Logotipo

Descripción generada automáticamente con confianza mediaImagen que contiene Icono

Descripción generada automáticamente

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Desarrollo de Aplicaciones Móviles Nativas

**“Lista de frases”**

Alumno:

Malagón Baeza Alan Adrian

Profesor:

M. en C. José Asunción Enríquez Zárate

Grupo: 7CM1

**Introducción**

En el marco de mi práctica, he trabajado en el desarrollo de una aplicación en Android que permite almacenar y mostrar frases junto con sus respectivos autores. He implementado un diseño de interfaz de usuario y he utilizado una base de datos SQLite para el almacenamiento de datos. A lo largo de este informe, presentaré el código relevante y explicaré cómo se implementaron las funcionalidades clave.

El código principal se divide en tres clases: Frase, FraseAdapter y MainActivity. La clase Frase es un modelo de datos que representa una frase con su autor. FraseAdapter es un adaptador personalizado que se utiliza para vincular los datos de frases con las vistas del RecyclerView. Por último, MainActivity es la clase principal que gestiona la interfaz de usuario y las interacciones del usuario.

En la actividad principal (MainActivity), se capturan las frases y sus autores ingresados por el usuario. Luego, se utiliza la clase SQLiteHelper para almacenar los datos en la base de datos SQLite. Además, se implementa la funcionalidad para mostrar las frases almacenadas en un RecyclerView a través del adaptador FraseAdapter.

