

Universidad Politécnica de Chiapas

Ingeniería en Tecnologías de la Información e Innovación Digital

[Programación para Móviles]

[C1 – A2 - Mi primera App]

[Nomenclatura del nombre de archivo: C1 – A2 - Mi primera App-223216-DanielPeregrinoPerez.pdf]

[Alumno – Peregrino Pérez Daniel] - [223216]

Docente: [José Alonso Macias Montoya]

Fecha de entrega: [07/05/2025]



1. DESCRIPCIÓN DE LA ACTIVIDAD

1.1. Enunciado del problema

Ejecutar la primera aplicación de Android Studio en un dispositivo móvil, desarrollando una app simple de gestión de nombre de usuario utilizando Jetpack Compose.

1.2. Objetivos de aprendizaje

Definición clara

- Ejecutar la primera aplicación de Android Studio en un dispositivo móvil

Objetivos específicos

- Aprender a configurar un proyecto básico en Android Studio
- Comprender los fundamentos de Jetpack Compose para desarrollo de interfaces
- Implementar navegación entre pantallas en una aplicación Android

2. FUNDAMENTOS TEÓRICOS

Jetpack Compose es un moderno toolkit de desarrollo de interfaces para Android que simplifica y agiliza la creación de interfaces de usuario nativas. Utiliza un enfoque declarativo, permitiendo a los desarrolladores describir su interfaz de usuario mediante funciones componibles, lo que reduce significativamente la cantidad de código necesario en comparación con los métodos tradicionales de XML.

La arquitectura MVVM (Model-View-ViewModel) implementada en esta aplicación permite una separación clara de responsabilidades, facilitando el mantenimiento y la escalabilidad del código. El uso de ViewModel para gestionar el estado de la aplicación contribuye a una gestión más eficiente de los datos y la lógica de la aplicación.

3. DESARROLLO DE LA ACTIVIDAD

3.1. Desarrollo

Creación de un nuevo proyecto en Android Studio

- Configuración de Jetpack Compose como framework de interfaz
- Implementación de dos pantallas: Pantalla de Inicio y Pantalla de Perfil
- Desarrollo de navegación entre pantallas
- Gestión del estado del nombre de usuario mediante ViewModel

3.3. Implementación (opcional)

```
package com.example.miprimeraapp
```

```
import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.layout.*
```

```
import androidx.compose.material3.*
```

```
import androidx.compose.runtime.*
```

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.lifecycle.ViewModel

import androidx.lifecycle.viewmodel.compose.viewModel

import androidx.navigation.NavController

import androidx.navigation.NavHostController

import androidx.navigation.compose.NavHost

import androidx.navigation.compose.composable

import androidx.navigation.compose.rememberNavController

```
// Punto de entrada principal de la app
```

```
class MainActivity : ComponentActivity() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContent {
```

```
            MyApp()
```

```
        }
```

```
    }
```

```
}
```

```
// Función raíz de la aplicación
```

```
@Composable
```

```
fun MyApp() {

    val navController = rememberNavController()

    // Controlador de navegación que define las pantallas disponibles

    NavHost(navController = navController, startDestination = "home") {

        composable("home") { HomeScreen(navController) }

        composable("profile") { ProfileScreen(navController) }

    }

}

// ViewModel para manejar el estado del nombre

class UserViewModel : ViewModel() {

    // Estado mutable que contiene el nombre del usuario
```

```
var userName by mutableStateOf("Estudiante")
```

```
private set
```

```
fun updateName(newName: String) {
```

```
    userName = newName
```

```
}
```

```
}
```

```
// Pantalla principal
```

```
@Composable
```

```
fun HomeScreen(navController: NavController, viewModel: UserViewModel = viewModel()) {
```

Column(

modifier = Modifier

.fillMaxSize()

.padding(16.dp),

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center

) {

Text("Hola, \${viewModel.userName}!", fontSize = 24.sp, fontWeight = FontWeight.Bold)

Spacer(modifier = Modifier.height(16.dp))

Button(onClick = { navController.navigate("profile") }) {

Text("Ir al perfil")

}


```
}
```

```
}
```

```
// Pantalla de perfil donde el usuario puede editar su nombre
```

```
@Composable
```

```
fun ProfileScreen(navController: NavController, viewModel: UserViewModel = viewModel()) {
```

```
    var text by remember { mutableStateOf(viewModel.userName) }
```

```
Collapse
```

```
Column(
```

```
    modifier = Modifier
```

```
.fillMaxSize()
```

```
.padding(16.dp),
```

```
verticalArrangement = Arrangement.Center,
```

```
horizontalAlignment = Alignment.CenterHorizontally
```

```
) {
```

```
OutlinedTextField(
```

```
    value = text,
```

```
    onChange = { text = it },
```

```
    label = { Text("Tu nombre") }
```

```
)
```

```
Spacer(modifier = Modifier.height(16.dp))
```

```
Button(onClick = {
```

```
viewModel.updateName(text)
```

```
navController.popBackStack()
```

```
}) {
```

```
Text("Guardar")
```

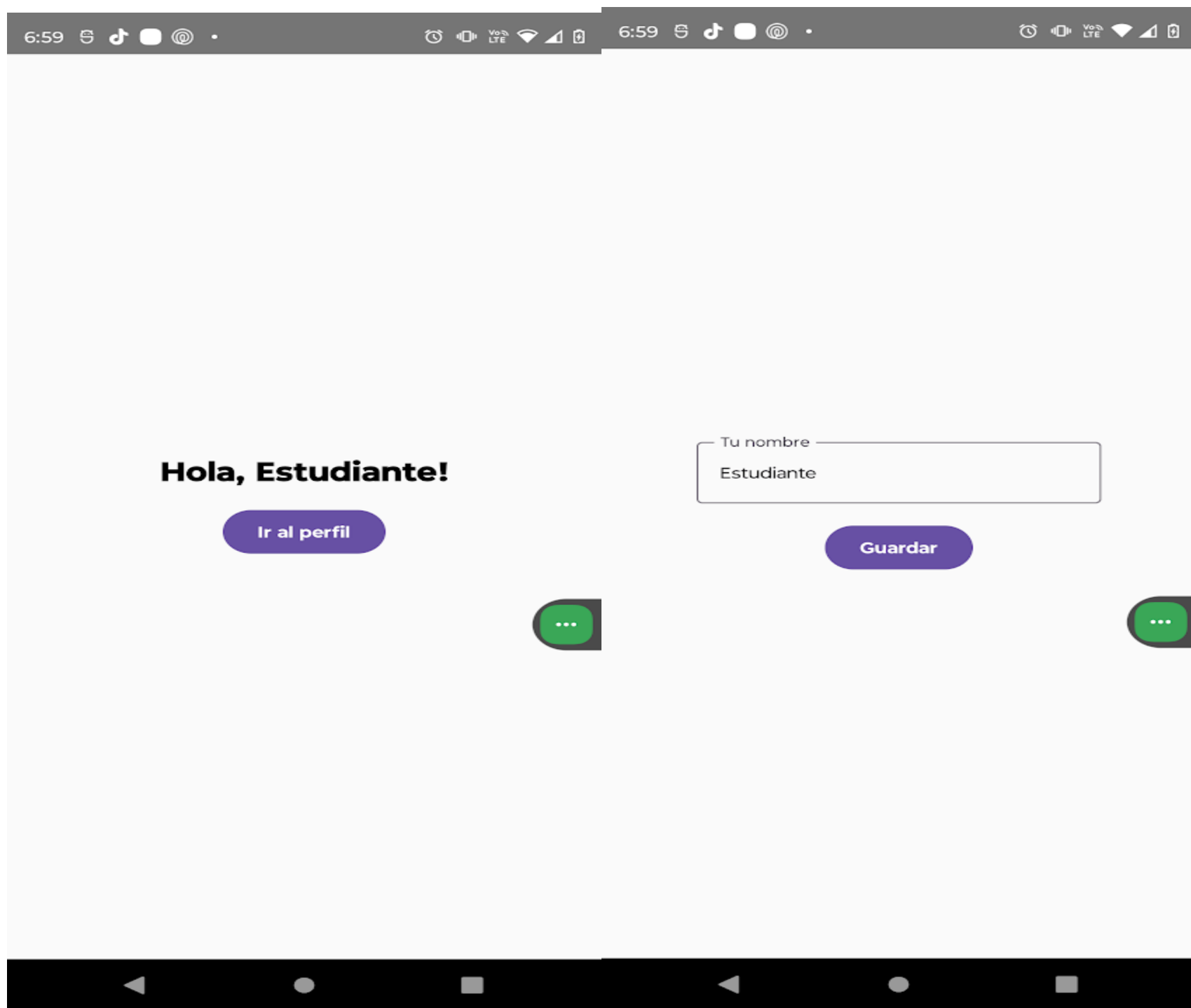
```
}
```

```
}
```

```
}
```

4. RESULTADOS

4.1. Resultados obtenidos



- **Aplicación funcional con dos pantallas**
- **Capacidad de editar y guardar nombre de usuario**
- **Navegación fluida entre pantallas**
- **Implementación exitosa de Jetpack Compose**

5. CONCLUSIONES

Aprendizaje:

- Adquirí conocimientos fundamentales sobre desarrollo móvil con Kotlin
- Comprendí los conceptos básicos de Jetpack Compose
- Desarrollé habilidades de programación en entorno móvil

Objetivos:

- Objetivos de aprendizaje completados satisfactoriamente
- Implementación exitosa de una aplicación funcional

Aplicación:

- Base sólida para futuros proyectos de desarrollo móvil
- Comprensión de arquitectura de aplicaciones Android

6. DIFICULTADES Y SOLUCIONES

Problema: Configuración inicial del entorno de desarrollo

Causas: Configuraciones de Android Studio y dependencias

Solución: Actualización de SDK, configuración de versiones compatibles

Verificación: Ejecución exitosa de la aplicación en dispositivo móvil

7. REFERENCIAS

Google. (s.f.). [Título de la clase o asignación]. Google Classroom. Recuperado de <https://classroom.google.com/u/2/c/Nzc1ODc3MDEEx/m/Nzc4NDI1MjQxMjgy/details>

RÚBRICA DE AUTOEVALUACIÓN

Criterio | Nivel alcanzado | Justificación

Comprensión del problema | Intermedio | Implementación completa de requisitos

Diseño de la solución | Intermedio | Arquitectura MVVM y navegación implementada

Implementación | Intermedio | Código funcional con Jetpack Compose

Pruebas realizadas | Básico | Pruebas básicas de funcionalidad

Documentación | Intermedio | Documentación detallada del proceso