UNIVERSIDAD DON BOSCO

Docente:

Julio Armando García Fabián

Asignatura:

Desarrollo de Software para Moviles

Integrantes:

Cristian Ricardo González González (GG222955)

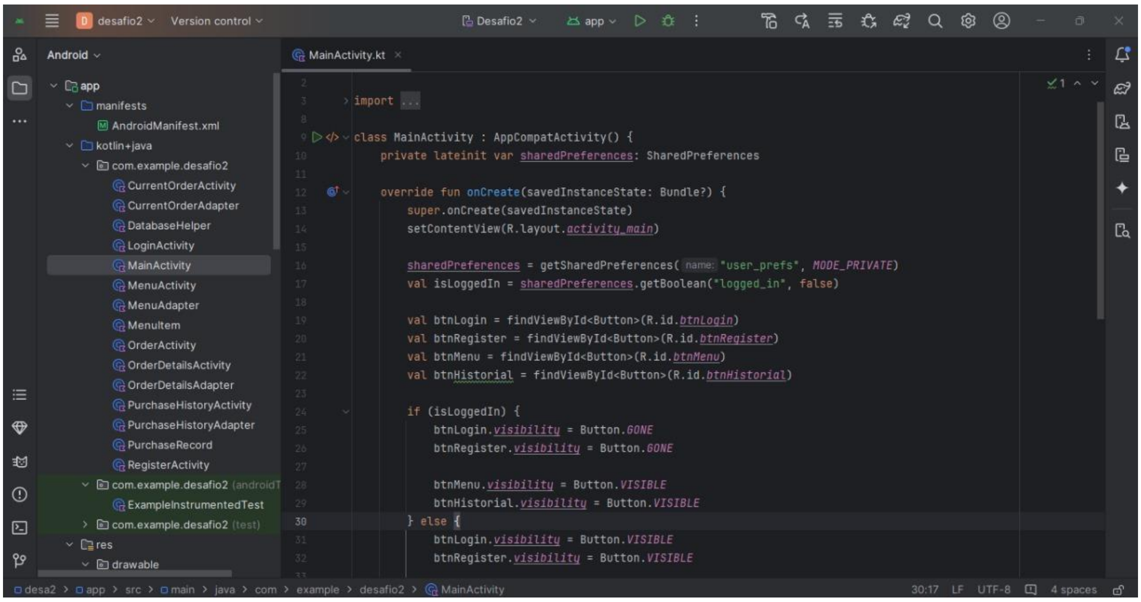
Dania Gorety Tejada Campos (TC232020)

Grupo:

G01L

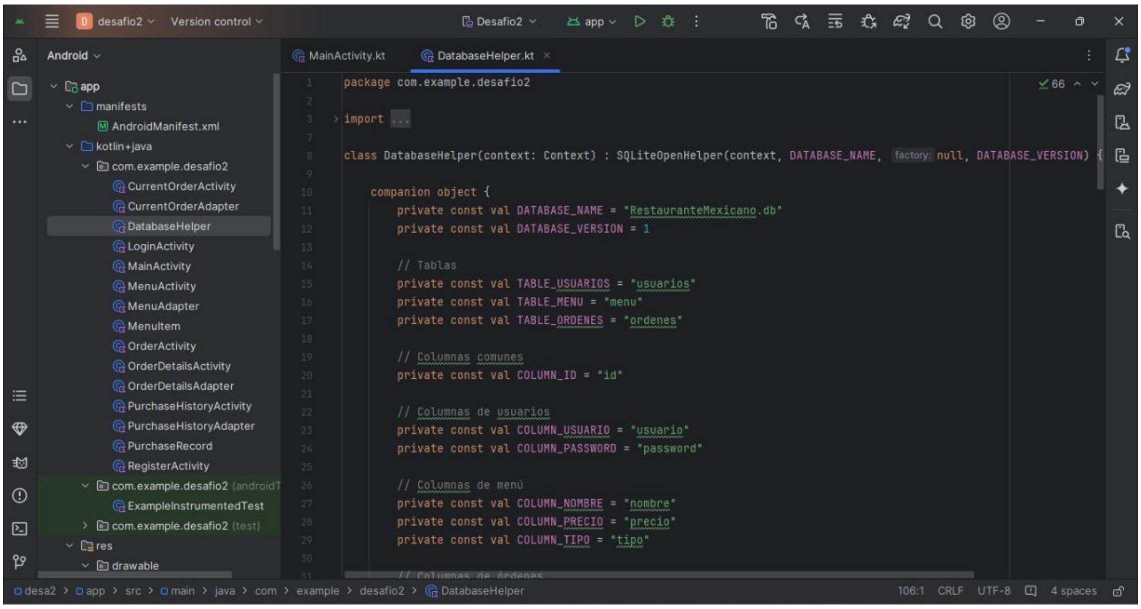
Desafío 2

Aplicación de **"Comida Mexicana"** pantalla.



```
import ...

class MainActivity : AppCompatActivity() {
    private lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        sharedPreferences = getSharedPreferences( name: "user_prefs", MODE_PRIVATE)
        val isLoggedIn = sharedPreferences.getBoolean("logged_in", false)

        val btnLogin = findViewById<Button>(R.id.btnLogin)
        val btnRegister = findViewById<Button>(R.id.btnRegister)
        val btnMenu = findViewById<Button>(R.id.btnMenu)
        val btnHistorial = findViewById<Button>(R.id.btnHistorial)

        if (isLoggedIn) {
            btnLogin.visibility = Button.GONE
            btnRegister.visibility = Button.GONE

            btnMenu.visibility = Button.VISIBLE
            btnHistorial.visibility = Button.VISIBLE
        } else {
            btnLogin.visibility = Button.VISIBLE
            btnRegister.visibility = Button.VISIBLE
```

