



UNIVERSIDAD TECNOLÓGICA DE LEÓN

Ingeniería en Desarrollo y Gestión de Software

Desarrollo para dispositivos inteligentes

Zoodiacochino

presenta:

Claudio Plasencia Luis Giovanni

IDGS903

Fecha: 03/07/2025

1. FormRegistro

```
Column(
    modifier = Modifier
        .padding(16.dp)
        .verticalScroll(rememberScrollState())
) {
    Text("Datos Personales", style = MaterialTheme.typography.titleLarge)
    OutlinedTextField(value = nombre, onValueChange = { nombre = it }, label = { Text("Nombre") })
    OutlinedTextField(value = apaterno, onValueChange = { apaterno = it }, label = { Text("Apellido Paterno") })
    OutlinedTextField(value = amaterno, onValueChange = { amaterno = it }, label = { Text("Apellido Materno") })
    Text("Fecha de Nacimiento:")
    Row {
        OutlinedTextField(value = dia, onValueChange = { dia = it }, label = { Text("Dia") }, modifier = Modifier.weight(1f))
        Spacer(modifier = Modifier.width(8.dp))
        OutlinedTextField(value = mes, onValueChange = { mes = it }, label = { Text("Mes") }, modifier = Modifier.weight(1f))
        Spacer(modifier = Modifier.width(8.dp))
        OutlinedTextField(value = anio, onValueChange = { anio = it }, label = { Text("Año") }, modifier = Modifier.weight(1f))
    }
    Text("Sexo:")
    Row {
        RadioButton(selected = sexo == "Masculino", onClick = { sexo = "Masculino" })
        Text("Masculino")
        Spacer(modifier = Modifier.width(16.dp))
        RadioButton(selected = sexo == "Femenino", onClick = { sexo = "Femenino" })
        Text("Femenino")
    }
    //Form de registro de datos

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

    Row {
        Button(onClick = {
            nombre = ""; apaterno = ""; amaterno = ""; dia = ""; mes = ""; anio = ""; sexo = "Masculino"
        }) { Text("Limpiar") }

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

        Button(onClick = {
            navController.navigate("exam?name=${nombre}%20${apaterno}%20${amaterno}&birthYear=${anio}")
        }) { Text("Siguiente") } //botones para la navegacion de la pagina
    }
}
```

2. FormExamen

```
@Composable
fun ExamScreen(navController: NavController, fullName: String, birthYear: Int) {
    val preguntas = listOf(
        Triple("¿Cuál es la suma de 2 + 2?", listOf("8", "6", "4", "3"), 2),
        Triple("¿Cuál es la capital de México?", listOf("Lima", "Bogotá", "CDMX", "Quito"), 2),
        Triple("¿Cuántos días tiene una semana?", listOf("5", "6", "7", "8"), 2),
        Triple("¿Cuál es el resultado de 3 * 3?", listOf("6", "9", "12", "15"), 1),
        Triple("¿Cuál es el color del cielo?", listOf("Rojo", "Azul", "Verde", "Amarillo"), 1),
        Triple("¿Cuántos meses tiene un año?", listOf("10", "12", "11", "13"), 1)
    ) //Lista de las preguntas

    val respuestas = remember { mutableStateListOf(-1, -1, -1, -1, -1, -1) }

    Column(
        modifier = Modifier
            .padding(16.dp)
            .verticalScroll(rememberScrollState())
    ) {
        Text("Examen", style = MaterialTheme.typography.titleLarge)
        Spacer(modifier = Modifier.height(16.dp))

        preguntas.forEachIndexed { index, triple ->
            val (pregunta, opciones, correcta) = triple
            Text("${index + 1}. $pregunta", style = MaterialTheme.typography.bodyLarge)
            opciones.forEachIndexed { i, opcion ->
                Row(verticalAlignment = Alignment.CenterVertically) {
                    RadioButton(
                        selected = respuestas[index] == i,
                        onClick = { respuestas[index] = i }
                    )
                    Text(opcion)
                }
            }
            Spacer(modifier = Modifier.height(8.dp))
        } //proceso de sumatoria de puntos de preguntas

        Spacer(modifier = Modifier.height(16.dp))
        Button(onClick = {
            val score = preguntas.indices.count { respuestas[it] == preguntas[it].third }
            navController.navigate("result/${fullName}/${birthYear}/${score}")
        }) {
            Text("Terminar")
        }
    }
}
```

3. Resultado

```
@Composable
fun ResultScreen(nombre: String, anioNacimiento: Int, calificacion: Int) {
    val context = LocalContext.current
    val edad = 2025 - anioNacimiento
    val signos = listOf(
        "mono", "gallo", "perro", "cerdo", "rata", "buey",
        "tigre", "conejo", "dragon", "serpiente", "caballo", "cabra"
    ) //Apartado que define que animal eres del horoscopo chino
    val signo = signos[anioNacimiento % 12]

    val imagenId = when (signo) {
        "rata" -> R.drawable.rata
        "buey" -> R.drawable.buey
        "tigre" -> R.drawable.tigre
        "conejo" -> R.drawable.conejo
        "dragon" -> R.drawable.dragon
        "serpiente" -> R.drawable.serpiente
        "caballo" -> R.drawable.caballo
        "cabra" -> R.drawable.cabra
        "mono" -> R.drawable.mono
        "gallo" -> R.drawable.gallo
        "perro" -> R.drawable.perro
        "cerdo" -> R.drawable.cerdo
        else -> R.drawable.rata
    } //Segun el animal, te colocara la imagen de tu signo

    LaunchedEffect(Unit) {
        val contenido = """
            Nombre: $nombre
            Año de nacimiento: $anioNacimiento
            Edad: $edad
            Signo: ${signo.capitalize()}
            Calificación: $calificacion
            -----
        """.trimIndent()

        guardarEnArchivo(context, contenido)
    } //Mensaje completo con datos, signo e imagen
```