



# Universidad Don Bosco

---

Escuela de Ingeniería en Computación

## Desarrollo de Software para Móviles – G01T

### Desafío # 3:

Aplicación móvil en Kotlin con Android

### Catedrático:

Tec. Julio Armando García Sánchez

### Equipo colaborativo:

Samuel Isaac Azucena Gómez - AG170932

Diana Jasmín García Fabian -GF230979

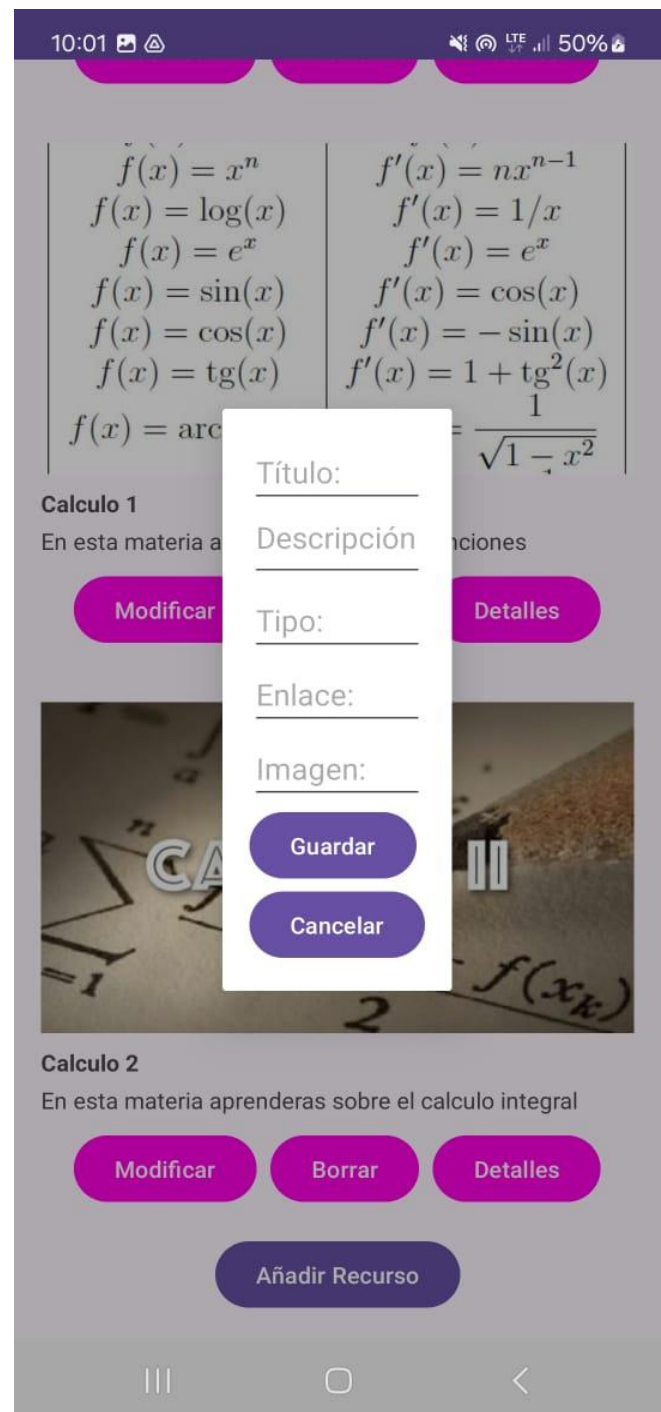
### Campus Soyapango

2024

### Carrera universitaria:

Técnico en Ingeniería en Computación

## APP FUNCIONAL.



10:01


 50%

$f(x) = x^n$	$f'(x) = nx^{n-1}$
$f(x) = \log(x)$	$f'(x) = 1/x$
$f(x) = e^x$	$f'(x) = e^x$
$f(x) = \sin(x)$	$f'(x) = \cos(x)$
$f(x) = \cos(x)$	$f'(x) = -\sin(x)$
$f(x) = \operatorname{tg}(x)$	$f'(x) = 1 + \operatorname{tg}^2(x)$
$f(x) = \arcsin(x)$	$f'(x) = \frac{1}{\sqrt{1-x^2}}$

**Calculo 1**  
 En esta materia aprenderas limites y funciones

Modificar

Borrar

Detalles

**Calculo 2**  
 En esta materia aprenderas sobre el calculo integral

Modificar

Borrar

Detalles

Añadir Recurso

10:02


 51%

$f(x) = x^n$	$f'(x) = nx^{n-1}$
$f(x) = \log(x)$	$f'(x) = 1/x$
$f(x) = e^x$	$f'(x) = e^x$
$f(x) = \sin(x)$	$f'(x) = \cos(x)$
$f(x) = \cos(x)$	$f'(x) = -\sin(x)$
$f(x) = \operatorname{tg}(x)$	$f'(x) = 1 + \operatorname{tg}^2(x)$
$f(x) = \arcsin(x)$	$f'(x) = \frac{1}{\sqrt{1-x^2}}$

