

**Universidad Don Bosco**

**Facultad de Ingeniería**



**“Trabajo de investigación Sobre El Patrón MVVM”**

**Materia:**

Desarrollo de Software para Móviles G04L

**Presentado por:**

Bryan Efraín Crespín Bran

CB180936

**Docente:**

Ing. Alexander Campos

San Salvador, 29 de Abril del 2023

# Índice

|                    |    |
|--------------------|----|
| Introducción ..... | 1  |
| Contenido.....     | 2  |
| Anexos .....       | 8  |
| Bibliografía ..... | 12 |

# Introducción

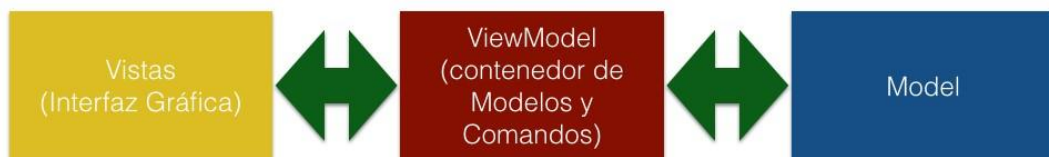
**En el presente trabajo se dará a conocer una de las estructuras de programación que es el patrón MVVM que se utiliza para darle modelo a una aplicación que está orientada a aplicaciones móviles. Se implementará este modelo en un ejemplo sencillo como demostración del patrón MVVM**

**El patrón MVVM También puede mejorar significativamente las oportunidades de reutilización del código y permite a los desarrolladores y diseñadores de interfaz de usuario colaborar más fácilmente al desarrollar sus respectivas partes de una aplicación**

# Contenido

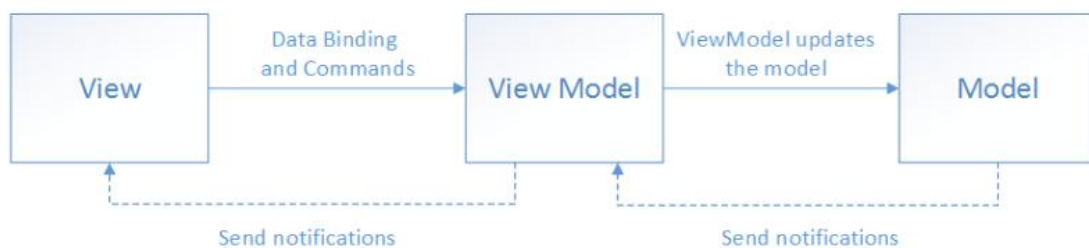
## ¿Qué es el patrón MVVM?

**El patrón MVVM ayuda a separar limpiamente la lógica de presentación y negocios de una aplicación de su interfaz de usuario (UI). Mantener una separación limpia entre la lógica de la aplicación y la interfaz de usuario ayuda a abordar numerosos problemas de desarrollo y facilita la prueba, el mantenimiento y la evolución de una aplicación. También puede mejorar significativamente las oportunidades de reutilización del código y permite a los desarrolladores y diseñadores de interfaz de usuario colaborar más fácilmente al desarrollar sus respectivas partes de una aplicación.**



# ¿Cuáles son sus componentes principales y cómo se relacionan entre sí?

Hay tres componentes principales en el patrón MVVM: el modelo, la vista y el modelo de vista. Cada uno sirve para un propósito distinto. En el diagrama siguiente se muestran las relaciones entre los tres componentes.