Base de datos MySQLlite



```
package com.example.desafio2

import ...

class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME,  factory: null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "RestauranteMexicano.db"
        private const val DATABASE_VERSION = 1

        // Tablas
        private const val TABLE_USUARIOS = "usuarios"
        private const val TABLE_MENU = "menu"
        private const val TABLE_ORDENES = "ordenes"

        // Columnas comunes
        private const val COLUMN_ID = "id"

        // Columnas de usuarios
        private const val COLUMN_USUARIO = "usuario"
        private const val COLUMN_PASSWORD = "password"

        // Columnas de menú
        private const val COLUMN_NOMBRE = "nombre"
        private const val COLUMN_PRECIO = "precio"
        private const val COLUMN_TIPO = "tipo"

        // Columnas de órdenes
```

Codigo:

package com.example.desafio2

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

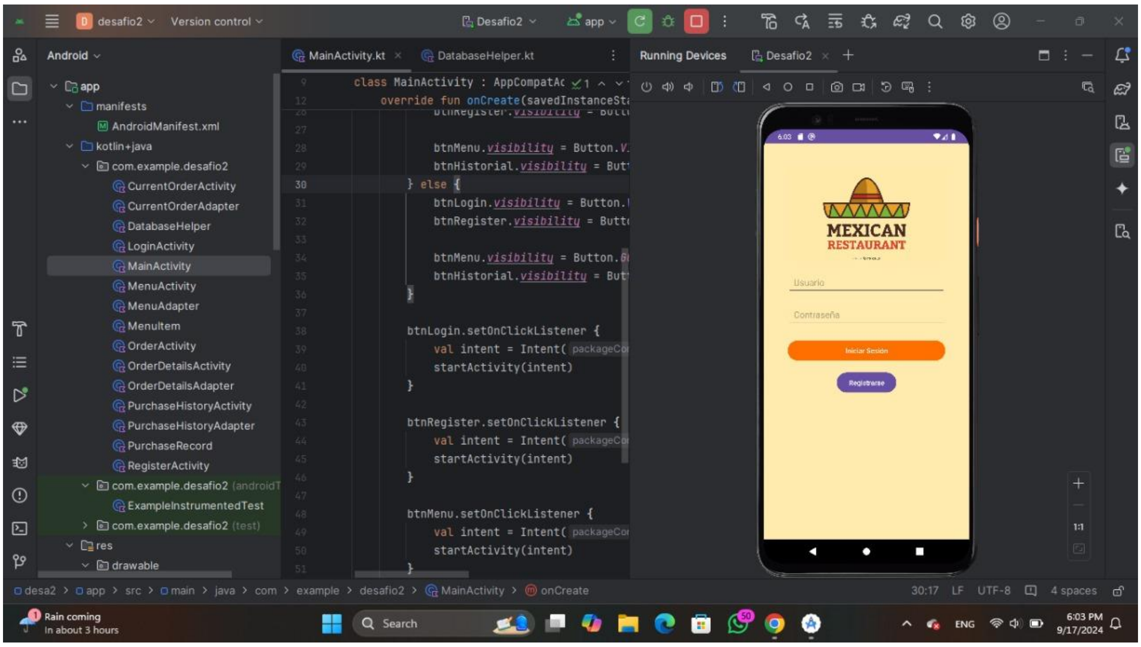
class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {

```
companion object {
    private const val DATABASE_NAME = "RestauranteMexicano.db"
    private const val DATABASE_VERSION = 1

    // Tablas
    private const val TABLE_USUARIOS = "usuarios"
    private const val TABLE_MENU = "menu"
    private const val TABLE_ORDENES = "ordenes"

    // Columnas comunes
    private const val COLUMN_ID = "id"

    // Columnas de usuarios
    private const val COLUMN_USUARIO = "usuario"
    private const val COLUMN_PASSWORD = "password"

    // Columnas de menú
    private const val COLUMN_NOMBRE = "nombre"
    private const val COLUMN_PRECIO = "precio"
    private const val COLUMN_TIPO = "tipo"

    // Columnas de órdenes
    private const val COLUMN_USUARIO_ID = "usuario_id"
    private const val COLUMN_FECHA = "fecha"
    private const val COLUMN_TOTAL = "total"
}

override fun onCreate(db: SQLiteDatabase) {
    // Crear tabla de usuarios
    val createUsuariosTable = """
        CREATE TABLE $TABLE_USUARIOS (
            $COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT,
            $COLUMN_USUARIO TEXT UNIQUE,
            $COLUMN_PASSWORD TEXT
        )
    """.trimIndent()
    db.execSQL(createUsuariosTable)

    // Crear tabla de menú
    val createMenuTable = """
        CREATE TABLE $TABLE_MENU (
            $COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT,
            $COLUMN_NOMBRE TEXT,
            $COLUMN_PRECIO REAL,
            $COLUMN_TIPO TEXT
        )
    """.trimIndent()
    db.execSQL(createMenuTable)

    // Crear tabla de órdenes
    val createOrdenesTable = """
```

```
            CREATE TABLE $TABLE_ORDENES (
                $COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT,
                $COLUMN_USUARIO_ID INTEGER,
                $COLUMN_FECHA TEXT,
                $COLUMN_TOTAL REAL,
                FOREIGN KEY($COLUMN_USUARIO_ID) REFERENCES
$TABLE_USUARIOS($COLUMN_ID)
            )
        """.trimIndent()
        db.execSQL(createOrdenesTable)

        // Insertar datos de ejemplo en el menú
        insertarDatosEjemplo(db)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
        // En caso de actualización de la base de datos
        db.execSQL("DROP TABLE IF EXISTS $TABLE_USUARIOS")
        db.execSQL("DROP TABLE IF EXISTS $TABLE_MENU")
        db.execSQL("DROP TABLE IF EXISTS $TABLE_ORDENES")
        onCreate(db)
    }


    private fun insertarDatosEjemplo(db: SQLiteDatabase) {
        // Insertar alimentos
        val alimentosEjemplo = arrayOf(
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Tacos al Pastor', 5.00, 'Alimento')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Enchiladas', 3.00, 'Alimento')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Guacamole', 1.75, 'Alimento')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Quesadillas', 4.50, 'Alimento')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Tamales', 2.00, 'Alimento')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Flautas', 4.00, 'Alimento')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Tortas', 5.00, 'Alimento')"
        )

        // Insertar bebidas
        val bebidasEjemplo = arrayOf(
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Agua Frescas', 2.50, 'Bebida')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Margarita', 4.50, 'Bebida')",
            "INSERT INTO $TABLE_MENU ($COLUMN_NOMBRE, $COLUMN_PRECIO,
$COLUMN_TIPO) VALUES ('Tequila', 4.00, 'Bebida')"
```

```
        )

        alimentosEjemplo.forEach { db.execSQL(it) }
        bebidasEjemplo.forEach { db.execSQL(it) }
    }


    fun savePurchase(usuarioId: Int, fecha: String, total: Double) {
        val values = ContentValues().apply {
            put(COLUMN_USUARIO_ID, usuarioId)
            put(COLUMN_FECHA, fecha)
            put(COLUMN_TOTAL, total)
        }
        writableDatabase.insert(TABLE_ORDENES, null, values)
    }

    fun getPurchaseHistory(usuarioId: Int): List<PurchaseRecord> {
        val purchaseList = mutableListOf<PurchaseRecord>()
        val query = """
            SELECT $COLUMN_FECHA, $COLUMN_TOTAL
            FROM $TABLE_ORDENES
            WHERE $COLUMN_USUARIO_ID = ?
            ORDER BY $COLUMN_FECHA DESC
        """.trimIndent()
        val cursor = readableDatabase.rawQuery(query, arrayOf(usuarioId.toString()))

        with(cursor) {
            while (moveToNext()) {
                val date = getString(getColumnIndexOrThrow(COLUMN_FECHA))
                val total = getDouble(getColumnIndexOrThrow(COLUMN_TOTAL))
                purchaseList.add(PurchaseRecord(date, total))
            }
        }
        cursor.close()
        return purchaseList
    }
}
```

