



Carrera:

Técnico en ingeniería en computación

Docente:

Ing. Manuel Alexander Jiménez García.

Materia:

Desarrollo de Software para Móviles DSM441 G02T

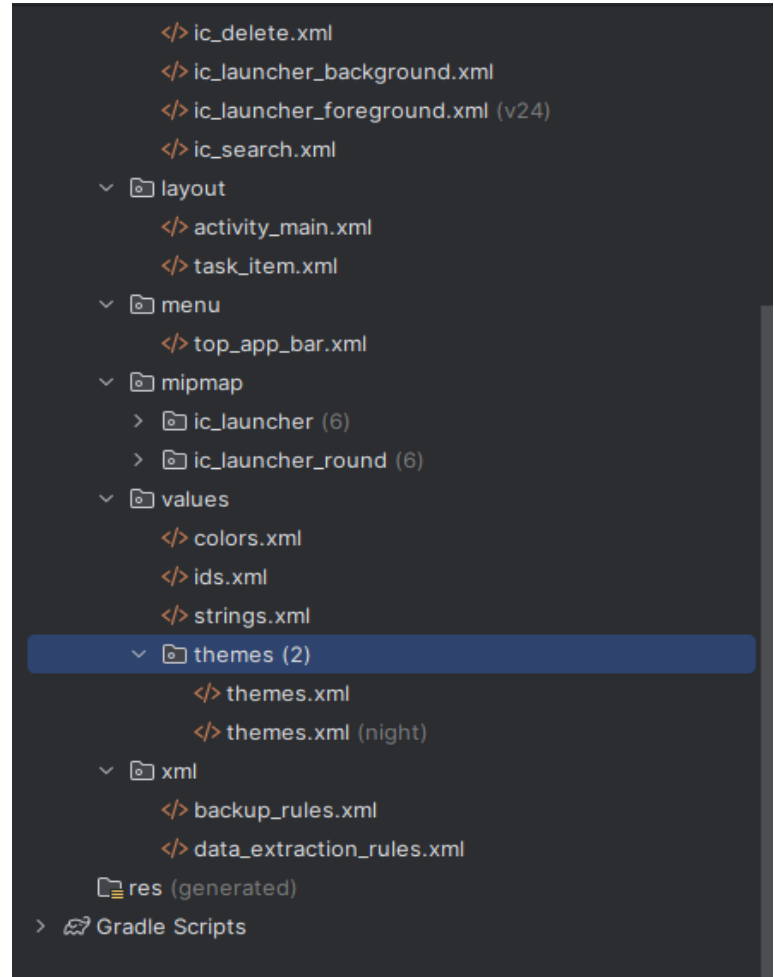
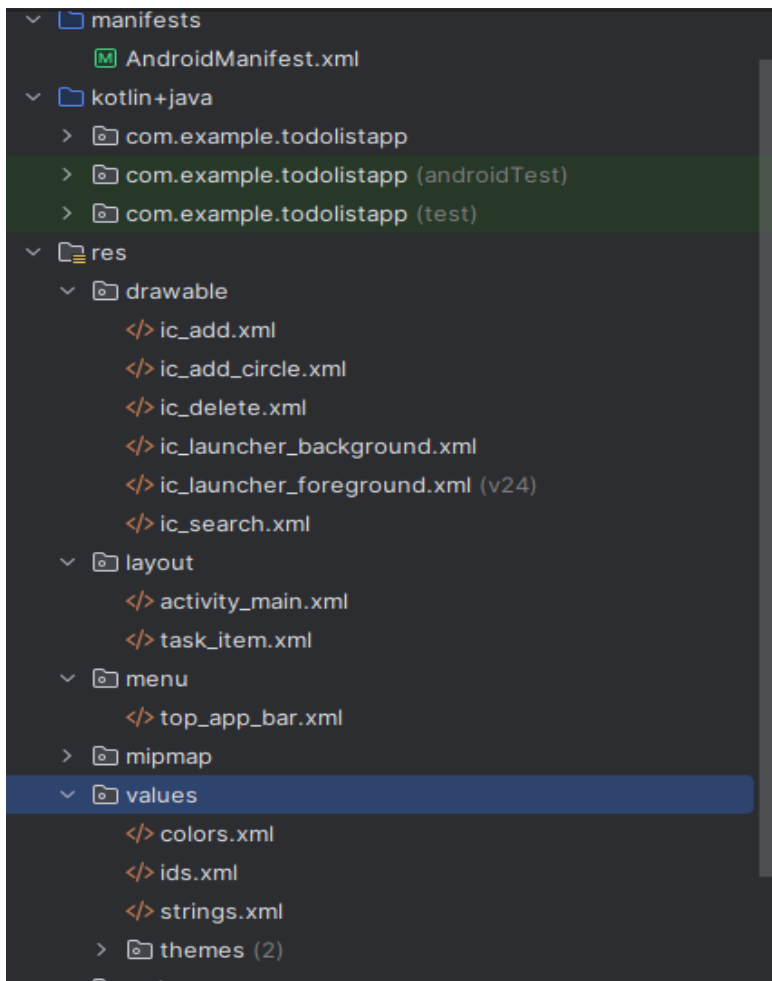
Integrantes:

