**INTRODUCCION**

En el presente informe, se abordará el desarrollo de una calculadora en Android Studio, que permitirá realizar operaciones aritméticas básicas en dispositivos móviles Android. El objetivo de este proyecto es crear una aplicación funcional y de fácil uso, que brinde a los usuarios la capacidad de realizar cálculos matemáticos de manera rápida y conveniente.

Para llevar a cabo el desarrollo de esta aplicación, se han utilizado diversas fuentes de información y recursos, que han proporcionado conocimientos fundamentales y orientación en la implementación de la calculadora. Entre las fuentes bibliográficas consultadas se encuentran los libros "Android Programming: The Big Nerd Ranch Guide" de Phillips y Stewart, y "Head First Android Development" de Griffiths y Griffiths. Estas obras han sido de gran utilidad para comprender los conceptos esenciales de la programación en Android y aprender a utilizar las herramientas proporcionadas por Android Studio.

Además de los libros, se han utilizado recursos en línea, como la documentación oficial de Android Developers, disponible en el sitio web oficial de Android Studio. Esta documentación ha servido como referencia fundamental para comprender la estructura y el funcionamiento de Android Studio, así como para obtener información detallada sobre las diferentes clases y métodos necesarios para implementar la lógica de la calculadora.

Asimismo, se han consultado tutoriales en video disponibles en plataformas como YouTube, que han proporcionado ejemplos prácticos y explicaciones detalladas sobre el desarrollo de aplicaciones en Android Studio. Estos tutoriales han sido una fuente adicional de conocimiento y han contribuido significativamente en la comprensión de los conceptos y la aplicación de buenas prácticas en el desarrollo de la calculadora.

Por último, se han explorado repositorios de código abierto en plataformas como GitHub, donde se han encontrado proyectos relacionados con calculadoras en Android. El análisis de estos repositorios ha brindado ideas sobre la arquitectura de la aplicación, la implementación de funciones específicas y las mejores prácticas en el diseño de interfaces de usuario.

**MARCO TEORICO**

**MainActivity.java**

El MainActivity es la clase principal de la aplicación "CalculadoraBin". Proporciona la funcionalidad de una calculadora binaria en una interfaz de usuario. A continuación se muestra una descripción de los elementos y funciones clave en este código:

Elementos importados:

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

En esta sección, se importan las clases necesarias para la funcionalidad de la actividad principal. AppCompatActivity es una clase base para actividades en aplicaciones de Android, proporcionando características de compatibilidad con versiones anteriores de Android. También se importan clases relacionadas con las vistas, como Button y TextView, necesarias para manipular los elementos de la interfaz de usuario.

Declaración de variables:

private *TextView* inputTextView, resultTextView;

private *StringBuilder* input = new StringBuilder();

private *double* num1 = 0, num2 = 0, result = 0;

private *String* operator = "";

private *String* operation = "";

En esta sección, se declaran las variables que se utilizarán en el MainActivity. Estas variables representan elementos de la interfaz de usuario y almacenan datos necesarios para realizar las operaciones. Aquí hay una descripción de las variables:

inputTextView y resultTextView: Son instancias de la clase TextView que se utilizan para mostrar el texto de entrada y el resultado en la interfaz de usuario.

input: Es una instancia de StringBuilder que se utiliza para almacenar el texto de entrada de la calculadora.

num1, num2 y result: Son variables de tipo double que almacenan los números utilizados en las operaciones de la calculadora.

operator: Es una cadena que representa el operador (suma, resta, multiplicación, división) seleccionado por el usuario.

operation: Es una cadena que representa la operación completa introducida por el usuario.

Método onCreate():

@*Override*

protected *void* onCreate(*Bundle* savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity\_main);

    inputTextView = findViewById(R.id.inputTextView);

    resultTextView = findViewById(R.id.resultTextView);

}

El método onCreate() se ejecuta cuando se crea la actividad. Aquí se realiza la inicialización de la actividad, se configuran las referencias a los elementos de la interfaz de usuario y se establecen los OnClickListener para cada botón.

Se obtienen las referencias a los elementos inputTextView y resultTextView utilizando sus ID asignados en el archivo de diseño XML (activity\_main.xml) mediante el método findViewById().

También se obtienen las referencias a todos los botones de la interfaz de usuario utilizando sus ID asignados en el archivo de diseño XML.