LoginActivity.kt (Pantalla de Login)



```
class MainActivity : AppCompatAc
    override fun onCreate(savedInstanceSta
            btnRegister.visibility = Butt

            btnMenu.visibility = Button.Vi
            btnHistorial.visibility = Butt
        } else {
            btnLogin.visibility = Button.
            btnRegister.visibility = Butt

            btnMenu.visibility = Button.G
            btnHistorial.visibility = Butt
        }

        btnLogin.setOnClickListener {
            val intent = Intent( packageCo
            startActivity(intent)
        }

        btnRegister.setOnClickListener {
            val intent = Intent( packageCo
            startActivity(intent)
        }

        btnMenu.setOnClickListener {
            val intent = Intent( packageCo
            startActivity(intent)
        }
```

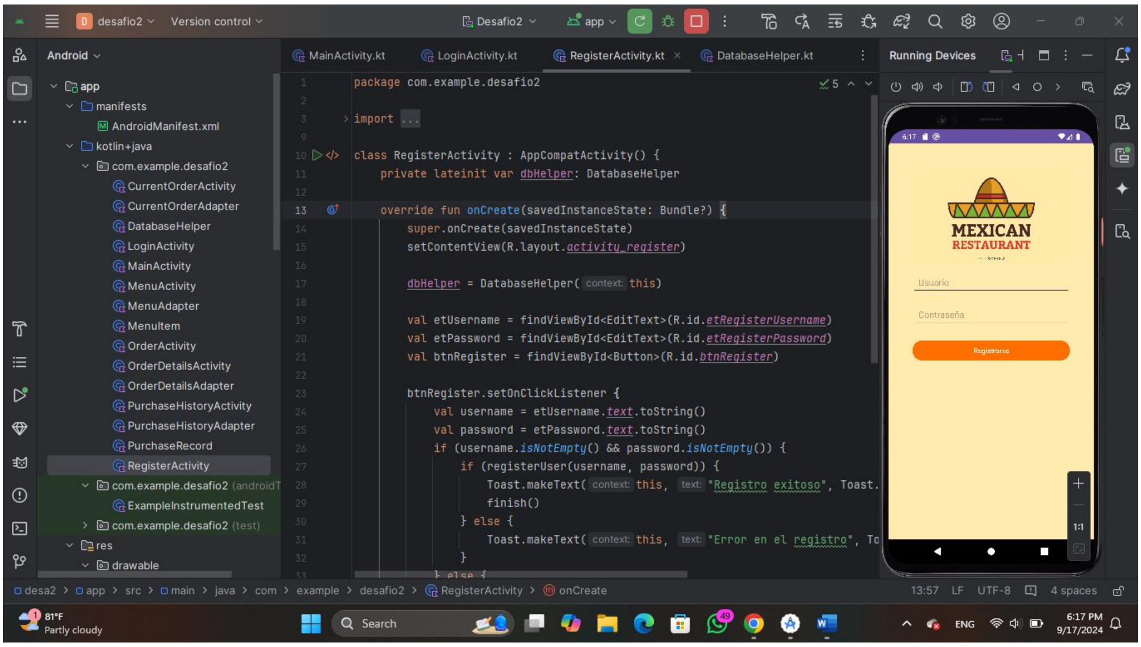
Codigo:

package com.example.desafio2

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import android.content.SharedPreferences

```
class LoginActivity : AppCompatActivity() {
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        dbHelper = DatabaseHelper(this)
        sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE)

        val etUsername = findViewById<EditText>(R.id.etUsername)
        val etPassword = findViewById<EditText>(R.id.etPassword)
        val btnLogin = findViewById<Button>(R.id.btnLogin)
        val btnGoToRegister = findViewById<Button>(R.id.btnGoToRegister)

        btnLogin.setOnClickListener {
            val username = etUsername.text.toString()
            val password = etPassword.text.toString()
            if (validateUser(username, password)) {
                saveLoginState(username)
                Toast.makeText(this, "Login exitoso", Toast.LENGTH_SHORT).show()
                val intent = Intent(this, MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                Toast.makeText(this, "Usuario o contraseña incorrectos",
Toast.LENGTH_SHORT).show()
            }
        }

        btnGoToRegister.setOnClickListener {
            val intent = Intent(this, RegisterActivity::class.java)
            startActivity(intent)
        }
    }

    private fun validateUser(username: String, password: String): Boolean {
        val db = dbHelper.readableDatabase
        val cursor = db.query(
            "usuarios",
            arrayOf("id"),
            "usuario = ? AND password = ?",
            arrayOf(username, password),
            null,
            null,
            null
        )
        val isValid = cursor.count > 0
        cursor.close()
        return isValid
    }

    private fun saveLoginState(username: String) {
        val editor = sharedPreferences.edit()
        editor.putBoolean("logged_in", true)
        editor.putString("username", username)
        editor.apply()
    }
}
```

RegisterActivity.kt (Pantalla de Registro)

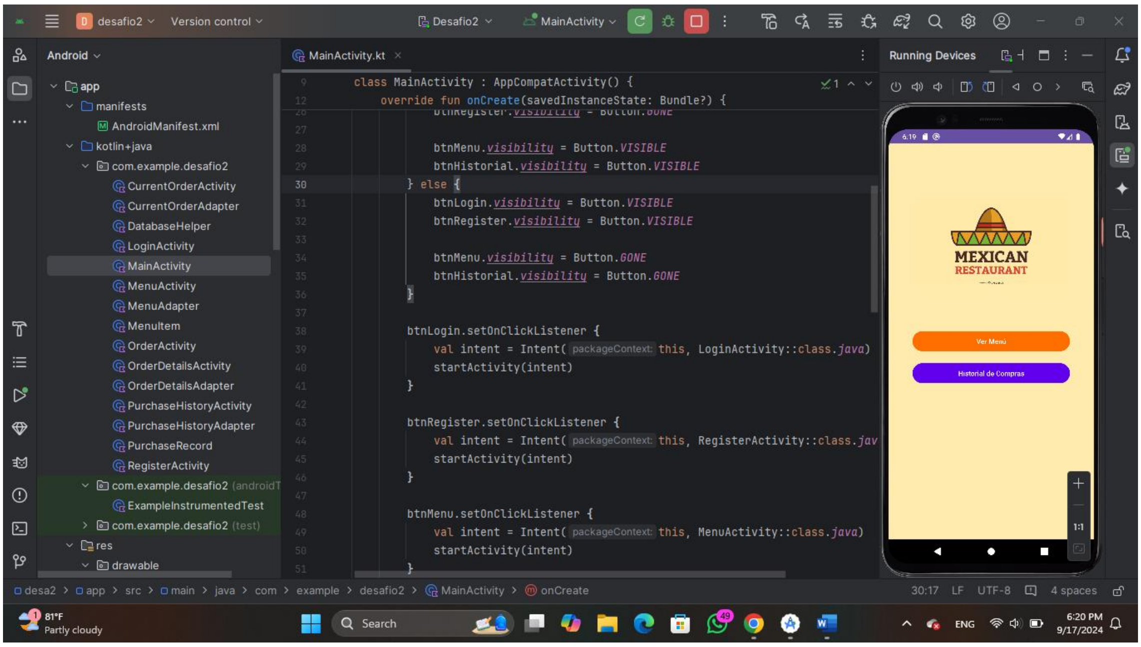


```
package com.example.desafio2

import ...

class RegisterActivity : AppCompatActivity() {
    private lateinit var dbHelper: DatabaseHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_register)

        dbHelper = DatabaseHelper( context: this)

        val etUsername = findViewById<EditText>(R.id.etRegisterUsername)
        val etPassword = findViewById<EditText>(R.id.etRegisterPassword)
        val btnRegister = findViewById<Button>(R.id.btnRegister)

        btnRegister.setOnClickListener {
            val username = etUsername.text.toString()
            val password = etPassword.text.toString()
            if (username.isNotEmpty() && password.isNotEmpty()) {
                if (registerUser(username, password)) {
                    Toast.makeText( context: this,  text: "Registro exitoso", Toast.
                    finish()
                } else {
                    Toast.makeText( context: this,  text: "Error en el registro", To
                }
```

