



# EXAMEN PARCIAL 2

Desarrollo para dispositivos inteligentes  
IDGS 901

Andrea Guadalupe Concha Diaz  
07/02/2025

# CÓDIGO

MainActivity.kt

```
package org.utl.dsm505.segundoparcial

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import org.utl.dsm505.segundoparcial.AppNavigation
import org.utl.dsm505.segundoparcial.ui.theme.SegundoParcialTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            SegundoParcialTheme {
                AppNavigation()
            }
        }
    }
}
```

AppScreens.kt

```
package org.utl.dsm505.segundoparcial

sealed class AppScreens {
    object PersonalDataScreen : AppScreens()
    object QuizScreen : AppScreens()
    object ResultsScreen : AppScreens()
}
```

QuizQuestion.kt

```
package org.utl.dsm505.segundoparcial

data class QuizQuestion(
    val id: Int,
    val question: String,
    val options: List<String>,
    val correctAnswer: Int
)
```

UserData.kt

```
package org.utl.dsm505.segundoparcial
```

```
data class UserData(  
    val name: String = "",  
    val paternalLastName: String = "",  
    val maternalLastName: String = "",  
    val birthDay: Int = 1,  
    val birthMonth: Int = 1,  
    val birthYear: Int = 2000,  
    val gender: String = ""  
) {  
    fun fullName(): String = "$name $paternalLastName $maternalLastName"  
  
    fun calculateAge(): Int {  
        val currentYear = java.util.Calendar.getInstance().get(java.util.Calendar.YEAR)  
        return currentYear - birthYear  
    }  
}
```

```
ZodiacSign.kt
```

```
package org.utl.dsm505.segundoparcial
```

```
data class ZodiacSign(  
    val name: String,  
    val imageRes: String,  
    val dateRange: String  
)
```

```
AppNavigation.kt
```

```
package org.utl.dsm505.segundoparcial  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.mutableStateOf  
import androidx.compose.runtime.remember  
import org.utl.dsm505.segundoparcial.UserData  
import org.utl.dsm505.segundoparcial.PersonalDataScreen  
import org.utl.dsm505.segundoparcial.QuizScreen  
import org.utl.dsm505.segundoparcial.ResultsScreen
```

```
import androidx.compose.runtime.getValue  
import androidx.compose.runtime.setValue
```

```
@Composable  
fun AppNavigation() {  
    var currentScreen by remember {  
        mutableStateOf<AppScreens>(AppScreens.PersonalDataScreen) }  
    var userData by remember { mutableStateOf(UserData()) }  
    var quizScore by remember { mutableStateOf(0) }
```

```

when (currentScreen) {
    AppScreens.PersonalDataScreen -> PersonalDataScreen(
        onNavigate = { screen ->
            currentScreen = screen
        },
        onSaveUserData = { data ->
            userData = data
        }
    )
    AppScreens.QuizScreen -> QuizScreen(
        onNavigate = { screen ->
            currentScreen = screen
        },
        calculateScore = { score ->
            quizScore = score
        }
    )
    AppScreens.ResultsScreen -> ResultsScreen(
        userData = userData,
        score = quizScore,
        onNavigate = { screen ->
            currentScreen = screen
            // Reseteo de datos si es necesario
            userData = UserData()
            quizScore = 0
        }
    )
}
}
}

```

FileManager.kt

```
package org.utl.dsm505.segundoparcial
```

```
import android.content.Context
import org.utl.dsm505.segundoparcial.UserData
```

```

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

    fun saveUserData(userData: UserData) {
        val data = """
            Nombre: ${userData.name}
            Apellido Paterno: ${userData.paternalLastName}
            Apellido Materno: ${userData.maternalLastName}
            Fecha de Nacimiento:
            ${userData.birthDay}/${userData.birthMonth}/${userData.birthYear}
            Género: ${userData.gender}
            """.trimIndent()

        context.openFileOutput(fileName, Context.MODE_PRIVATE).use {
            it.write(data.toByteArray())
        }
    }

    fun readUserData(): String {
        return try {
            context.openFileInput(fileName).bufferedReader().use { it.readText() }
        } catch (e: Exception) {
            "No hay datos guardados"
        }
    }
}

```

PersonalDataScreen.kt

```
package org.utl.dsm505.segundoparcial
```

```

import android.content.Context
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.*

