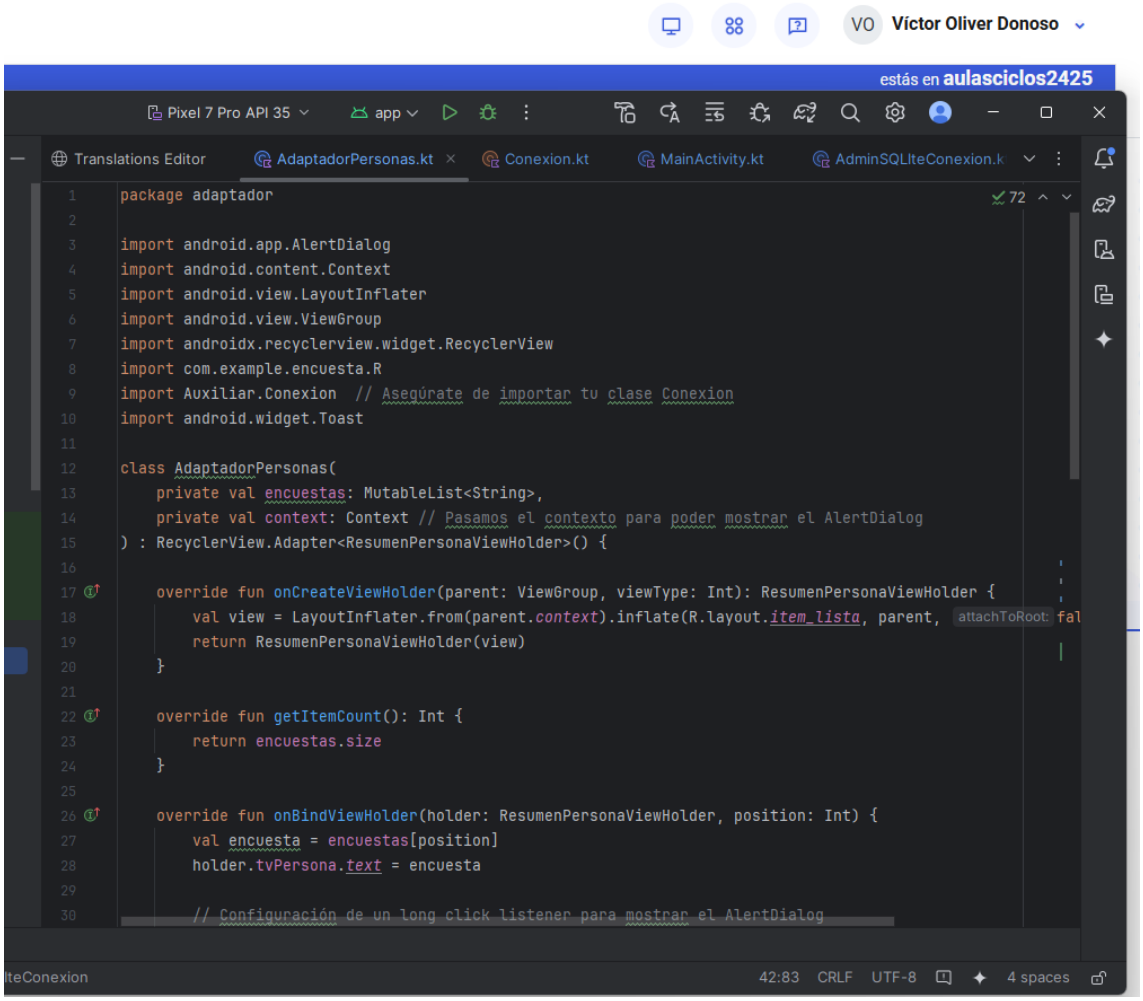
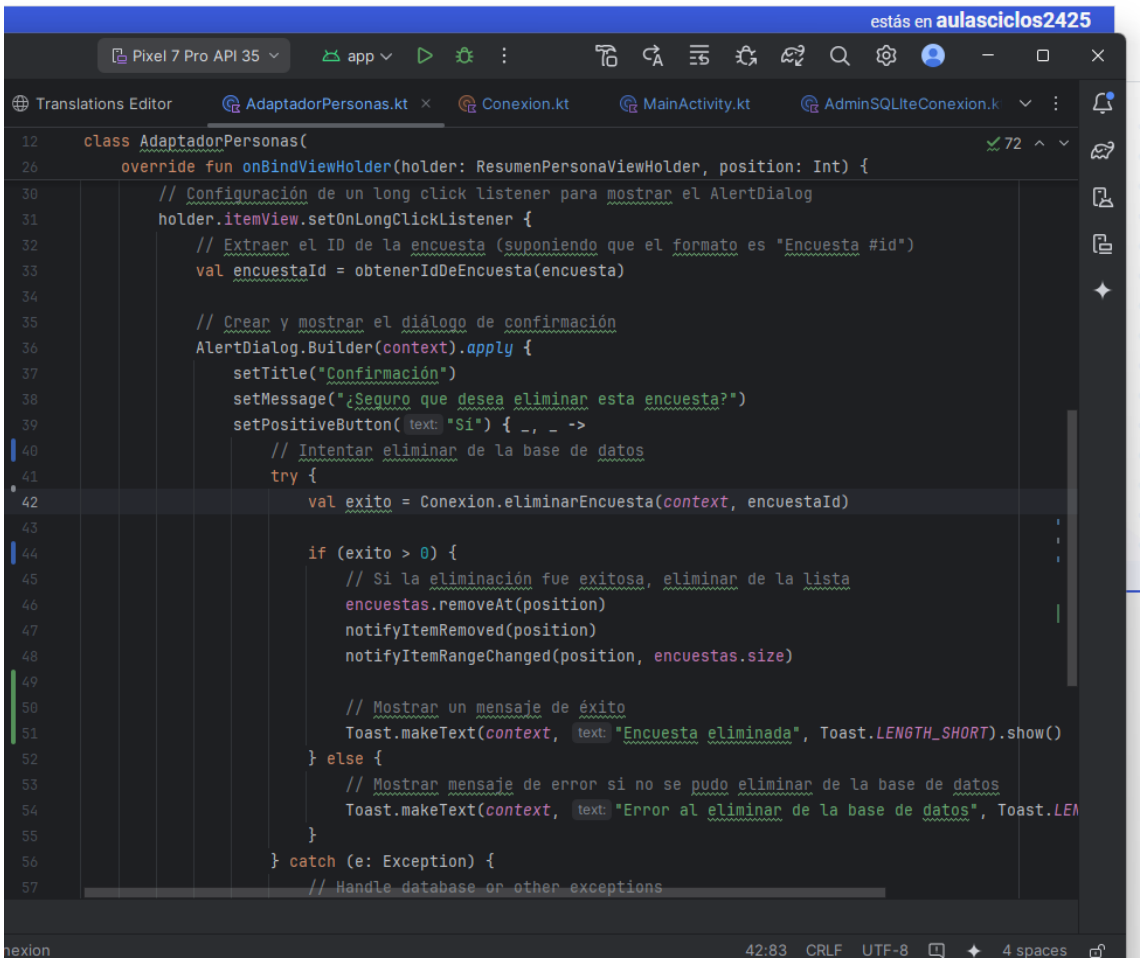
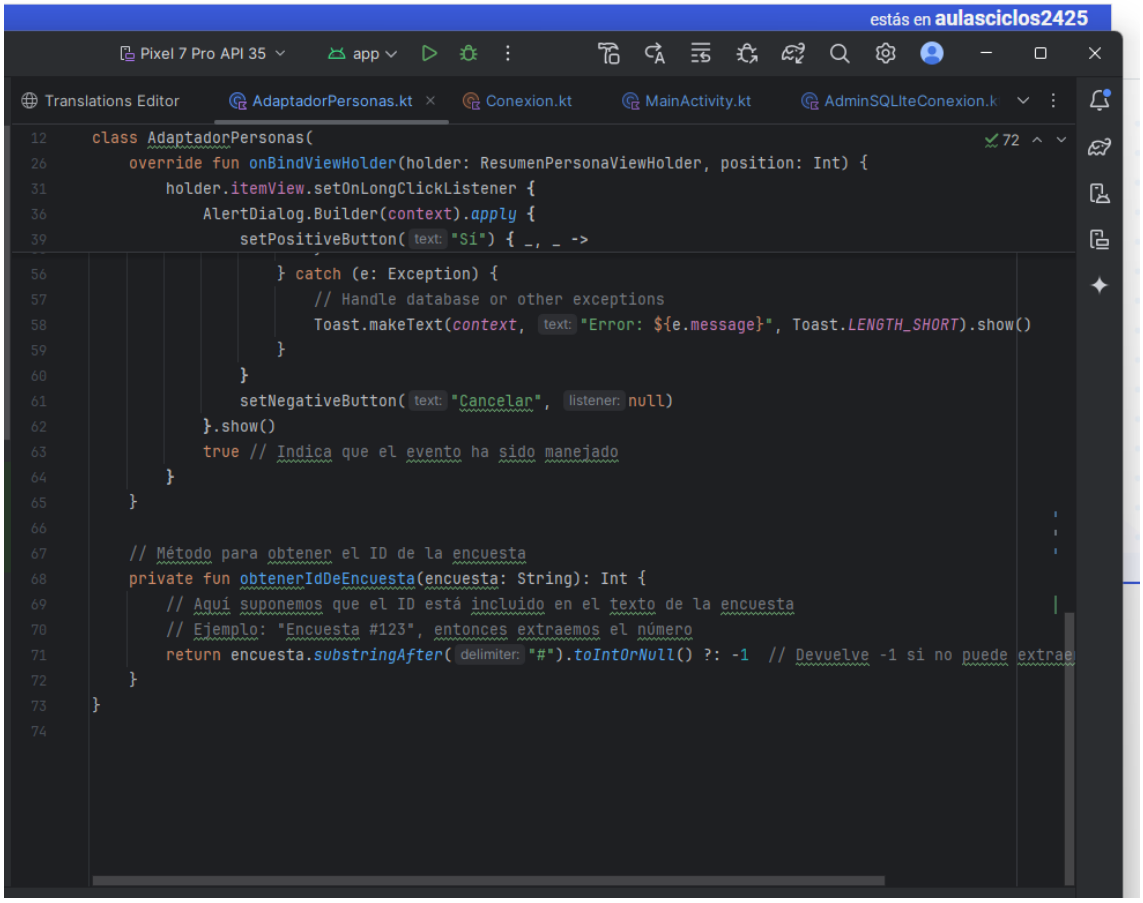
ENCUESTA SQLITE

Solo subiré a este pdf las capturas que tengan que ver con sqlite

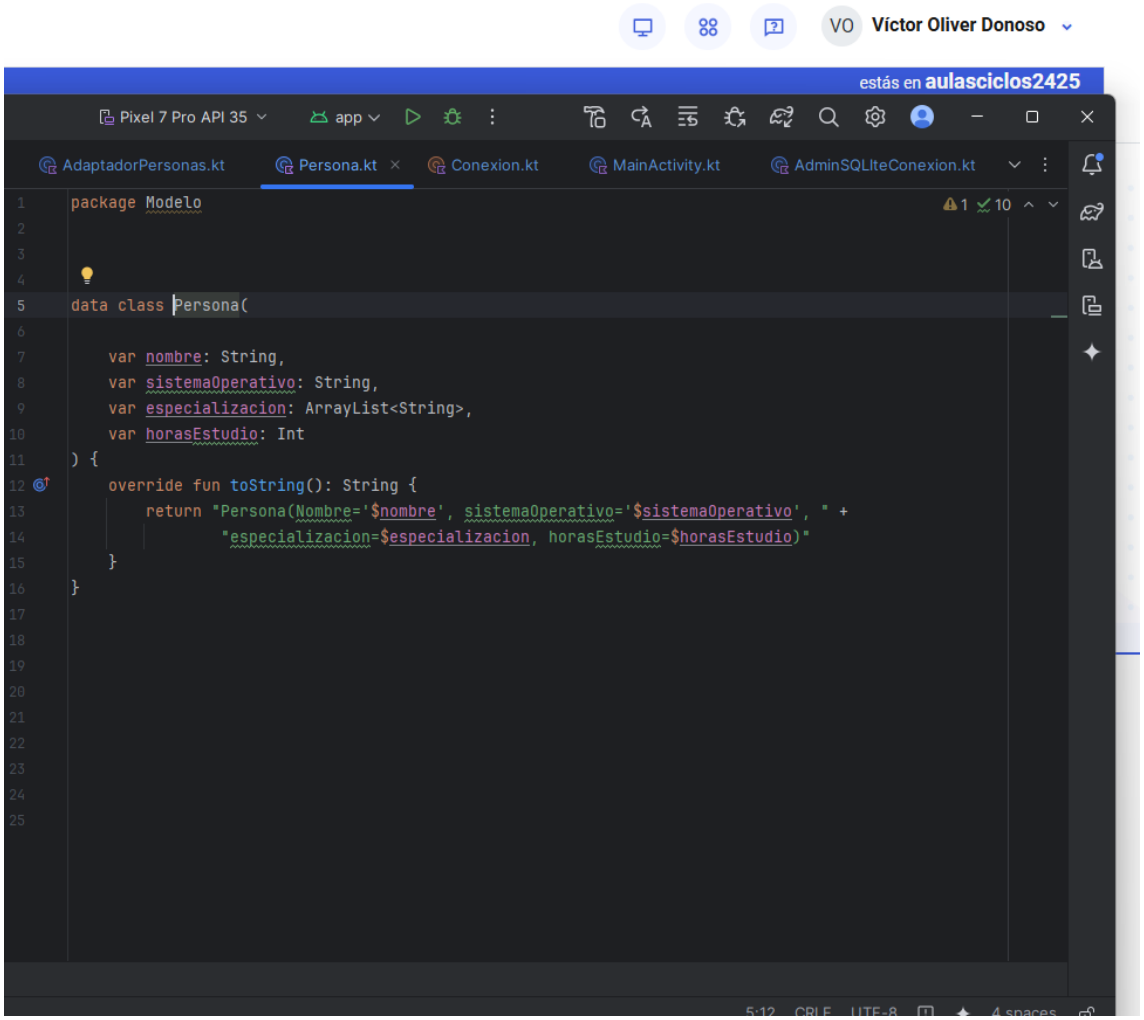
AdminSQLIteConexion



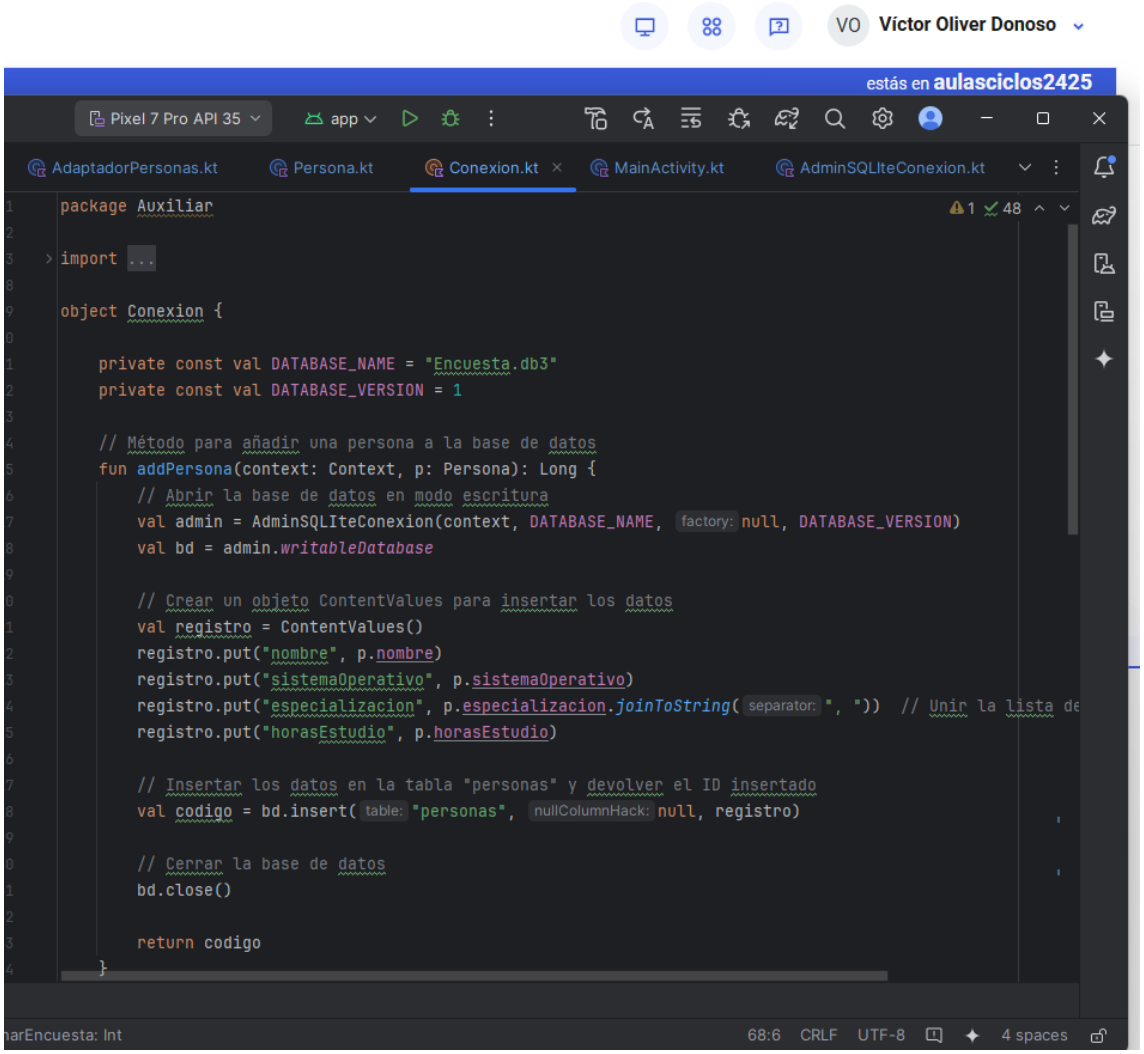

Pixel 7 Pro API 35
app
Translations Editor
AdaptadorPersonas.kt
Conexion.kt
MainActivity.kt
AdminSQLiteConexion.k
72
class AdaptadorPersonas(
override fun onBindViewHolder(holder: ResumenPersonaViewHolder, position: Int) {
// Configuración de un long click listener para mostrar el AlertDialog
holder.itemView.setOnLongClickListener {
// Extraer el ID de la encuesta (suponiendo que el formato es "Encuesta #id")
val encuestaId = obtenerIdDeEncuesta(encuesta)
// Crear y mostrar el diálogo de confirmación
AlertDialog.Builder(context).apply {
setTitle("Confirmación")
setMessage("¿Seguro que desea eliminar esta encuesta?")
