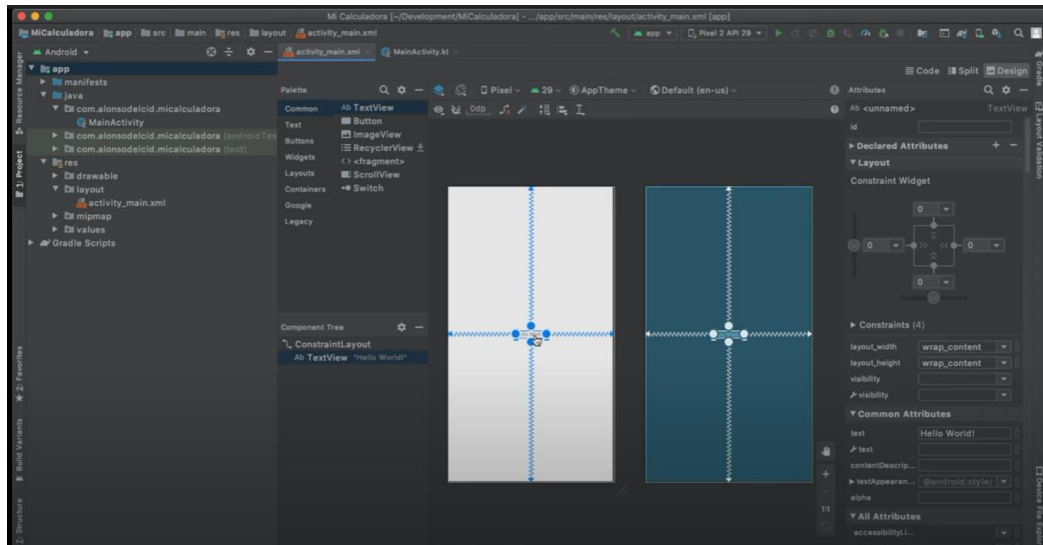


Hacer una calculadora con Android studio

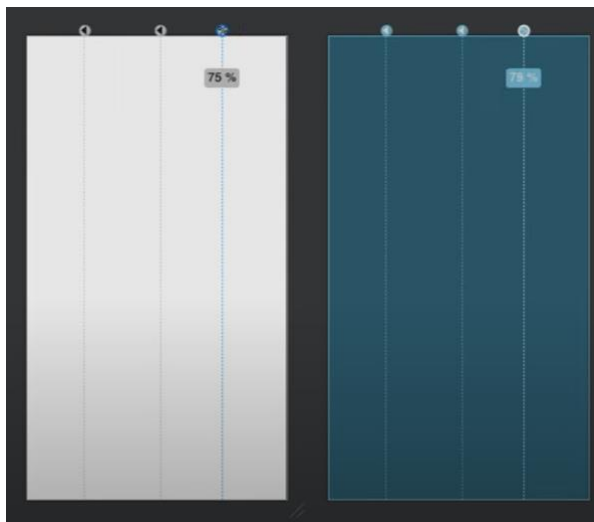
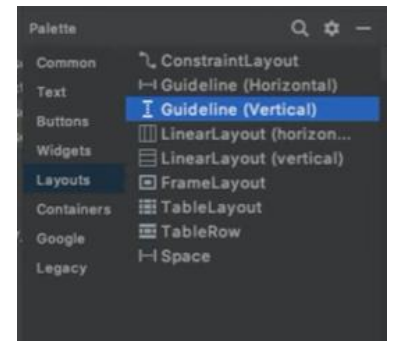
El primer paso es abrir un nuevo proyecto desde cero, al estar con el nuevo proyecto tenemos que localizar el archivo .xml activity_main, donde observaremos dos pantallas para empezar el diseño.



Como se puede ver hay una columna con varias opciones del lado izquierdo, de esa columna vamos a arrastrar los botonees necesarios para realizar la calculadora.

También utilizaremos los guidelines para acomodarlos de una manera más fácil

En este caso utilizaremos las líneas verticales, como vamos a tener 4 botones por fila, colocaremos 3 guidelines verticales como se muestra en la siguiente imagen

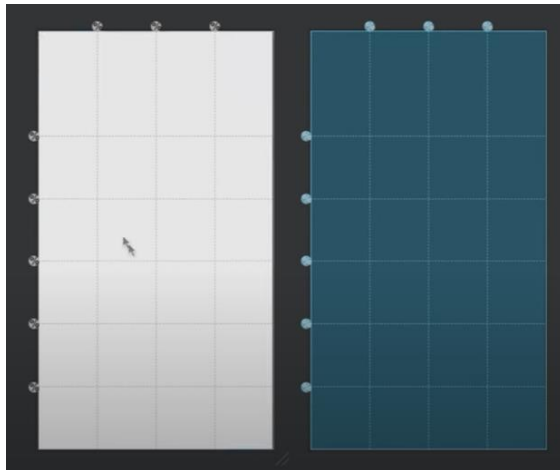


En la parte superior se pueden observar unas flechas dentro de unos círculos.

