

**Ejercicio 2: Crea una nueva App con Android Studio para calcular el enésimo número primo.****1. Descripción General de la Aplicación**

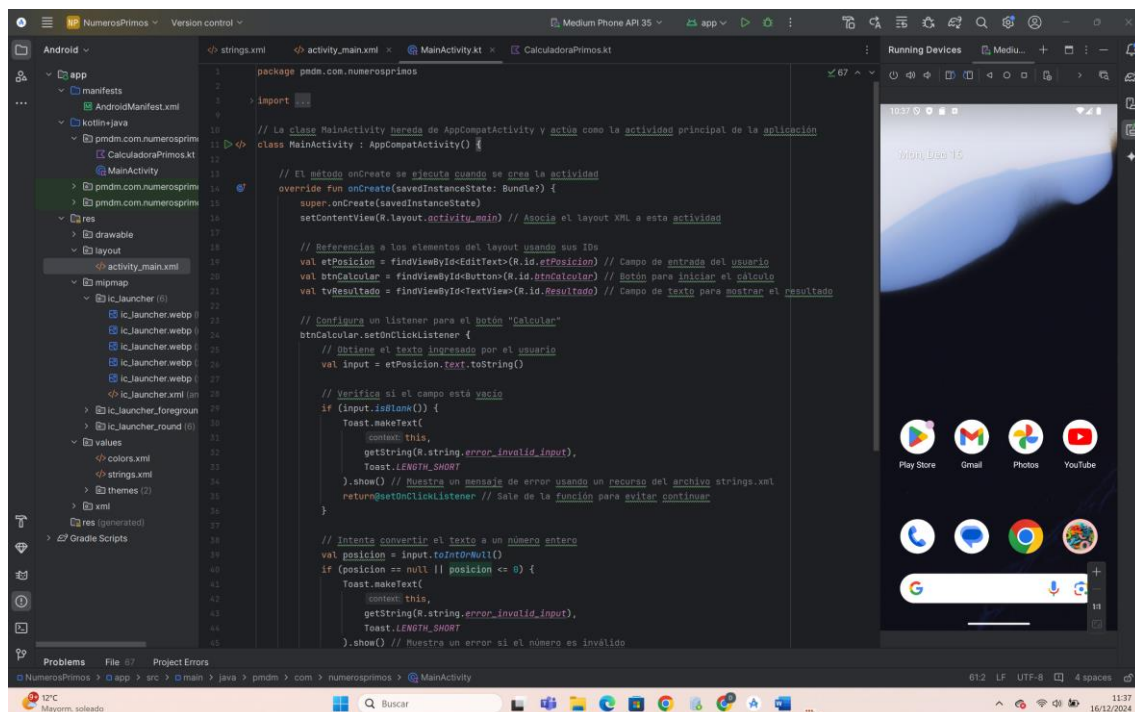
La aplicación "Calculadora de Números Primos" permite al usuario calcular el enésimo número primo. El usuario introduce un valor entero positivo en el campo de texto, pulsa el botón "Calcular", y la aplicación muestra el resultado en pantalla. Además, la aplicación está implementada con buenas prácticas de programación y recursos reutilizables, cumpliendo estándares profesionales.

