

**UNIVERSIDAD ACADEMICA DE LA REGION NORTE DE GUERRERO UT**

**Docente; ING; Abel Jerónimo Vargas**

**Alumno:**

* **Roberto Chauteco Bello**
* **Documentación Act.8 Componentes de ListViews**

**Chilapa de Álvarez Guerrero, OCTUBRE 2023**

[Ilustración 0‑1estructura y interfaz de la apliacion 3](#_Toc148917588)

[Ilustración 0‑2 importaciones de los componentes de la estructura 4](#_Toc148917589)

[Ilustración 0‑1 hacer llamdo ala estructura 5](#_Toc148917590)

[Ilustración 0‑1 Estructura de OnCreate 6](#_Toc148917591)

[Ilustración 0‑1Estructura de los componentes 7](#_Toc148917592)

[Ilustración 0‑1Agregación del Listiew 8](#_Toc148917593)

[Ilustración 0‑1Funcion Restore\_Views 9](#_Toc148917594)

[Ilustración 0‑1Botton Calcúlate 10](#_Toc148917595)

[Ilustración 0‑1 Gestion y Validacion 10](#_Toc148917596)

Ilustración ‑estructura y interfaz de la apliacion

Principalmente se crea en Diseño de la interfaz de usuario en el (archivo XML), en esta parte en el archivo XML es donde definimos la estructura de la interfaz de usuario utilizando un RelativeLayout, es decir en este diseño consta de varios elementos, como tfNumero1, tfNumero2 para ingresar números, posteriormente agregamos un Button para calcular, y finalmente creamos un ListView para mostrar resultados.

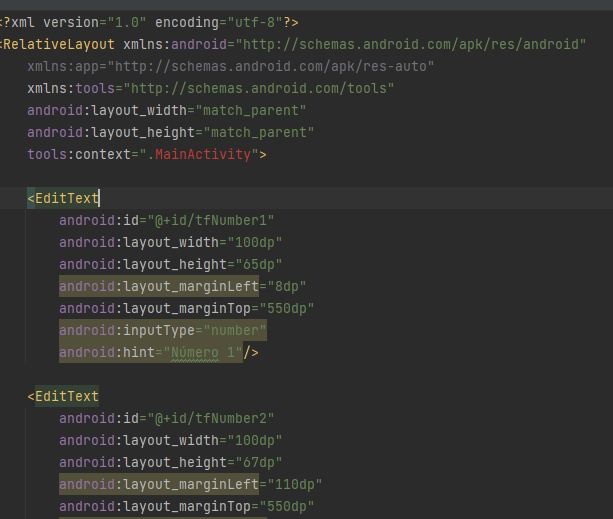




Ilustración ‑ importaciones de los componentes de la estructura

# **Importaciones**

La aplicación contiene una lista simple de elementos texto que pueden agregarse a través de un campo de entrada y un botón, Cuando se hace clic en un elemento de la lista, se muestra un mensaje de Toast, Además, el código contiene e incluye su funcionalidad para controlar la aparición y desaparición del teclado virtual en la interfaz de usuario.

Como lo son los Paquetes y Importaciones es decir este bloque de código comienza con la declaración del paquete y la importación de clases y bibliotecas necesarias para el desarrollo de la aplicación, también algunas de las importaciones son muy importantes y son claves para clases relacionadas con elementos de la interfaz de usuario y la gestión de vistas.

La Clase Principal: es MainActivity y extiende AppCompatActivity, esto requiere que es la actividad principal de la aplicación, que generalmente es la interfaz de usuario principal.