Codigo:

```
package com.example.desafio2

import android.content.ContentValues
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class RegisterActivity : AppCompatActivity() {
    private lateinit var dbHelper: DatabaseHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_register)

        dbHelper = DatabaseHelper(this)

        val etUsername = findViewById<EditText>(R.id.etRegisterUsername)
        val etPassword = findViewById<EditText>(R.id.etRegisterPassword)
        val btnRegister = findViewById<Button>(R.id.btnRegister)

        btnRegister.setOnClickListener {
            val username = etUsername.text.toString()
            val password = etPassword.text.toString()
            if (username.isNotEmpty() && password.isNotEmpty()) {
                if (registerUser(username, password)) {
                    Toast.makeText(this, "Registro exitoso", Toast.LENGTH_SHORT).show()
                    finish()
                } else {
                    Toast.makeText(this, "Error en el registro",
Toast.LENGTH_SHORT).show()
                }
            } else {
                Toast.makeText(this, "Por favor, complete todos los campos",
Toast.LENGTH_SHORT).show()
            }
        }
    }

    private fun registerUser(username: String, password: String): Boolean {
        val db = dbHelper.writableDatabase
        val values = ContentValues().apply {
            put("usuario", username)
            put("password", password)
        }
        val newRowId = db.insert("usuarios", null, values)
        return newRowId != -1L
    }
}
```

MainActivity.kt (Pantalla del usuario ya logueado, puede ver Menu e historial)

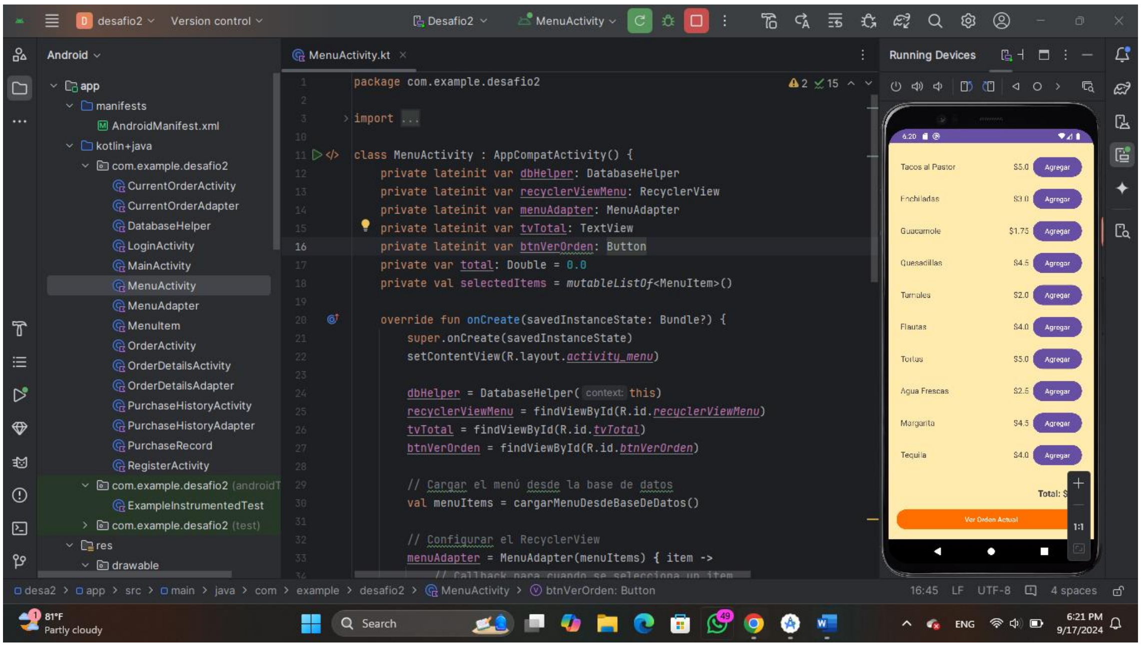


```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
            btnRegister.visibility = Button.GONE

            btnMenu.visibility = Button.VISIBLE
            btnHistorial.visibility = Button.VISIBLE
        } else {
            btnLogin.visibility = Button.VISIBLE
            btnRegister.visibility = Button.VISIBLE

            btnMenu.visibility = Button.GONE
            btnHistorial.visibility = Button.GONE
        }

        btnLogin.setOnClickListener {
            val intent = Intent( packageContext: this, LoginActivity::class.java)
            startActivity(intent)
        }

        btnRegister.setOnClickListener {
            val intent = Intent( packageContext: this, RegisterActivity::class.jav
            startActivity(intent)
        }

        btnMenu.setOnClickListener {
            val intent = Intent( packageContext: this, MenuActivity::class.java)
            startActivity(intent)
        }
```