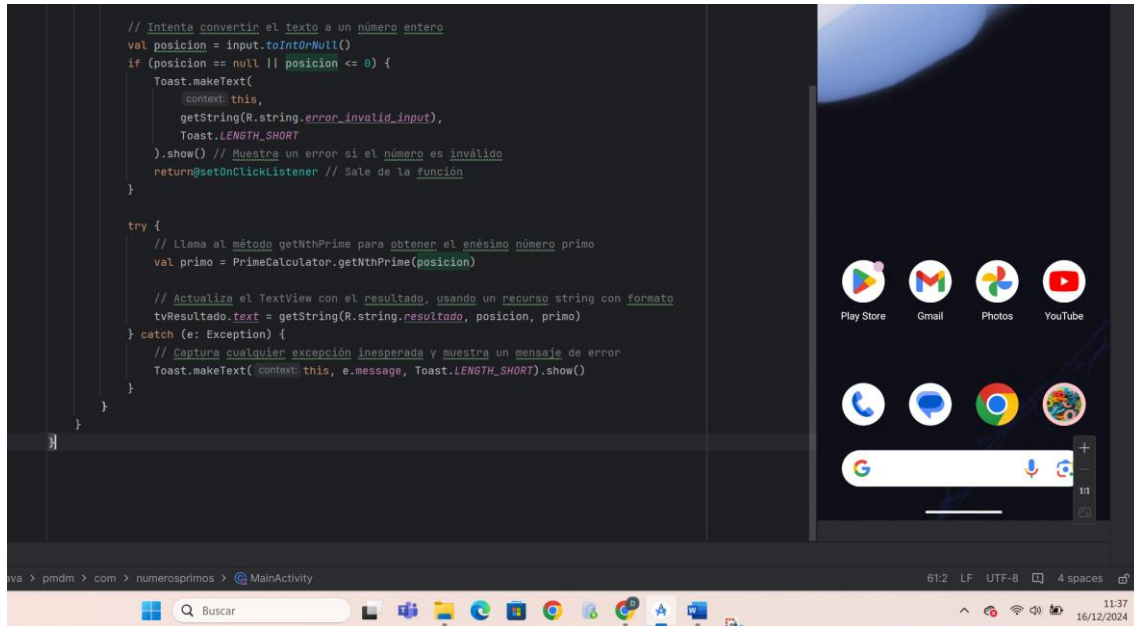
**2. Componentes de la Aplicación****a. Clase MainActivity**

La clase principal MainActivity es la encargada de manejar la interacción del usuario con la interfaz y comunicarse con la lógica de cálculo.

Resumen de las responsabilidades:

- Recuperar las referencias a los componentes del layout utilizando findViewById.
- Validar la entrada del usuario asegurándose de que sea un número entero positivo.
- Manejar eventos del botón "Calcular" mediante un setOnClickListener.
- Llamar al método PrimeCalculator.getNthPrime() para obtener el resultado.
- Mostrar el resultado en la interfaz o un mensaje de error en caso de entrada inválida.



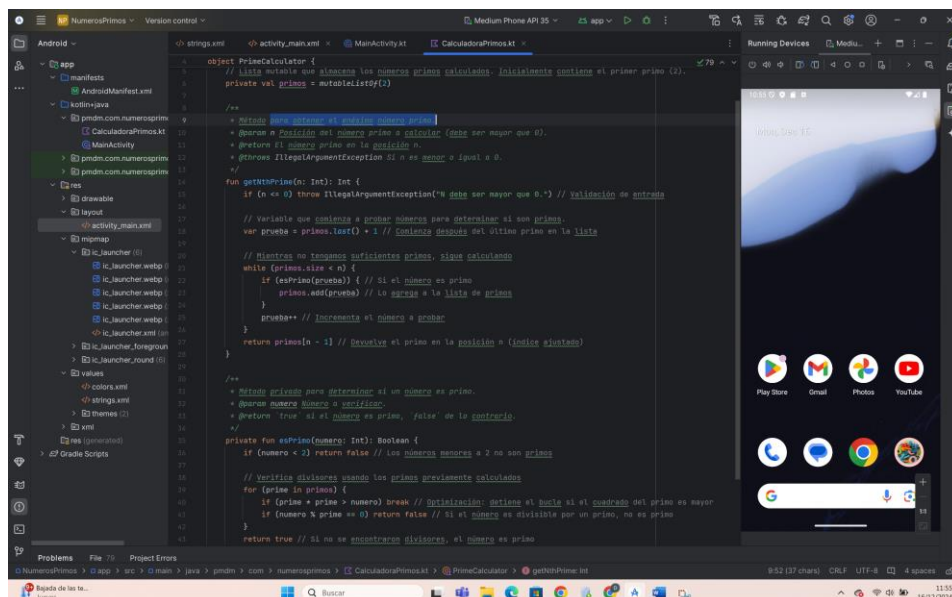


## b. Clase PrimeCalculator

La clase PrimeCalculator es una clase de utilidad (object) que contiene la lógica para calcular números primos.

Resumen de las responsabilidades:

- Incluye el método PrimeCalculator.getNthPrime() para obtener el enésimo número primo.
- Crea una lista de números primos incluyendo los ya calculados para evitar cálculos redundantes.
- Calcula nuevos números primos solo cuando sea necesario.
- Utilizar una función eficiente esPrimo para verificar si un número es primo.