Manuel Ezequiel Guerrero Granados GG241501

Fecha:

24/07/2025

Proceso de la App de tareas



Principalmente aquí se muestra la rama y los nombres de los archivos de cómo fue compuesto el archivo de tareas.

La creación de código fuente, así como las importaciones que fueron usadas en el archivo de MainActivity.kt

```
class MainActivity : AppCompatActivity() {
    private fun addTask() {
        saveTasks()
    } else {
        taskInputLayout.error = "Por favor ingrese una tarea"
    }
}

private fun onTaskChecked(position: Int, isChecked: Boolean) {
    taskList[position].completed = isChecked
    adapter.notifyItemChanged(position)
    saveTasks()
}

private fun onTaskDeleted(position: Int) {
    val deletedTask = taskList[position]
    taskList.removeAt(position)
    adapter.notifyItemRemoved(position)
    saveTasks()

    // Mostrar Snackbar con opción de deshacer
    Snackbar.make(taskRecyclerView, TEXT "Tarea eliminada", Snackbar.LENGTH_LONG)
        .setAction(TEXT "DESHACER") {
            taskList.add(position, deletedTask)
            adapter.notifyItemInserted(position)
            saveTasks()
        }
        .show()
}

private fun createItemTouchHelperCallback(): ItemTouchHelper.SimpleCallback {
    return object : ItemTouchHelper.SimpleCallback(
        dragDirs: 0, swipeDirs: ItemTouchHelper.LEFT or ItemTouchHelper.RIGHT
    ) {
        override fun onMove(
            recyclerView: RecyclerView,
            viewHolder: RecyclerView.ViewHolder,
            target: RecyclerView.ViewHolder
        ): Boolean = false

        override fun onSwiped(viewHolder: RecyclerView.ViewHolder, direction: Int) {