setPositiveButton(text: "Sí") { _, _ ->
// Intentar eliminar de la base de datos
try {
val exito = Conexion.eliminarEncuesta(context, encuestaId)
if (exito > 0) {
// Si la eliminación fue exitosa, eliminar de la lista
encuestas.removeAt(position)
notifyItemRemoved(position)
notifyItemRangeChanged(position, encuestas.size)
// Mostrar un mensaje de éxito
Toast.makeText(context, text: "Encuesta eliminada", Toast.LENGTH_SHORT).show()
} else {
// Mostrar mensaje de error si no se pudo eliminar de la base de datos
Toast.makeText(context, text: "Error al eliminar de la base de datos", Toast.LEN
}
} catch (e: Exception) {
// Handle database or other exceptions
nexion
42:83
CRLF
UTF-8
4 spaces

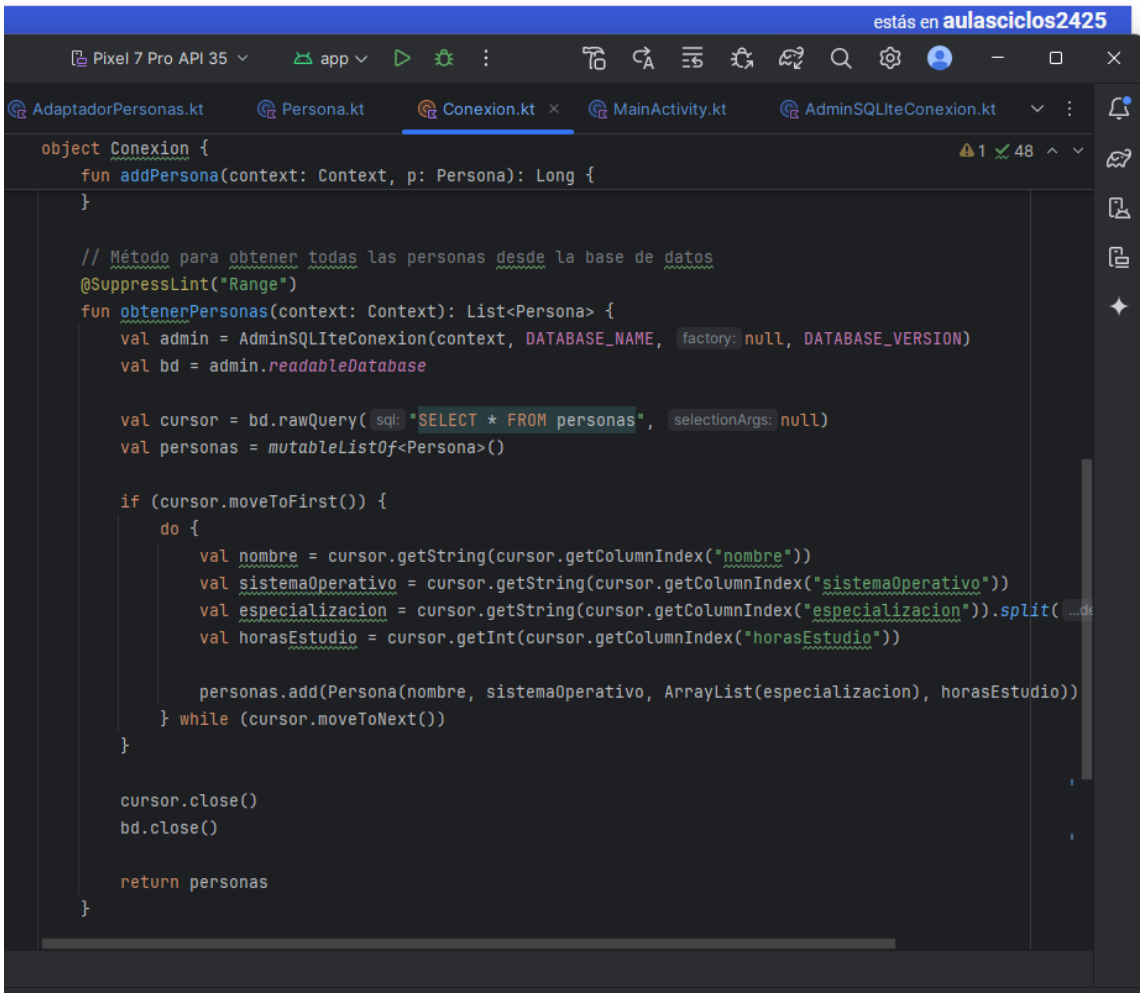

Pixel 7 Pro API 35
app
Translations Editor
AdaptadorPersonas.kt
Conexion.kt
MainActivity.kt
AdminSQLiteConexion.k
72
class AdaptadorPersonas(
override fun onBindViewHolder(holder: ResumenPersonaViewHolder, position: Int) {
holder.itemView.setOnLongClickListener {
AlertDialog.Builder(context).apply {
setPositiveButton(text: "Si") { _, _ ->
} catch (e: Exception) {
// Handle database or other exceptions
Toast.makeText(context, text: "Error: ${e.message}", Toast.LENGTH_SHORT).show()
}
}
setNegativeButton(text: "Cancelar", listener: null)
}.show()
true // Indica que el evento ha sido manejado
}
}
