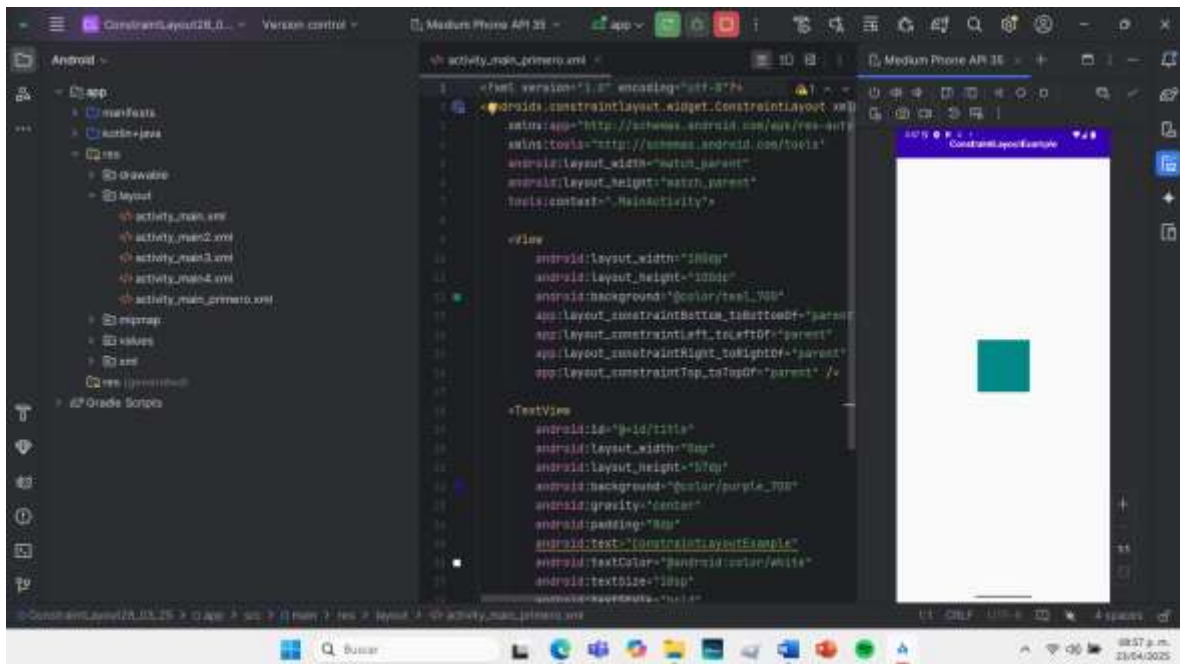


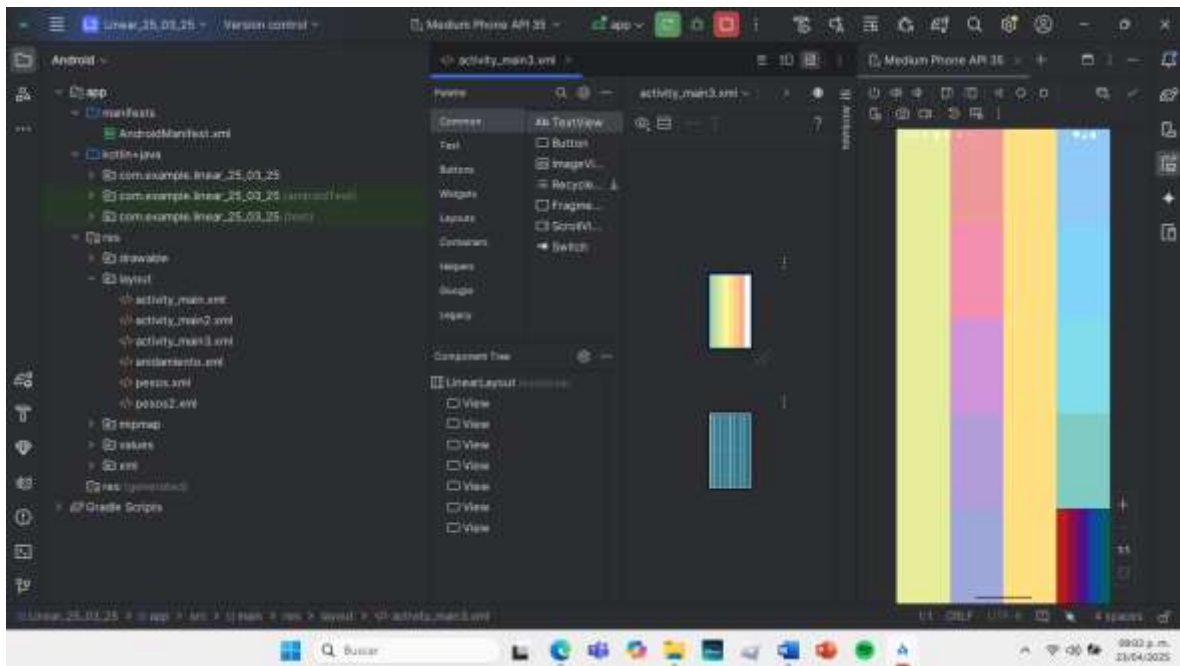
Programa 1:



Documentación:

Este layout XML define una interfaz básica usando ConstraintLayout como contenedor principal, en el cual se incluyen dos vistas: una View cuadrada de 100dp por 100dp con fondo color teal (@color/teal_700), centrada en la pantalla al estar anclada a todos los bordes del padre, y un TextView identificado como @+id/title, que se ubica en la parte superior, extendiéndose horizontalmente (con layout_width="0dp" para aprovechar todo el espacio entre Start y End del padre) y mostrando el texto "ConstraintLayoutExample" centrado, en color blanco y con fondo púrpura (@color/purple_700). Este diseño es útil para mostrar un título fijo en la parte superior mientras se mantiene un componente decorativo o funcional centrado en pantalla.

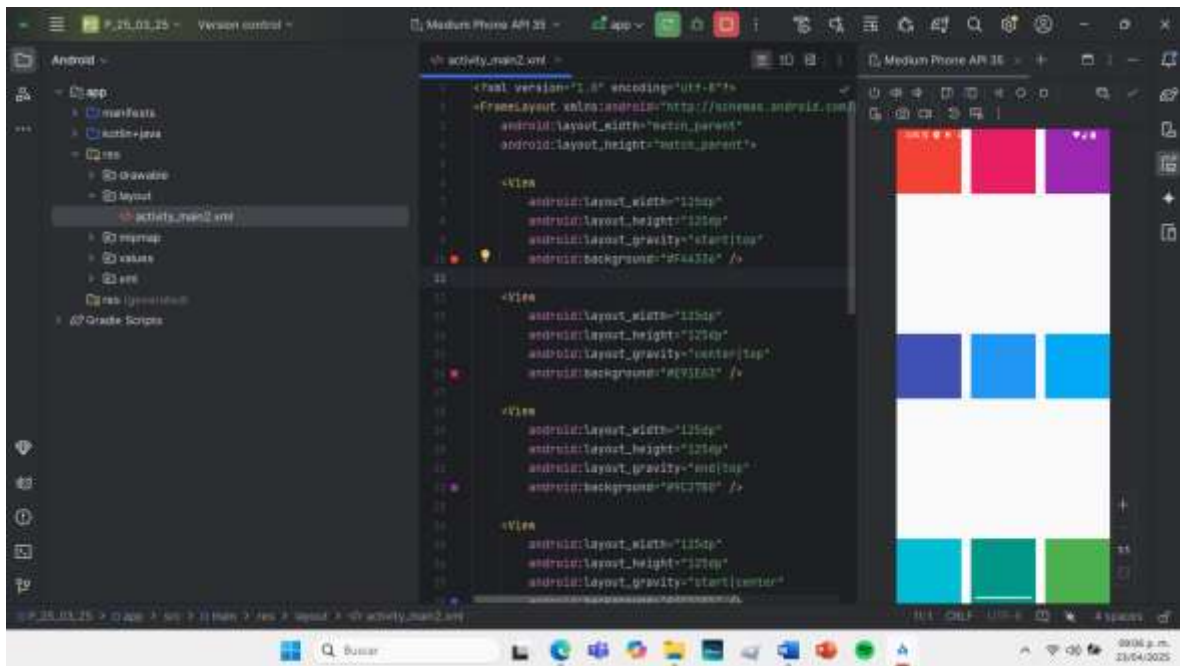
Programa 2:



Documentación:

Este layout XML utiliza un `LinearLayout` con orientación horizontal para mostrar una secuencia de siete `View` verticales, cada una con un ancho fijo de 50dp y altura que ocupa toda la pantalla (`match_parent`). Cada `View` tiene un color de fondo diferente en tonos pastel, lo que genera un diseño visualmente atractivo tipo paleta de colores o barra segmentada. Este tipo de estructura puede servir como ejemplo para prácticas de distribución en línea, visualización de categorías, indicadores de progreso o simplemente para propósitos decorativos dentro de una interfaz gráfica.

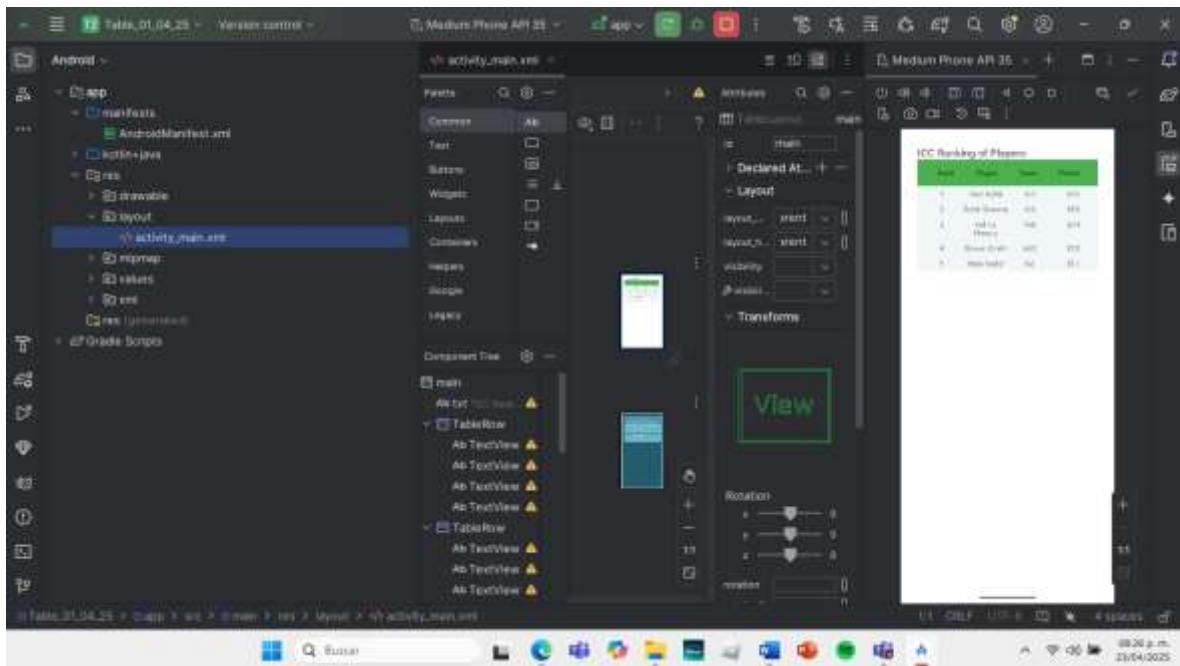
Programa 3:



Documentación:

Este layout XML utiliza un `FrameLayout` como contenedor principal para superponer y posicionar nueve `View` cuadradas de 125dp por 125dp en diferentes regiones de la pantalla usando el atributo `layout_gravity`, que permite ubicar cada vista en posiciones como esquinas (`start|top`, `end|bottom`), bordes centrales (`center|top`, `center|bottom`) y centro absoluto (`center`). Cada `View` tiene un color de fondo distinto, creando una cuadrícula visual de 3x3 sin usar un `GridLayout`, lo cual demuestra cómo `FrameLayout` puede aprovecharse para distribuir elementos con precisión visual sin requerir estructuras complejas, ideal para prototipos visuales o elementos decorativos.