```

```
override fun onCreate(savedInstanceState: Bundle?) {
    // Inicializar vistas con findViewById
    taskRecyclerView = findViewById(R.id.taskRecyclerView)
    addButton = findViewById(R.id.addButton)
    fab = findViewById(R.id.fab)
    taskInputLayout = findViewById(R.id.taskInputLayout)
    taskInput = findViewById(R.id.taskInput)

    // Configurar RecyclerView
    taskRecyclerView.layoutManager = LinearLayoutManager(this)
    adapter = TaskAdapter(taskList, this::onTaskChecked, this::onTaskDeleted)
    taskRecyclerView.adapter = adapter

    // Configurar el ItemTouchHelper para deslizar para eliminar
    val itemTouchHelper = ItemTouchHelper(createItemTouchHelperCallback())
    itemTouchHelper.attachToRecyclerView(taskRecyclerView)

    // Cargar tareas guardadas
    loadTasks()

    // Configurar el botón para agregar tareas
    addButton.setOnClickListener { addTask() }

    // Configurar el FAB para agregar tareas
    fab.setOnClickListener {
        taskInputLayout.requestFocus()
        showKeyboard(taskInput)
    }

    // Configurar el icono de limpiar texto
    taskInputLayout.setEndIconOnClickListener {
        taskInput.text?.clear()
    }
}

private fun addTask() {
    val taskText = taskInput.text.toString().trim()
    if (taskText.isNotEmpty()) {
        val newTask = Task(taskText, completed: false)
        taskList.add(INDEX 0, newTask)
    }
}
```

```
package com.example.todoListApp

import android.content.Context
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.inputmethod.InputMethodManager
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.ItemTouchHelper
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.checkbox.MaterialCheckBox
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.android.material.snackbar.Snackbar
import com.google.android.material.textfield.TextInputLayout
import org.json.JSONArray
import org.json.JSONObject
import java.io.*

class MainActivity : AppCompatActivity() {
    private lateinit var taskRecyclerView: RecyclerView
    private lateinit var addButton: Button
    private lateinit var fab: FloatingActionButton
    private lateinit var taskInputLayout: TextInputLayout
    private lateinit var taskInput: EditText

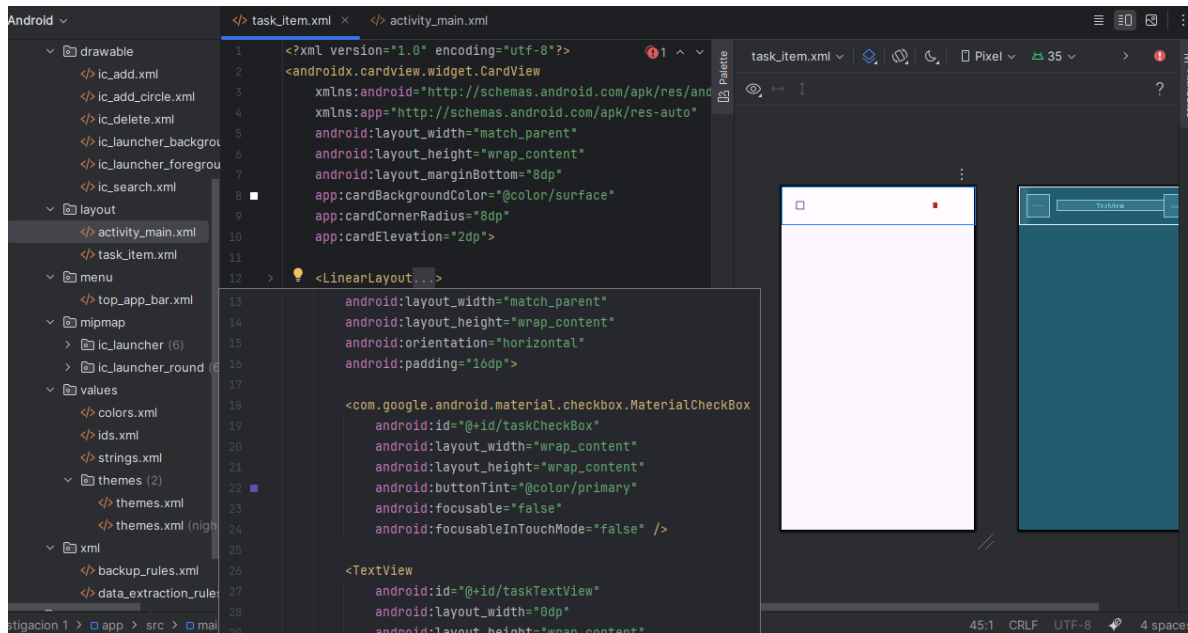
    private val taskList = mutableListOf<Task>()
    private lateinit var adapter: TaskAdapter
    private val FILENAME = "tasks.json"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

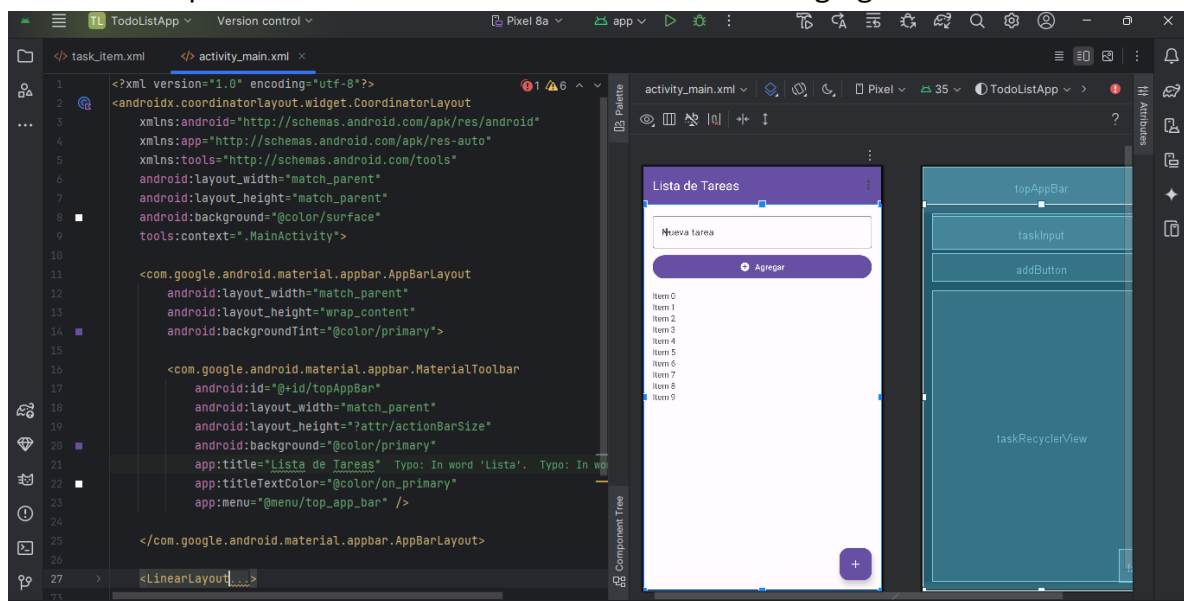
        // Inicializar vistas con findViewById
        taskRecyclerView = findViewById(R.id.taskRecyclerView)
        addButton = findViewById(R.id.addButton)