// Método para obtener el ID de la encuesta
private fun obtenerIdDeEncuesta(encuesta: String): Int {
// Aquí suponemos que el ID está incluido en el texto de la encuesta
// Ejemplo: "Encuesta #123", entonces extraemos el número
return encuesta.substringAfter(delimiter: "#").toIntOrNull() ?: -1 // Devuelve -1 si no puede extrae
}
}

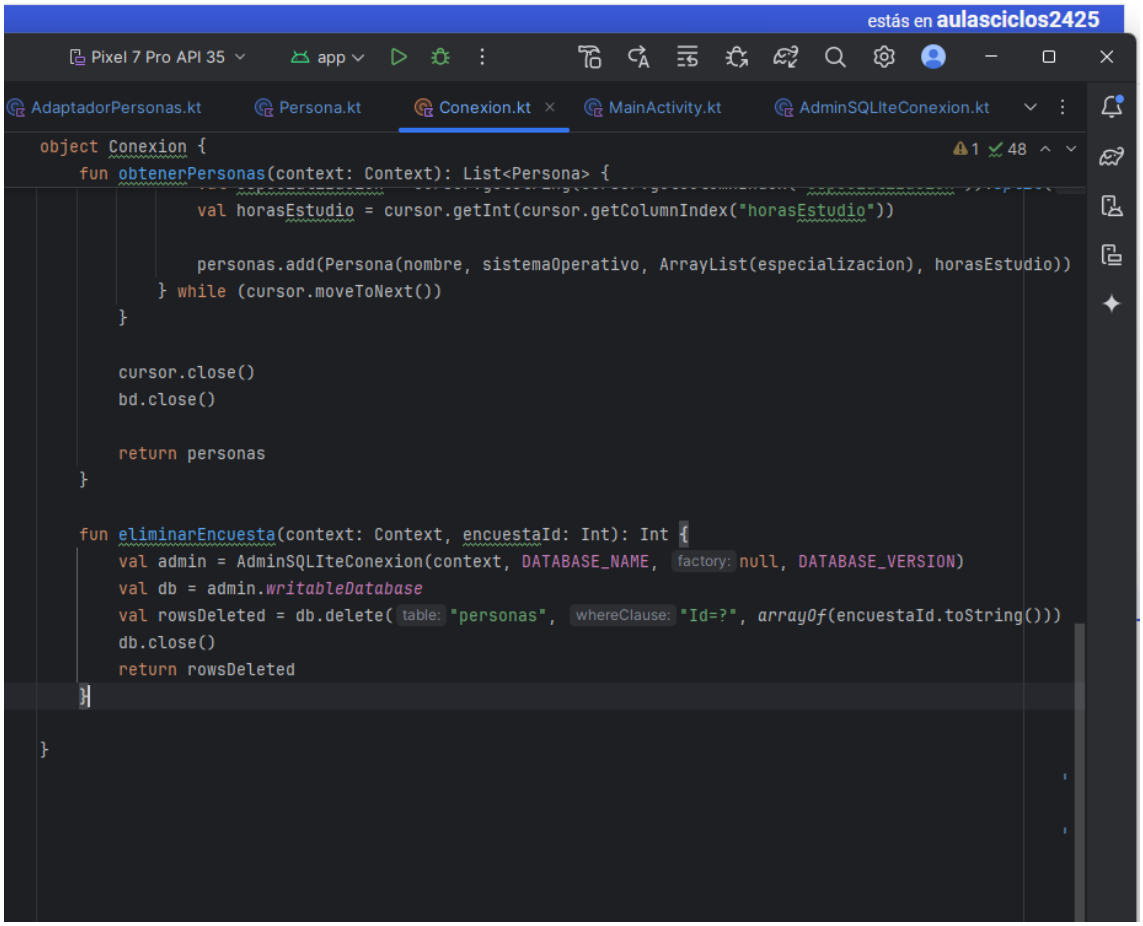
Clase Persona



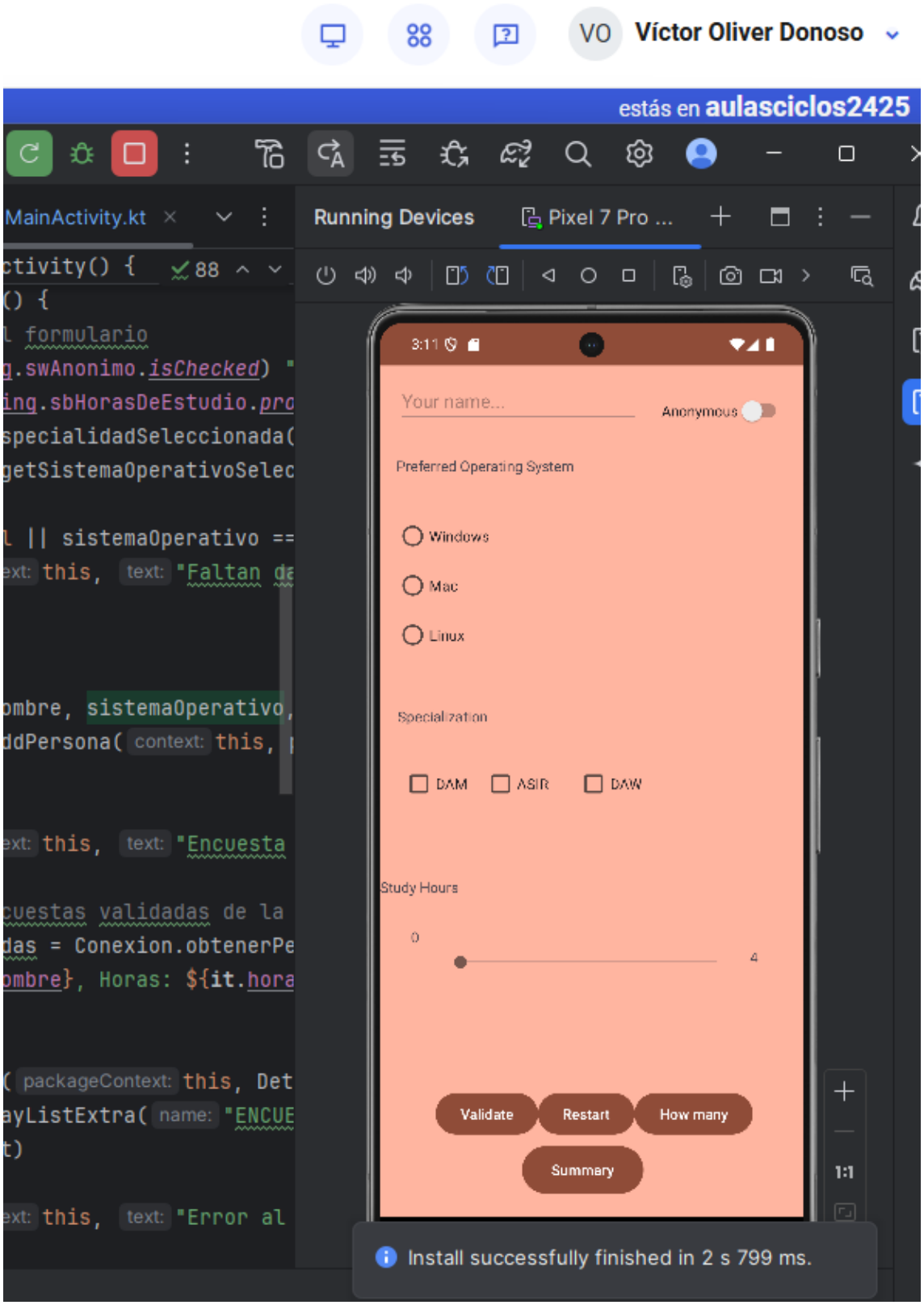
Conexión

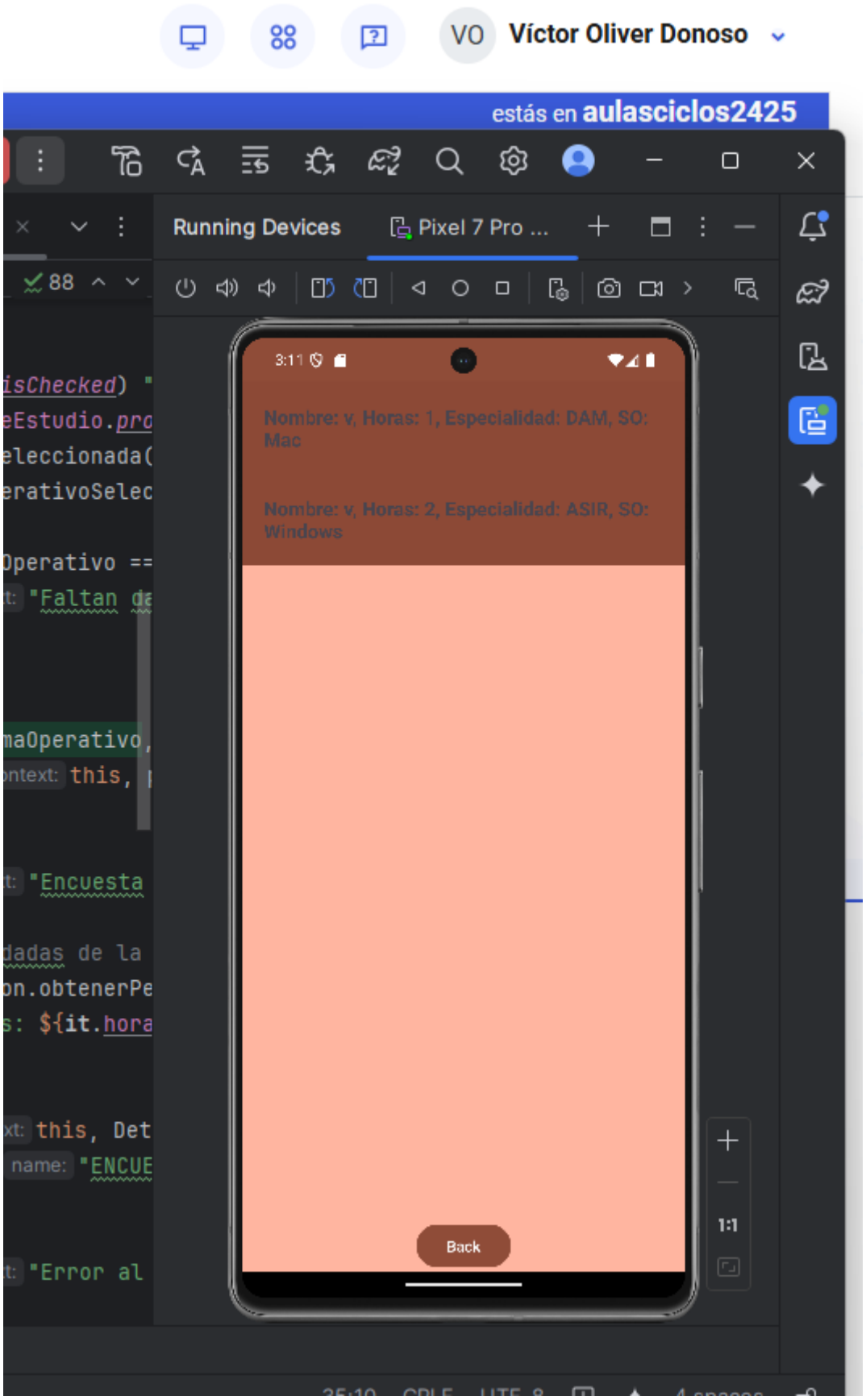

VO Víctor Oliver Donoso
estás en aulasciclos2425
Pixel 7 Pro API 35
app
AdaptadorPersonas.kt
Persona.kt
Conexion.kt
MainActivity.kt
AdminSQLIteConexion.kt
package Auxiliar
import ...
