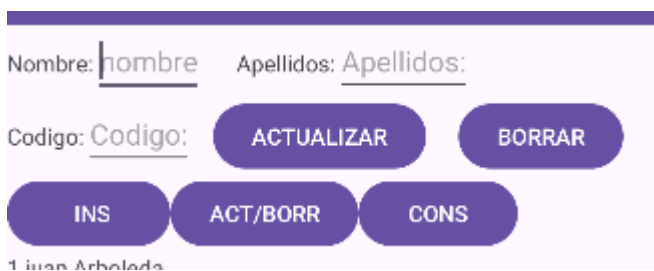
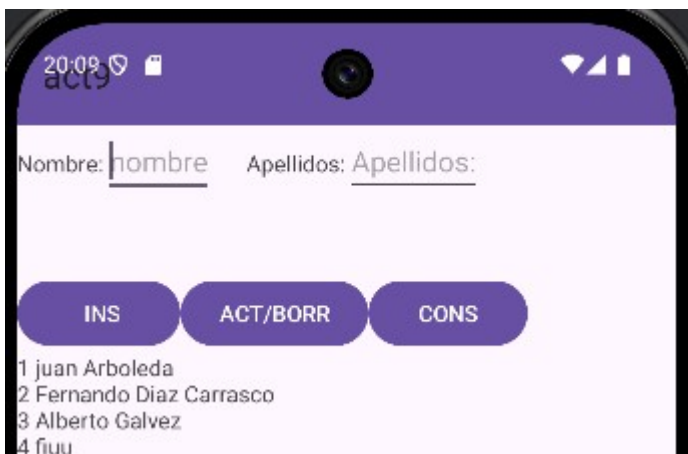
insertar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        contador++;
        String nom = nombre.getText().toString();
        String ap = apellido.getText().toString();

        ContentValues content = new ContentValues();
        content.put("codigo", contador);
        content.put("nombre", nom);
        content.put("apellidos", ap);
        db.insert(table: "Registro", nullColumnHack: null, content);

        nombre.setText("");
        apellido.setText("");
    }
}

```



```

act.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        ly.setVisibility(View.VISIBLE);
    }
});

```


Nombre: JUU Apellidos: SASA

Codigo: 4 **ACTUALIZAR** **BORRAR**

INS **ACT/BORR** **CONS**

1 juan Arboleda
 2 Fernando Diaz Carrasco
 3 Alberto Galvez
 4 JUJ SASA

```
actualizar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        ContentValues c = new ContentValues();
        c.put("nombre", nombre.getText().toString());
        c.put("apellidos", apellido.getText().toString());
        db.update( table: "Registro", c, whereClause: "codigo="+codigo.getText().toString(), whereArgs: null);
    }
});
```

Nombre: nombre Apellidos: Apellidos:

Codigo: Codigo: **ACTUALIZAR** **BORRAR**

INS **ACT/BORR** **CONS**

1 juan Arboleda
 2 Fernando Diaz Carrasco
 3 Alberto Galvez

```
borrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        db.delete( table: "Registro", whereClause: "codigo=" + codigo.getText().toString(), whereArgs: null);
        Toast.makeText( context: MainActivity.this, text: "Se elimino el usuario" + nombre.getText().toString(), Toast.LENGTH_SHORT).show();
        codigo.setText("");
    }
});
```