```

```

import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import org.utl.dsm505.segundoparcial.R
import org.utl.dsm505.segundoparcial.UserData
import org.utl.dsm505.segundoparcial.AppScreens

```

@Composable

```

fun PersonalDataScreen(
    onNavigate: (AppScreens) -> Unit,
    onSaveUserData: (UserData) -> Unit
){
    var name by remember { mutableStateOf("") }
    var paternalLastName by remember { mutableStateOf("") }
    var maternalLastName by remember { mutableStateOf("") }
    var birthDay by remember { mutableStateOf("") }
    var birthMonth by remember { mutableStateOf("") }
    var birthYear by remember { mutableStateOf("") }
    var gender by remember { mutableStateOf("") }

    val context = LocalContext.current
    val fileManager = remember { FileManager(context) }

```

```

Column(
    modifier = Modifier
        .fillMaxSize()
        .padding(16.dp)
        .verticalScroll(rememberScrollState()),
    horizontalAlignment = Alignment.CenterHorizontally
){
    Text(
        text = "Datos Personales",
        fontSize = 24.sp,
        fontWeight = FontWeight.Bold,
        modifier = Modifier.padding(bottom = 16.dp)
    )

```

```

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

```

```
        .fillMaxWidth()
        .padding(bottom = 8.dp)
    )
```

```
OutlinedTextField(
    value = paternalLastName,
    onChange = { paternalLastName = it },
    label = { Text("Apellido Paterno") },
    modifier = Modifier
        .fillMaxWidth()
        .padding(bottom = 8.dp)
)
```

```
OutlinedTextField(
    value = maternalLastName,
    onChange = { maternalLastName = it },
    label = { Text("Apellido Materno") },
    modifier = Modifier
        .fillMaxWidth()
        .padding(bottom = 8.dp)
)
```

```
Text(
    text = "Fecha de Nacimiento",
    modifier = Modifier
        .fillMaxWidth()
        .padding(top = 8.dp, bottom = 4.dp)
)
```

```
Row(
    modifier = Modifier.fillMaxWidth(),
    horizontalArrangement = Arrangement.SpaceBetween
) {
    OutlinedTextField(
        value = birthDay,
        onChange = { if (it.length <= 2) birthDay = it },
        label = { Text("Día") },
        modifier = Modifier.weight(0.3f)
    )
    Spacer(modifier = Modifier.width(8.dp))
    OutlinedTextField(
        value = birthMonth,
        onChange = { if (it.length <= 2) birthMonth = it },
        label = { Text("Mes") },
        modifier = Modifier.weight(0.3f)
    )
    Spacer(modifier = Modifier.width(8.dp))
}
```

```

        OutlinedTextField(
            value = birthYear,
            onValueChange = { if (it.length <= 4) birthYear = it },
            label = { Text("Año") },
            modifier = Modifier.weight(0.4f)
        )
    }

    Text(
        text = "Sexo",
        modifier = Modifier
            .fillMaxWidth()
            .padding(top = 16.dp, bottom = 8.dp)
    )

    Row(
        modifier = Modifier.fillMaxWidth(),
        verticalAlignment = Alignment.CenterVertically
    ) {
        RadioButton(
            selected = gender == "Masculino",
            onClick = { gender = "Masculino" }
        )
        Text("Masculino", modifier = Modifier.padding(start = 8.dp))

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

        RadioButton(
            selected = gender == "Femenino",
            onClick = { gender = "Femenino" }
        )
        Text("Femenino", modifier = Modifier.padding(start = 8.dp))
    }

    Row(
        modifier = Modifier
            .fillMaxWidth()
            .padding(top = 24.dp),
        horizontalArrangement = Arrangement.SpaceBetween
    ) {
        Button(
            onClick = {
                name = ""
                paternalLastName = ""
                maternalLastName = ""
                birthDay = ""
                birthMonth = ""
            }
        )
    }

