APP COMPARAR PALABRAS

CURSO DESARROLLO APP MOBILES

FERNANDO CEAGLIO

2023

Contenido

[CONSIGNA 2](#_Toc144597497)

[CODIGO 3](#_Toc144597498)

[MODEL 3](#_Toc144597499)

[VIEW (VIEW MODEL) 4](#_Toc144597500)

[VIEW (ACTIVITY) 5](#_Toc144597501)

[CAPTURAS DE PANTALLA 6](#_Toc144597502)

[TEST UNITARIO 9](#_Toc144597503)

[TEST DE UI 10](#_Toc144597504)

## CONSIGNA

Para la realización de este proyecto, te invitamos a imaginarte que sos un desarrollador recién incorporado en un nuevo emprendimiento tecnológico.

Te solicitan que desarrolles una aplicación que tenga un solo activity que cumpla con las siguientes premisas

Una única pantalla (sin importar el layout elegido) con:

2 cuadros de textos (EditText)

1 botón con el texto “comparar”

1 texto (TextView) que en el que se escriba el resultado de la acción al presionar el botón.

Asegurate de que:

- Utiliza MVVM

- Tiene al menos un test unitario

- Tiene al menos un test de UI

**Función de la app:** Cuando el usuario hace click en el botón “comparar” debe comparar la entrada de ambos cuadros de texto y escribir en el texto (TextView) si ambas cadenas de caracteres son iguales o no.

Para subir la carpeta de tu proyecto, te proponemos:

Usar un repositorio de GitHub pages haciendo la entrega del link.

En caso de tener alguna duda o tengas algún problema con la subida, te dejamos aquí un video instructivo extra que te ayudará paso a paso con GitHub Pages:

En el último modulo del curso, encontrarás el apartado "Proyecto Final" con las etapas para poder realizarlo, recomendaciones, guías y condiciones para la entrega.

¡Te recomendamos que no dejes de consultarlo!

## CODIGO

### MODEL

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

### VIEW (VIEW MODEL)

class MainViewModel: ViewModel() {  
  
 // Solo queremos que se pueda leer palabra  
 val palabra: LiveData<Words> get() = \_palabras  
 // no nos interesa que se modifique por fuera del ViewModel  
 private var \_palabras = MutableLiveData<Words>(Words("","",""))  
  
 fun compararPalabras(pal11: String,pal22: String, esta2:String) {  
  
 val pal3 = pal11  
 val pal4 = pal22  
 var estado = "sin evaluar"  
  
 if (pal3 == pal4)  
 {  
 *println*("Equal")  
 *println*("Las palabras '$pal11' y '$pal22' son iguales")  
 estado = "SON IGUALES"  
 }  
 else  
 {  
 *println*("Not Equal")  
 *println*("'$pal11' es distinta de '$pal22'")  
 *println*("Las palabras '$pal11' y '$pal22' son distintas")  
 estado = "SON DISTINTAS"  
 }  
  
 updatePalabras(pal11, pal22, estado)  
 }  
  
 private fun updatePalabras(pal1: String, pal2: String, est2:String ){  
 \_palabras.*value* = Words(pal1,pal2, est2)  
 }  
}

### VIEW (ACTIVITY)

class MainActivity : AppCompatActivity() {  
  
 private lateinit var binding: ActivityMainBinding  
 private val mainViewModel: MainViewModel by *viewModels*()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 binding = ActivityMainBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 mainViewModel.palabra.observe(this)**{** *println*("Se generan cambios en viewModel. $**it**")  
 *println*("Evalua binding text")  
 *println*("Evalua estado")  
 binding.textResultado.*text* = **it**.estado  
 **}** binding.btnComparar.setOnClickListener **{** val pepe = binding.palabra1.*text*.*trim*().toString()  
 val pepe2 = binding.palabra2.*text*.*trim*().toString()  
 binding.textResultado.*text* = binding.palabra1.*text*.toString()  
 mainViewModel.compararPalabras(pepe,pepe2,"")  
 **}** }  
}

## CAPTURAS DE PANTALLA

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

## TEST UNITARIO

El test unitario se pudo realizar en forma correcta en el VM

Tabla

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

## TEST DE UI

Este test me arroja un problema que al momento no pude encontrar como solucionar, pero se entienden el tests que más abajo se comparte esta bien diseñado.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente