



UNIVERSIDAD TECNOLÓGICA DE LEÓN

INGENIERIA EN DESARROLLO Y GESTIÓN DE SOFTWARE

DESARROLLO PARA DISPOSITIVOS INTELIGENTES

Examen 2do Parcial

presenta:

Miranda Ramírez Viviana

IDGS903

Fecha: 03 /07/2025

MainActivity

```
package com.viviana.zodiacochino
```

```
import android.os.Build
import androidx.annotation.RequiresApi
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.navigation.NavHostController
import com.viviana.zodiacochino.R
import java.time.LocalDate
```

```
@RequiresApi(Build.VERSION_CODES.O)
```

```
@Composable
```

```
fun PantallaResultado(navController: NavHostController) {
    val context = LocalContext.current
    var resultadoTexto by remember { mutableStateOf("") }
    var signo by remember { mutableStateOf("") }
    var imagenZodiacold by remember { mutableStateOf<Int?>(null) }
```

```
    LaunchedEffect(Unit) {
```

```
        val usuario =
```

```
        context.openFileInput("usuario.txt").bufferedReader().readText().split("|")
```

```
        val nombre = usuario[0]
```

```
        val dia = usuario[1].toInt()
```

```
val mes = usuario[2].toInt()
```

```
val anio = usuario[3].toInt()
```

```
val edad = calcularEdad(anio, mes, dia)
```

```
signo = calcularSignoChino(anio)
```

```
val calificacion =
```

```
context.openFileInput("resultado.txt").bufferedReader().readText()
```

```
resultadoTexto = "Hola $nombre\nTienes $edad años y tu signo zodiacal  
chino es $signo\nCalificación: $calificacion"
```

```
imagenZodiacal = when (signo.lowercase()) {
```

```
    "rata" -> R.drawable.rata
```

```
    "buey" -> R.drawable.buey
```

```
    "tigre" -> R.drawable.tigre
```

```
    "conejo" -> R.drawable.conejo
```

```
    "dragón", "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 -> null
```

```
}
```

```
}
```

```
Column(modifier = Modifier.padding(16.dp)) {
```

```
    Text(resultadoTexto)
```

```

        Spacer(modifier = Modifier.height(16.dp))
        imagenZodiacold?.let {
            Image(
                painter = painterResource(id = it),
                contentDescription = "Signo Chino",
                modifier = Modifier.size(200.dp)
            )
        }
    }
}

```

```

@RequiresApi(Build.VERSION_CODES.O)
fun calcularEdad(anio: Int, mes: Int, dia: Int): Int {
    val today = LocalDate.now()
    val birthDate = LocalDate.of(anio, mes, dia)
    return today.year - birthDate.year - if (today.dayOfYear < birthDate.dayOfYear) 1
    else 0
}

```

```

fun calcularSignoChino(anio: Int): String {
    val signos = listOf(
        "Rata", "Buey", "Tigre", "Conejo", "Dragón", "Serpiente",
        "Caballo", "Cabra", "Mono", "Gallo", "Perro", "Cerdo"
    )
    return signos[(anio - 1900) % 12]
}

```

PantallaExamen

```
package com.viviana.zodiacochino
```

```
import android.content.Context
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.unit.dp
import androidx.navigation.NavHostController
import java.io.File
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.ui.Alignment
```

```
data class Pregunta(val texto: String, val opciones: List<String>, val correcta: Int)
```

```
val preguntas = listOf(
    Pregunta("¿Capital de Canadá?", listOf("Toronto", "Vancouver", "Ottawa",
    "Montreal"), 2),
    Pregunta("¿Cuál es el océano más grande?", listOf("Atlántico", "Índico",
    "Pacífico", "Ártico"), 2),
    Pregunta("¿Quién escribió 'Cien años de soledad'?", listOf("Borges",
    "Cervantes", "García Márquez", "Neruda"), 2),
    Pregunta("¿Elemento con símbolo 'O'?", listOf("Oro", "Osmio", "Oxígeno",
    "Plomo"), 2),
    Pregunta("¿Cuántos huesos tiene el cuerpo humano adulto?", listOf("206",
    "210", "250", "180"), 0),
    Pregunta("¿Cuál es el planeta más grande?", listOf("Tierra", "Saturno", "Júpiter",
    "Marte"), 2)
)
```

@Composable

```
fun PantallaExamen(navController: NavHostController) {
```

```
    val context = LocalContext.current
```

```
    val respuestas = remember { mutableStateListOf<Int?>().apply {  
repeat(preguntas.size) { add(null) } } }
```

```
    val scrollState = rememberScrollState()
```

```
    Column(
```

```
        modifier = Modifier
```