```

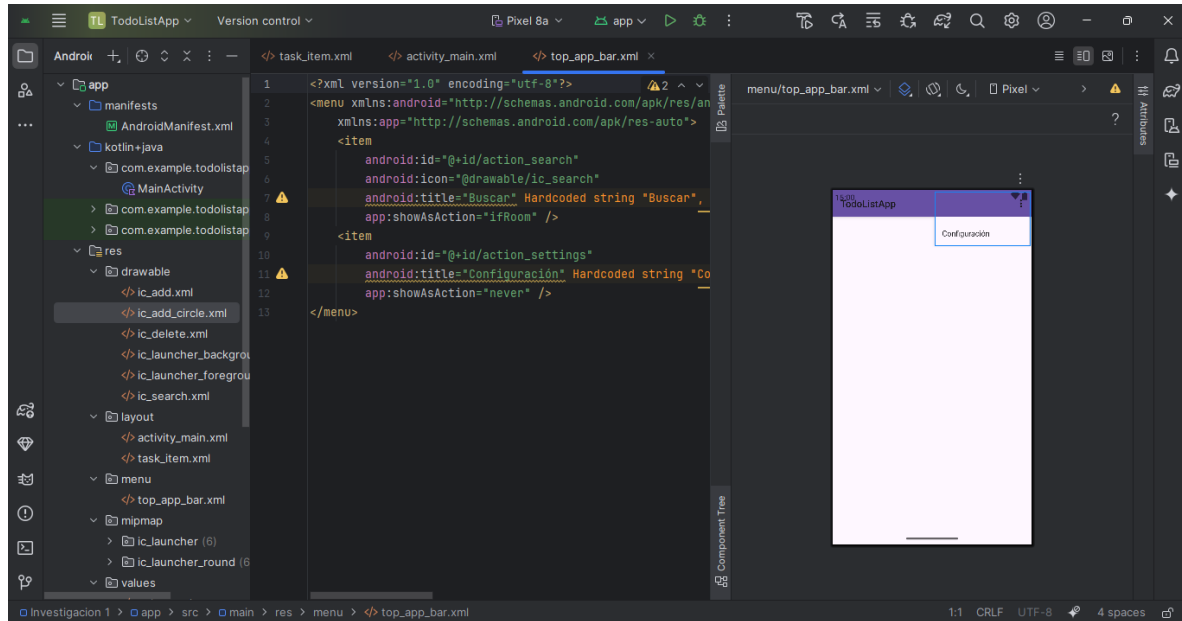
El Archivo task_item.xml serían los ítems que crea los iconos de borrado y apuntado (el encargado de borrar las tareas y de marcar las casillas mostrando al usuario que la tarea que ha realizado se ha completado).