**Desarrollo**

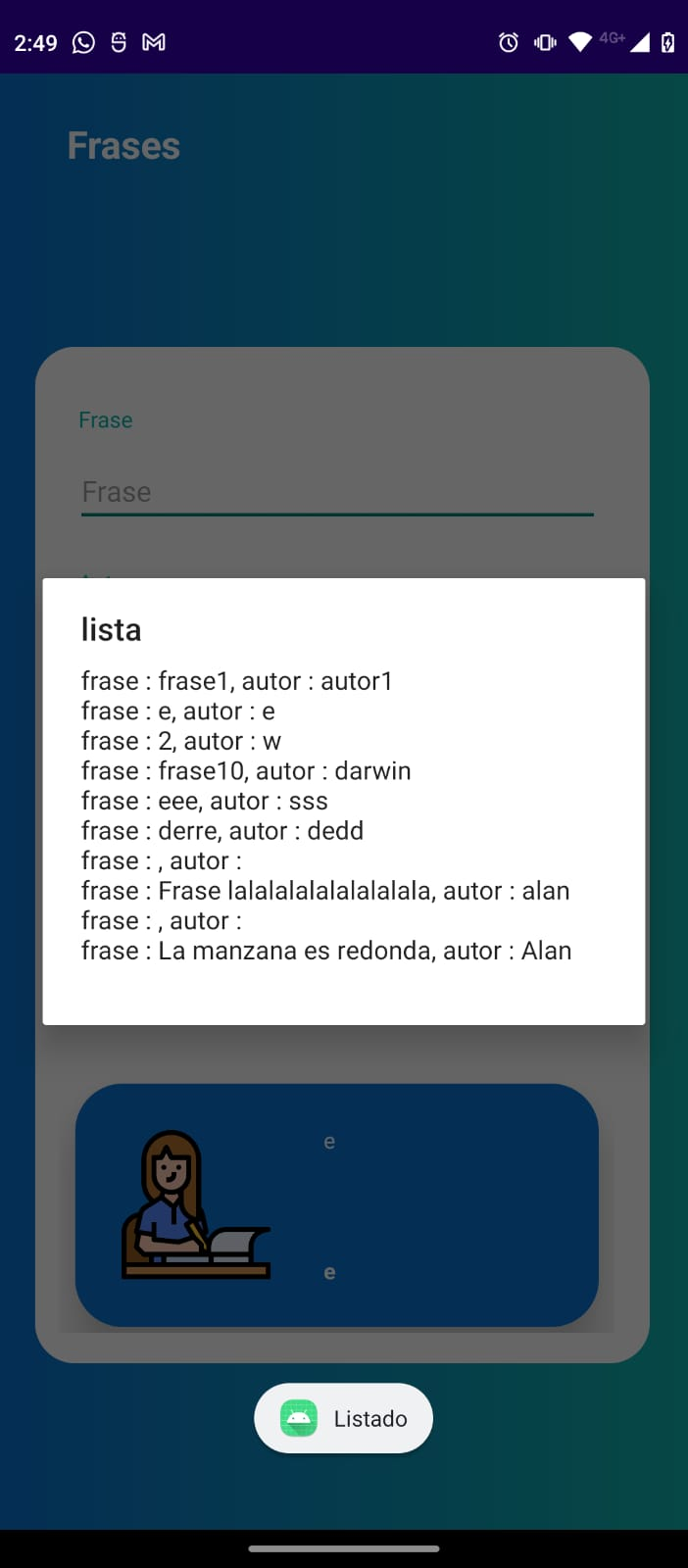
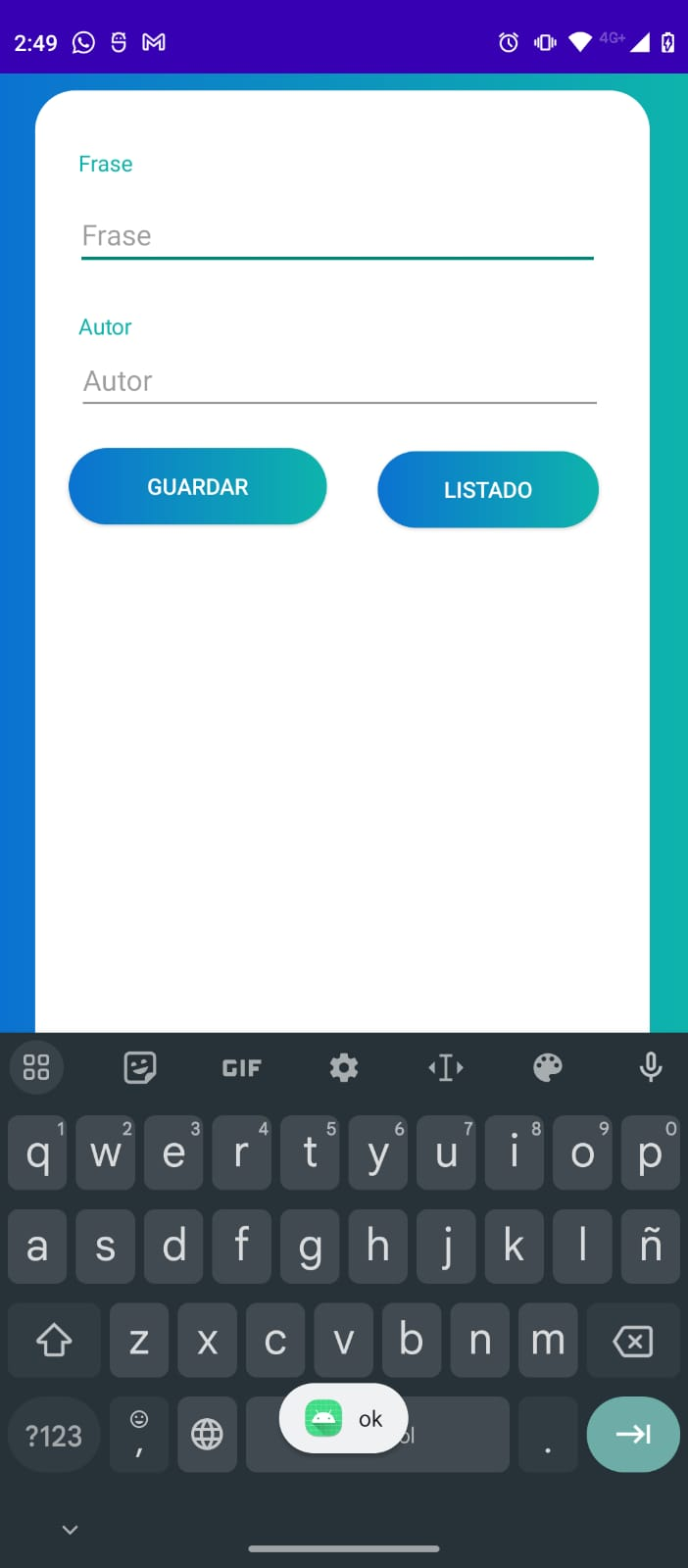
class Frase {  
 var id : Int = 0  
 var frase : String = ""  
 var autor: String=""  
  
 constructor(frase:String,autor : String){  
 this.frase = frase  
 this.autor = autor  
 }  
}

class FraseAdapter(private val datalist: List<Frase>) : RecyclerView.Adapter<FraseAdapter.ManejadorVista>() {  
  
  
 class ManejadorVista(elementoVista: View):  
 RecyclerView.ViewHolder(elementoVista){  
 var frase: TextView  
 var autor: TextView  
 init{  
 frase = elementoVista.findViewById(R.id.*frase*)  
 autor = elementoVista.findViewById(R.id.*autor*)  
 }  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ManejadorVista {  
 val vista = LayoutInflater.from(parent.*context*)  
 .inflate(R.layout.*card\_item\_layout*, parent, false)  
 return ManejadorVista(vista)  
 }  
  
 override fun getItemCount(): Int {  
 // crear un arreglo de elementos a mostrar  
 // return elArreglo size  
 return datalist.size  
 }  
  
 override fun onBindViewHolder(holder: ManejadorVista, position: Int) {  
 val frase = datalist[position]  
 holder.frase.*text* = frase.frase  
 holder.autor.*text* = frase.autor  
 }  
  