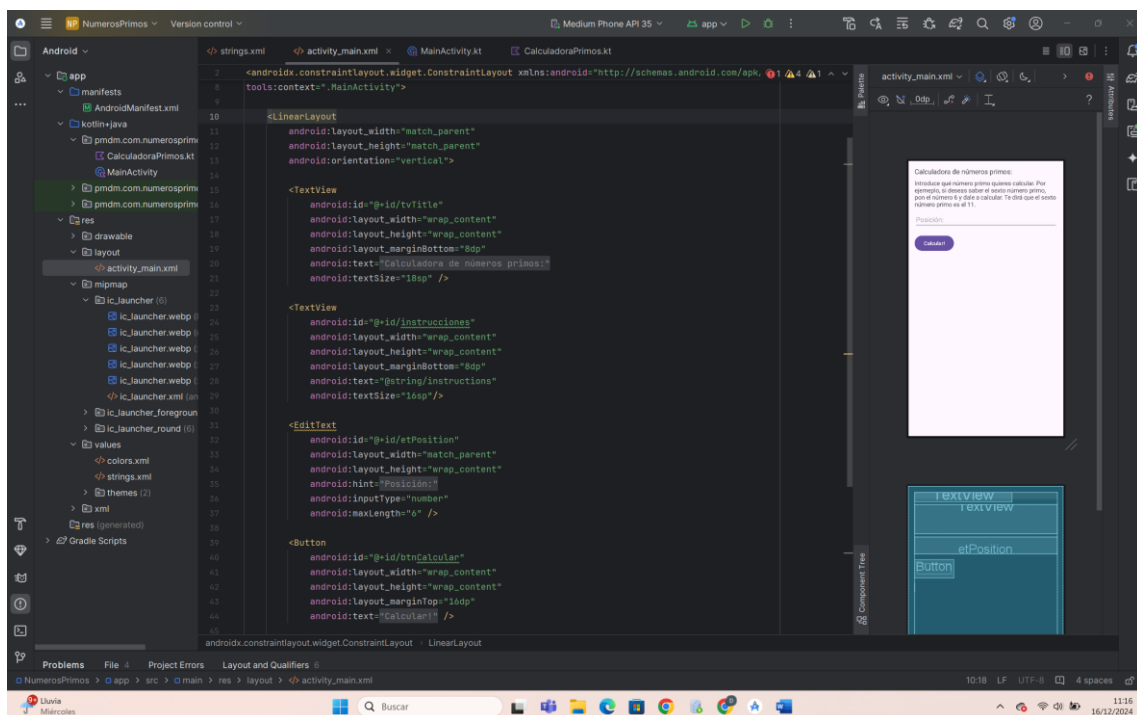
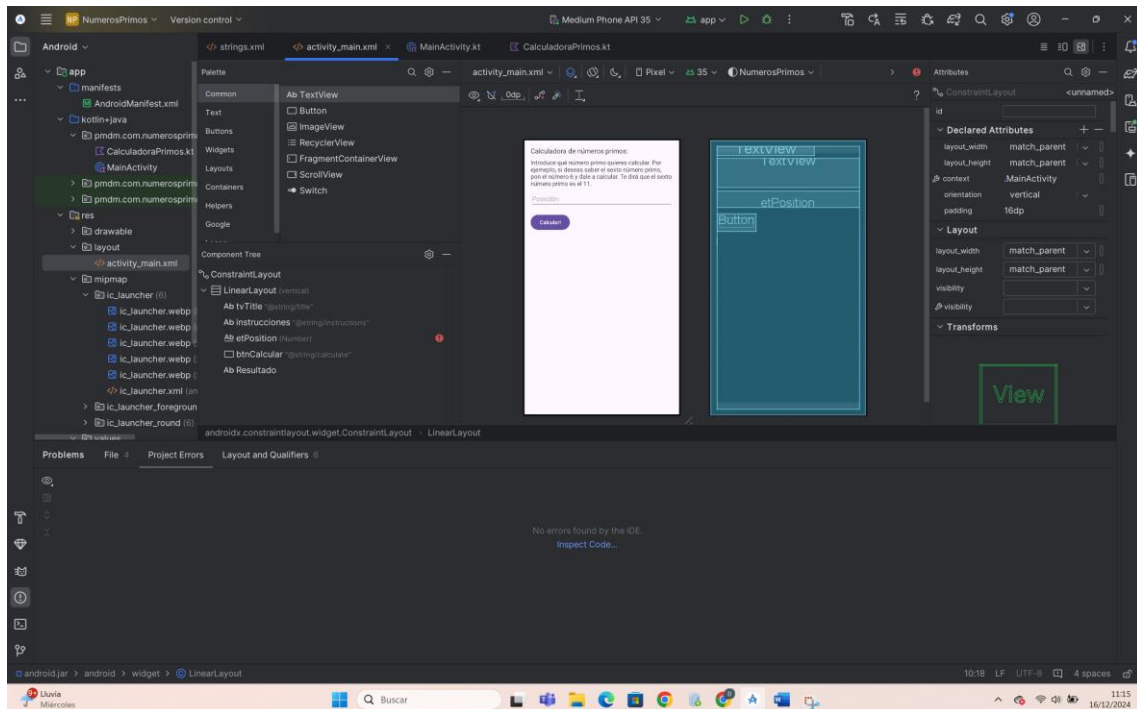