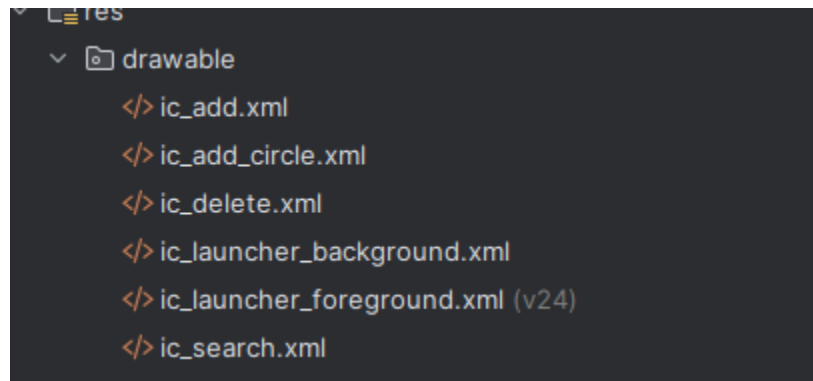
El archivo Activity_main.xml conformaría lo que sería el orden que debe ir al momento de agregar una tarea, así como que se guarde y muestre satisfactoriamente al usuario la tarea que se ah inscrito al seleccionar la casilla de agregar una nueva tarea