}

class MainActivity : AppCompatActivity() {  
  
 lateinit var frase: EditText  
 lateinit var autor: EditText  
 lateinit var btnGuardar: Button  
 lateinit var btnListado: Button  
 lateinit var db: SQLiteHelper  
 lateinit var texto: String  
 lateinit var tvListado: TextView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 frase = findViewById(R.id.*txtFrase*)  
 autor = findViewById(R.id.*txtAutor*)  
  
 tvListado = findViewById(R.id.*tvListado*)  
  
 btnGuardar = findViewById(R.id.*btnAceptar*)  
 btnListado = findViewById(R.id.*btnListado*)  
 texto = ""  
  
 db = SQLiteHelper(this)  
  
 btnGuardar.setOnClickListener() **{** val laFrase = frase.*text*.toString()  
 val elAutor = autor.*text*.toString()  
  
 val f = Frase(laFrase, elAutor)  
  
 val almacenado = db.create(f)  
  
 if (almacenado == true) {  
 Toast.makeText(this, "ok", Toast.*LENGTH\_SHORT*).show()  
 frase.*text*.clear()  
 autor.*text*.clear()  
 }  
 **}** btnListado.setOnClickListener()**{** Toast.makeText(this, "Listado", Toast.*LENGTH\_SHORT*).show()  
  
 val cursor = db.readAll()  
 val sb = StringBuilder()  
 if(cursor!=null){  
 if(cursor.moveToFirst()){  
 do {  
 sb.append("frase : ").append(cursor.getString(1))  
 sb.append(", autor : ").append(cursor.getString(2)).append("\n")  
 }while (cursor.moveToNext())  
 }  
 }  
 val alertDialog = AlertDialog.Builder(this)  
 alertDialog.setTitle("lista")  
 alertDialog.setMessage(sb.toString())  
 alertDialog.show()  
  
 texto=""  
 muestraTabla()  
 tvListado.*text* = texto  
  
 val recyclerView = findViewById<RecyclerView>(R.id.*recyclerView*)  
  
 val cursorA = db.readAll()  
 val adapter = FraseAdapter(getFrasesFromCursor(cursorA))  
  
 recyclerView.*layoutManager* = LinearLayoutManager(this)  
 recyclerView.*adapter* = adapter  
 **}** }  
  
 fun muestraTabla(){  
 val cursor = db.readAll()  
 val numeroFilas : Int = cursor.*count* cursor.moveToFirst()  
 for (i in 1 .. numeroFilas){  
 var id : Int = cursor.getInt(0)  
 var frase : String = cursor.getString(1)  
 var autor : String = cursor.getString(2)  
 texto = texto +"\n"+id+", "+frase+", "+autor+"\n"  
 cursor.moveToNext()  
 }  
 }  
  
 fun getFrasesFromCursor(cursor: Cursor): List<Frase> {  
 val frasesList = *mutableListOf*<Frase>()  
 while (cursor.moveToNext()) {  
 val frase = cursor.getString(1)  
 val autor = cursor.getString(2)  
 val frasea = Frase(frase, autor)  
 frasesList.add(frasea)  
 }  
 return frasesList  
 }  
}

class SQLiteHelper(context: Context) :  
 SQLiteOpenHelper(context,"FrasesChuck",null,1)  
{  
 override fun onCreate(db: SQLiteDatabase?) {  
 val sql="create table Frases(idFrase integer primary key autoincrement, " +  
 "frase text, autor text)"  
 db?.execSQL(sql)  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
 db!!.execSQL("drop table if exists Frases")  
 onCreate(db)  
 }  
  
 fun create(frase:Frase):Boolean{  
 val db = *writableDatabase* val contentValues = ContentValues()  
 contentValues.put("frase",frase.frase)  
 contentValues.put("autor",frase.autor)  
 val resultado :Long = db.insert("Frases",null,contentValues)  
 return resultado != -1.toLong()  
 }  
  
 fun readAll(): Cursor {  
 val db = *readableDatabase* val cursor = db.query("Frases", null, null, null, null, null, null)  
 cursor.moveToFirst()  
 return cursor  
 }  
}

**Resultados**

****

****

**Conclusiones**

En conclusión, el desarrollo de esta aplicación en Android para almacenar y mostrar frases junto con sus autores ha sido una experiencia enriquecedora durante mi práctica. He aprendido a utilizar SQLite para el almacenamiento de datos y a implementar un RecyclerView con un adaptador personalizado para mostrar los datos de manera eficiente.

El código presentado demuestra cómo utilizar una base de datos SQLite en Android, así como también cómo crear y utilizar un adaptador personalizado para mostrar los datos en un RecyclerView. Además, se ha implementado la funcionalidad para mostrar las frases almacenadas en un cuadro de diálogo y en un TextView en la actividad principal.

En resumen, esta práctica me ha permitido adquirir habilidades importantes en el desarrollo de aplicaciones móviles en Android, específicamente en el manejo de bases de datos SQLite y la implementación de RecyclerViews con adaptadores personalizados. Estoy satisfecho con los resultados obtenidos y considero que esta experiencia ha sido de gran valor para mi formación en el desarrollo de aplicaciones móviles.