Le daremos click a cada una de ellas para poner en modo de porcentaje la división que queremos que tenga dentro del área asignada.

Como tenemos tres líneas, se dividirá el 100 en 4 partes, es decir la primera línea al 25%, la segunda línea al 50% y finalmente la tercera línea al 75%, de esta forma tendremos a todos los botones a la misma distancia.

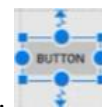
Ahora realizaremos las divisiones horizontales, aproximadamente tenemos 7 divisiones, empezaremos con una línea al 30%, después al 45% y así sucesivamente. Al final obtendremos el resultado que se muestra en la imagen.



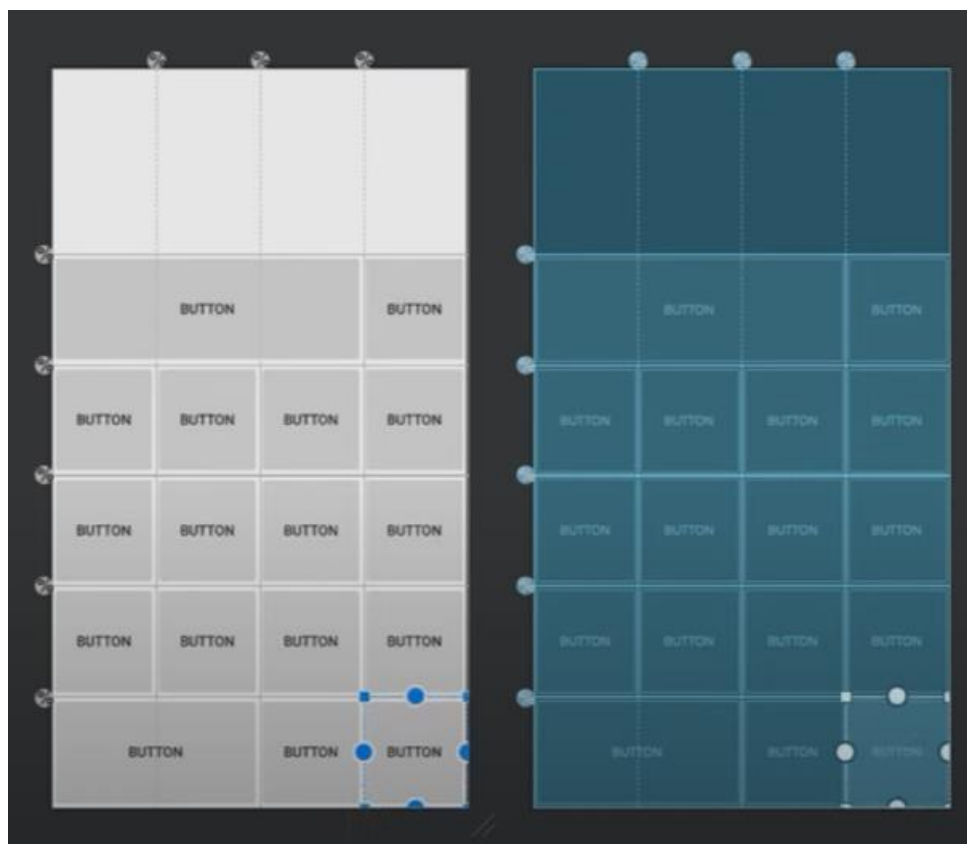
Ahora sí, proseguimos a colocar los botones correspondientes en cada cuadro que le corresponda.



Para que cada botón este centrado en cada una de sus posiciones, utilizaremos los puntitos azules y lo



jalaremos hasta el guideline superior e inferior que le corresponda. en la parte derecha, en la seccion de `layout_width`, cambiaremos el valo que tiene por default de `wrap_content` a `0dp`. Obteniendo el siguiente resultado.



Como existe la parte de programación y facilitar nuestro trabajo, tenemos que identificar o cambiar el nombre de los botones, es recomendable realizarlo con la posición o su acción dentro del programa.

A cada botón tenemos que cambiarle, tanto el id como el nombre que se muestra de manera visual, esto se realiza a través de Common Attributes, en text. Dentro de este cuadro se escribe el texto o números que queremos que el botón muestre al usuario, en este ejemplo de la calculadora se colocaran los números del 0 al 9, un botón de clear y las cuatro operaciones básicas.