**c. Archivo activity\_main.xml**

El archivo XML define la interfaz de usuario de la aplicación.

Elementos destacados:

- I. **ConstraintLayout como contenedor principal:**
  - Es el contenedor raíz y proporciona flexibilidad en la disposición de los elementos.
  - Se aplica un padding general de 16dp para dar espacio alrededor del contenido.
- II. **LinearLayout interno:**
  - Se utiliza un LinearLayout vertical para organizar los elementos en una disposición de arriba hacia abajo.
- III. **TextView del título:**
  - Muestra el título de la aplicación. El texto se obtiene desde el archivo strings.xml mediante @string/titulo.
  - Se aplica un textSize de 18sp para destacar el título.
- IV. **TextView de instrucciones:**
  - Explica al usuario cómo utilizar la aplicación.
  - Tiene un tamaño de texto ligeramente más pequeño (16sp) para no competir con el título.
- V. **EditText para la posición del número primo:**
  - Permite que el usuario introduzca únicamente números (inputType="number").
  - La longitud máxima se limita a 6 dígitos usando android:maxLength="6".
  - El campo incluye un hint para guiar al usuario (texto: "Posición").
- VI. **Button para iniciar el cálculo:**
  - Botón con el texto "Calcular", definido en strings.xml.
  - Tiene un layout\_marginTop="16dp" para mantener espacio entre el campo de texto y el botón.
- VII. **TextView para mostrar el resultado:**
  - Inicialmente vacío, su contenido será actualizado dinámicamente desde el código de la actividad.
  - Posee un margen superior (16dp) para separar visualmente del botón.



