

FACULTAD DE INGENIERÍAS Y ARQUITECTURA

NOTA	

EXAMEN PARCIAL

APELLIDOS Y NOMBRES			
CURSO	Programación para Dispositivos Móviles	SEMESTRE	Sexto
DOCENTE	Josue Miguel Flores Parra	FECHA: 16 de Octubre del 2025	
DURACIÓN	3 horas.		

Examen Parcial: Juego de Colores

Desarrolla una aplicación usando Kotlin y Android Studio. El juego pondrá a prueba tu capacidad de crear una app interactiva con múltiples fragments, navegación, gestión de estados del ciclo de vida, la interacción avanzada con RecyclerView, manejo de ventanas de diálogo, lógica condicional y elementos visuales.

Objetivo:

Crear un juego donde el usuario debe presionar el botón que coincida con el color que aparece en pantalla. El objetivo es obtener la mayor cantidad de aciertos en 30 segundos, mostrar el historial de puntajes en la sesión (sin usar Room/BD).

☼ ¿Qué debe tener la aplicación?

- 1. Fragmento de bienvenida (WelcomeFragment):
 - Título del juego.
 - Mensaje de bienvenida y reglas del juego (Mostrar en un AlertDialog)
 - Botón "Iniciar juego" para comenzar.
 - Interfaz libre pueden incluir imágenes.
- 2. Fragmento del juego (GameFragment):
 - Cuadro que muestra un color aleatorio a adivinar (rojo, verde, azul, amarillo, etc).
 - Botones de colores para responder.
 - Cada acierto suma un punto y cambia el color.
 - Temporizador de 30 segundos.
 - Muestre el puntaje actual y el tiempo restante.
 - El juego finaliza al agotar el tiempo.

- 3. Fragmento de resultados (ResultFragment):
 - Muestra el puntaje final de la partida actual.
 - Muestra el puntaje más alto histórico (Usar SharedPreferences).
 - Contiene un RecyclerView para mostrar el Historial de Puntajes sin persintencia. Puede solo visualizar el historial de las partidas de la sesión.
 - Botón para volver a jugar.

Funcionalidades Adicionales (elige al menos una, es obligatoria):

- Agrega sonidos al juego (por ejemplo, cuando acierta, se equivoca o música de fondo).
- Harda el historial de puntajes en una base de datos usando **Room**, recupera el puntaje más alto con una consulta a la base de datos usando **Room**.
- Paplica animaciones a los botones o colores que aparecen.
- Implementa **niveles de dificultad** (Ejemplos, **Efecto Stroop**: Muestra un texto que dice "Rojo" en color azul, El jugador debe presionar el color **del texto**, no el que dice la palabra, aumentar el número de colores conforme se avanza, etc).

¿Qué se debe practicar?

- Navegación entre Fragments y paso de datos.
- Uso de temporizador (CountDownTimer).
- Interacción con el usuario a través de botones y feedback.
- Manejo de estados y lógica condicional básica.
- Diseño de UI con ConstraintLayout.
- Control de errores y validación del input del usuario.
- Organización del código y buenas prácticas de programación en Kotlin.
- Uso adecuado de recursos visuales (colores, imágenes, etc).
- Almacenamiento básico de datos usando SharedPreferences
- Uso de AlertDialog.
- Almacenamiento de datos usando Room (si eliges esa funcionalidad adicional).
- Incorporación de medios como sonidos o animaciones opcionales (si eliges esa funcionalidad adicional).

Tips útiles:

- Usa recursos de cadenas en strings.xml.
- Usa recursos de colores en colors.xml.
- Puedes usar una función para generar el color aleatorio.
- Para el temporizador, revisa la clase CountDownTimer.
- Para el puntaje más alto histórico, revisa SharedPreferences.