Main Activity

```
package com.example.examenapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import com.example.examenapp.navigation.NavGraph
import com.example.examenapp.ui.theme.ExamenAppTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            ExamenAppTheme(darkTheme = false) {
                NavGraph()
            }
        }
    }
}
```

NavGraph.kt

```
package com.example.examenapp.navigation

import androidx.compose.runtime.Composable
import androidx.navigation.NavHostController
import androidx.navigation.NavType
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import androidx.navigation.navArgument
import com.example.examenapp.screens.ExamenScreen
import com.example.examenapp.screens.FormularioScreen
import com.example.examenapp.screens.ResultadoScreen
```

```
@Composable
fun NavGraph(navController: NavHostController = rememberNavController()) {
  NavHost(navController = navController, startDestination = "formulario") {
    composable("formulario") { FormularioScreen(navController) }
    composable("examen") { ExamenScreen(navController) }

    composable(
      route = "resultado/{calificacion}",
      arguments = listOf(navArgument("calificacion") { type = NavType.IntType })
    ) { backStackEntry ->
      val calificacion = backStackEntry.arguments?.getInt("calificacion") ?: 0
      ResultadoScreen(navController, calificacion)
    }
  }
}
```

FormularioScreen.kt

```
package com.example.examenapp.screens

import android.content.Context
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.Button
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.RadioButton
import androidx.compose.material3.Text
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.NavController
import java.io.OutputStreamWriter

fun saveToFile(context: Context, filename: String, data: String) {
```

```
  val outputStreamWriter = OutputStreamWriter(context.openFileOutput(filename,
Context.MODE_PRIVATE))
  outputStreamWriter.use {
    it.write(data)
  }
}

@Composable
fun FormularioScreen(navController: NavController) {
  val context = LocalContext.current
  var nombre by remember { mutableStateOf("") }
  var apaterno by remember { mutableStateOf("") }
  var amaterno 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
    .padding(16.dp)
    .fillMaxSize(),
    verticalArrangement = Arrangement.spacedBy(10.dp)
  ) {
    Text("Formulario", style = MaterialTheme.typography.titleLarge, color =
MaterialTheme.colorScheme.onBackground)

    OutlinedTextField(
      value = nombre,
      onValueChange = { nombre = it },
      label = { Text("Nombre") },
      modifier = Modifier.fillMaxWidth()
    )

    OutlinedTextField(
      value = apaterno,
      onValueChange = { apaterno = it },
      label = { Text("Apellido Paterno") },
      modifier = Modifier.fillMaxWidth()
```

```
    )

    OutlinedTextField(
      value = amaterno,
      onValueChange = { amaterno = it },
      label = { Text("Apellido Materno") },
      modifier = Modifier.fillMaxWidth()
    )


    Row(horizontalArrangement = Arrangement.SpaceBetween) {
      OutlinedTextField(
        value = dia,
        onValueChange = { dia = it },
        label = { Text("Día") },
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),
        modifier = Modifier.weight(1f)
      )
      Spacer(modifier = Modifier.width(8.dp))
      OutlinedTextField(
        value = mes,
        onValueChange = { mes = it },
        label = { Text("Mes") },
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),
        modifier = Modifier.weight(1f)
      )
      Spacer(modifier = Modifier.width(8.dp))
      OutlinedTextField(
        value = anio,
        onValueChange = { anio = it },
        label = { Text("Año") },
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),
        modifier = Modifier.weight(1f)
      )
    }


    Text("Sexo:")
```

```
Row {
    RadioButton(
        selected = sexo == "Hombre",
        onClick = { sexo = "Hombre" }
    )
    Text("Hombre", modifier = Modifier.padding(end = 16.dp))
    RadioButton(
        selected = sexo == "Mujer",
        onClick = { sexo = "Mujer" }
    )
    Text("Mujer")
}

Row(
    horizontalArrangement = Arrangement.SpaceBetween,
    modifier = Modifier.fillMaxWidth()
) {
    Button(onClick = {
        nombre = ""
        apaterno = ""
        amaterno = ""
        dia = ""
        mes = ""
        anio = ""
        sexo = "Hombre"
    }) {
        Text("Limpiar")
    }

    Button(onClick = {
        val data = "$nombre,$apaterno,$amaterno,$dia,$mes,$anio,$sexo"
        // Guardar en archivo
        saveToFile(context, "usuario.txt", data)
        navController.navigate("examen")
    }) {
        Text("Siguiente")
    }
}
```

```
  }
}
```