object Conexion {
private const val DATABASE_NAME = "Encuesta.db3"
private const val DATABASE_VERSION = 1
// Método para añadir una persona a la base de datos
fun addPersona(context: Context, p: Persona): Long {
// Abrir la base de datos en modo escritura
val admin = AdminSQLIteConexion(context, DATABASE_NAME, factory: null, DATABASE_VERSION)
val bd = admin.writableDatabase
// Crear un objeto ContentValues para insertar los datos
val registro = ContentValues()
registro.put("nombre", p.nombre)
registro.put("sistemaOperativo", p.sistemaOperativo)
registro.put("especializacion", p.especializacion.joinToString(separator: ", "))
registro.put("horasEstudio", p.horasEstudio)
// Insertar los datos en la tabla "personas" y devolver el ID insertado
val codigo = bd.insert(table: "personas", nullColumnHack: null, registro)
// Cerrar la base de datos
bd.close()
return codigo
}
68:6
CRLF
UTF-8
4 spaces


Pixel 7 Pro API 35
app
AdaptadorPersonas.kt
Persona.kt
Conexion.kt
MainActivity.kt
AdminSQLIteConexion.kt
1
48
object Conexion {
fun addPersona(context: Context, p: Persona): Long {
}
// Método para obtener todas las personas desde la base de datos
@SuppressLint("Range")
fun obtenerPersonas(context: Context): List<Persona> {
val admin = AdminSQLIteConexion(context, DATABASE_NAME, factory: null, DATABASE_VERSION)
val bd = admin.readableDatabase
val cursor = bd.rawQuery(sql: "SELECT * FROM personas", selectionArgs: null)
val personas = mutableListOf<Persona>()
if (cursor.moveToFirst()) {
do {
val nombre = cursor.getString(cursor.getColumnIndex("nombre"))
val sistemaOperativo = cursor.getString(cursor.getColumnIndex("sistemaOperativo"))
val especializacion = cursor.getString(cursor.getColumnIndex("especializacion")).split(
val horasEstudio = cursor.getInt(cursor.getColumnIndex("horasEstudio"))
personas.add(Persona(nombre, sistemaOperativo, ArrayList(especializacion), horasEstudio))
} while (cursor.moveToNext())
}
cursor.close()
bd.close()
return personas
}


Pixel 7 Pro API 35
app
AdaptadorPersonas.kt
Persona.kt
Conexion.kt
MainActivity.kt
AdminSQLIteConexion.kt
1
48
object Conexion {
fun obtenerPersonas(context: Context): List<Persona> {
val horasEstudio = cursor.getInt(cursor.getColumnIndex("horasEstudio"))
personas.add(Persona(nombre, sistemaOperativo, ArrayList(especializacion), horasEstudio))
} while (cursor.moveToNext())
}
cursor.close()
bd.close()
return personas
}
fun eliminarEncuesta(context: Context, encuestaId: Int): Int {
val admin = AdminSQLIteConexion(context, DATABASE_NAME, factory: null, DATABASE_VERSION)
val db = admin.writableDatabase
val rowsDeleted = db.delete(table: "personas", whereClause: "Id=?", arrayOf(encuestaId.toString()))
db.close()
return rowsDeleted
}
}

Funcionamiento, sin introducir ninguna encuesta, solo con las que se guardaron




Víctor Oliver Donoso
estás en aulasciclos2425
Running Devices
Pixel 7 Pro ...
3:11
Nombre: v, Horas: 1, Especialidad: DAM, SO: Mac
Nombre: v, Horas: 2, Especialidad: ASIR, SO: Windows
Back
1:1