Configuración de OnClickListener para los botones:

button0.setOnClickListener(new View.OnClickListener()

button1.setOnClickListener(new View.OnClickListener()

buttonPlus.setOnClickListener(new View.OnClickListener()

buttonEquals.setOnClickListener(new View.OnClickListener()

Para cada botón de la interfaz de usuario, se configura un OnClickListener personalizado. Cuando se hace clic en un botón, se ejecuta el código dentro del método onClick() correspondiente.

En cada OnClickListener, se realiza una acción específica según el botón clicado. Por ejemplo, para los botones numéricos (0-9), se agrega el dígito correspondiente a la cadena de entrada (input) y se actualiza el TextView de entrada (inputTextView) para mostrar la operación actualizada.

Para los botones de operación (suma, resta, multiplicación, división), se almacenan los números y el operador seleccionado, y se actualiza el TextView de entrada para mostrar la operación actualizada.

El botón "Igual" (buttonEquals) realiza la operación seleccionada (suma, resta, multiplicación, división) y muestra el resultado en el TextView de resultado (resultTextView).

Botones para conversión binaria, octal y hexadecimal:

En la parte final del código, se configuran los OnClickListener para los botones de conversión binaria, octal y hexadecimal. Estos botones convierten el número de entrada actual a su representación correspondiente en binario, octal o hexadecimal, respectivamente. El resultado se muestra en el TextView de resultado (resultTextView).

Botón "Limpiar":

buttonClear.setOnClickListener(new View.OnClickListener() {

    @*Override*

    public *void* onClick(*View* v) {

        input.delete(0, input.length());

        num1 = 0;

        num2 = 0;

        result = 0;

        operator = "";

        operation = "";

        inputTextView.setText("");

        resultTextView.setText("");

    }

});

El botón "Limpiar" (buttonClear) restablece todas las variables y textos relacionados con la calculadora, dejándola en un estado inicial limpio.

**Conversión a binario**

El botón "Binario" (buttonBinario) está asociado a un OnClickListener que realiza la conversión de un número decimal a su representación binaria. El proceso de conversión se lleva a cabo de la siguiente manera:

* Si no hay ningún número en la entrada, la conversión no se realiza.
* Se obtiene el número decimal de la entrada utilizando Double.parseDouble(input.toString()).
* El método Integer.toBinaryString((int) num) se utiliza para convertir el número decimal (num) a su representación binaria como una cadena de texto (binario).
* El resultado binario se muestra en el TextView de resultado (resultTextView).
* Se agrega un espacio al TextView de entrada (inputTextView) para indicar que se ha realizado la conversión.

**Conversión a octal**

El botón "Octal" (buttonOctal) realiza la conversión de un número decimal a su representación octal. El proceso de conversión es similar al de la conversión binaria:

* Si no hay ningún número en la entrada, la conversión no se realiza.
* Se obtiene el número decimal de la entrada utilizando Double.parseDouble(input.toString()).
* El método Integer.toOctalString((int) num) se utiliza para convertir el número decimal (num) a su representación octal como una cadena de texto (octalString).
* El resultado octal se muestra en el TextView de resultado (resultTextView).
* Se agrega un espacio al TextView de entrada (inputTextView) para indicar que se ha realizado la conversión.

**Conversión a hexadecimal**

El botón "Hexadecimal" (buttonHexadecimal) realiza la conversión de un número decimal a su representación hexadecimal. El proceso de conversión es similar al de las conversiones anteriores:

* Si no hay ningún número en la entrada, la conversión no se realiza.
* Se obtiene el número decimal de la entrada utilizando Double.parseDouble(input.toString()).
* El método Integer.toHexString((int) num).toUpperCase() se utiliza para convertir el número decimal (num) a su representación hexadecimal como una cadena de texto (hexString).
* El resultado hexadecimal se muestra en el TextView de resultado (resultTextView).
* Se agrega un espacio al TextView de entrada (inputTextView) para indicar que se ha realizado la conversión.

**ActivityMain.xml**

El diseño de la aplicación se ha implementado utilizando un archivo XML en el que se definen los elementos de la interfaz de usuario y sus propiedades. A continuación, se detalla cómo se ha utilizado este archivo para lograr el diseño de la aplicación.

**Contenedor principal:**

Se utiliza un ConstraintLayout como contenedor principal de la interfaz de usuario. Este layout permite definir restricciones entre los elementos para posicionarlos correctamente en la pantalla.

**Guías:**

Se emplean guías (Guideline) para establecer líneas de referencia que ayudan a posicionar los elementos de manera precisa. Se definen guías horizontales y verticales, especificando su posición relativa en relación con los bordes y otros elementos.