ExamenScreen.kt

```
package com.example.examenapp.screens

import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp
import androidx.navigation.NavController

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

@Composable
fun ExamenScreen(navController: NavController) {
  val preguntas = listOf(
    Pregunta("¿Cuál es el océano más grande del mundo?", listOf("Atlántico", "Índico",
"Pacífico", "Ártico"), 2),
    Pregunta("¿Cuál es la capital de México?", listOf("Lima", "Madrid", "CDMX",
"Bogotá"), 2),
    Pregunta("¿Cuál es el planeta más cercano al Sol?", listOf("Venus", "Tierra",
"Mercurio", "Marte"), 2),
    Pregunta("¿Cuál es el símbolo químico del oro?", listOf("Au", "Ag", "Fe", "O"), 0),
    Pregunta("¿Qué animal es conocido como el rey de la selva?", listOf("Tigre",
"Elefante", "León", "Jaguar"), 2),
    Pregunta("¿Qué instrumento tiene teclas, cuerdas y martillos?", listOf("Violín",
"Guitarra", "Piano", "Arpa"), 2)
  )
```

```
  val respuestasUsuario = remember { mutableStateListOf(-1, -1, -1, -1, -1, -1) }

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

    Text("Examen", style = MaterialTheme.typography.titleLarge)

    LazyColumn(
      modifier = Modifier
        .weight(1f)
        .padding(top = 16.dp),
      verticalArrangement = Arrangement.spacedBy(16.dp)
    ) {
      items(preguntas.size) { index ->
        val pregunta = preguntas[index]
        Column {
          Text("${index + 1}. ${pregunta.texto}")
          pregunta.opciones.forEachIndexed { i, opcion ->
            Row(verticalAlignment = androidx.compose.ui.Alignment.CenterVertically)
{
              RadioButton(
                selected = respuestasUsuario[index] == i,
                onClick = { respuestasUsuario[index] = i }
              )
              Text(opcion)
            }
          }
        }
      }
    }

    Button(
      onClick = {
        val correctas = preguntas.zip(respuestasUsuario).count {
          it.second == it.first.respuestaCorrecta
        }
        // Aquí podrías guardar el resultado y usarlo en la Pantalla 3
```

```
            navController.navigate("resultado/$correctas")
        },
        modifier = Modifier.fillMaxWidth()
    ) {
        Text("Terminar")
    }
  }
}
```

ResultadoScreen.kt

```
package com.example.examenapp.screens

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.ui.Alignment
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.NavController
import com.example.examenapp.utils.readFromFile
import com.example.examenapp.R
import com.example.examenapp.model.Usuario
import com.example.examenapp.utils.calcularEdad
import com.example.examenapp.utils.signoZodiacoChino

@Composable
fun ResultadoScreen(navController: NavController, calificacion: Int) {
  val context = LocalContext.current
  var datos by remember { mutableStateOf<String?>(null) }
```

```
  LaunchedEffect(Unit) {
    datos = readFromFile(context, "usuario.txt")
  }

  if (datos == null) {
    Text("Cargando datos...")
    return
  }

  val partes = datos!!.split(",")
  if (partes.size < 7) {
    Text("Datos corruptos")
    return
  }

  val usuario = Usuario(
    nombre = partes[0],
    apaterno = partes[1],
    amaterno = partes[2],
    dia = partes[3].toIntOrNull() ?: 1,
    mes = partes[4].toIntOrNull() ?: 1,
    anio = partes[5].toIntOrNull() ?: 2000,
    sexo = partes[6]
  )

  val edad = calcularEdad(usuario.dia, usuario.mes, usuario.anio)
  val signo = signoZodiacoChino(usuario.anio)

  Column(
    modifier = Modifier
      .fillMaxSize()
      .padding(24.dp),
    verticalArrangement = Arrangement.spacedBy(20.dp),
    horizontalAlignment = Alignment.CenterHorizontally
  ) {
    Text(
      "Hola ${usuario.nombre} ${usuario.apaterno} ${usuario.amaterno}",
```

```
            style = MaterialTheme.typography.titleLarge
        )
        Text("Tienes $edad años")
        Text("Tu signo zodiacal chino es: ${signo.replaceFirstChar { it.uppercase() }}")
        Text("Calificación: $calificacion / 6")

        val recursoImagen = 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
        }

        Image(
            painter = painterResource(id = recursoImagen),
            contentDescription = "Signo Zodiacal",
            modifier = Modifier.size(150.dp)
        )
    }
}
```

Usuario

```
package com.example.examenapp.model

data class Usuario(
    val nombre: String,
    val apaterno: String,
```

```
  val amaterno: String,
  val dia: Int,
  val mes: Int,
  val anio: Int,
  val sexo: String
)
```

Zodiaco.kt

```
package com.example.examenapp.utils

fun calcularEdad(dia: Int, mes: Int, anio: Int): Int {
  val today = java.util.Calendar.getInstance()
  var edad = today.get(java.util.Calendar.YEAR) - anio
  if (mes > today.get(java.util.Calendar.MONTH) + 1 ||
    (mes == today.get(java.util.Calendar.MONTH) + 1 && dia >
today.get(java.util.Calendar.DAY_OF_MONTH))) {
    edad--
  }
  return edad
}

fun signoZodiacoChino(anio: Int): String {
  val signos = listOf(
    "rata", "buey", "tigre", "conejo", "dragon", "serpiente",
    "caballo", "cabra", "mono", "gallo", "perro", "cerdo"
  )
  return signos[anio % 12]
}
```