El diseño de los botones también se puede cambiar, en la parte de component Tree seleccionas los botones que quieres personalizar, puedes buscar las siguientes opciones

- Textsize: te permite modificar el tamaño de la letra de cada botón
- textColor: te permite cambiar el color de la letra
- backgroundColor: permite cambiar el color de los botones

Para poder mostrar los resultados tenemos que colocar un TextView en la parte superior de nuestra calculadora, realizaremos el mismo procedimiento con el layout height and width a 0dp. En este textview se harán cambios en “gravity”, seleccionando la opción de centrado.

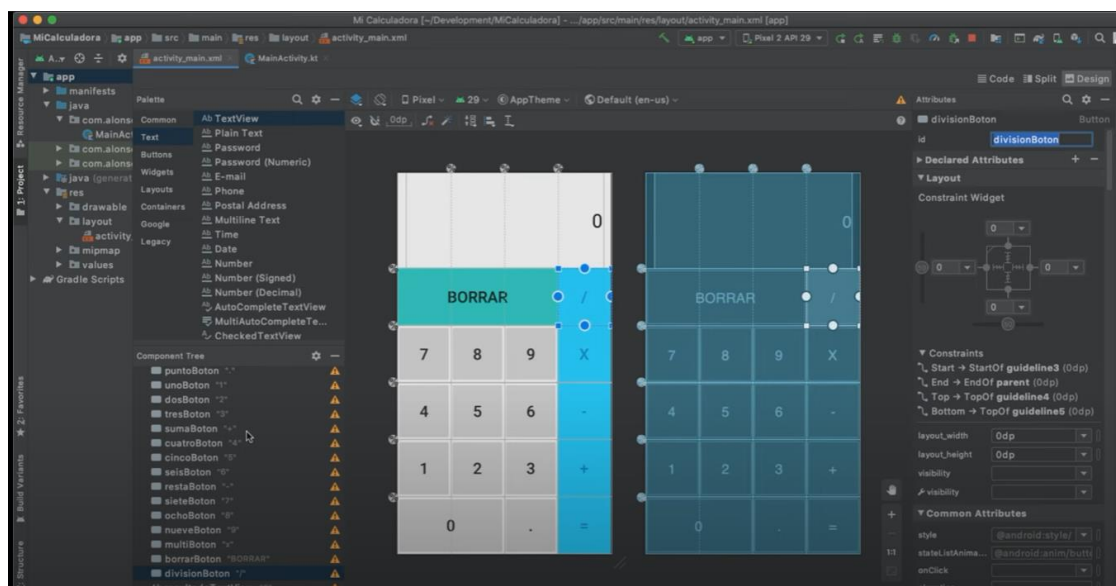
Después en la sección layout_constraint, en la propiedad de textalign pondremos la opción de textEnd.

De igual manera puedes personalizar el textview como más te guste con las propiedades mostradas anteriormente.



Para cambiar los id de los botones se tiene que hacer uno por uno, el proceso que se seguirá es seleccionar el botón ir a la parte superior derecha y cambiar el nombre id del botón. Es recomendable poner id como “divisionBoton” para las operaciones o en el caso de los números se puede poner “unoBoton”.

Al final tendremos un resultado así:



Hemos acabado con el diseño de nuestra calculadora, ahora empezaremos con el código para que nuestra computadora funcione de manera correcta.

Comenzaremos con la función de los botones de los números. Ponemos todos los botones de la siguiente forma:

```
unoBoton.setOnClickListener: { }
```

Necesitamos una función que nos diga que tenemos que hacer cuando se presione ese botón:

```
Private fun numeroPresionado(digito: String) {  
    resultadoTextView.text = "${resultadoTextView.text}$digito"  
}
```

Esta función nos dice que recibe un digito como string y recibirá un resultado en modo texto junto con el digito

En la primera función que creamos se uno Boton, llamaremos esta función de numeroPresionado y el digito que se recibe, quedando así:

```
unoBoton.setOnClickListener: { numeroPresionado(digito:"1") }
```

este proceso se tiene que hacer con todos los botones con numero de nuestra calculadora, teniendo 10 funciones

```
unoBoton.setOnClickListener: { numeroPresionado(digito:"1") }  
dosBoton.setOnClickListener: { numeroPresionado(digito:"2") }  
tresBoton.setOnClickListener: { numeroPresionado(digito:"3") }  
cuatroBoton.setOnClickListener: { numeroPresionado(digito:"4") }  
cincoBoton.setOnClickListener: { numeroPresionado(digito:"5") }  
seisBoton.setOnClickListener: { numeroPresionado(digito:"6") }  
sieteBoton.setOnClickListener: { numeroPresionado(digito:"7") }  
ochoBoton.setOnClickListener: { numeroPresionado(digito:"8") }  
nueveBoton.setOnClickListener: { numeroPresionado(digito:"9") }  
ceroBoton.setOnClickListener: { numeroPresionado(digito:"0") }
```

si se corre este código en el emulador debería aparecer en el textview el numero correspondiente al botón que se está presionando.