**Calculo 1**  
 En esta materia a... nciones

Modificar

Detalles

**Calculo 2**  
 En esta materia aprenderas sobre el calculo integral

Modificar

Borrar

Detalles

Añadir Recurso

Astronomia

NASA

1

.gov/

61527291

Guardar

Cancelar

$f(x) = x^n$	$f'(x) = nx^{n-1}$
$f(x) = \log(x)$	$f'(x) = 1/x$
$f(x) = e^x$	$f'(x) = e^x$
$f(x) = \sin(x)$	$f'(x) = \cos(x)$
$f(x) = \cos(x)$	$f'(x) = -\sin(x)$
$f(x) = \operatorname{tg}(x)$	$f'(x) = 1 + \operatorname{tg}^2(x)$
$f(x) = \arcsin(x)$	$f'(x) = \frac{1}{\sqrt{1-x^2}}$

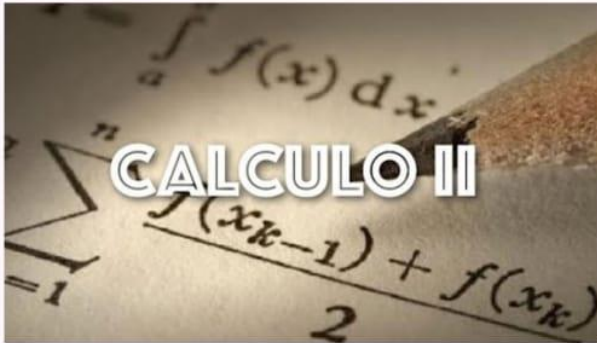
**Calculo 1**

En esta materia aprenderas limites y funciones

Modificar

Borrar

Detalles

**Calculo 2**

En esta materia aprenderas sobre el calculo integral

Modificar

Borrar

Detalles



Recurso creado

Añadir Recurso

**Fisic de messi moderna de messi**

albert eisntein es la mera mona

Modificar

Borrar

Detalles

$f(x) = x^n$	$f'(x) = nx^{n-1}$
$f(x) = \log(x)$	$f'(x) = 1/x$
$f(x) = e^x$	$f'(x) = e^x$
$f(x) = \sin(x)$	$f'(x) = \cos(x)$
$f(x) = \cos(x)$	$f'(x) = -\sin(x)$
$f(x) = \operatorname{tg}(x)$	$f'(x) = 1 + \operatorname{tg}^2(x)$
$f(x) = \arcsin(x)$	$f'(x) = \frac{1}{\sqrt{1-x^2}}$

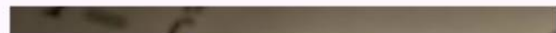
**Calculo 1**

En esta materia aprenderas limites y funciones

Modificar

Borrar

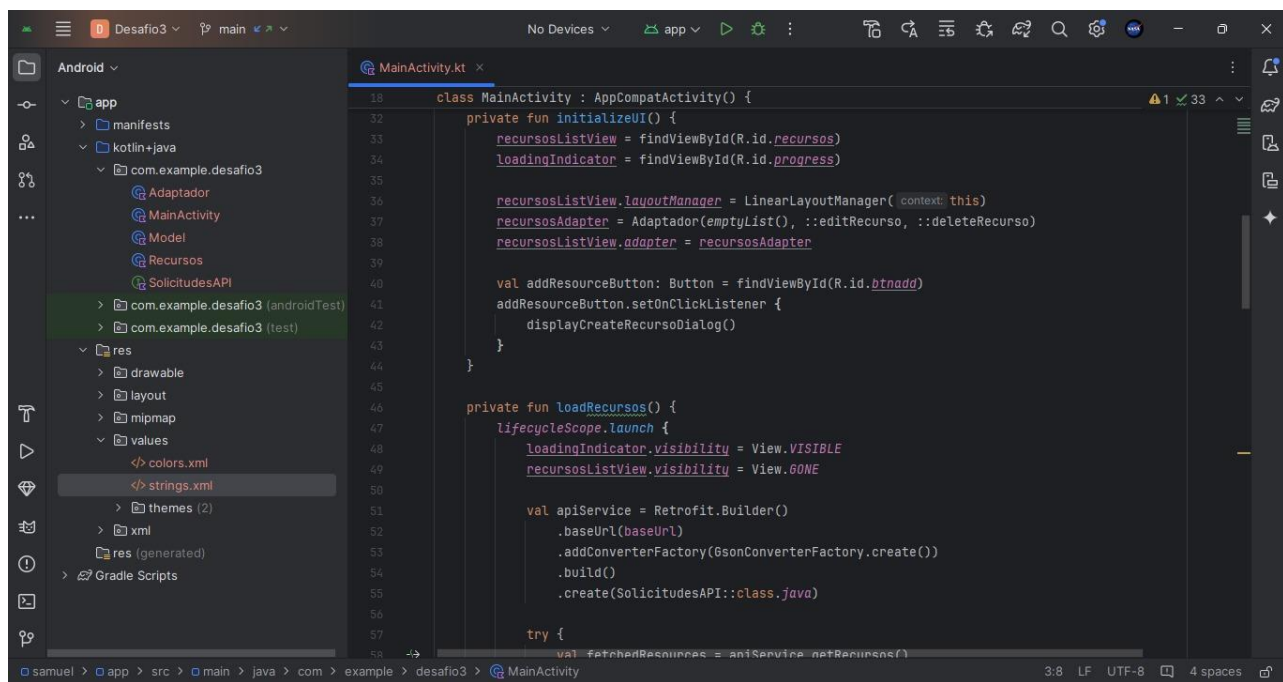
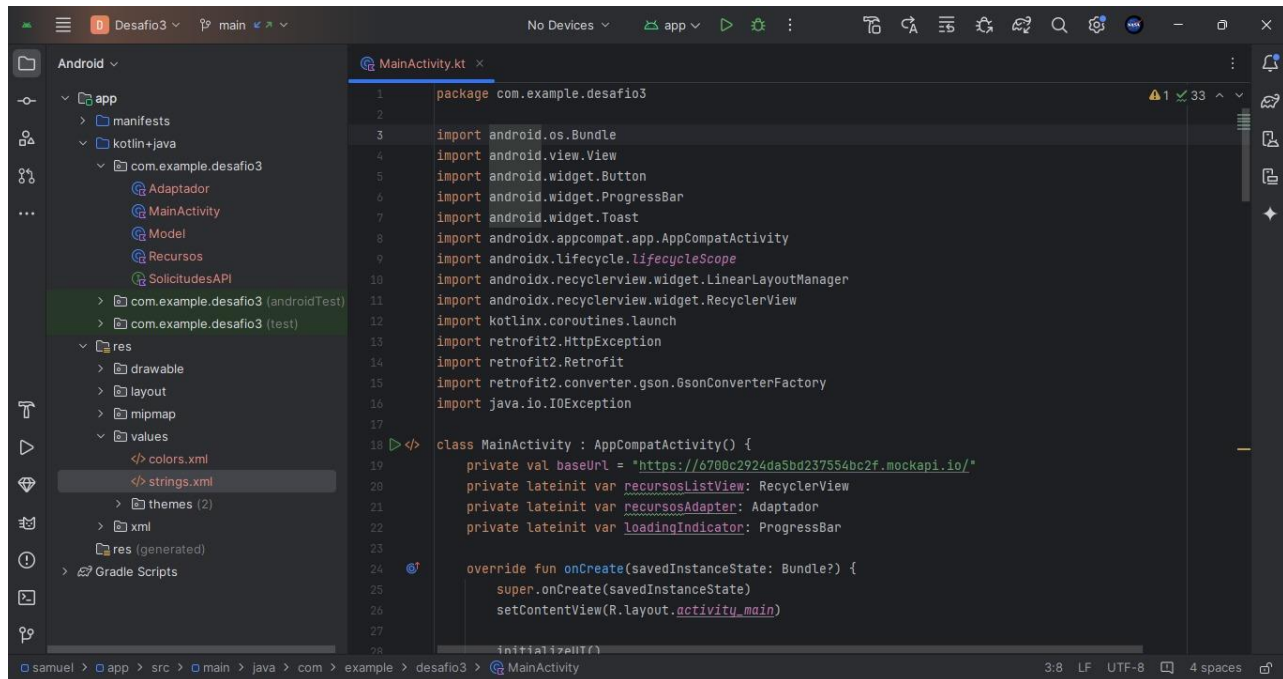
Detalles



