



3-7-2025

Examen 2do Parcial: Zodiaco Chino

Ana Paulina Duran Martinez



UNIVERSIDAD TECNOLOGICA DE LEON

```
package com.paulina.examen2doparcialidgs903
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.unit.dp
import androidx.navigation.NavController
import com.paulina.examen2doparcialidgs903.Pregunta
import com.paulina.examen2doparcialidgs903.calcularEdad
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
```

```
@Composable
fun PantallaExamen(
    navController: NavController,
    nombre: String,
    dia: Int,
    mes: Int,
    anio: Int,
    context: Context
){
    val preguntas = listOf(
        Pregunta("¿Cuál es la suma de 2 + 2?", listOf("8", "6", "4", "3"), 2),
        Pregunta(
            "¿Cuál es la capital de Francia?",
            listOf("Londres", "Berlín", "París", "Madrid"),
            2
        ),
        Pregunta(
            "¿Qué planeta es el más cercano al sol?",
            listOf("Venus", "Mercurio", "Tierra", "Marte"),
            1
        ),
        Pregunta("¿Cuántos continentes hay?", listOf("5", "6", "7", "8"), 2),
        Pregunta(
            "¿Quién pintó la Mona Lisa?",
            listOf("Picasso", "Van Gogh", "Da Vinci", "Miguel Ángel"),
            2
        ),
        Pregunta(
            "¿Cuál es el lenguaje oficial de Brasil?",
            listOf("Español", "Portugués", "Inglés", "Francés"),
```

```

        1
    )
)

val respuestasSeleccionadas = remember { mutableStateMapOf<Int, Int>() }

val scrollState = rememberScrollState()

Column(
    modifier = Modifier
        .fillMaxSize()
        .verticalScroll(scrollState)
        .padding(16.dp),
    verticalArrangement = Arrangement.spacedBy(12.dp)
)
{
    Text("Examen", style = MaterialTheme.typography.headlineMedium)

    preguntas.forEachIndexed { index, pregunta ->
        Card(modifier = Modifier.fillMaxWidth()) {
            Column(modifier = Modifier.padding(12.dp)) {
                Text("${index + 1}. ${pregunta.texto}")
                pregunta.opciones.forEachIndexed { opcionIndex, opcion ->
                    Row {
                        RadioButton(
                            selected = respuestasSeleccionadas[index] == opcionIndex,
                            onClick = { respuestasSeleccionadas[index] = opcionIndex }
                        )
                        Text("${'a' + opcionIndex} $opcion")
                    }
                }
            }
        }
    }
}

Button(
    onClick = {
        val correctas =
            respuestasSeleccionadas.count { it.value == preguntas[it.key].respuestaCorrecta }
        val calificacion = (correctas * 10) / preguntas.size
        val edad = calcularEdad(anio)
        val signo = determinarSignoChino(dia, mes, anio)

        guardarResultado(context, "$nombre,$edad,$signo,$calificacion")
    }
)

```

```

        navController.navigate("resultados/$nombre/$edad/$signo/$calificacion")

    },
    modifier = Modifier.fillMaxWidth()
){
    Text("Terminar Examen")
}
}
}

```

```
package com.paulina.examen2doparcialidgs903
```

```
import android.content.Context
import java.io.*
```

```
class FileManager(private val context: Context) {
    private val fileName = "datos_usuario.txt"

    fun saveData(data: String) {
        try {
            context.openFileOutput(fileName, Context.MODE_PRIVATE).use {
                it.write(data.toByteArray())
            }
        } catch (e: IOException) {
            e.printStackTrace()
        }
    }

    fun readData(): String {
        return try {
            context.openFileInput(fileName).bufferedReader().useLines { lines ->
                lines.fold("") { acc, line -> acc + line }
            }
        } catch (e: FileNotFoundException) {
            ""
        } catch (e: IOException) {
            ""
        }
    }

    fun clearData() {
        context.deleteFile(fileName)
    }
}

```

```
}  
}
```

```
package com.paulina.examen2doparcialidgs903
```

```
import android.content.Context  
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.unit.dp  
import androidx.navigation.NavController
```

```
@Composable
```

```
fun PantallaFormulario(navController: NavController, context: Context) {  
    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("Hombre") }
```

```
    Column(modifier = Modifier  
        .fillMaxSize()  
        .padding(16.dp),  
        verticalArrangement = Arrangement.spacedBy(12.dp)
```

```
    ){  
        Text("Formulario", style = MaterialTheme.typography.headlineMedium)
```

```
        OutlinedTextField(value = nombre, onValueChange = { nombre = it }, label = {  
Text("Nombre Completo") }, modifier = Modifier.fillMaxWidth())  
        OutlinedTextField(value = dia, onValueChange = { dia = it }, label = { Text("Día") }, modifier  
= Modifier.fillMaxWidth())  
        OutlinedTextField(value = mes, onValueChange = { mes = it }, label = { Text("Mes") },  
modifier = Modifier.fillMaxWidth())  
        OutlinedTextField(value = anio, onValueChange = { anio = it }, label = { Text("Año") },  
modifier = Modifier.fillMaxWidth())
```

```
    Row(verticalAlignment = Alignment.CenterVertically) {  
        RadioButton(selected = sexo == "Hombre", onClick = { sexo = "Hombre" })  
        Text("Hombre")  
        Spacer(modifier = Modifier.width(16.dp))  
        RadioButton(selected = sexo == "Mujer", onClick = { sexo = "Mujer" })
```

```

        Text("Mujer")
    }

    Row(horizontalArrangement = Arrangement.spacedBy(8.dp)) {
        Button(onClick = {
            nombre = ""
            dia = ""
            mes = ""
            anio = ""
            sexo = "Hombre"
        }) {
            Text("Limpiar")
        }

        Button(onClick = {
            if (nombre.isNotEmpty() && dia.isNotEmpty() && mes.isNotEmpty() &&
                anio.isNotEmpty()) {
                navController.navigate("examen/$nombre/$dia/$mes/$anio")
            } else {
                print("Error: Todos los campos deben estar llenos.")
            }
            navController.navigate("examen/$nombre/$dia/$mes/$anio")
        }) {
            Text("Siguiente")
        }
    }
}