Los tres componentes se relacionan entre sí de la siguiente manera:

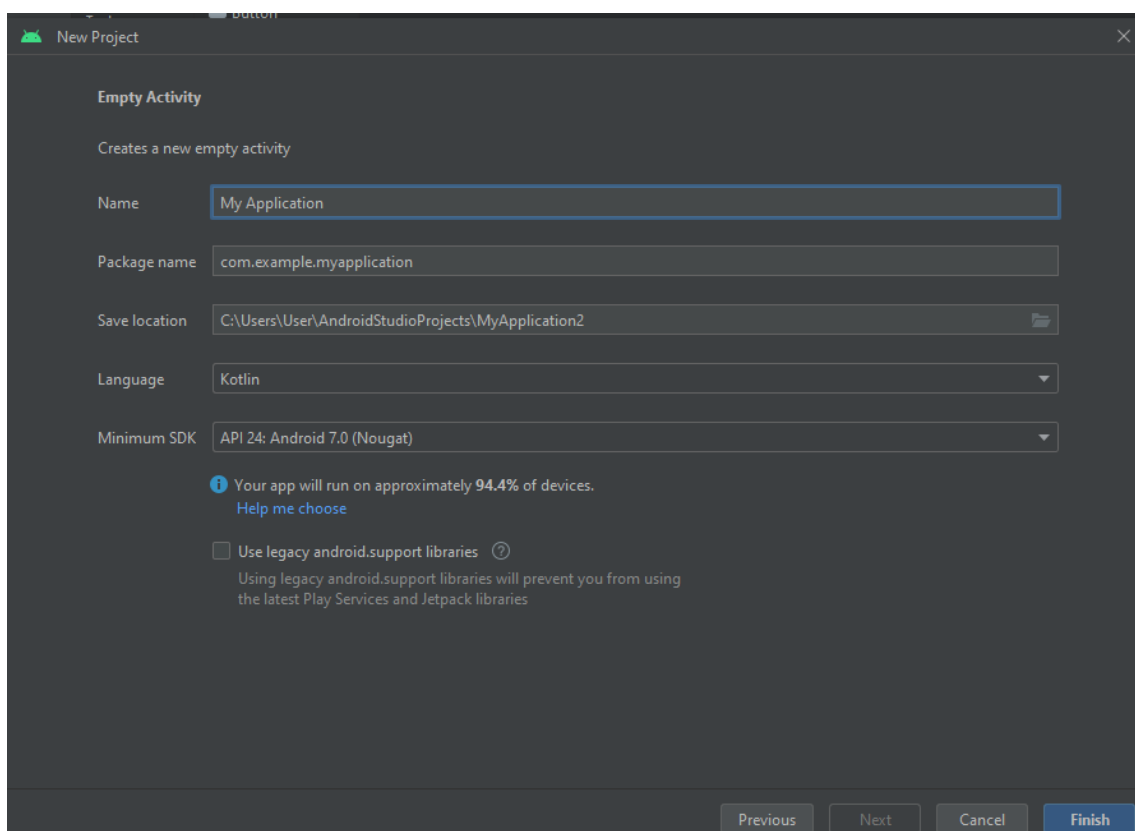
**Model:** esta es la capa en la que se establecerán los modelos para el almacenamiento de datos. **View:** se trata de la capa encargada de mostrar los datos que se han almacenado. **ViewModel:** es la capa responsable de envolver el modelo y preparar los datos observables requeridos por la vista. Cabe resaltar que los cambios en el ViewModel influyen directamente en la vista y viceversa.

# ¿Cómo se aplica el patrón MVVM en Android con Kotlin?

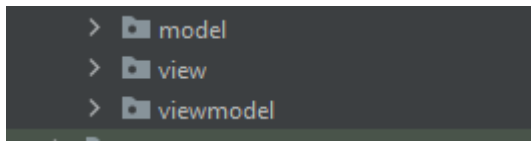
## Pasos para la implementación de MVVM en Android:

Después de comprender en qué consiste la arquitectura MVVM, podemos entrar en materia sobre los pasos que debes tener en cuenta para la implementación de MVVM en Android.

**Crear un nuevo proyecto:** en Android Studio según las necesidades de tu app, por ejemplo, el lenguaje de programación y el SDK que vas a usar.