Añadir Recurso



## MainActivity.kt





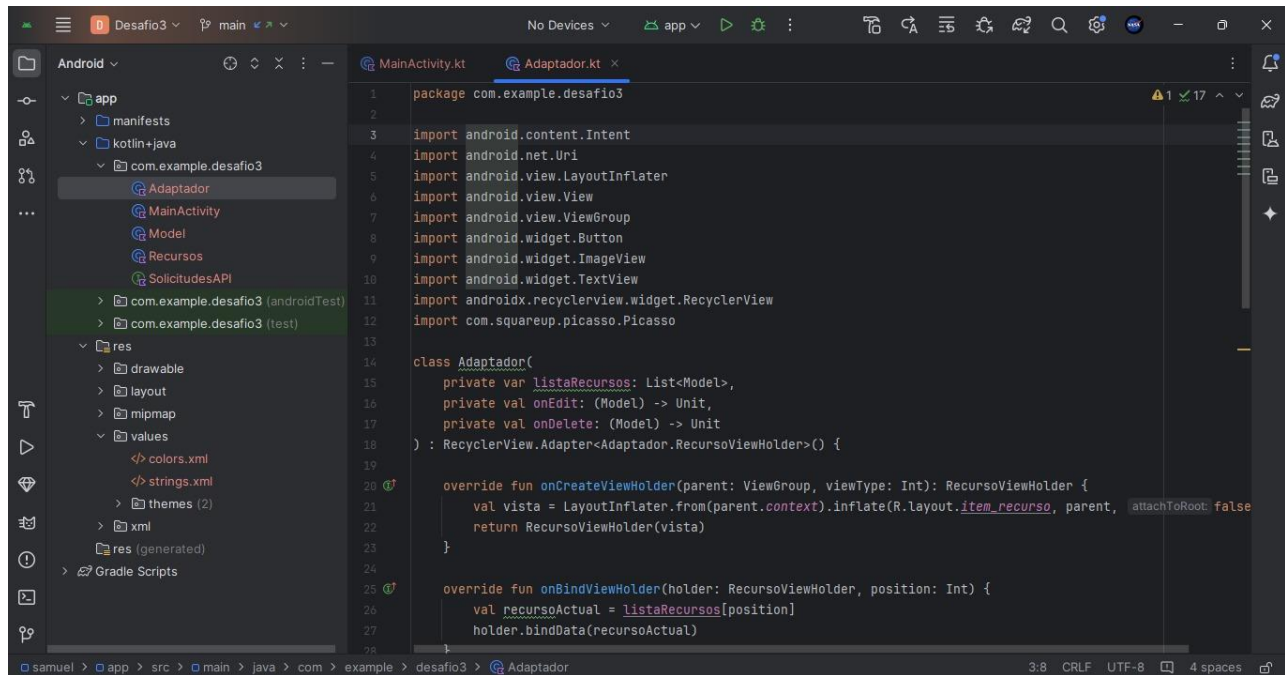
```
18 class MainActivity : AppCompatActivity() {
46     private fun loadRecursos() {
47         lifecycleScope.launch {
58             val fetchedResources = apiService.getRecursos()
59             recursoAdapter.actualizarRecursos(fetchedResources)
60             recursosListView.visibility = View.VISIBLE
61
62             if (fetchedResources.isEmpty()) {
63                 showToastMessage("No se encontraron recursos")
64             }
65         } catch (exception: Exception) {
66             handleException(exception)
67         } finally {
68             loadingIndicator.visibility = View.GONE
69         }
70     }
71
72     private fun displayCreateRecursoDialog() {
73         val dialog = Recursos(context: this) { nuevoRecurso ->
74             addNewRecurso(nuevoRecurso)
75         }
76         dialog.show()
77     }
78
79     private fun editRecurso(recurso: Model) {
80         val dialog = Recursos(context: this, recurso) { updatedRecurso ->
81             modifyRecurso(updatedRecurso)
82         }
```

```
109     private fun modifyRecurso(recurso: Model) {
110         val retrofit = Retrofit.Builder().baseUrl(baseUrl).addConverterFactory(GsonConverterFactory.create())
111         val apiService = retrofit.create(SolicitudesAPI::class.java)
112
113         lifecycleScope.launch {
114             try {
115                 apiService.updateRecurso(recurso.id, recurso)
116                 showToastMessage("Recurso actualizado correctamente")
117                 loadRecursos()
118             } catch (e: Exception) {
119                 showToastMessage("Error: ${e.message}")
120             }
121         }
122
123     private fun removeRecurso(recurso: Model) {
124         val retrofit = Retrofit.Builder().baseUrl(baseUrl).addConverterFactory(GsonConverterFactory.create())
125         val apiService = retrofit.create(SolicitudesAPI::class.java)
126
127         lifecycleScope.launch {
128             try {
129                 apiService.deleteRecurso(recurso.id)
130                 showToastMessage("Recurso eliminado correctamente")
131                 loadRecursos()
132             } catch (e: Exception) {
133                 showToastMessage("Error: ${e.message}")
134             }
```

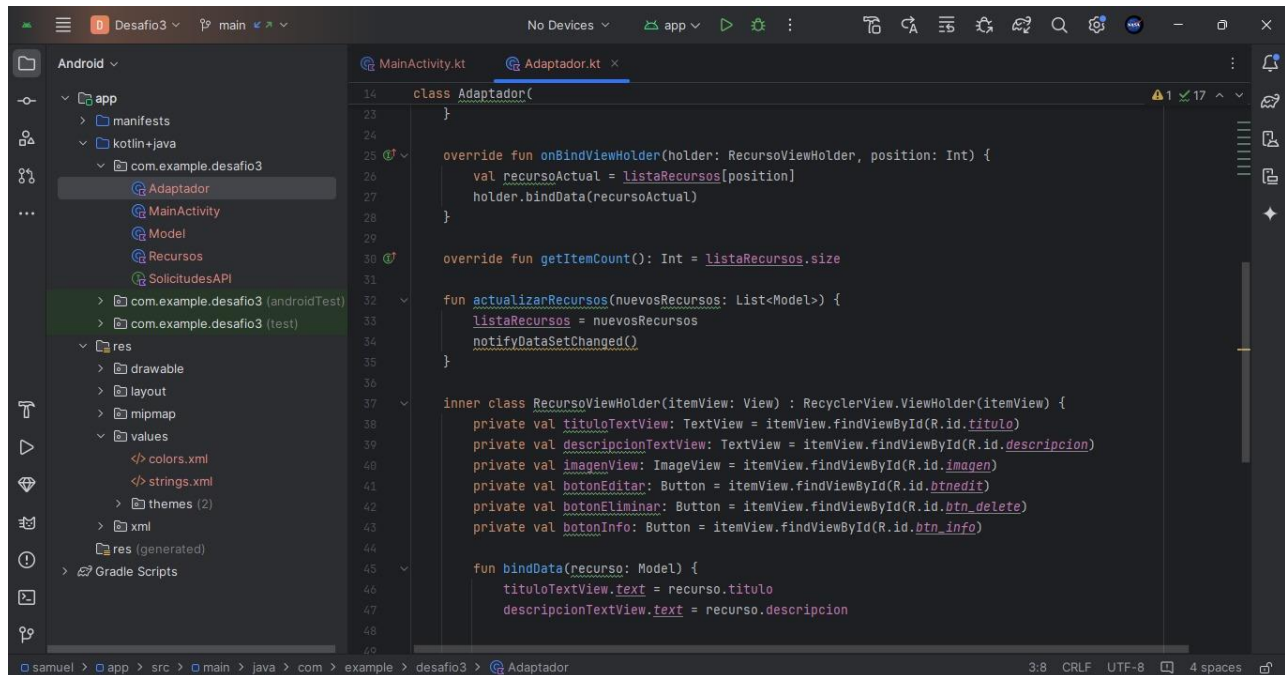
```
18 class MainActivity : AppCompatActivity() {
124 private fun removeRecurso(recurso: Model) {
128     LifecycleScope.launch {
129         try {
130             apiService.deleteRecurso(recurso.id)
131             showMessage("Recurso eliminado correctamente")
132             loadRecursos()
133         } catch (e: Exception) {
134             showMessage("Error: ${e.message}")
135         }
136     }
137 }
138
139 private fun showMessage(message: String) {
140     Toast.makeText(context, this@MainActivity, message, Toast.LENGTH_SHORT).show()
141 }
142
143 private fun handleException(exception: Exception) {
144     val message = when (exception) {
145         is IOException -> "verifica tu conexión de internet"
146         is HttpException -> "Error de la api: ${exception.code()}"
147         else -> "Error: ${exception.message}"
148     }
149     Toast.makeText(context, this@MainActivity, message, Toast.LENGTH_LONG).show()
150 }
151
152 }
```