```

```
package com.paulina.examen2doparcialidgs903
```

```

import android.content.Context
import java.io.OutputStreamWriter
import java.util.Calendar

```

```

fun calcularEdad(anio: Int): Int {
    val anioActual = Calendar.getInstance().get(Calendar.YEAR)
    return anioActual - anio
}

// funcion para determianr el signo chino
fun determinarSignoChino(dia: Int, mes: Int, anio: Int): String {
    val signos = listOf(
        "Rata", "Buey", "Tigre", "Conejo", "Dragón", "Serpiente",

```

```

        "Caballo", "Cabra", "Mono", "Gallo", "Perro", "Cerdo"
    )

    val fechasInicio = mapOf(
        2000 to Pair(5, 2),
        2001 to Pair(24, 1),
        2002 to Pair(12, 2),
        2003 to Pair(1, 2),
        2004 to Pair(22, 1),
        2005 to Pair(9, 2),
        2006 to Pair(29, 1),
        2007 to Pair(18, 2),
        2008 to Pair(7, 2),
        2009 to Pair(26, 1),
        2010 to Pair(14, 2)
    )

    val (inicioDia, inicioMes) = fechasInicio[anio] ?: return "Desconocido"

    val esAntesDelAñoNuevo = (mes < inicioMes) || (mes == inicioMes && dia < inicioDia)

    val anioZodiaco = if (esAntesDelAñoNuevo) anio - 1 else anio

    val indice = (anioZodiaco - 2000) % 12
    return signos[(indice + 4) % 12]
}

fun guardarResultado(context: Context, datos: String) {
    val file = OutputStreamWriter(context.openFileOutput("resultados.txt",
Context.MODE_APPEND))
    file.write("$datos\n")
    file.close()
}

package com.paulina.examen2doparcialidgs903
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.material3.*
import androidx.navigation.compose.rememberNavController

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {

```

```

        super.onCreate(savedInstanceState)
        setContent {
            val navController = rememberNavController()
            Navigation(navController, applicationContext)
        }
    }
}

```

package com.paulina.examen2doparcialidgs903

```

import android.content.Context
import androidx.compose.runtime.Composable
import androidx.navigation.NavHostController
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable

```

@Composable

```

fun Navigation(navController: NavHostController, context: Context) {
    NavHost(navController = navController, startDestination = "formulario") {
        composable("formulario") {
            PantallaFormulario(navController, context)
        }
        composable("examen/{nombre}/{dia}/{mes}/{anio}") { backStackEntry ->
            val nombre = backStackEntry.arguments?.getString("nombre") ?: ""
            val dia = backStackEntry.arguments?.getString("dia")?.toIntOrNull() ?: 1
            val mes = backStackEntry.arguments?.getString("mes")?.toIntOrNull() ?: 1
            val anio = backStackEntry.arguments?.getString("anio")?.toIntOrNull() ?: 2000

            PantallaExamen(navController, nombre, dia, mes, anio, context)
        }
        composable("resultados/{nombre}/{edad}/{signo}/{calificacion}") { backStackEntry ->
            val nombre = backStackEntry.arguments?.getString("nombre") ?: ""
            val edad = backStackEntry.arguments?.getString("edad")?.toInt() ?: 0
            val signo = backStackEntry.arguments?.getString("signo") ?: ""
            val calificacion = backStackEntry.arguments?.getString("calificacion")?.toInt() ?: 0
            PantallaResultados(nombre, edad, signo, calificacion)
        }
    }
}

```

package com.paulina.examen2doparcialidgs903

```

data class Pregunta(
    val texto: String,
    val opciones: List<String>,

```



```

        val respuestaCorrecta: Int
    )

package com.paulina.examen2doparcialidgs903

```

```

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.res.painterResource
import androidx.compose.material3.*
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign

```

```
@Composable
```

```

fun PantallaResultados(nombre: String, edad: Int, signo: String, calificacion: Int) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(24.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.spacedBy(24.dp)
    ) {
        Text("Hola $nombre", fontSize = 22.sp, fontWeight = FontWeight.Bold)
        Text("Tienes $edad años y tu signo zodiacal", fontSize = 18.sp)
        Text("Es $signo", fontSize = 18.sp, fontWeight = FontWeight.Bold)

        Image(
            painter = painterResource(id = getSignoDrawable(signo.lowercase())),
            contentDescription = null,
            modifier = Modifier.size(150.dp)
        )

        Text("Calificación: $calificacion", fontSize = 20.sp, fontWeight = FontWeight.Bold,
            textAlign = TextAlign.Center)
    }
}

fun getSignoDrawable(signo: String): Int {
    return 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 -> R.drawable.signo_desconocido
}
}
```