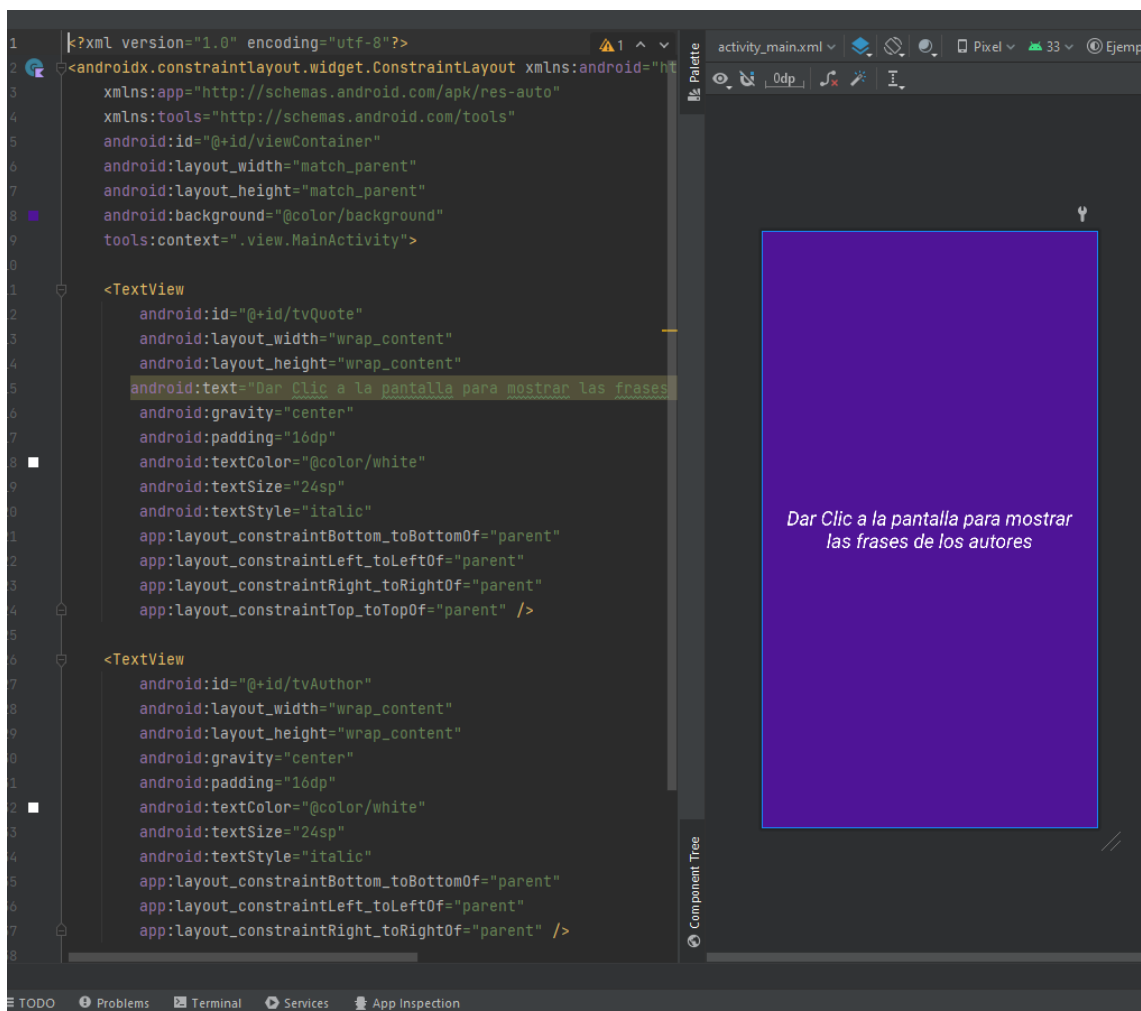


**Crear la clase Modelo:** con el fin de que el programa sepa la estructura que va a definir la presentación de los datos de tu app.