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

```
            .verticalScroll(scrollState)
```

```
    ) {
```

```
        preguntas.forEachIndexed { index, pregunta ->
```

```
            Text("${index + 1}. ${pregunta.texto}", style =
```

```
MaterialTheme.typography.titleMedium)
```

```
            Spacer(modifier = Modifier.height(4.dp))
```

```
            pregunta.opciones.forEachIndexed { i, opcion ->
```

```
                Row(verticalAlignment = Alignment.CenterVertically) {
```

```
                    RadioButton(
```

```
                        selected = respuestas[index] == i,
```

```
                        onClick = { respuestas[index] = i }
```

```
                    )
```

```
                    Text(opcion)
```

```
                }
```

```
            }
```

```
            Spacer(modifier = Modifier.height(16.dp)) // separación entre preguntas
```

```
    }
```

```
    Button(onClick = {
```

```
        val score = respuestas.mapIndexed { i, r -> if (r == preguntas[i].correcta) 1
```

```

else 0 }.sum()
        context.openFileOutput("resultado.txt", Context.MODE_PRIVATE).use {
            it.write(score.toString().toByteArray())
        }
        navController.navigate("resultado")
    }) {
        Text("Terminar")
    }

    Spacer(modifier = Modifier.height(32.dp))
}
}

```

PantallaFormulario

```

package com.viviana.zodiacochino

import android.content.Context
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.unit.dp
import androidx.navigation.NavHostController
import java.io.File

@Composable
fun PantallaFormulario(navController: NavHostController) {
    val context = LocalContext.current

```

```
var nombre by remember { mutableStateOf("") }
var dia by remember { mutableStateOf("") }
var mes by remember { mutableStateOf("") }
var anio by remember { mutableStateOf("") }
var sexo by remember { mutableStateOf("Masculino") }
```

```
Column(Modifier.padding(16.dp)) {
    TextField(value = nombre, onValueChange = { nombre = it }, label = {
Text("Nombre Completo") })
    Row {
        TextField(value = dia, onValueChange = { dia = it }, label = { Text("Día") },
modifier = Modifier.weight(1f),
        keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Number))
        Spacer(modifier = Modifier.width(8.dp))
        TextField(value = mes, onValueChange = { mes = it }, label = { Text("Mes")
}, modifier = Modifier.weight(1f),
        keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Number))
        Spacer(modifier = Modifier.width(8.dp))
        TextField(value = anio, onValueChange = { anio = it }, label = { Text("Año")
}, modifier = Modifier.weight(1f),
        keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Number))
    }
    Row {
        RadioButton(selected = sexo == "Masculino", onClick = { sexo =
"Masculino" })
        Text("Masculino")
        Spacer(modifier = Modifier.width(8.dp))
        RadioButton(selected = sexo == "Femenino", onClick = { sexo =
```



```

"Femenino" })
    Text("Femenino")
}

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

    Spacer(modifier = Modifier.width(8.dp))

    Button(onClick = {
        guardarDatos(context, nombre, dia, mes, anio, sexo)
        navController.navigate("examen")
    }) {
        Text("Siguiente")
    }
}
}
}
}

```

```

fun guardarDatos(context: Context, nombre: String, dia: String, mes: String, anio:
String, sexo: String) {
    val texto = "$nombre|$dia|$mes|$anio|$sexo"
    context.openFileOutput("usuario.txt", Context.MODE_PRIVATE).use {
        it.write(texto.toByteArray())
    }
}
}

```

PantallaResultado

```
package com.viviana.zodiacochino
```

```
import android.os.Build
import androidx.annotation.RequiresApi
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.navigation.NavHostController
import com.viviana.zodiacochino.R
import java.time.LocalDate
```

```
@RequiresApi(Build.VERSION_CODES.O)
```

```
@Composable
```

```
fun PantallaResultado(navController: NavHostController) {
    val context = LocalContext.current
    var resultadoTexto by remember { mutableStateOf("") }
    var signo by remember { mutableStateOf("") }
    var imagenZodiacold by remember { mutableStateOf<Int?>(null) }
```

```
    LaunchedEffect(Unit) {
```

```
        val usuario =
```

```
context.openFileInput("usuario.txt").bufferedReader().readText().split("|")
```

```
        val nombre = usuario[0]
```

```
        val dia = usuario[1].toInt()
```

```
        val mes = usuario[2].toInt()
```

```
        val anio = usuario[3].toInt()
```

```
val edad = calcularEdad(anio, mes, dia)
```

```
signo = calcularSignoChino(anio)
```

```
val calificacion =
```

```
context.openFileInput("resultado.txt").bufferedReader().readText()
```

```
resultadoTexto = "Hola $nombre\nTienes $edad años y tu signo zodiacal  
chino es $signo\nCalificación: $calificacion"
```

```
imagenZodiacold = when (signo.lowercase()) {
```

```
    "rata" -> R.drawable.rata
```

```
    "buey" -> R.drawable.buey
```

```
    "tigre" -> R.drawable.tigre
```

```
    "conejo" -> R.drawable.conejo
```

```
    "dragón", "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 -> null
```

```
}
```

```
}
```

```
Column(modifier = Modifier.padding(16.dp)) {
```

```
    Text(resultadoTexto)
```