Codigo:

```
package com.example.desafio2

import android.content.Intent
import android.content.SharedPreferences
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    private lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE)
        val isLoggedIn = sharedPreferences.getBoolean("logged_in", false)

        val btnLogin = findViewById<Button>(R.id.btnLogin)
        val btnRegister = findViewById<Button>(R.id.btnRegister)
        val btnMenu = findViewById<Button>(R.id.btnMenu)
        val btnHistorial = findViewById<Button>(R.id.btnHistorial)

        if (isLoggedIn) {
            btnLogin.visibility = Button.GONE
            btnRegister.visibility = Button.GONE

            btnMenu.visibility = Button.VISIBLE
            btnHistorial.visibility = Button.VISIBLE
        } else {
            btnLogin.visibility = Button.VISIBLE
            btnRegister.visibility = Button.VISIBLE

            btnMenu.visibility = Button.GONE
            btnHistorial.visibility = Button.GONE
        }

        btnLogin.setOnClickListener {
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
        }

        btnRegister.setOnClickListener {
            val intent = Intent(this, RegisterActivity::class.java)
            startActivity(intent)
        }

        btnMenu.setOnClickListener {
            val intent = Intent(this, MenuActivity::class.java)
            startActivity(intent)
        }


        btnHistorial.setOnClickListener {
            val intent = Intent(this, PurchaseHistoryActivity::class.java)
            startActivity(intent)
        }
    }
}
```

MenuActivity.kt (Pantalla del menú y también puede ver la orden actual)



```
package com.example.desafio2

import ...

class MenuActivity : AppCompatActivity() {
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var recyclerViewMenu: RecyclerView
    private lateinit var menuAdapter: MenuAdapter
    private lateinit var tvTotal: TextView
    private lateinit var btnVerOrden: Button
    private var total: Double = 0.0
    private val selectedItems = mutableListOf<MenuItem>()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_menu)

        dbHelper = DatabaseHelper( context: this)
        recyclerViewMenu = findViewById(R.id.recyclerViewMenu)
        tvTotal = findViewById(R.id.tvTotal)
        btnVerOrden = findViewById(R.id.btnVerOrden)

        // Cargar el menú desde la base de datos
        val menuItems = cargarMenuDesdeBaseDeDatos()

        // Configurar el RecyclerView
        menuAdapter = MenuAdapter(menuItems) { item ->
            // Callback para cuando se selecciona un item
```

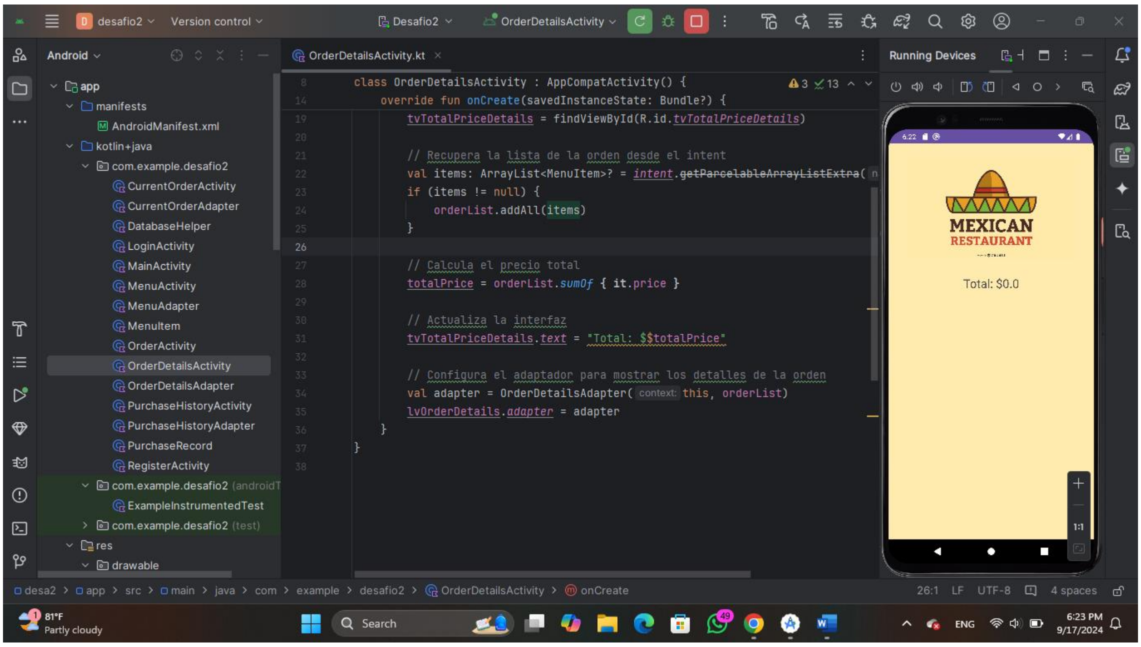
Tacos al Pastor	$5.0	Agregar
Enchiladas	$3.0	Agregar
Guacamole	$1.75	Agregar
Quesadillas	$4.5	Agregar
Tamales	$2.0	Agregar
Flautas	$4.0	Agregar
Tortas	$5.0	Agregar
Agua Frescas	$2.5	Agregar
Margarita	$4.5	Agregar
Tequila	$4.0	Agregar