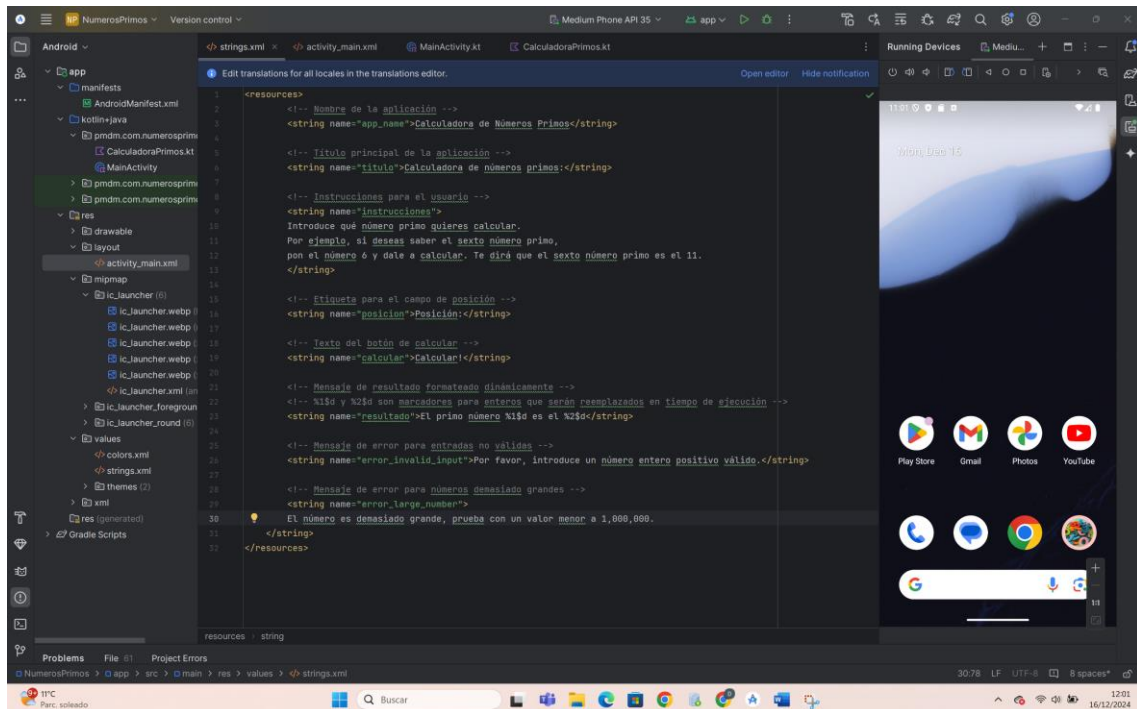
#### d. Archivo strings.xml

El archivo strings.xml centraliza todas las cadenas de texto de la aplicación, facilitando el mantenimiento y la internacionalización.

Aquí tienes el archivo strings.xml con comentarios explicativos para cada elemento y sus buenas prácticas.

#### Explicación Detallada de las Claves:

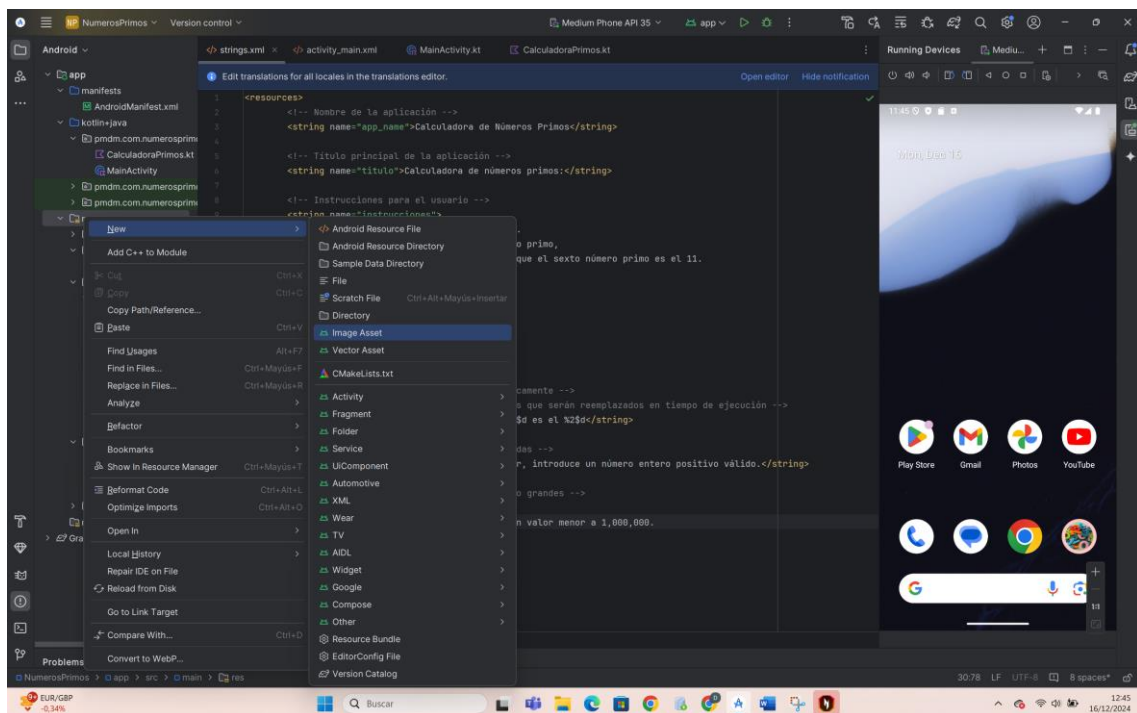
- I. **app\_name**: Define el nombre de la aplicación que se muestra en el lanzador y en la barra superior.
- II. **título**: Contiene el título principal de la interfaz de usuario.
- III. **instrucciones**: Proporciona una guía clara y detallada para que el usuario sepa cómo utilizar la aplicación.
- IV. **posicion**: Etiqueta para el campo de entrada de la posición del número primo.
- V. **calcular**: Texto que aparece en el botón que inicia el cálculo.
- VI. **resultado**:
  - Contiene texto formateado con dos marcadores **%1\$d** y **%2\$d**:
    - **%1\$d** se reemplaza por la posición introducida por el usuario.
    - **%2\$d** se reemplaza por el número primo calculado.
- VII. **error\_invalid\_input**: Mensaje de error que aparece si la entrada es nula, negativa o no es un número válido.
- VIII. **error\_large\_number**: Mensaje que se muestra si el usuario introduce un número demasiado grande que podría afectar el rendimiento.