Y las Variables y Objeto del código, se declaran variables y objetos que se utilizan para diferentes propósitos. Algunas de las variables notables son:

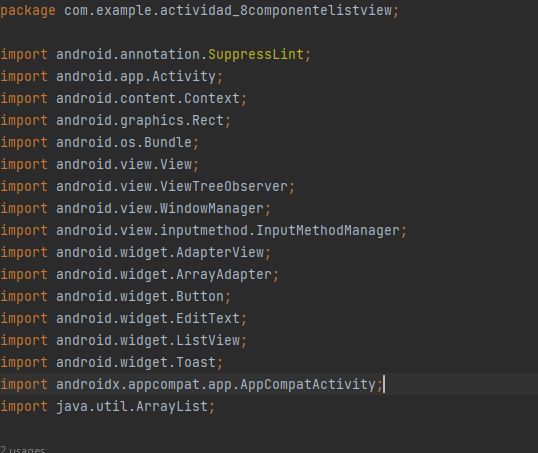
* ListView: Representa un componente de lista en la interfaz de usuario.
* Editex: Representa un campo de entrada de texto en la interfaz de usuario.
* Button buttonAdd: Representa un botón en la interfaz de usuario.
* ArrayList<Sting> ítems: Almacena elementos de texto que se mostrarán en la lista.

Ilustración ‑ hacer llamdo ala estructura

# **llamado de la estructura**

Principalmente visualizamos lo que es la Actividad principal (Java), que lleva por nombre.

public class MainActivity extends AppCompatActivity {

Estos objetos Editex son campos de entrada de texto en los que los usuarios pueden introducir números. Por sus nombres, parece que se espera que ingresen dos números.

Atributos:

* Editex tfNumber1
* Editex tfNumber2
* Button buttonCalculate: Este es un botón que, se utiliza para realizar algún tipo de cálculo con los números ingresados en los campos de texto.
* ListView listView: Este es un componente de lista en el que se mostrarán los resultados o elementos generados por la aplicación.
* ArrayList<String> resultsList: Esta es una lista de cadenas (String) que se utilizará para almacenar resultados u otros elementos que se mostrarán en el ListView.
* ArrayAdapter<String> adapter: El adaptador se utiliza para vincular los datos de resultsList con el ListView, lo que permite mostrar los elementos en la lista de la interfaz de usuario.

Posteriormente la actividad principal es una clase que hereda de AppCompatActivity que representa la aplicación es decir lo que hereda a la aplicación, y estas son las variables que se utilizan en la actividad para interactuar con los elementos de la interfaz para almacenar cada uno de los resultados de las sumas.

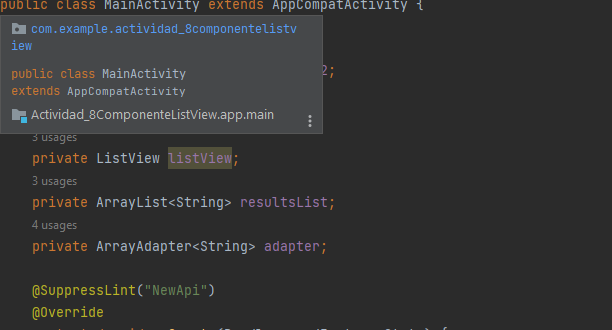


Ilustración ‑ Estructura de OnCreate

# **Estructura de clase OnCreate**

En el método onCreate se encarga de configurar la interfaz de usuario de la actividad, inicializar elementos, vincular un adaptador al ListView y asegurarse de que el teclado virtual no oculte las vistas cuando está activo.

Su funcionamiento específico de la actividad dependerá de que se agregue en la aplicación, como el manejo de eventos del botón buttonCalculate y la actualización de resultsList.

Como lo son.

Mandar a llamada al onCreate(savedInstanceState), este es el método onCreate y es necesario para inicializar correctamente la actividad.

setContentView (R. layout. activity\_main) es establece la vista de la actividad principal utilizando el archivo de diseño XML llamado activity\_main, en este archivo de diseño, se define la disposición y los elementos visuales que forman la interfaz de usuario de la actividad.

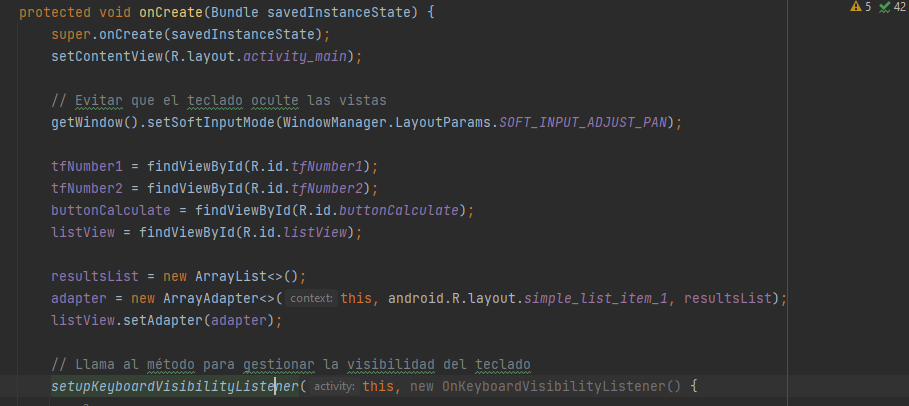


Ilustración ‑Estructura de los componentes

# **Estructura Componentes**

En esta sección de la visibilidad del teclado virtual en la actividad actual va dependiendo de si el teclado se muestra u oculta, se ejecutarán acciones específicas para ajustar o restaurar las vistas de la interfaz de usuario, garantizando una experiencia de usuario adecuada cuando se utiliza el teclado virtual.

Es decir setupKeyboardVisibilityListener(this, new OnKeyboardVisibilityListener() {...});: y esta línea de código llama a la función setupKeyboardVisibilityListener pasando dos argumentos:

This este se refiere al contexto actual que se refiere a la actividad en la que se encuentra, el que contexto se utiliza para acceder a recursos y realizar acciones dentro de la actividad

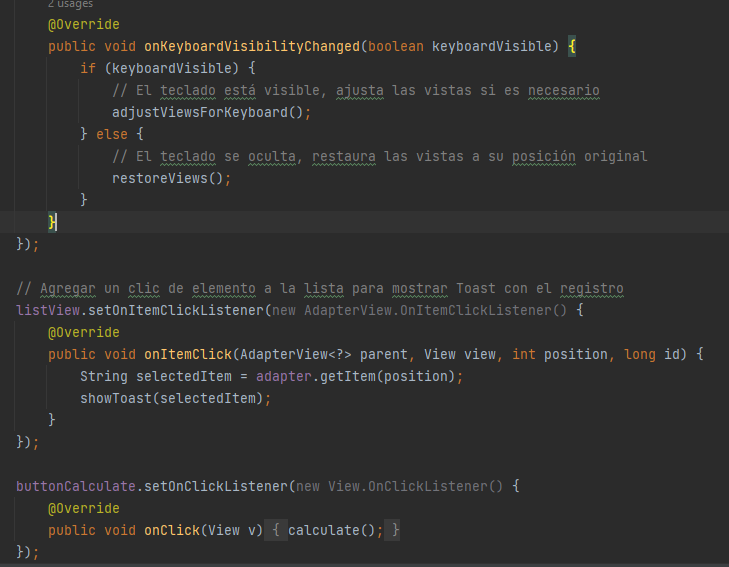


Ilustración ‑Agregación del Listiew

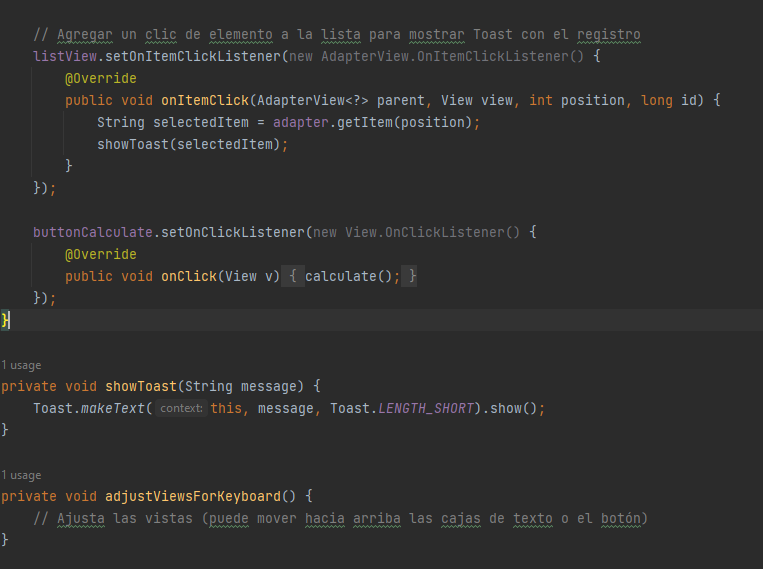
# **Agregación del Listiew**

En elementos de la lista (ListView) para mostrar mensajes de Toast cuando se hace clic en un elemento. También configura un clic en el botón "Calcular" para realizar algún tipo de cálculo o procesamiento, de la función calcúlate no se han proporcionado en el código.