Total: $

Ver Orden Actual

```
package com.example.desafio2

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class MenuActivity : AppCompatActivity() {
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var recyclerViewMenu: RecyclerView
    private lateinit var menuAdapter: MenuAdapter
    private lateinit var tvTotal: TextView
    private lateinit var btnVerOrden: Button
    private var total: Double = 0.0
    private val selectedItems = mutableListOf<MenuItem>()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_menu)

        dbHelper = DatabaseHelper(this)
        recyclerViewMenu = findViewById(R.id.recyclerViewMenu)
        tvTotal = findViewById(R.id.tvTotal)
        btnVerOrden = findViewById(R.id.btnVerOrden)

        // Cargar el menú desde la base de datos
        val menuItems = cargarMenuDesdeBaseDeDatos()

        // Configurar el RecyclerView
        menuAdapter = MenuAdapter(menuItems) { item ->
            // Callback para cuando se selecciona un item
            selectedItems.add(item)
            total += item.price
            actualizarTotal()
        }
        recyclerViewMenu.adapter = menuAdapter
        recyclerViewMenu.layoutManager = LinearLayoutManager(this)

        btnVerOrden.setOnClickListener {
            val intent = Intent(this, CurrentOrderActivity::class.java)
            intent.putParcelableArrayListExtra("selected_items", ArrayList(selectedItems))
            intent.putExtra("total", total)
            startActivity(intent)
        }
    }

    private fun cargarMenuDesdeBaseDeDatos(): ArrayList<MenuItem> {
        val menuItems = ArrayList<MenuItem>()
        val db = dbHelper.readableDatabase

        val cursor = db.rawQuery("SELECT nombre, precio FROM menu", null)

        if (cursor.moveToFirst()) {
            do {
                val nombre = cursor.getString(cursor.getColumnIndexOrThrow("nombre"))
                val precio = cursor.getDouble(cursor.getColumnIndexOrThrow("precio"))
                menuItems.add(MenuItem(nombre, precio))
            } while (cursor.moveToNext())
        }

        cursor.close()
        return menuItems
    }

    private fun actualizarTotal() {
        tvTotal.text = String.format("Total: $%.2f", total)
    }
}
```

OrdenDetailsActivity.kt (Pantalla donde se vera la orden actual)



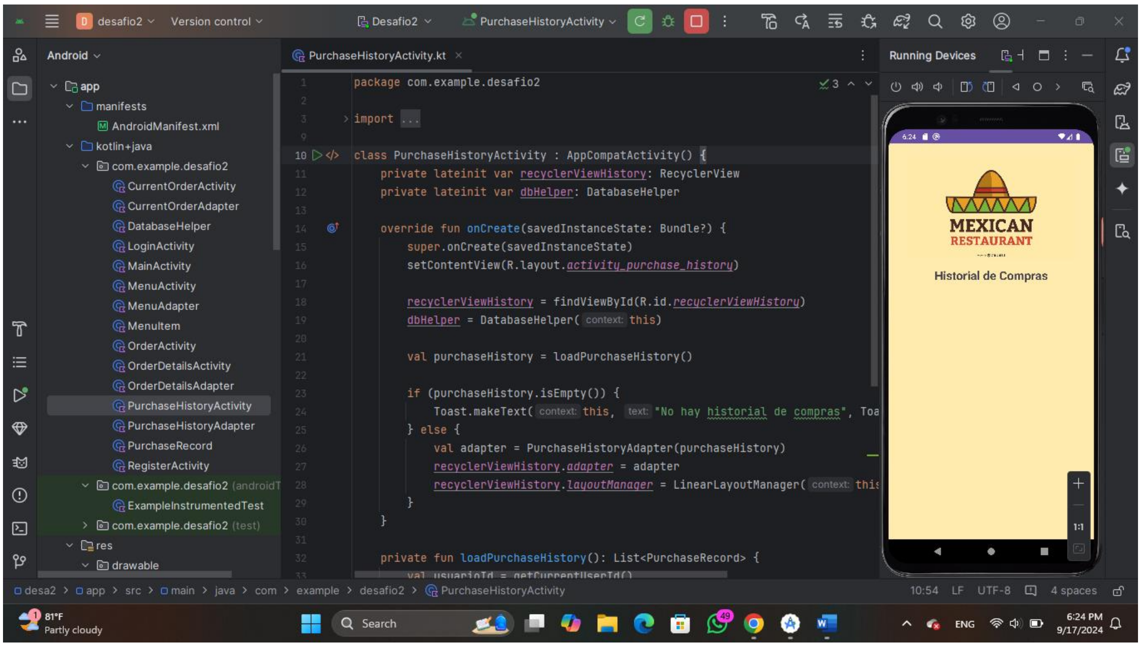
```
class OrderDetailsActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        tvTotalPriceDetails = findViewById(R.id.tvTotalPriceDetails)

        // Recupera la lista de la orden desde el intent
        val items: ArrayList<MenuItem>? = intent.getParcelableArrayListExtra( n
        if (items != null) {
            orderList.addAll(items)
        }

        // Calcula el precio total
        totalPrice = orderList.sumOf { it.price }

        // Actualiza la interfaz
        tvTotalPriceDetails.text = "Total: $$totalPrice"

        // Configura el adaptador para mostrar los detalles de la orden
        val adapter = OrderDetailsAdapter( context: this, orderList)
        lvOrderDetails.adapter = adapter
    }
}
```

Codigo:

```
package com.example.desafio2

import android.os.Bundle
import android.widget.ListView
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
```

```
class OrderDetailsActivity : AppCompatActivity() {
    private lateinit var lvOrderDetails: ListView
    private lateinit var tvTotalPriceDetails: TextView
    private var totalPrice: Double = 0.0
    private val orderList: MutableList<MenuItem> = mutableListOf()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_order_details)

        lvOrderDetails = findViewById(R.id.lvOrderDetails)
        tvTotalPriceDetails = findViewById(R.id.tvTotalPriceDetails)

        // Recupera la lista de la orden desde el intent
        val items: ArrayList<MenuItem>? = intent.getParcelableArrayListExtra("order_list")
        if (items != null) {
            orderList.addAll(items)
        }

        // Calcula el precio total
        totalPrice = orderList.sumOf { it.price }

        // Actualiza la interfaz
        tvTotalPriceDetails.text = "Total: $$totalPrice"

        // Configura el adaptador para mostrar los detalles de la orden
        val adapter = OrderDetailsAdapter(this, orderList)
        lvOrderDetails.adapter = adapter
    }
}
```

