



DSM L Evaluado 3

Autores: Mario Steven Cabrera Velasco CV230385

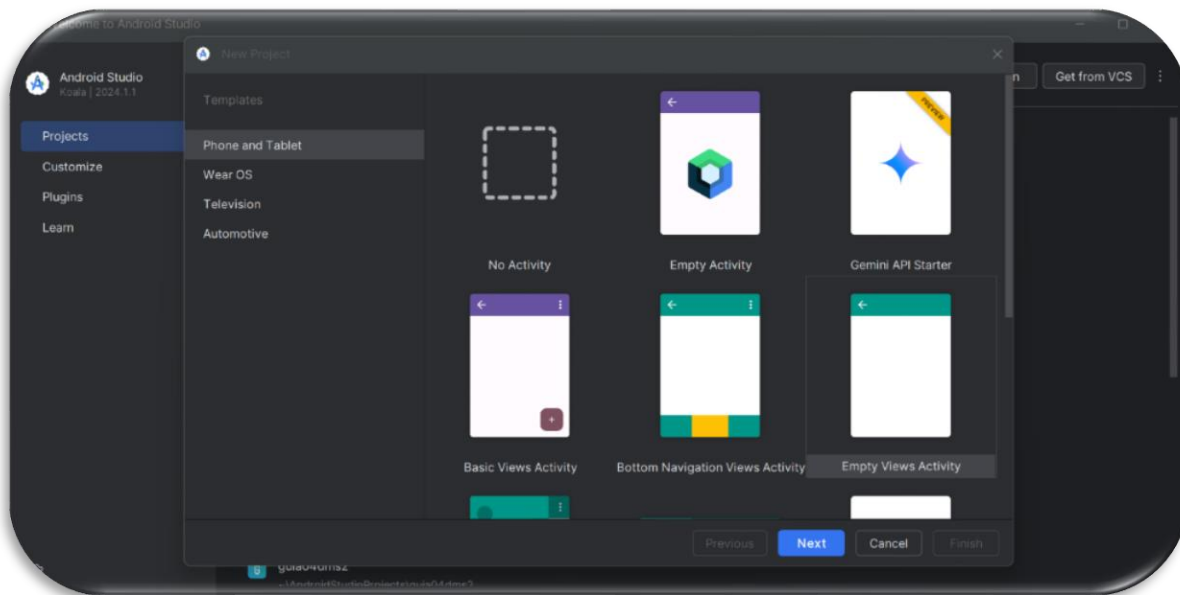
David Alejandro Menjivar Batres MB180762

10/9/2024

Nuestra aplicación consiste en Una app que les permite a los estudiantes de ingeniería en sistemas acceder una lista de recursos como libros, videos y tutoriales.