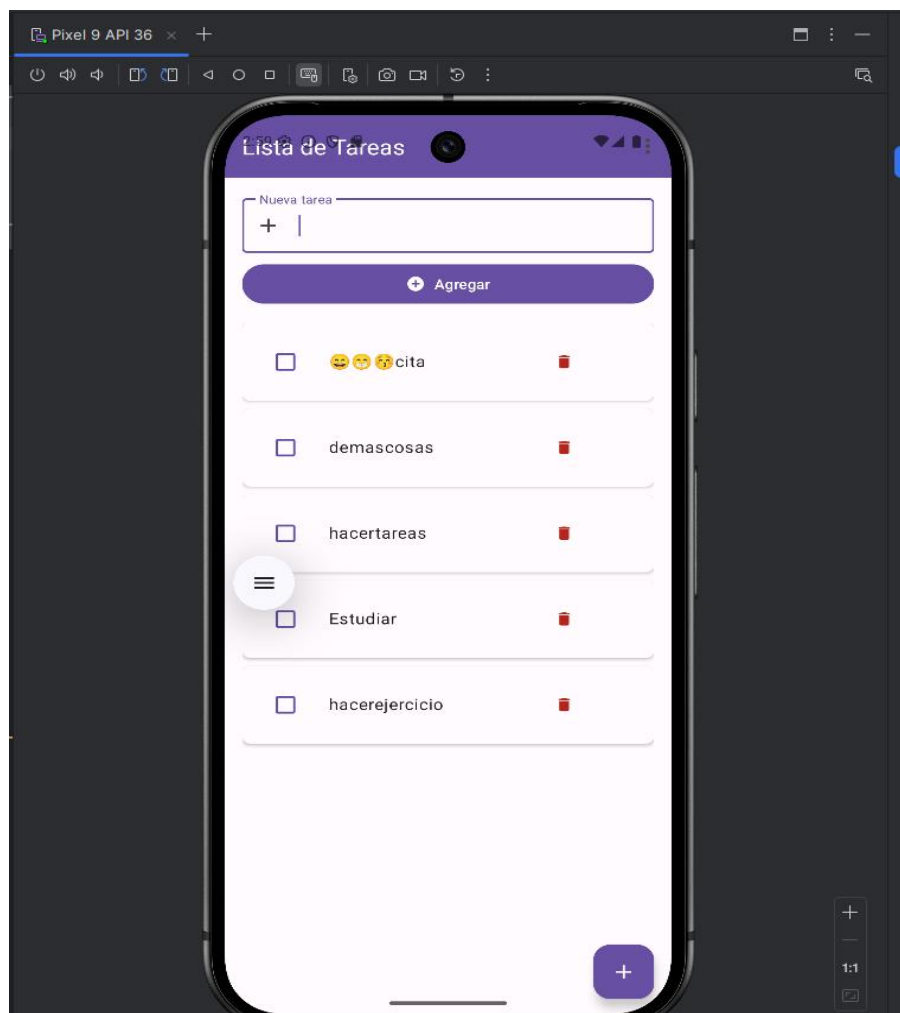
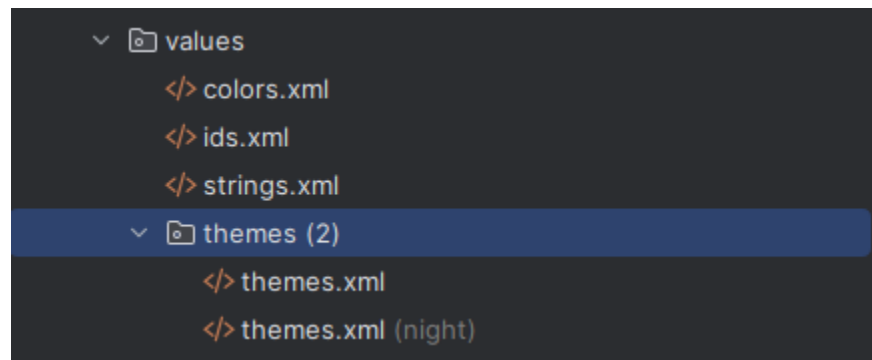
El archivo `top_app_bar.xml` sería el que cumple la función de ser la configuración ofreciendo al usuario las diferentes funciones que puede hacer para en el momento de agregar una nueva tarea ya sea agregarle emojis hablar por voz y de ser el traductor también.



En la carpeta `drawable` serian donde residen los iconos que se muestran en la App



En las carpetas values y themes serian donde residen los colores que le dan mejor diseño de color visual para el usuario



una vez dentro de la app el usuario podrá agregar las siguientes tareas que se muestran (solo es un ejemplo) solamente el usuario debe presionar la casilla poner dicha tarea que deberá posponer para realizarlo en otro momento una vez ingresada la tarea tendrá la opción de marcar la casilla como completo oh borrar la tarea si fue

un error suyo agregar algo que ya había realizado , el usuario al momento de seleccionar el cuadro de barra(el menú) que al abrir se mostrar al diferentes opciones como que puede comentar por voz la tarea a agregar sin necesidad de escribir también de agregar emojis, cuenta con un traductor y demás.

Error al momento de la ejecución.

El único error que se podrá encontrar seria que al momento de marcar la casilla de la tarea automáticamente hará que saque al usuario de la App