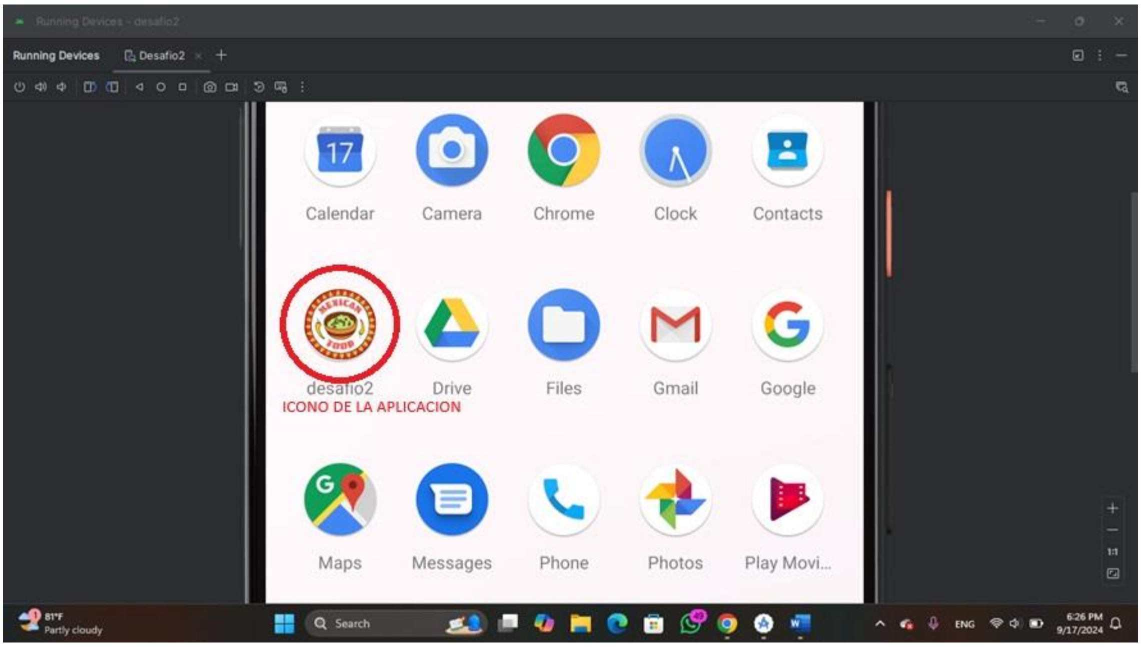
PurchaseHistoryActivity.kt (Pantalla para ver el historial de compras)



```
package com.example.desafio2

import ...

class PurchaseHistoryActivity : AppCompatActivity() {
    private lateinit var recyclerViewHistory: RecyclerView
    private lateinit var dbHelper: DatabaseHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_purchase_history)

        recyclerViewHistory = findViewById(R.id.recyclerViewHistory)
        dbHelper = DatabaseHelper( context: this)

        val purchaseHistory = loadPurchaseHistory()

        if (purchaseHistory.isEmpty()) {
            Toast.makeText( context: this,  text: "No hay historial de compras", Toa
        } else {
            val adapter = PurchaseHistoryAdapter(purchaseHistory)
            recyclerViewHistory.adapter = adapter
            recyclerViewHistory.layoutManager = LinearLayoutManager( context: this
        }
    }

    private fun loadPurchaseHistory(): List<PurchaseRecord> {
        val usuarioId = getCurrentUserId()
```

```
package com.example.desafio2

import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class PurchaseHistoryActivity : AppCompatActivity() {
    private lateinit var recyclerViewHistory: RecyclerView
    private lateinit var dbHelper: DatabaseHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_purchase_history)

        recyclerViewHistory = findViewById(R.id.recyclerViewHistory)
        dbHelper = DatabaseHelper(this)

        val purchaseHistory = loadPurchaseHistory()

        if (purchaseHistory.isEmpty()) {
            Toast.makeText(this, "No hay historial de compras",
Toast.LENGTH_SHORT).show()
        } else {
            val adapter = PurchaseHistoryAdapter(purchaseHistory)
            recyclerViewHistory.adapter = adapter
            recyclerViewHistory.layoutManager = LinearLayoutManager(this)
        }
    }

    private fun loadPurchaseHistory(): List<PurchaseRecord> {
        val usuarioId = getCurrentUserId()
        val history = dbHelper.getPurchaseHistory(usuarioId)
        Log.d("PurchaseHistory", "Loaded ${history.size} records for user $usuarioId")
        return history
    }

    private fun getCurrentUserId(): Int {
        // TODO: Implement proper user ID retrieval
        return 1 // Using a default value for now
    }


}
```

Iconos de la aplicación



ICONO DE LA APLICACION

Imágenes

MEXICAN
FOOD

Link del video:

https://udbedu-my.sharepoint.com/:v:/g/personal/gg222955_alumno_udb_edu_sv/EaC_Vr0CaihEieZ-yPMYzdUBMBm_3Ng7wI8eDYd1-i9vXQ?e=hrShuj&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D