



Guías Prácticas N° 1

Asignatura

DESARROLLO DE APLICACIONES MÓVILES

Docente : Mg. Guillermo Peña García
Ing. Yan Casayco Contreras

Facultad : Ingeniería

Sección: _____

Fecha: _____

1. Creación de un Nuevo Proyecto en Android Studio

Ahora crearemos un proyecto base para nuestra aplicación móvil.

Pasos:


1. Abre Android Studio y haz clic en **"New Project"** (Nuevo Proyecto).
2. Selecciona **"Empty Activity"** y presiona **Next**.
3. Configura el proyecto:
 - **Name:** MiPrimeraApp
 - **Package name:** com.tuequipo.miprimeraapp
 - **Save location:** Selecciona una carpeta en tu computadora.
 - **Language:** Kotlin
4. **Minimum SDK:** API 23: Android 6.0 (Marshmallow)
 - Haz clic en **Finish** y espera a que Android Studio configure el proyecto.

2. Explorando la Estructura del Proyecto

Después de crear el proyecto, verás diferentes carpetas y archivos. Aquí te explicamos los más importantes:

 **app/src/main/java/com.tuequipo.miprimeraapp/** → Aquí se encuentra el código de la aplicación.

 **app/src/main/res/layout/activity_main.xml** → Define la interfaz de usuario de la actividad principal.

 **AndroidManifest.xml** → Contiene la configuración de permisos y actividades de la app.

Tómate unos minutos para explorar la estructura y familiarizarte con el entorno.

3. Modificación de la Interfaz de Usuario

Vamos a modificar la pantalla principal para que muestre un mensaje de bienvenida.

Pasos:

1. Abre el archivo **activity_main.xml** en la carpeta **res/layout/**.
2. Reemplaza el código existente con lo siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:id="@+id/txtMensaje"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="¡Bienvenido a MiPrimeraApp!"
        android:textSize="24sp"
        android:textColor="@android:color/black"/>

</LinearLayout>
```

3. Guarda los cambios presionando **Ctrl + S** (Windows) o **Cmd + S** (Mac).

4. Ejecutando la Aplicación en un Emulador

Vamos a probar nuestra aplicación en un dispositivo virtual.

Pasos:

1. En la barra superior, haz clic en **Device Manager**.
2. Presiona **Create Device** y selecciona un modelo de teléfono.
3. Elige una versión de Android y descárgala si es necesario.
4. Configura el emulador y haz clic en **Finish**.
5. Regresa a Android Studio y presiona **Run** (🔴▶) para ejecutar la aplicación.

Si todo está bien, verás la pantalla con el mensaje "**¡Bienvenido a MiPrimeraApp!**" en el emulador.

5. Nueva Función: Botón para Cambiar Texto

Vamos a agregar un botón que, al presionarlo, cambie el mensaje en TextView.

Pasos:

1. Modificar `activity_main.xml`

- Abre `res/layout/activity_main.xml`.
- Agrega un botón debajo del TextView:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:id="@+id/txtMensaje"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="¡Bienvenido a MiPrimeraApp!"
        android:textSize="24sp"
        android:textColor="@android:color/black"
        android:padding="16dp"/>

    <Button
        android:id="@+id/btnCambiarTexto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cambiar Mensaje"/>

</LinearLayout>
```

2. Modificar MainActivity.kt

- Abre MainActivity.kt y agrega el código para cambiar el texto al presionar el botón:

```
package com.tuequipo.miprimeraapp

import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

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

        val txtMensaje = findViewById<TextView>(R.id.txtMensaje)
        val btnCambiarTexto = findViewById<Button>(R.id.btnCambiarTexto)

        btnCambiarTexto.setOnClickListener {
            txtMensaje.text = "¡Texto cambiado con éxito!"
        }
    }
}
```

3. Ejecutar la aplicación

- Presiona **Run** (▶) y prueba la funcionalidad.
- Cuando presiones el botón, el texto del TextView cambiará.

6. Ejercicio Adicional

Desafío 1: Personaliza el botón

- **Tarea:** Modifica el botón para que cambie su propio texto cuando es presionado.
- **Pista:** Usa `setText()` para cambiar el texto del botón dentro del `setOnClickListener()`.

Desafío 2: Cambia el color del texto dinámicamente

- **Tarea:** Al presionar el botón, además de cambiar el texto, el color del TextView debe cambiar a rojo.
- **Pista:** Usa `txtMensaje.setTextColor()` y `Color.RED`.

```
import android.graphics.Color
...
btnCambiarTexto.setOnClickListener {
    txtMensaje.text = "¡Texto cambiado con éxito!"
    txtMensaje.setTextColor(Color.RED)
}
```

Desafío 3: Agrega un segundo botón

- **Tarea:** Agrega un segundo botón que, cuando se presione, cambie el mensaje a "¡Hola, [tu nombre]!".
- **Pista:** Sigue el mismo proceso que el primer botón, pero con un nuevo Button.

Desafío 4: Añadir un contador de clics

- **Tarea:** Muestra en pantalla cuántas veces se ha presionado el botón.
- **Pista:** Usa una variable contador y actualízala en cada `setOnClickListener()`.

```
var contador = 0
btnCambiarTexto.setOnClickListener {
    contador++
    txtMensaje.text = "Has presionado el botón $contador veces."
}
```