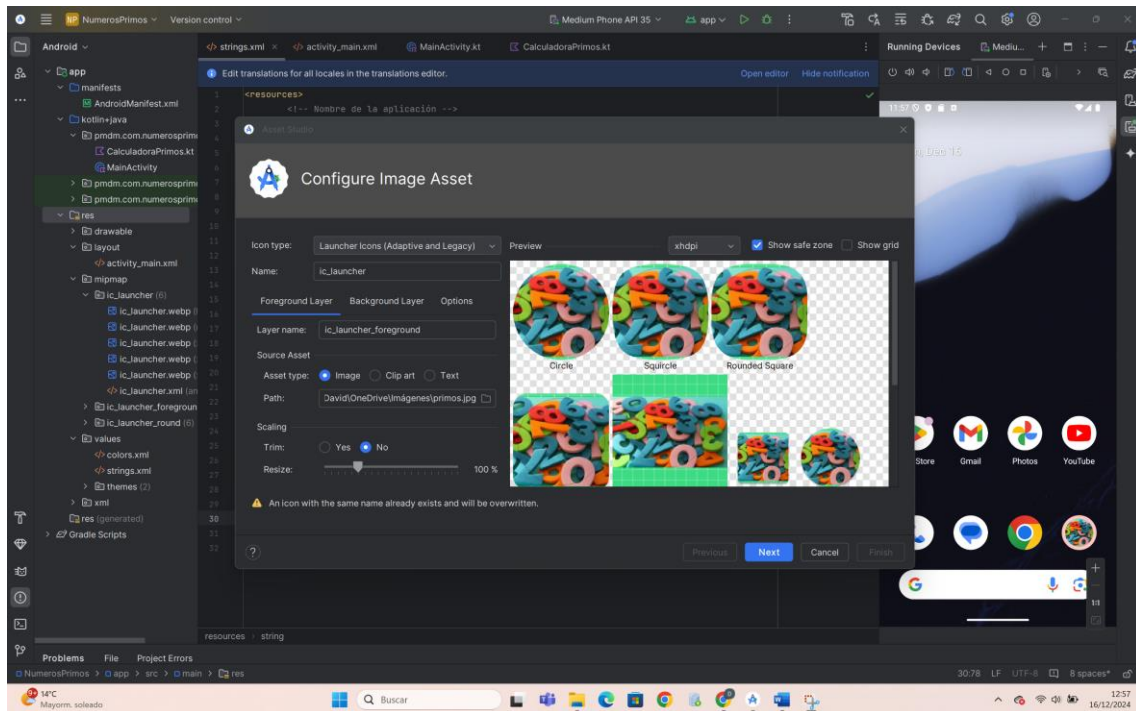
### 3. Incluir la Imagen del Icono de la Aplicación

Par cambiar el icono de la aplicación hacemos click derecho en la carpeta res/New/Image Asset.

En el asistente Asset Type seleccionamos nuestra imagen en Source Asset y Android Studio generará automáticamente los tamaños adecuados y coloca los archivos en las carpetas mipmap.







#### 4. Prueba de la aplicación