153:1 LF UTF-8 4 spaces

# Adaptador.kt

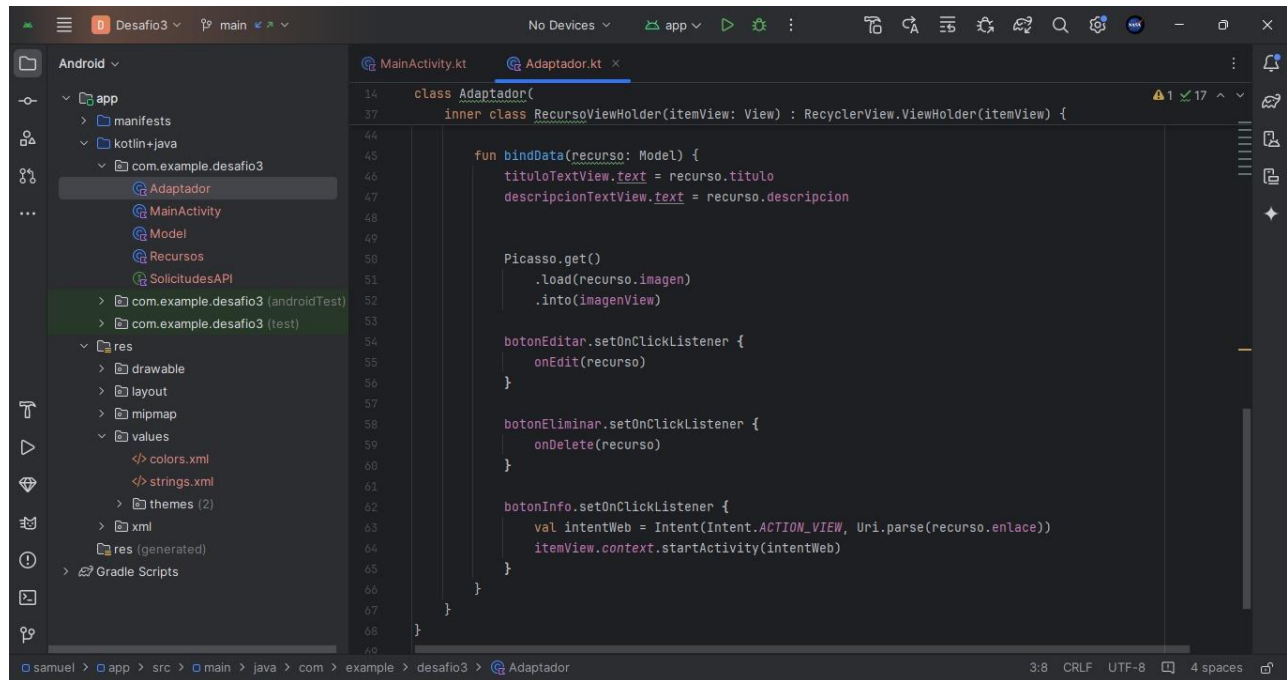


```
1 package com.example.desafio3
2
3 import android.content.Intent
4 import android.net.Uri
5 import android.view.LayoutInflater
6 import android.view.View
7 import android.view.ViewGroup
8 import android.widget.Button
9 import android.widget.ImageView
10 import android.widget.TextView
11 import androidx.recyclerview.widget.RecyclerView
12 import com.squareup.picasso.Picasso
13
14 class Adaptador(
15     private var listaRecurso: List<Model>,
16     private val onEdit: (Model) -> Unit,
17     private val onDelete: (Model) -> Unit
18 ) : RecyclerView.Adapter<Adaptador.RecursoViewHolder>() {
19
20     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecursoViewHolder {
21         val vista = LayoutInflater.from(parent.context).inflate(R.layout.item_recurso, parent, attachToRoot: false)
22         return RecursoViewHolder(vista)
23     }
24
25     override fun onBindViewHolder(holder: RecursoViewHolder, position: Int) {
26         val recursoActual = listaRecurso[position]
27         holder.bindData(recursoActual)
28     }
29 }
```

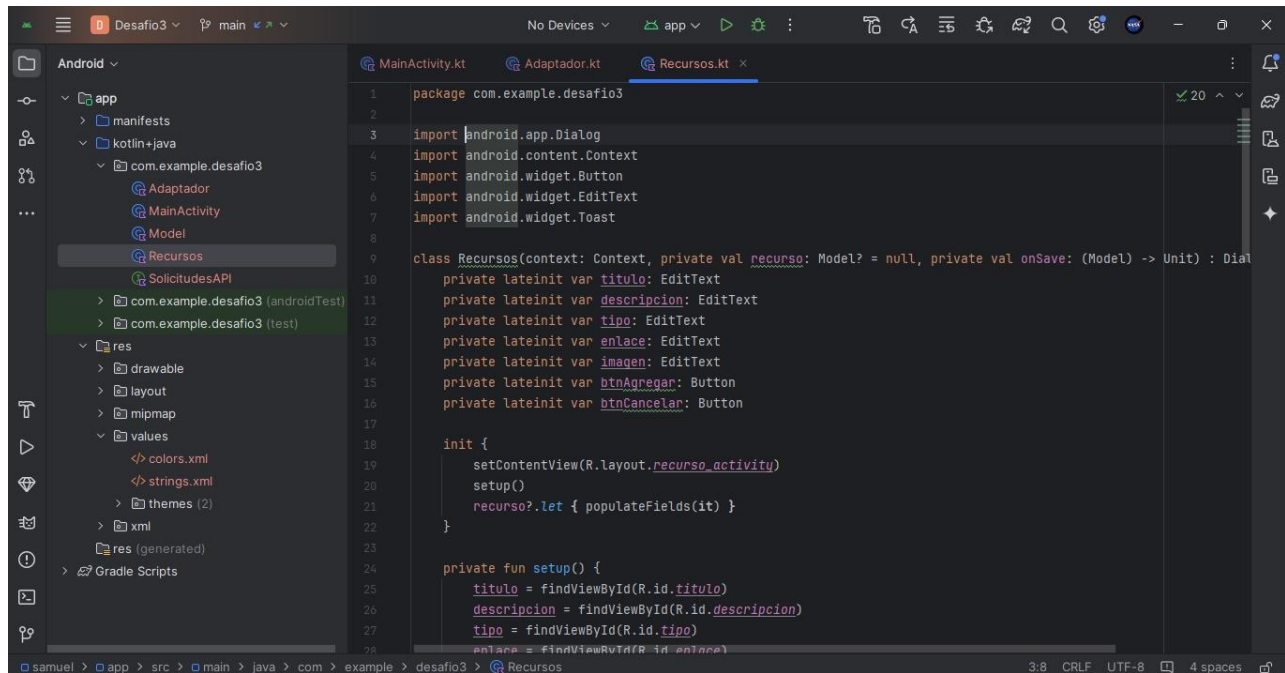


```
14 class Adaptador(
15     private var listaRecurso: List<Model>,
16     private val onEdit: (Model) -> Unit,
17     private val onDelete: (Model) -> Unit
18 ) : RecyclerView.Adapter<Adaptador.RecursoViewHolder>() {
19
20     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecursoViewHolder {
21         val vista = LayoutInflater.from(parent.context).inflate(R.layout.item_recurso, parent, attachToRoot: false)
22         return RecursoViewHolder(vista)
23     }
24
25     override fun onBindViewHolder(holder: RecursoViewHolder, position: Int) {
26         val recursoActual = listaRecurso[position]
27         holder.bindData(recursoActual)
28     }
29
30     override fun getItemCount(): Int = listaRecurso.size
31
32     fun actualizarRecurso(nuevosRecurso: List<Model>) {
33         listaRecurso = nuevosRecurso
34         notifyDataSetChanged()
35     }
36
37     inner class RecursoViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
38         private val tituloTextView: TextView = itemView.findViewById(R.id.titulo)
39         private val descripcionTextView: TextView = itemView.findViewById(R.id.descripcion)
40         private val imagenView: ImageView = itemView.findViewById(R.id.imagen)
41         private val botonEditar: Button = itemView.findViewById(R.id.btn_edit)
42         private val botonEliminar: Button = itemView.findViewById(R.id.btn_delete)
43         private val botonInfo: Button = itemView.findViewById(R.id.btn_info)
44
45         fun bindData(recurso: Model) {
46             tituloTextView.text = recurso.titulo
47             descripcionTextView.text = recurso.descripcion
48         }
49     }
50 }
```

