



INSTITUTO POLITÉCNICO NACIONAL  
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS  
SOCIALES Y ADMINISTRATIVAS  
PRIMER EXAMEN PARCIAL – DISPOSITIVOS MÓVILES/PROGR. MÓVIL



Pérez	García	Jenny Alondra	SECUENCIA:	6NM61
APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE(S)		
CALIF. EXAMEN: <b>(58)</b>	EVAL. CONTINUA (Tareas + Prácticas + Participaciones + Extras): <b>1.5+2+0.5</b>		CALIF. FINAL: <b>(P)</b>	

**INSTRUCCIONES:**

- Se permite solamente el uso de **bolígrafo de tinta azul o verde**, así como de goma y sacapuntas.
- **Toda respuesta de teoría que esté resuelta con lápiz se considera incorrecta.**
- Lee detenidamente cada pregunta y selecciona la respuesta correcta.

PREGUNTA	OPCIONES	RESPUESTA
1. Los inicios de los dispositivos móviles surgieron durante:	a) II guerra mundial b) Guerra fría c) Guerra de Kwait d) Terremoto de Chile.	b
2. El propósito de los primeros dispositivos móviles fue:	a) Comunicar a distintos presidentes durante actos importantes b) Mantener comunicación entre distintos rangos militares para establecer estrategias de guerra c) Desarrollar una máquina capaz de calcular operaciones complejas. d) Ninguna de las anteriores	b ✓
3. Un dispositivo móvil se define como:	a) Elemento electrónico que envía y recibe imágenes y datos b) Elemento electrónico capaz de establecer una comunicación entre dos puntos a distancias largas y brindar entretenimiento en momentos de ocio c) Artefacto de uso exclusivo para amenazas y tomas de evidencias en formato electrónico d) Todas las anteriores	d ✓
4. Son tipos de plataformas de desarrollos para dispositivos móviles:	a) Nativas, multiplataforma y de computadora b) Híbridas, hibernales y web c) Nativas, multiplataforma e híbridas d) Web, nativas y stand-alone	c ✓
5. De acuerdo con este principio, permite controlar los datos que maneja el usuario	a) Principio de incertidumbre b) Principio de Arquímedes c) Principio de control de interfaz de usuario d) Principio de separación de problemas	c ✓
6. Nos permite manejar distintos elementos para que conformen el todo dentro de la aplicación	a) Principio de control de interfaz de usuario b) Principio de Lewis c) Principio de separación de problemas d) Principio de oferta y demanda	c ✓
7. Establece que son todos los elementos con lo que contará una app	a) Arquitectura empresarial b) Arquitectura organizacional c) Arquitectura móvil d) Arquitectura industrial	c ✓



INSTITUTO POLITÉCNICO NACIONAL  
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS  
SOCIALES Y ADMINISTRATIVAS



8. Dentro de este nivel, permite concentrar las decisiones y validaciones de la app	a) Capa de ozono b) Capa de datos c) Capa de negocio d) Capa de grasa	C
9. A través de ella, es lo que el usuario puede visualizar para el manejo de información	a) Capa de grasa b) Capa de tierra c) Capa de presentación d) Capa de datos	C
10. Nos permite identificar si la app cuenta con fallas antes de ser publicada	a) Etapa de planeación b) Etapa de mantenimiento c) Etapa de codificación d) Etapa de pruebas	d

Valor de la sección: 1 puntos

II. De acuerdo con los siguientes puntos solicitados, realice:

- A) El código correspondiente
  - B) Código funcionando y salida de pantalla
  - C) El algoritmo de la solución
  - D) El diagrama de flujo de la solución
  - E) El prototipo a utilizar para la interfaz de la solución
1. Crea un programa que te pida una cantidad en miligramos para una poción multijugos, el valor debe ser de tipo entero, si el valor es mayor a 100 imprime “¡Felicitaciones, es una buena poción multijugos!”, de lo contrario imprime “La poción es mediocre, sangre sucia inmunda”. Deberá aparecer un botón con la etiqueta “Nuevo” para reiniciar el programa una vez concluida la validación. Además, la información deberá ingresarse mediante el teclado. Si el programa se reinicia tres veces, éste en la última iteración deberá cerrarse en automático una vez mostrado el resultado.

Valor: 5 puntos.