```





```
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import org.utl.dsm505.segundoparcial.QuizQuestion
import org.utl.dsm505.segundoparcial.AppScreens
```

```
@Composable
fun QuizScreen(
    onNavigate: (AppScreens) -> Unit,
    calculateScore: (Int) -> Unit
){
    val questions = remember {
        listOf(
            QuizQuestion(
                id = 1,
                question = "¿Cuál es el planeta más grande del sistema solar?",
                options = listOf("Tierra", "Marte", "Júpiter", "Venus"),
                correctAnswer = 2
            ),
            QuizQuestion(
                id = 2,
                question = "¿Qué autor escribió Cien años de soledad?",
                options = listOf(
                    "Mario Vargas Llosa",
                    "Gabriel García Márquez",
                    "Julio Cortázar",
                    "Isabel Allende"
                ),
                correctAnswer = 1
            ),
            QuizQuestion(
                id = 3,
                question = "¿Cuál es el resultado de  $8 \times 7$ ?",
                options = listOf("54", "56", "64", "48"),
                correctAnswer = 1
            ),
            QuizQuestion(
                id = 4,
                question = "¿En qué año comenzó la Segunda Guerra Mundial?",
                options = listOf("1918", "1939", "1945", "1929"),
                correctAnswer = 1
            ),
            QuizQuestion(
                id = 5,
                question = "¿Qué órgano del cuerpo humano bombea la sangre?",
                options = listOf("Pulmones", "Estómago", "Hígado", "Corazón"),
                correctAnswer = 3
            )
        )
    }
}
```

```

    ),
    QuizQuestion(
        id = 6,
        question = "¿Cuál es el lenguaje de programación más utilizado para desarrollo
web front-end?",
        options = listOf("Python", "Java", "JavaScript", "C++"),
        correctAnswer = 2
    )
)
}

```

```

val userAnswers = remember { mutableStateMapOf<Int, Int?>() }
var showResults by remember { mutableStateOf(false) }
var score by remember { mutableStateOf(0) }

```

```

Column(
    modifier = Modifier
        .fillMaxSize()
        .padding(16.dp)
        .verticalScroll(rememberScrollState())
) {
    Text(
        text = "Examen de Conocimientos",
        fontSize = 24.sp,
        fontWeight = FontWeight.Bold,
        modifier = Modifier.padding(bottom = 16.dp)
    )
}

```

```

questions.forEach { question ->
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .padding(vertical = 8.dp)
    ) {
        Text(
            text = "${question.id}. ${question.question}",
            fontWeight = FontWeight.Bold,
            modifier = Modifier.padding(bottom = 8.dp)
        )

        question.options.forEachIndexed { index, option ->
            Row(
                verticalAlignment = Alignment.CenterVertically,
                modifier = Modifier.padding(vertical = 4.dp)
            ) {
                RadioButton(
                    selected = userAnswers[question.id] == index,

```



```
package org.utl.dsm505.segundoparcial
```

```
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
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.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import org.utl.dsm505.segundoparcial.R
import org.utl.dsm505.segundoparcial.UserData
import org.utl.dsm505.segundoparcial.AppScreens
import org.utl.dsm505.segundoparcial.ZodiacCalculator
import androidx.compose.runtime.remember
```

```
@Composable
```

```
fun ResultsScreen(
    userData: UserData,
    score: Int,
    onNavigate: (AppScreens) -> Unit
){
    val zodiacSign = ZodiacCalculator.calculateChineseZodiac(userData.birthYear)
    val context = LocalContext.current
    val fileManager = remember { FileManager(context) }
```

```
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        horizontalAlignment = Alignment.CenterHorizontally
    ){
        Text(
            text = "Resultados",
            fontSize = 24.sp,
            fontWeight = FontWeight.Bold,
            modifier = Modifier.padding(bottom = 16.dp)
        )
```

```
        Text(
            text = "Hola ${userData.fullName()}",
            fontSize = 20.sp,
            modifier = Modifier.padding(bottom = 8.dp)
        )
    }
}
```

```

)

Text(
    text = "Tienes ${userData.calculateAge()} años y tu signo zodiacal es:",
    modifier = Modifier.padding(bottom = 16.dp)
)

// Mostrar imagen del signo zodiacal
val imageRes = context.resources.getIdentifier(
    zodiacSign.imageRes,
    "drawable",
    context.packageName
)

Image(
    painter = painterResource(id = imageRes),
    contentDescription = zodiacSign.name,
    modifier = Modifier
        .size(120.dp)
        .padding(16.dp)
)

Text(
    text = zodiacSign.name,
    fontSize = 20.sp,
    fontWeight = FontWeight.Bold,
    modifier = Modifier.padding(bottom = 8.dp)
)

Text(
    text = "Años correspondientes: ${zodiacSign.years}",
    modifier = Modifier.padding(bottom = 16.dp)
)

Text(
    text = "Calificación del examen: $score de 6",
    fontSize = 18.sp,
    fontWeight = FontWeight.Bold,
    modifier = Modifier.padding(bottom = 24.dp)
)

Button(
    onClick = {
        onNavigate(AppScreens.PersonalDataScreen)
    },
    modifier = Modifier.fillMaxWidth()
){