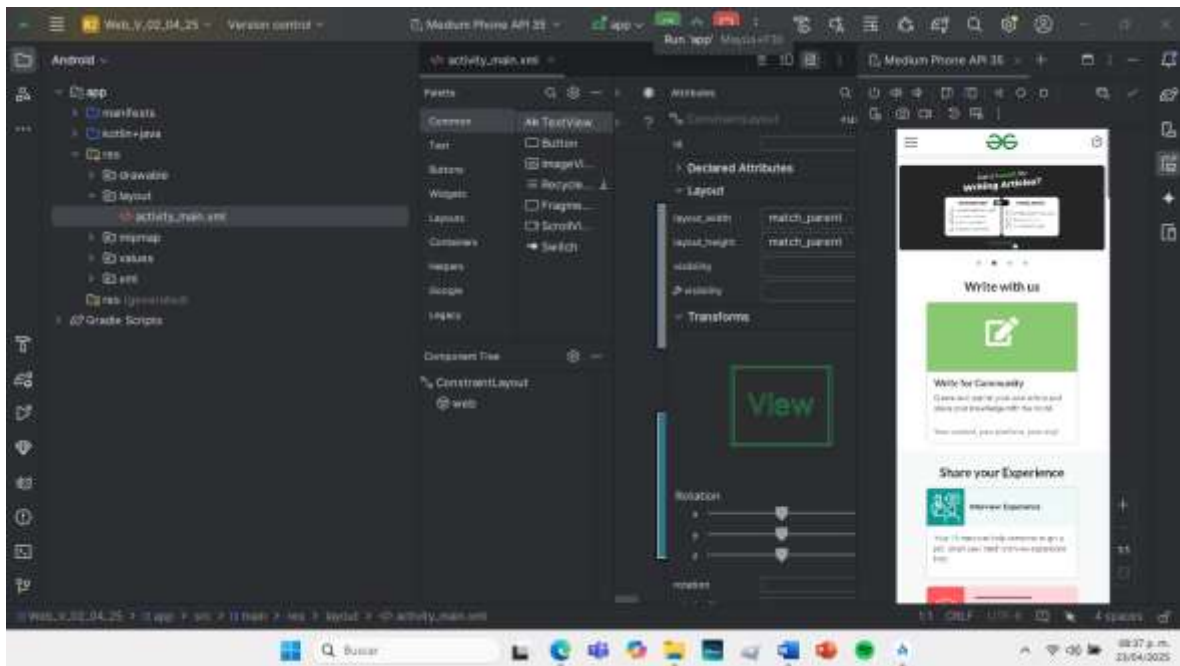
Programa 4:



Documentación:

Este layout XML define una tabla utilizando `TableLayout` para mostrar un ranking de jugadores de cricket (ICC Ranking), con un encabezado destacado y cinco filas de datos. Cada `TableRow` contiene cuatro `TextView` distribuidos equitativamente usando `layout_weight="1"` para mostrar las columnas de Rank, Player, Team y Points, con el texto centrado. El encabezado tiene un fondo verde y texto en negrita, mientras que las filas de datos alternan un fondo claro para mejorar la legibilidad. El diseño proporciona una manera clara y estructurada de presentar datos tabulares en una interfaz Android, ideal para rankings, estadísticas o resultados.

Programa 5:



Documentación:

Este layout XML define una interfaz de usuario simple en Android utilizando un `ConstraintLayout` como contenedor raíz y un único componente `WebView` que ocupa toda la pantalla. El `ConstraintLayout` permite posicionar el `WebView` con restricciones en sus cuatro bordes (Top, Bottom, Start y End) fijadas al contenedor padre, asegurando que se ajuste completamente al tamaño del dispositivo. El `WebView`, identificado con el ID `web`, está diseñado para mostrar contenido web directamente dentro de la aplicación, como páginas HTML, recursos locales o URLs externas. Este diseño es ideal para aplicaciones que requieren integrar navegación o visualización de contenido web sin salir del entorno nativo de la app.