# Algoritmo de Solución

Inicio

reiniciar = 1

repetir:

    Imprimir "Ingresar cantidad de miligramos para la opción"

    Leer cantidad

    Si cantidad > 100:

        Imprimir "Felicitaciones, es una buena opción multilugos"

    Si no:

        Imprimir "La opción es mediocre, sangre Siria inmundas"

    Si se presiona "Nuevo":

        Si reiniciar <= 3:

            Limpiar cantidad

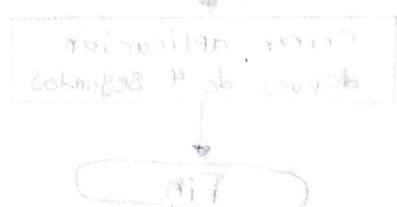
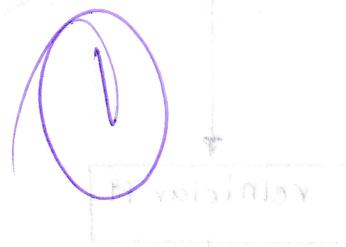
            Limpiar salida

            reiniciar = reiniciar + 1

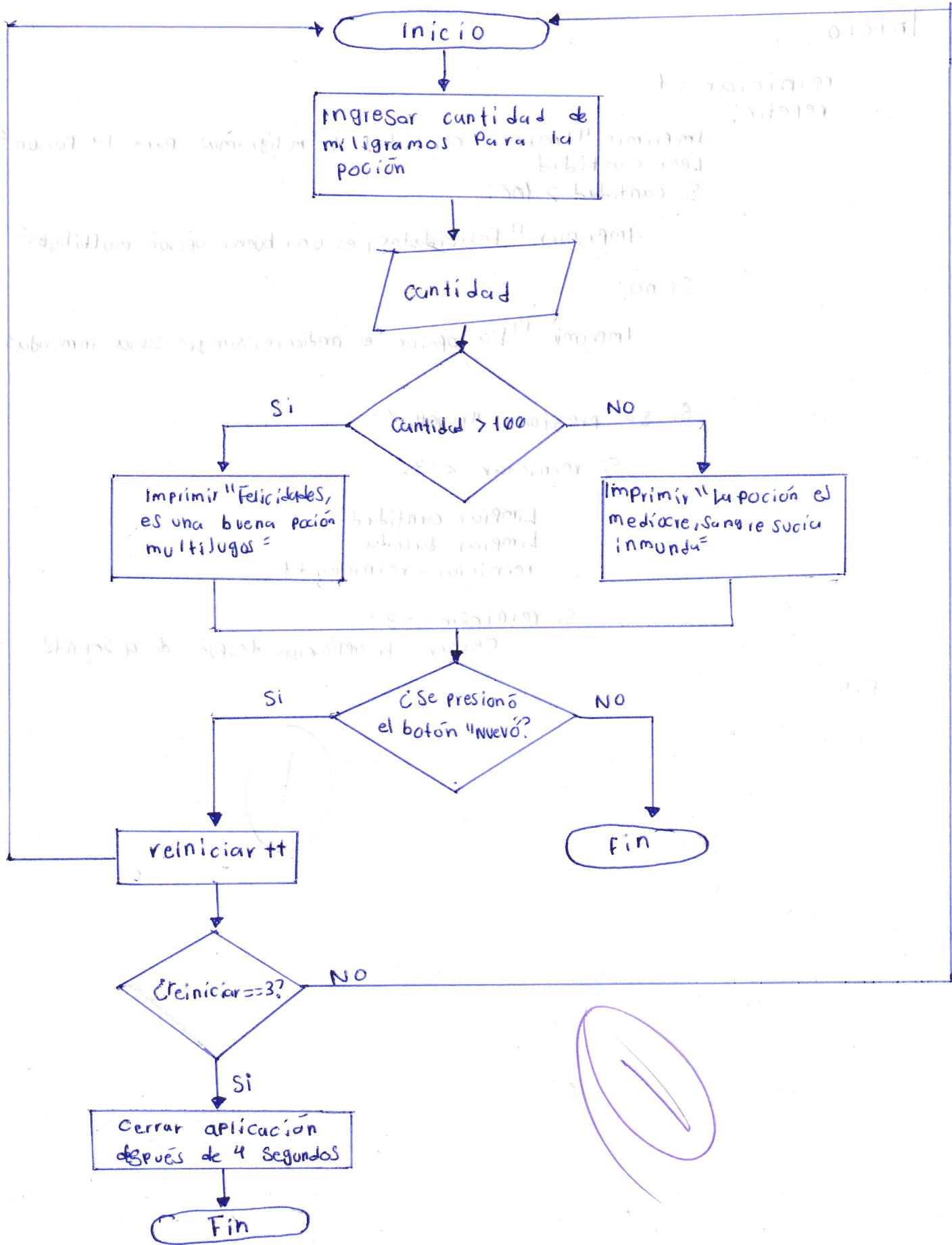
        Si reiniciar == 3:

            Cerrar la aplicación después de 4 segundos

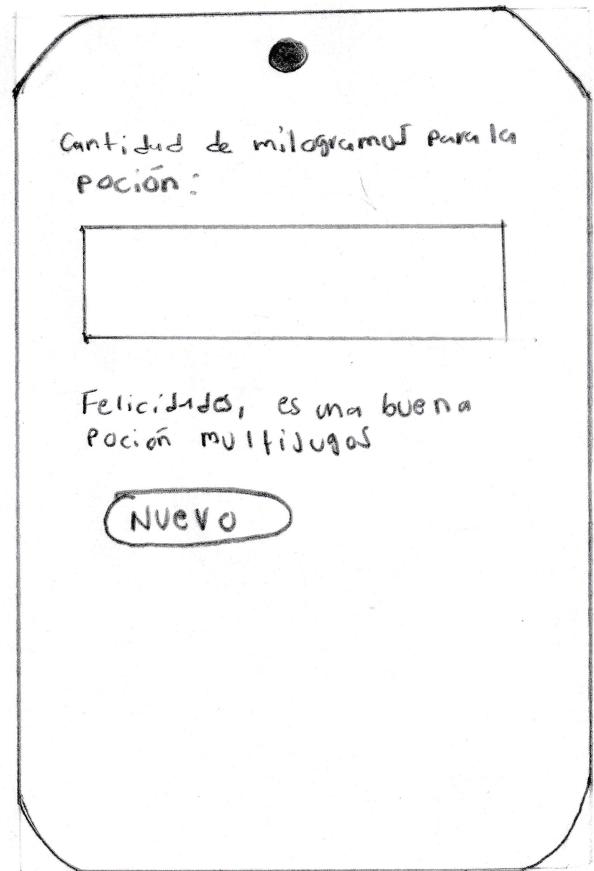
Fin



# Diagrama de flujo de la solución



## E) Prototipo a usar para la interfaz de la solución



## Código

a) MainActivity.kt

```
package com.Examen

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardActions
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.input.ImeAction
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.platform.LocalSoftwareKeyboardController
import androidx.compose.ui.Alignment
import androidx.compose.ui.graphics.Color
import kotlin.system.exitProcess

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            PantallaExamen()
        }
    }
}

@Composable
fun PantallaExamen() {
    var cantidad by remember { mutableStateOf("") }
    var salida by remember { mutableStateOf("") }
    var reiniciar by remember { mutableStateOf(0) }

    val teclado = LocalSoftwareKeyboardController.current

    Box(
        modifier = Modifier
            .padding(45.dp),
        contentAlignment = Alignment.Center
    ) {

```

```

Column(modifier = Modifier.padding(45.dp)) {
    Text("Cantidad de miligramos para la poción:")

    TextField(
        value = cantidad,
        onValueChange = { cantidad = it },
        modifier = Modifier.fillMaxWidth(),
        keyboardOptions = KeyboardOptions.Default.copy(imeAction = ImeAction.Done),
        keyboardActions = KeyboardActions(
            onDone = { //Funciona cuando se da enter
                //Condicion
                val cantidaddf = cantidad.toFloat()
                if (cantidaddf > 100) {
                    salida = "Felicidades, es una buena poción multijugos"
                } else {
                    salida = "La poción es mediocre, sangre sucia inmunda"
                }
                println(salida)
                teclado?.hide()
            }
        )
    )

    Text(text = salida)
    Button(
        onClick = {
            if (reiniciar <=3) { //Condiciones del botón
                cantidad = ""
                salida = ""
                reiniciar++
                if (reiniciar ==3) {
                    android.os.Handler().postDelayed({
                        exitProcess(0) // Cerrar la app despues de 4 seg
                    }, 4000)
                }
            }
        },
        modifier = Modifier
            .padding(top = 22.dp),
        colors = ButtonDefaults.buttonColors( //Colores del boton
            containerColor = Color.Blue,
            contentColor = Color.White
        )
    )
}

```

```

        ) { Text("Nuevo") }
    }
}
}

@Preview(showBackground = true)
@Composable
fun PantallaExamenPreview() {
    PantallaExamen()
}

```

b) AndroidManifest.xml

```

</manifest>

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Examen"
        tools:targetApi="31">
        <activity
            android:name="com.Examen.MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.Examen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

## Código funcionando y salida de pantalla