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

```
    imagenZodiacold?.let {
```

```

        Image(
            painter = painterResource(id = it),
            contentDescription = "Signo Chino",
            modifier = Modifier.size(200.dp)
        )
    }
}
}

```

```

@RequiresApi(Build.VERSION_CODES.O)
fun calcularEdad(anio: Int, mes: Int, dia: Int): Int {
    val today = LocalDate.now()
    val birthDate = LocalDate.of(anio, mes, dia)
    return today.year - birthDate.year - if (today.dayOfYear < birthDate.dayOfYear) 1
    else 0
}

```

```

fun calcularSignoChino(anio: Int): String {
    val signos = listOf(
        "Rata", "Buey", "Tigre", "Conejo", "Dragón", "Serpiente",
        "Caballo", "Cabra", "Mono", "Gallo", "Perro", "Cerdo"
    )
    return signos[(anio - 1900) % 12]
}

```

10:48

Nombre Completo
Jose Miranda

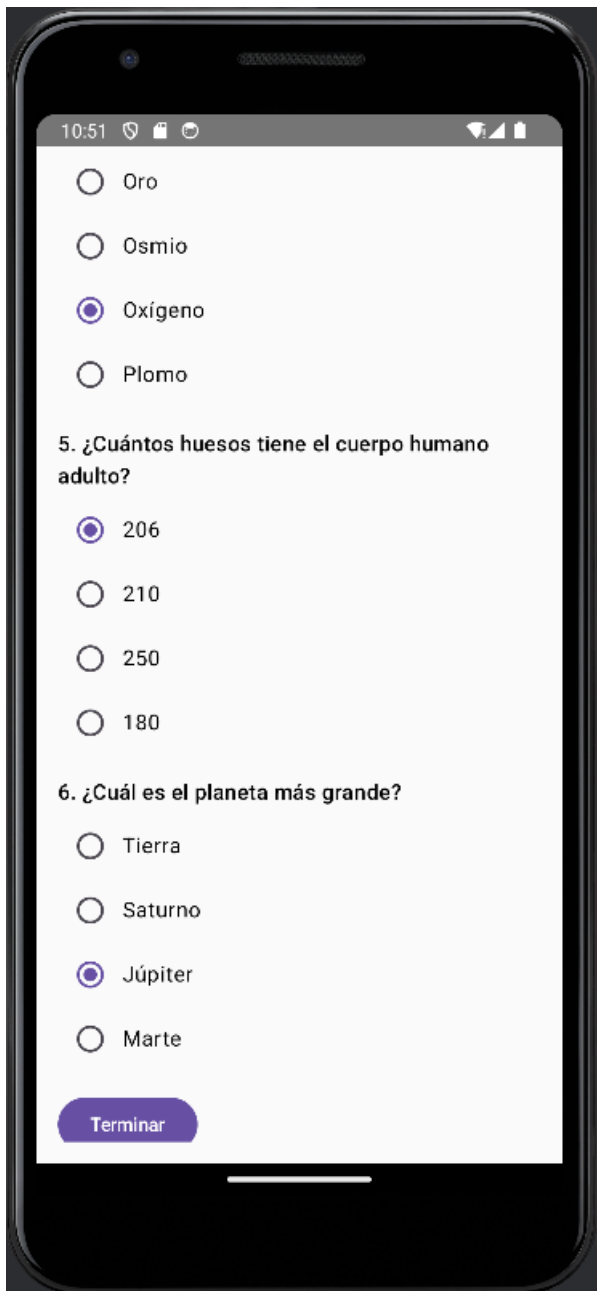
Día 08	Mes 11	Año 2000
-----------	-----------	-------------

☒ Masculino ☐ Femenino

Limpiar Siguiente

1 2 3 -
4 5 6 _
7 8 9 ×
, 0 . ←





10:51

Hola Jose Miranda

Tienes 24 años y tu signo zodiacal chino es
Dragón

Calificación: 4