Dentro de las carpetas están las clases

**Trabajar con el archivo activity\_main.xml:** para establecer las diferentes entradas y componentes de la app.



En esta parte de aquí definimos el diseño del XML

**Crear la clase ViewModel:** para indicar los métodos que se deben llamar para el buen funcionamiento de la app.

**Clase con sus parámetros**

```
package com.example.ejemplo_mvvm_dsm104.viewmodel

import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import com.example.ejemplo_mvvm_dsm104.model.QuoteModel
import com.example.ejemplo_mvvm_dsm104.model.QuoteProvider

class QuoteViewModel : ViewModel() {

    val quoteModel = MutableLiveData<QuoteModel>()

    fun randomQuote(){
        val currentQuote = QuoteProvider.random()
        quoteModel.postValue(currentQuote)
    }
}
```

**Definir las funcionalidades de View en el fichero de MainActivity.**

```
package com.example.ejemplo_mvvm_dsm104.view

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.activity.viewModels
import androidx.lifecycle.Observer
import com.example.ejemplo_mvvm_dsm104.databinding.ActivityMainBinding
import com.example.ejemplo_mvvm_dsm104.model.QuoteModel
import com.example.ejemplo_mvvm_dsm104.viewmodel.QuoteViewModel

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    private val quoteViewModel: QuoteViewModel by viewModels()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding=ActivityMainBinding.inflate(layoutInflater)

        setContentView(binding.root)

        quoteViewModel.quoteModel.observe( owner: this, Observer { currentQuote ->

            binding.tvQuote.text=currentQuote.quote
            binding.tvAuthor.text=currentQuote.author

        })

        binding.viewContainer.setOnClickListener{quoteViewModel.randomQuote()}
    }
}
```

**Ejecutar la app.**



**En general, podemos decirte que esos son los pasos que debes tener presentes en el momento de querer trabajar con la arquitectura mvvm android. Por último, es necesario destacar que la cantidad de pasos a seguir depende de las necesidades de tu proyecto o la finalidad que este tenga, así que los que te hemos presentado hacen referencia a una guía básica de trabajo con MVVM. Ahora te queda aprender en qué casos se usa la arquitectura mvvm**

## **¿Cuáles son las ventajas y desventajas de utilizar el patrón MVVM en el desarrollo de aplicaciones móviles?**

### **Ventajas**

**La lógica de negocio está desacoplada de la interfaz de usuario.**

**Es más fácil de mantener y probar. Puedes hacer pruebas unitarias para el modelo y para vista-modelo, sin necesidad de hacer referencia a la vista.**

**Los componentes pueden ser reutilizados**

**El mantenimiento de los sistemas es simplificado.**

### **Desventajas**

**La curva de aprendizaje para nuevos desarrolladores es un poco superior a los otros modelos que son más simples.**

**La distribución de componentes nos obliga a la creación y mantenimiento de un mayor número de ficheros.**

**Debes de adaptarte a una estructura predefinida y eso incrementa la complejidad del sistema.**

# Anexos

## Ejemplo sencillo utilizando MVVM en android studio

### Funcionalidad al 100%

### Capturas de pantalla

Se muestra la pantalla inicial a la hora de ejecutar el programa.

Nos muestra un mensaje para indicarle al usuario que es lo que debe de hacer. Con el dedo le damos clic sobre la pantalla para ver las frases de los autores



Al darle clic de  
nuevo a la  
pantalla se  
muestra la frase  
y el nombre del  
autor. Y así  
sucesivamente  
con las 10 frases  
y el autor que se  
muestran  
aleatoriamente

*Parece, cuando se ama, que el  
mundo entero tiene rumor de  
primavera*

*Juan Ramón Jiménez*

*Nada es real hasta que se  
experimenta, aún un proverbio  
no lo es hasta que la vida lo haya  
ilustrado*

*John Keats*

Le damos clic  
sobre la pantalla  
para ver las  
frases y el  
nombre del  
autor

*El libro es fuerza, es valor, es  
poder, es alimento, antorcha del  
pensamiento y manantial del  
amor*

*Rubén Darío*

Le damos clic  
sobre la pantalla  
para ver las  
frases y el  
nombre del  
autor. Así  
sucesivamente  
hasta ver todas  
las frases y  
autores

# Bibliografía

<https://www.abatic.es/arquitecturas-de-software-mvc-y-mvvm/>

<https://keepcoding.io/blog/que-es-mvc-mvp-y-mvvm-en-android/>

<https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>

<https://www.youtube.com/watch?v=6hIvPCkwMds>