Primero creamos nuestro proyecto en Android Studio

El nombre de nuestra aplicación es: IngeniLearn

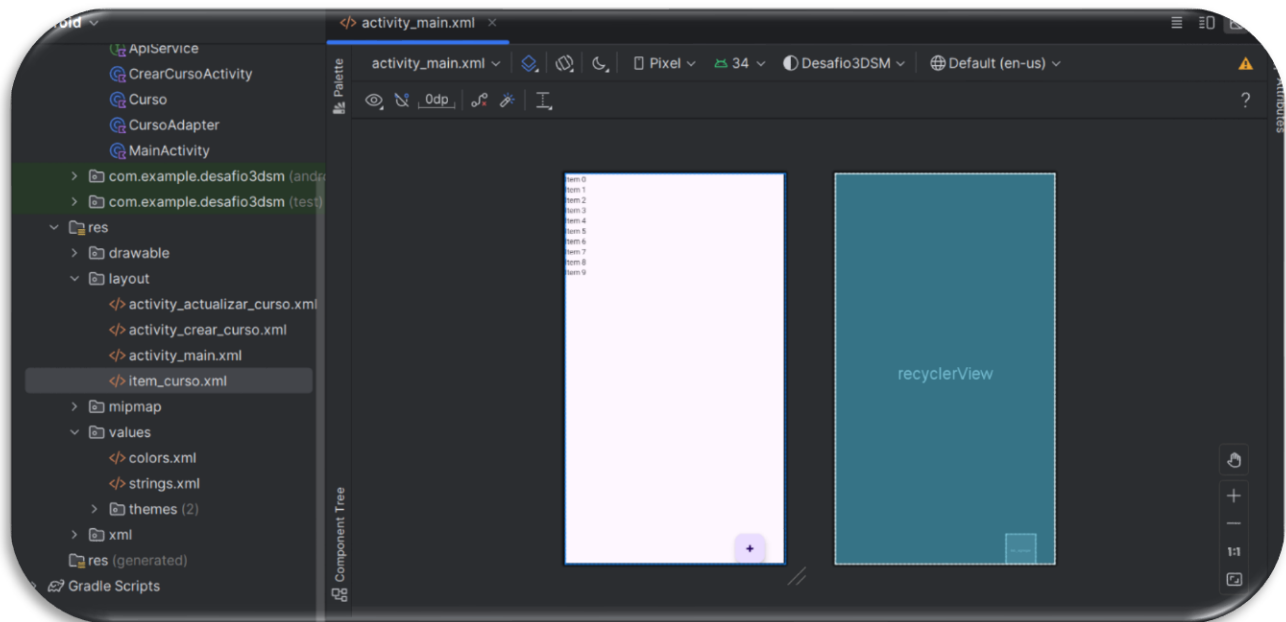


Creamos nuestro logo

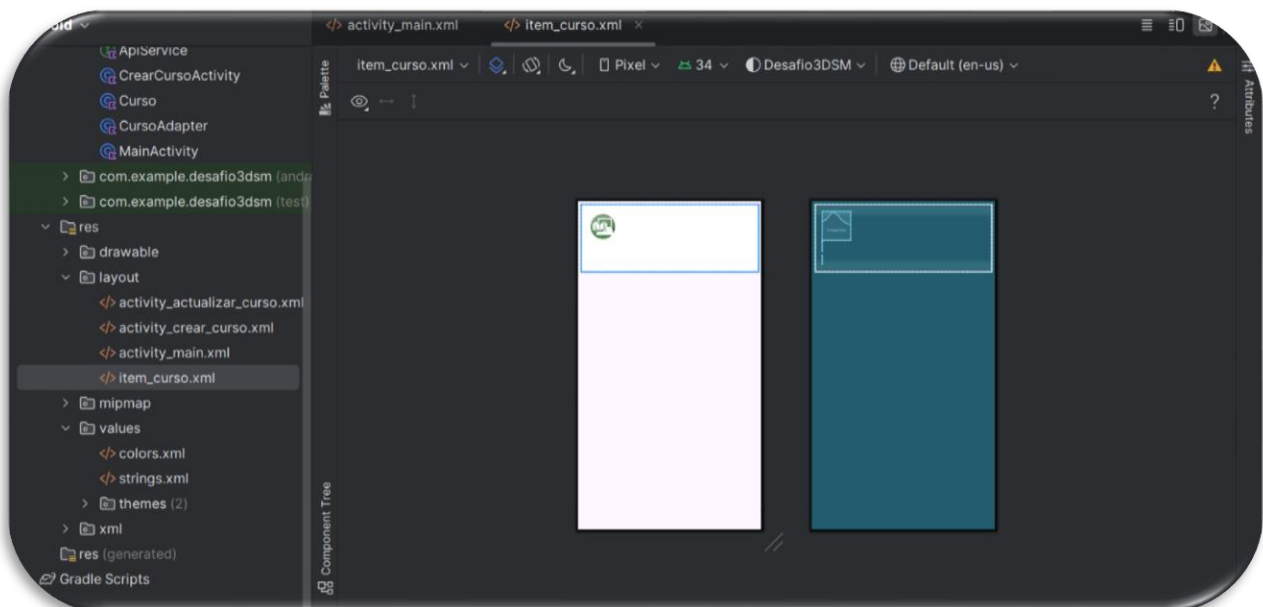


Creamos nuestras vistas que serian lo siguiente

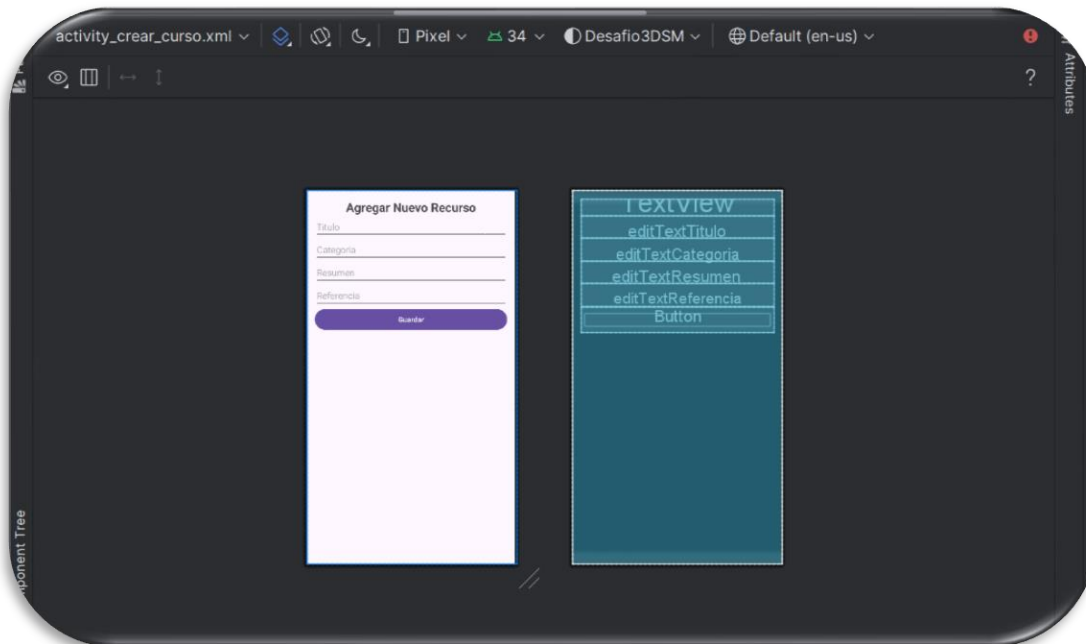
Creamos el Activity_Main.xml



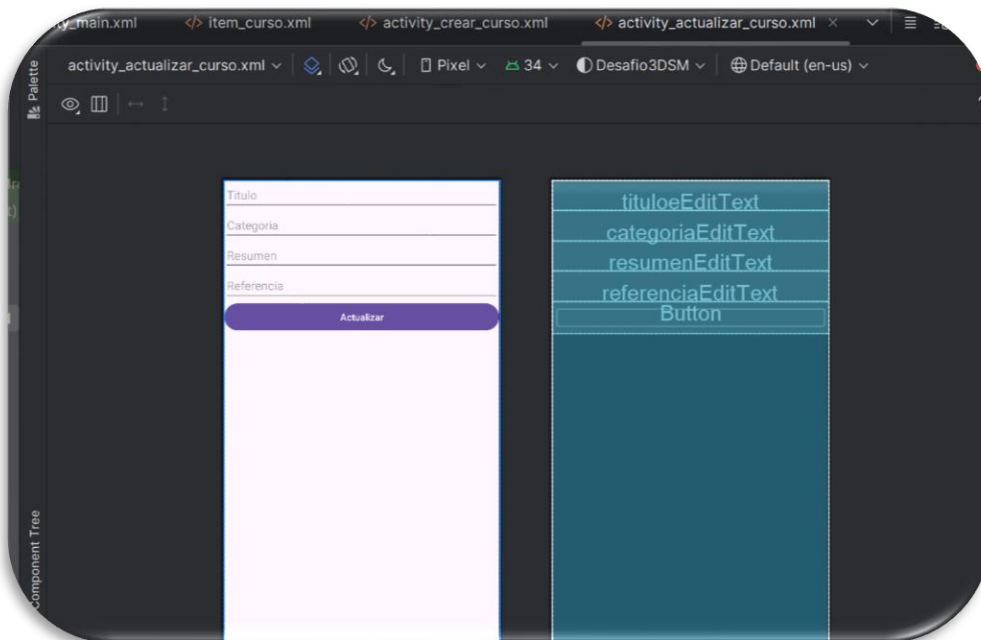
Creamos el item.xml



Creamos la vista crea_curso.xml



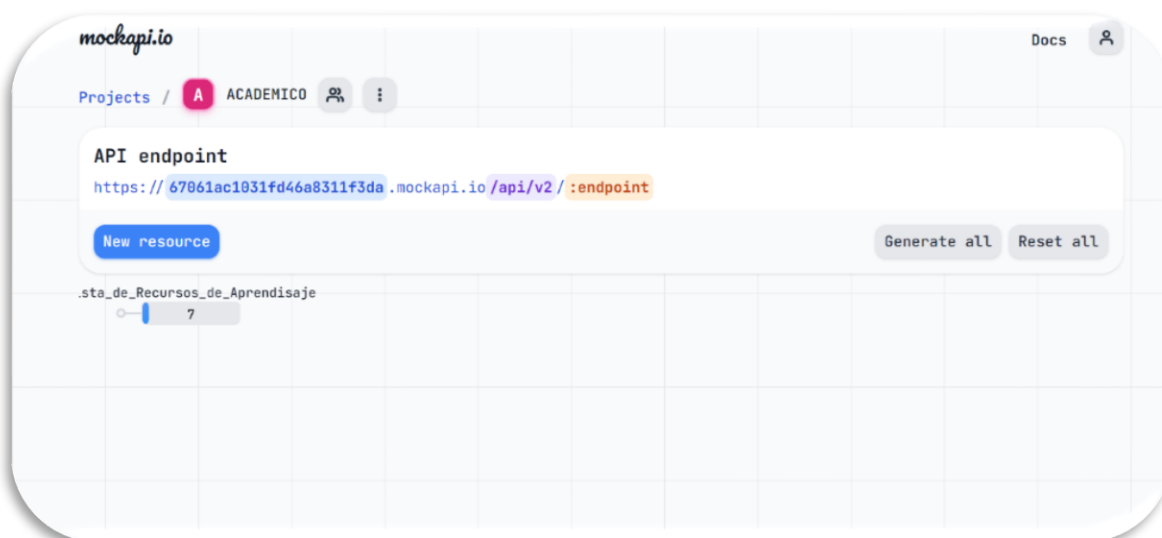
Creamos la vista Actualizar_curso.xml nos permite modificar un curso disponible



Ahora realizaremos la parte del funcionamiento la lógica para eso tenemos que irnos al archivo MainActivity.Kt aquí comenzamos a darle funcionamiento

```
MainActivity.kt x
20 class MainActivity : AppCompatActivity() {
31     override fun onCreate(savedInstanceState: Bundle?) {
36         cardDetalles = findViewById(R.id.card_detalles)
37         textTitulo = findViewById(R.id.text_titulo)
38         textCategoria = findViewById(R.id.text_categoria)
39         textResumen = findViewById(R.id.text_resumen)
40         textReferencia = findViewById(R.id.text_referencia)
41
42         cardDetalles.visibility = View.GONE
43
44
45         val fabAgregar: FloatingActionButton = findViewById(R.id.fab_agregar)
46
47         recyclerView = findViewById(R.id.recyclerView)
48         recyclerView.layoutManager = LinearLayoutManager(context, this)
49
50
51         // Configurar Retrofit
52         val retrofit : Retrofit = Retrofit.Builder()
53             .baseUrl("https://67061ac1031fd46a8311f3da.mockapi.io/api/v2/")
54             .addConverterFactory(GsonConverterFactory.create())
55             .build()
56
57         apiService = retrofit.create(ApiService::class.java)
58     }
59 }
```