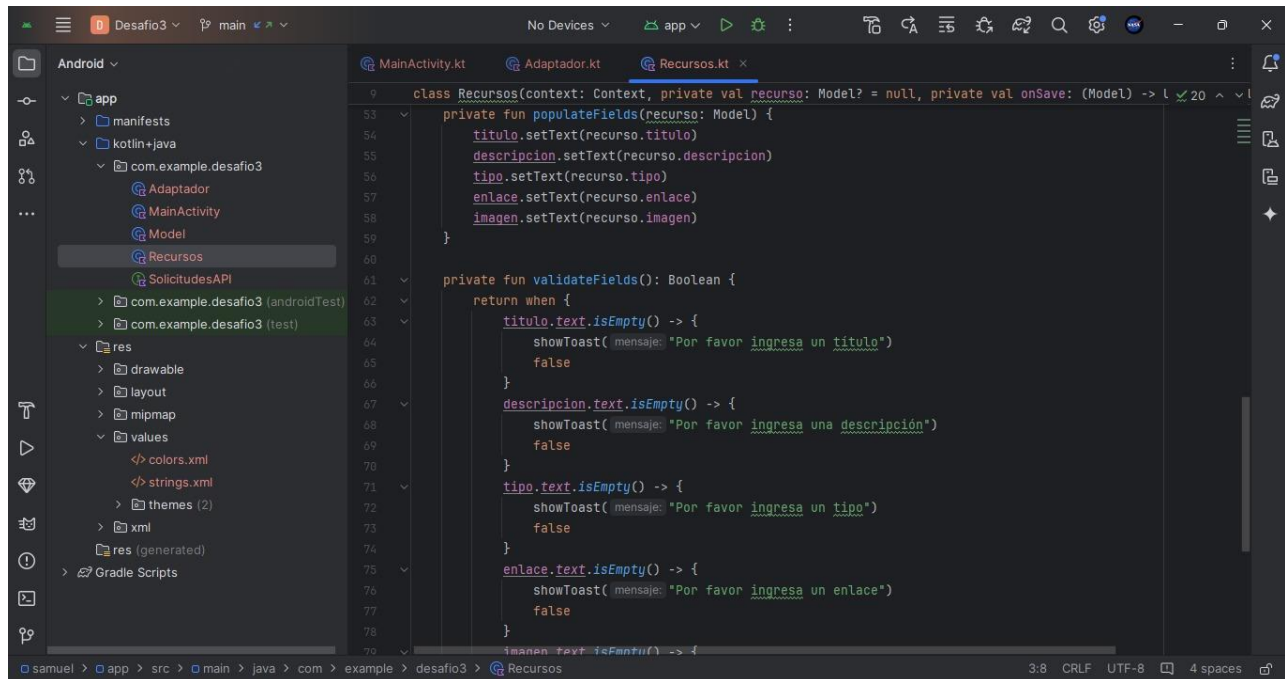




# Recursos.kt

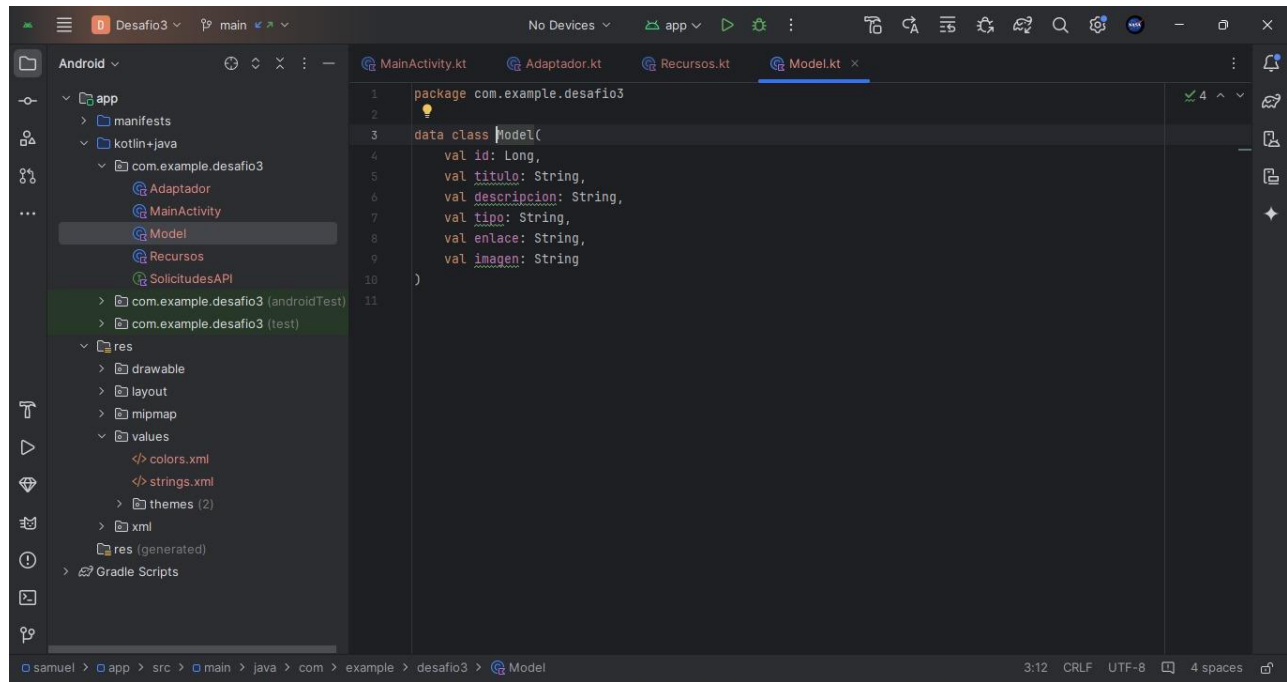


```
1 package com.example.desafio3
2
3 import android.app.Dialog
4 import android.content.Context
5 import android.widget.Button
6 import android.widget.EditText
7 import android.widget.Toast
8
9 class Recursos(context: Context, private val recurso: Model? = null, private val onSave: (Model) -> Unit) : Dialog() {
10     private lateinit var titulo: EditText
11     private lateinit var descripcion: EditText
12     private lateinit var tipo: EditText
13     private lateinit var enlace: EditText
14     private lateinit var imagen: EditText
15     private lateinit var btnAgregar: Button
16     private lateinit var btnCancelar: Button
17
18     init {
19         setContentView(R.layout.recurso_actividad)
20         setup()
21         recurso?.let { populateFields(it) }
22     }
23
24     private fun setup() {
25         titulo = findViewById(R.id.titulo)
26         descripcion = findViewById(R.id.descripcion)
27         tipo = findViewById(R.id.tipo)
28         enlace = findViewById(R.id.enlace)
29     }
30 }
```