**Botones:**

Se definen varios botones (Button) que representan los dígitos, operadores y funciones de la calculadora. Cada botón tiene un identificador único (android:id) que se utiliza para identificarlo en el código Java.

Se establecen las propiedades de ancho (android:layout\_width) y alto (android:layout\_height) como "0dp" para permitir que los botones se ajusten automáticamente al espacio disponible.

Se especifica el color de fondo (android:backgroundTint) utilizando un valor hexadecimal, lo que proporciona un aspecto oscuro.

Se establece el texto del botón (android:text), el tamaño de texto (android:textSize) y el color del texto (android:textColor).

Las propiedades app:layout\_constraintStart\_toStartOf, app:layout\_constraintEnd\_toEndOf, app:layout\_constraintTop\_toTopOf y app:layout\_constraintBottom\_toBottomOf se utilizan para establecer las restricciones de posicionamiento del botón en relación con otras vistas o guías.

**TextViews:**

Se definen dos TextViews para mostrar el número de entrada y el resultado de las operaciones.

Al igual que los botones, se especifican propiedades como ancho, alto, color de fondo, color de texto, tamaño de texto y restricciones de posicionamiento utilizando las mismas propiedades mencionadas anteriormente.

Además, se establecen las propiedades android:hint y android:textColorHint para mostrar un texto de sugerencia en caso de que los TextViews estén vacíos.

**Colores y estilo:**

Se utilizan atributos como android:background y android:textColor para establecer colores de fondo y colores de texto respectivamente. Los valores de color se definen utilizando valores hexadecimales.

En resumen, el archivo XML del MainActivity utiliza un ConstraintLayout como contenedor principal y define botones, guías y TextViews con sus respectivas propiedades para lograr el diseño de la aplicación. Las guías ayudan a posicionar los elementos de manera precisa, mientras que las propiedades de los botones y TextViews se establecen para lograr la apariencia deseada.

**CONCLUSION**

En este informe, se ha llevado a cabo el desarrollo de una calculadora en Android Studio, con el objetivo de proporcionar a los usuarios una herramienta funcional y conveniente para realizar cálculos matemáticos básicos en dispositivos móviles Android. A través de la investigación y la utilización de diversas fuentes de información, se logró implementar una aplicación que cumple con los requisitos establecidos y ofrece una experiencia de usuario satisfactoria.

Durante el proceso de desarrollo, se ha empleado una variedad de recursos bibliográficos y en línea, que han sido fundamentales para comprender los conceptos y las técnicas necesarias en la programación en Android Studio. Los libros "Android Programming: The Big Nerd Ranch Guide" de Phillips y Stewart, y "Head First Android Development" de Griffiths y Griffiths, han proporcionado una base sólida en la programación de aplicaciones para dispositivos Android, abarcando desde los aspectos básicos hasta los más avanzados.

Además, la documentación oficial de Android Developers y los tutoriales en video han sido fuentes valiosas de información, brindando detalles sobre las herramientas y funciones específicas de Android Studio, así como ejemplos prácticos de implementación. La exploración de repositorios de código abierto en plataformas como GitHub ha permitido obtener ideas y mejores prácticas para el diseño de interfaces de usuario y la arquitectura de la aplicación.

El resultado final es una calculadora funcional, diseñada con una interfaz de usuario intuitiva y atractiva, que permite realizar operaciones aritméticas básicas de manera eficiente. La aplicación cumple con los estándares de usabilidad y ofrece una experiencia fluida al interactuar con los elementos de la calculadora.

En conclusión, el desarrollo de esta calculadora en Android Studio ha sido un proceso enriquecedor, donde se han aplicado los conocimientos adquiridos a partir de fuentes bibliográficas, recursos en línea y análisis de código abierto. La implementación exitosa de la aplicación demuestra la importancia de contar con una base sólida de conocimientos y utilizar recursos adecuados para el desarrollo de aplicaciones móviles en Android Studio.

**BIBLIOGRAFIA**

*Android Developer*. (n.d.). Retrieved from Android Studio: https://developer.android.com/studio

Androide, Y. (2019, Diciembre 25 ). ANDROID STUDIO APLICACIÓN CALCULADORA DESDE CERO BIEN EXPLICADO .

Griffiths, D. G. (2015). Head First Android Development. In D. G. Griffiths, *Head First Android Development. 2ª edición. O'Reilly Media.*

Phillips, B. S. (2017). Android Programming. In B. S. Phillips, *The Big Nerd Ranch Guide. 3ª edición. Big Nerd Ranch.*