Creamos nuestro api utilizando mockapi.io para almacenar los datos de los recursos



```

{
  "Titulo": "Curso de LÓGICA DE PROGRAMACIÓN Desde Cero",
  "Categoria": "Video",
  "Resumen": "Este curso va dirigido a aquellos que quieren iniciar en el mundo de la programación, ya que tener una buena lógica al momento de programar es importante.",
  "Referencia": "https://www.youtube.com/watch?v=TdITcV064zI",
  "id": "3"
},
{
  "Titulo": "Curso de SQL y BASES DE DATOS Desde Cero",
  "Categoria": "Video",
  "Resumen": "Aprender a trabajar con bases de datos y lenguaje SQL es una habilidad esencial como programador; por eso, aprender a manejar bases de datos es importante.",
  "Referencia": "https://www.youtube.com/watch?v=0UJerKzV5T0",
  "id": "4"
},
{
  "Titulo": "Introducción a la Programación con Java",
  "Categoria": "Libro",
  "Resumen": "Este libro es perfecto para aquellos que quieren comenzar a usar Java y conocerlo más a profundidad.",
  "Referencia": "https://portafoliodeevidenciasderick.weebly.com/uploads/2/5/4/9/25498533/introduccion_a_la_programacion_con_java.pdf",
  "id": "5"
},
{
  "Titulo": "Código Limpio",
  "Categoria": "Libro",
  "Resumen": "Con este libro aprenderás a escribir código más limpio, siguiendo buenas prácticas y aplicándolo a nuestros proyectos de programación.",
  "Referencia": "https://elhacker.info/manuales/Lenguajes%20de%20Programacion/Codigo%20limpio%20-%20Robert%20Cecil%20Martin.pdf",
  "id": "6"
},
{
  "Titulo": "Curso del Lenguaje de Programación C#",
  "Categoria": "Video",
  "Resumen": "C# es uno de los lenguajes de programación más populares de internet; este curso te enseñará todo lo necesario para dominar el lenguaje C#.",
  "Referencia": "https://www.youtube.com/watch?v=ynevmXveJUM",
  "id": "7"
}

```

Creamos nuestra clase llamada curso define los datos de api

```

MainActivity.kt  CursoAdapter.kt  Curso.kt x
1 package com.example.desafio3dsm
2
3 data class Curso(
4     val Titulo: String,
5     val Categoria: String,
6     val Resumen: String,
7     val Referencia: String,
8     val id: String
9 )
10
11
12

```

Creamos nuestra clase cursoAdapter que siempre va relacionado con api muestra los ítems

```

app  manifests  kotlin+java  com.example.desafio3dsm  ActualizarCursoActivity  ApiService  CrearCursoActivity  Curso  CursoAdapter  MainActivity
  > com.example.desafio3dsm (android)
  > com.example.desafio3dsm (test)
  res  drawable  layout
    <? activity_actualizar_curso.xml
    <? activity_crear_curso.xml
    <? activity_main.xml
    <? item_curso.xml
  mipmap  values  colors.xml

12 class CursoAdapter(private val cursos: List<Curso>) {
13
14     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CursoViewHolder {
15         val view: View = LayoutInflater.from(parent.context)
16             .inflate(R.layout.item_curso, parent, attachToRoot = false)
17         return CursoViewHolder(view)
18     }
19
20     override fun onBindViewHolder(holder: CursoViewHolder, position: Int) {
21         val curso: Curso = cursos[position]
22
23         // Verifica si los datos no son nulos ni vacíos
24         Log.d("CursoAdapter", "Titulo: ${curso.Titulo}, Resumen: ${curso.Resumen}")
25
26         holder.tituloTextView.text = curso.Titulo
27         holder.resumenTextView.text = curso.Resumen
28
29         holder.itemView.setOnClickListener {
30             onItemClick?.onItemClick(curso)
31         }
32     }
33
34     override fun getItemCount(): Int = cursos.size
35 }