```

```

        Text("Finalizar")
    }

    // Mostrar datos guardados en archivo (opcional)
    Divider(modifier = Modifier.padding(vertical = 16.dp))
    Text(
        text = "Datos guardados:",
        fontWeight = FontWeight.Bold,
        modifier = Modifier.padding(bottom = 8.dp)
    )
    Text(text = fileManager.readUserData())
}
}

```

ZodiacCalculator.kt

```
package org.utl.dsm505.segundoparcial
```

```

object ZodiacCalculator {
    fun calculateChineseZodiac(birthYear: Int): ChineseZodiacSign {
        val zodiacs = listOf(
            ChineseZodiacSign("Rata", "rata", "2008, 1996, 1984, 1972, 1960"),
            ChineseZodiacSign("Buey", "buey", "2009, 1997, 1985, 1973, 1961"),
            ChineseZodiacSign("Tigre", "tigre", "2010, 1998, 1986, 1974, 1962"),
            ChineseZodiacSign("Conejo", "conejo", "2011, 1999, 1987, 1975, 1963"),
            ChineseZodiacSign("Dragón", "dragon", "2012, 2000, 1988, 1976, 1964"),
            ChineseZodiacSign("Serpiente", "serpiente", "2013, 2001, 1989, 1977, 1965"),
            ChineseZodiacSign("Caballo", "caballo", "2014, 2002, 1990, 1978, 1966"),
            ChineseZodiacSign("Cabra", "cabra", "2015, 2003, 1991, 1979, 1967"),
            ChineseZodiacSign("Mono", "mono", "2016, 2004, 1992, 1980, 1968"),
            ChineseZodiacSign("Gallo", "gallo", "2017, 2005, 1993, 1981, 1969"),
            ChineseZodiacSign("Perro", "perro", "2018, 2006, 1994, 1982, 1970"),
            ChineseZodiacSign("Cerdo", "cerdo", "2019, 2007, 1995, 1983, 1971")
        )

        val index = (birthYear - 1960) % 12
        return zodiacs[index]
    }
}

data class ChineseZodiacSign(
    val name: String,
    val imageRes: String,
    val years: String
)

```

## PANTALLAS

**Datos Personales**

Nombre

Apellido Paterno

Apellido Materno

**Fecha de Nacimiento**

Día  Mes  Año

**Sexo**

☐ Masculino ☒ Femenino



# Examen de Conocimientos

1. ¿Cuál es el planeta más grande del sistema solar?

- ☐ Tierra
- ☐ Marte
- ☐ Júpiter
- ☐ Venus

---

2. ¿Qué autor escribió Cien años de soledad?

- ☐ Mario Vargas Llosa
- ☐ Gabriel García Márquez
- ☐ Julio Cortázar
- ☐ Isabel Allende

---

3. ¿Cuál es el resultado de  $8 \times 7$ ?

- ☐ 54
- ☐ 56



17:45

85%



1929

---

**5. ¿Qué órgano del cuerpo humano bombea la sangre?**



Pulmones



Estómago



Hígado



Corazón

---

**6. ¿Cuál es el lenguaje de programación más utilizado para desarrollo web front-end?**



Python



Java



JavaScript



C++

---

Terminar Examen



---

**5. ¿Qué órgano del cuerpo humano bombea la sangre?**

- ☐ Pulmones
- ☐ Estómago
- ☐ Hígado
- ☒ Corazón

---

**6. ¿Cuál es el lenguaje de programación más utilizado para desarrollo web front-end?**

- ☐ Python
- ☐ Java
- ☒ JavaScript
- ☐ C++

---

**Tu puntuación: 6 de 6**

Ver Resultados



10:17

## Resultados

37%

Hola Andrea Guadalupe Concha  
Diaz

Tienes 21 años y tu signo zodiacal es:



### Mono

Años correspondientes: 2016, 2004, 1992,  
1980, 1968

**Calificación del examen: 6 de 6**

Finalizar

---

#### Datos guardados:

Nombre: Andrea Guadalupe

Apellido Paterno: Concha

Apellido Materno: Diaz

Fecha de Nacimiento: 3/1/2004

Género: Femenino

# ARCHIVO DE TEXTO