También el listView.setOnItemClickListener - Clic en elementos del ListView, es decir que Cuando se establece un clic en elementos del ListView, se crea un escuchador de clic utilizando setOnItemClickListener.

El buttonCalculate.setOnClickListener, Clic en el botón "Calcular, que Cuando se establece un clic en el botón "Calcular" (buttonCalculate), se crea un escuchador de clic utilizando setOnClickListener.

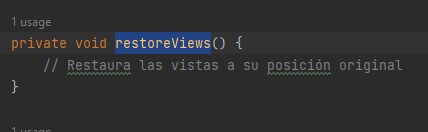
El showToast(String message), Este método se encarga de mostrar un mensaje de Toast (Toast) en la pantalla, recibe un mensaje como argumento y lo muestra en forma de mensaje emergente en la parte inferior de la pantalla utilizando Toast.makeText.



# Restore\_Views

Ilustración ‑Funcion Restore\_Views

En este campo esto restaura las vistas a su posición original es decir una vez terminado su proceso esto vuelve a su estado inicio.Carga el calcúlate



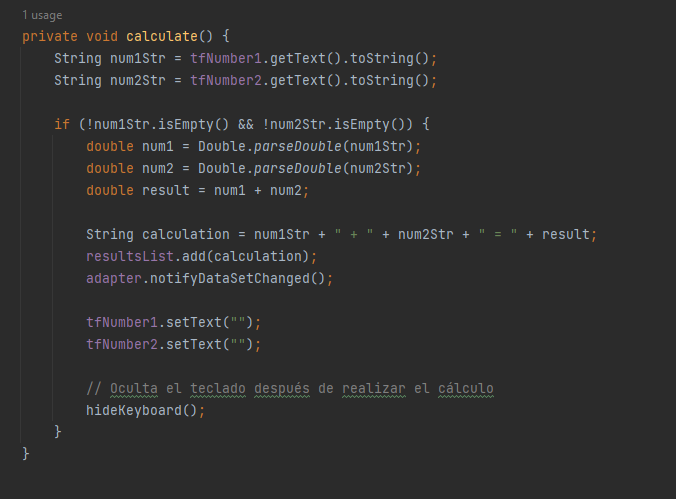
# Botton Calcúlate

Ilustración ‑Botton Calcúlate

Esta función calcúlate() se encarga de obtener valores ingresados por el usuario, realizar un cálculo simple, agregar el resultado a una lista de resultados, actualizar la vista del ListView, borrar los campos de entrada y ocultar el teclado virtual.

Procedimiento.

Se de los valores de los campos de entrada de texto (tfNumber1 y tfNumber2), los valores ingresados por el usuario en los campos de entrada de texto tfNumber1 y tfNumber2 se obtienen y almacenan en las cadenas num1Str y num2Str, respectivamente, es decir estos valores se obtienen como cadenas de texto utilizando tfNumber1.getText().toString() y tfNumber2.getText().toString().



# Gestion y Validacion

Ilustración ‑ Gestion y Validacion

Esta utilidad detecta la visibilidad del teclado virtual en una actividad de Android y notifica a un listener cuando se produce un cambio en la visibilidad del teclado, para realizar ajustes en la interfaz de usuario cuando el teclado se muestra , oculta.

Procedimiento

Esta línea de código cumple con la función para ser llamado desde cualquier actividad de Androi, toma dos argumentos setupKeyboardVisibilityListener(Activity activity, OnKeyboardVisibilityListener listener)::

activity: La actividad actual a la que se aplica la gestión de la visibilidad del teclado.

listener: crea un objete que implementa la interfaz OnKeyboardVisibilityListener, lo que permite a la actividad que llama ser notificada cuando cambia la visibilidad del teclado.