```

Creamos nuestra clase CrearCursosActivity aquí iría toda la lógica para crear nuevos recursos para el api

```
class CrearCursoActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        crearButton.setOnClickListener {
            api.crearCurso(nuevoCurso).enqueue(object : Callback<Curso> {
                override fun onResponse(call: Call<Curso>, response: Response<Curso>) {
                    if (response.isSuccessful) {
                        Toast.makeText(
                            context: this@CrearCursoActivity,
                            text: "Curso creado exitosamente",
                            Toast.LENGTH_SHORT
                        ).show()

                        // Inicia la actividad principal después de crear el curso
                        val intent = Intent(packageContext: this@CrearCursoActivity, MainActivity::class)
                        startActivity(intent)
                        finish()
                    } else {
                        val error :String? = response.errorBody()?.string()
                        Log.e(tag: "API", msg: "Error al crear curso: $error")
                        Toast.makeText(
                            context: this@CrearCursoActivity,
                            text: "Error al crear el curso",
                            Toast.LENGTH_SHORT
                        ).show()
                    }
                }
            })
        }
    }
}
```

Creamos la clase ActualizarCursoActivity es para actualizar los datos de la Api

```
class ActualizarCursoActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        // Cargar los datos del curso a actualizar
        cargarCurso()

        // Configuración botón de actualizar
        actualizarButton.setOnClickListener {
            actualizarCurso()
        }
    }

    private fun cargarCurso() {
        // Obtener el ID del curso desde el Intent
        val cursoId :String? = intent.getStringExtra(name: "curso_id") // Este debería ser un String

        // Verificar que el cursoId no sea null
        if (cursoId != null) {
            // Llamar a la API para obtener los datos del curso
            apiService.obtenerCursoPorId(cursoId).enqueue(object : Callback<Curso> {
                override fun onResponse(call: Call<Curso>, response: Response<Curso>) {
                    if (response.isSuccessful) {
                        val curso :Curso? = response.body()
                        // Verificar que el curso no sea null antes de acceder a sus propiedades
                    }
                }
            })
        }
    }
}
```

Y luego aquí definimos los métodos del api como el Post,Get,Delete y Put

```
Activity.kt  CursoAdapter.kt  ApiService.kt  CrearCursoActivity.kt  ActualizarCursoActivity.kt x
14      class ActualizarCursoActivity : AppCompatActivity() {
23      override fun onCreate(savedInstanceState: Bundle?) {
36
37          // Cargar los datos del curso a actualizar
38          cargarCurso()
39
40          // Configuración botón de actualizar
41          actualizarButton.setOnClickListener {
42              actualizarCurso()
43          }
44      }
45
46      private fun cargarCurso() {
47          // Obtener el ID del curso desde el Intent
48          val cursoId :String? = intent.getStringExtra( name: "curso_id") // Esto debería ser un String
49
50          // Verificar que el cursoId no sea null
51          if (cursoId != null) {
52              // Llamar a la API para obtener los datos del curso
53              apiService.obtenerCursoPorId(cursoId).enqueue(object : Callback<Curso> {
54                  override fun onResponse(call: Call<Curso>, response: Response<Curso>) {
55                      if (response.isSuccessful) {
56                          val curso :Curso? = response.body()
57                          // Verificar que el curso no sea null antes de acceder a sus propiedades
58                      }
59                  }
60              })
61          }
62      }
63
64      private fun actualizarCurso() {
65          // Obtener el ID del curso desde el Intent
66          val cursoId :String? = intent.getStringExtra( name: "curso_id") // Esto debería ser un String
67
68          // Verificar que el cursoId no sea null
69          if (cursoId != null) {
70              // Llamar a la API para actualizar el curso
71              apiService.actualizarCurso(cursoId, curso).enqueue(object : Callback<Curso> {
72                  override fun onResponse(call: Call<Curso>, response: Response<Curso>) {
73                      if (response.isSuccessful) {
74                          val curso :Curso? = response.body()
75                          // Verificar que el curso no sea null antes de acceder a sus propiedades
76                      }
77                  }
78              })
79          }
80      }
81  }
```

Resultados