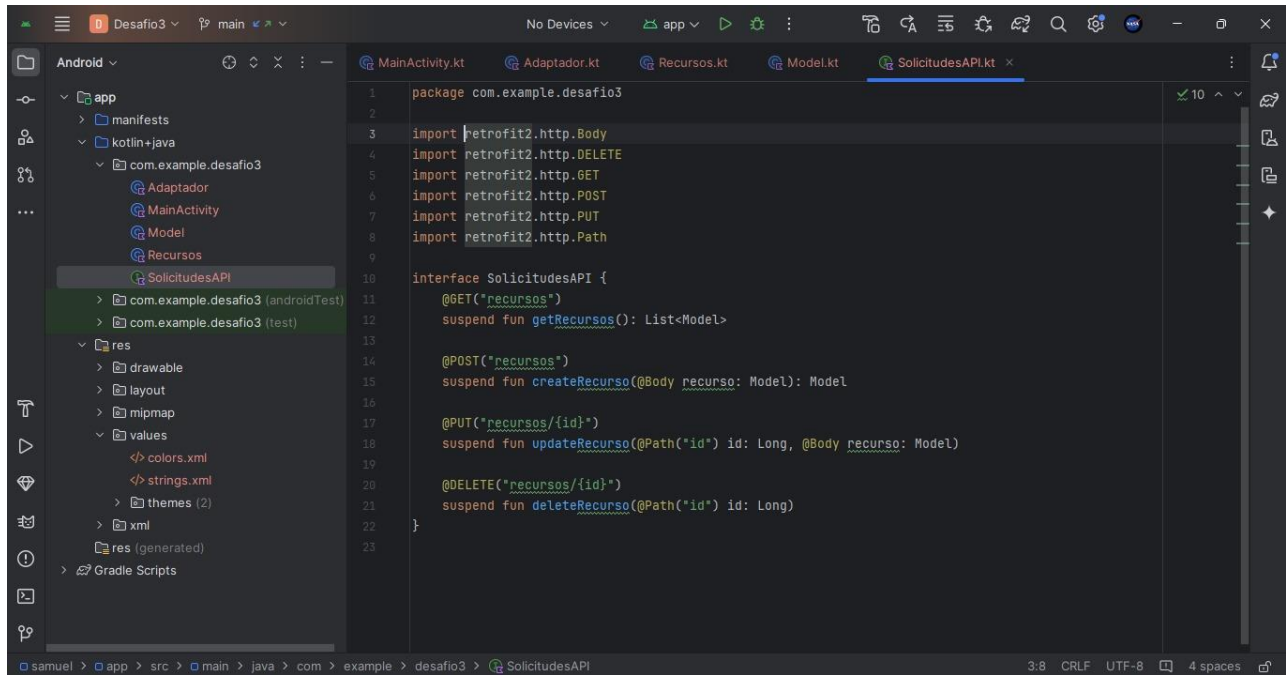


```
31 private fun populateFields(recurso: Model) {
32     titulo.setText(recurso.titulo)
33     descripcion.setText(recurso.descripcion)
34     tipo.setText(recurso.tipo)
35     enlace.setText(recurso.enlace)
36     imagen.setText(recurso.imagen)
37 }
38
39 private fun validateFields(): Boolean {
40     return when {
41         titulo.text.isEmpty() -> {
42             showToast(mensaje: "Por favor ingresa un titulo")
43             false
44         }
45         descripcion.text.isEmpty() -> {
46             showToast(mensaje: "Por favor ingresa una descripción")
47             false
48         }
49         tipo.text.isEmpty() -> {
50             showToast(mensaje: "Por favor ingresa un tipo")
51             false
52         }
53         enlace.text.isEmpty() -> {
54             showToast(mensaje: "Por favor ingresa un enlace")
55             false
56         }
57         imagen.text.isEmpty() -> {
58             showToast(mensaje: "Por favor ingresa una imagen")
59             false
60         }
61     }
62 }
```

# Model.kt



## SolicitudesAPI.kt



```
1 package com.example.desafio3
2
3 import retrofit2.http.Body
4 import retrofit2.http.DELETE
5 import retrofit2.http.GET
6 import retrofit2.http.POST
7 import retrofit2.http.PUT
8 import retrofit2.http.Path
9
10 interface SolicitudesAPI {
11     @GET("recursos")
12     suspend fun getRecursos(): List<Model>
13
14     @POST("recursos")
15     suspend fun createRecurso(@Body recurso: Model): Model
16
17     @PUT("recursos/{id}")
18     suspend fun updateRecurso(@Path("id") id: Long, @Body recurso: Model)
19
20     @DELETE("recursos/{id}")
21     suspend fun deleteRecurso(@Path("id") id: Long)
22 }
23
```

## LINK DE GITHUB.

[https://github.com/GF230979/Desafio3\\_DMS/](https://github.com/GF230979/Desafio3_DMS/)

## LINK DE VIDEO FUNCIONAL.

[https://www.youtube.com/watch?v=RxAE\\_9Q00NM](https://www.youtube.com/watch?v=RxAE_9Q00NM)