Ahora necesitamos una función que indique cuando una operación fue presionada, y nos va a decir que operación fue, en este caso vamos a usar la operación con ayuda de unas constantes de tipo int,

```
Private fun operacionPresionada(digito: String){  
  
}  
  
Companion object {  
    Const val SUMA = 1  
    Const val RESTA = 2  
    Const val MULTIPLICACION = 3  
    Const val DIVISION = 4  
}
```

Dependiendo de cual operación se esta realizando o recibiendo se va a realizar diferentes acciones en la función operacionPresionada.

En la parte superior vamos a agregar las variables

```
Private var num1: Double = 0.0  
  
Private var num2: Double = 0.0
```

Estas variables sirven, cuando presionamos nuestros números la operación la vamos a guardar en el número uno o en el número dos, para saber en cual de los dos se va a guardar se agregará una variable que indique la operación en curso.

```
Private var operación: Int = 0
```

Que esta tercera variable tenga el valor de cero, hace entender que no existe la operación, también la tenemos que agregar a nuestro companion object como

```
Const val NO_OPERACION = 0
```

Cuando se presiona la operación, tenemos que hacer que la operación que este en curso va a ser la operación que recibamos como las variables se llaman igual.

Al tener el mismo nombre, le agregaremos un this. Para indicar que es de las variables globales.

El número uno va a ser igual a lo que tengamos en nuestra pantalla, pero convertido a doble, finalmente nuestro resultado lo vamos a dejar listo para lo siguiente, se marca como en cero

```

Private fun operacionPresionada(digito: String){

    this.operacion = operación

    num1 = resultadoTextView.text.toString().toDouble()

    resultadoTextView.text = "0"

}

```

Ahora cuando queramos vamos a mandar a llamar a la función operacionPresionada con la constante que marcamos de la siguiente forma

```

sumaBoton.setOnClickListener { operacionPresionada(SUMA)}
restaBoton.setOnClickListener { operacionPresionada(RESTA)}
multiBoton.setOnClickListener { operacionPresionada(MULTIPLICACION)}
diviBoton.setOnClickListener { operacionPresionada(DIVISION)}

```

Ahora falta agregar la logica en código del boton de borrar

```

borrarBoton.setOnClickListener { it:View!

    num1 = 0.0

    num2 = 0.0

    resultadoTextView.text = "0"

    operación = NO_OPERACION

}

```

Al hacer esto pusimos en el estaco inicial de nuestra calculadora cuando iniciamos el programa o cuando presionemos el botón borrar.

El ultimo botón que falta es el botón igual, vamos a utilizar la función when, tomando en cuenta la operación, donde se indica que dependiendo de la operación se realizara tal operación a los num1 y num2.

```

igualBoton.setOnClickListener(){ it:View

    var resultado = when(operación){

        SUMA -> num1 + num2

        RESTA -> num1 - num2

        MULTIPLICACION -> num1 * num2
    }
}

```

```

        DIVISION -> num1 / num2
        else -> 0
    }
    resultadoTextView.text = resultado.toString()
}

```

Si observamos bien el código hasta ahora, no tenemos el número dos por lo que no se guardará al momento de presionar los botones

```

    if (operación == NO_OPERACION) {
        num1 = resultadoTextView.text.toString().toDouble()
    } else {
        num2 = resultadoTextView.text.toString().toDouble()
    }
}

```

Con esto hemos terminado el código y puedes hacer pruebas con la calculadora final.

CODIGO FINAL

```

    Private var num1: Double = 0.0
    Private var num2: Double = 0.0
    Private var operación: Int = 0

    Override fun onCreate(savedInstanceState: Bundle?) {
        Super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        unoBoton.setOnClickListener { numeroPresionado(digito:"1") }
        dosBoton.setOnClickListener { numeroPresionado(digito:"2") }
        tresBoton.setOnClickListener { numeroPresionado(digito:"3") }
        cuatroBoton.setOnClickListener { numeroPresionado(digito:"4") }
        cincoBoton.setOnClickListener { numeroPresionado(digito:"5") }
    }

```

```

seisBoton.setOnClickListener: { numeroPresionado(digito:"6") }
sieteBoton.setOnClickListener: { numeroPresionado(digito:"7") }
ochoBoton.setOnClickListener: { numeroPresionado(digito:"8") }
nueveBoton.setOnClickListener: { numeroPresionado(digito:"9") }
ceroBoton.setOnClickListener: { numeroPresionado(digito:"0") }

sumaBoton.setOnClickListener { operacionPresionada(SUMA)}
restaBoton.setOnClickListener { operacionPresionada(RESTA)}
multiBoton.setOnClickListener { operacionPresionada(MULTIPLICACION)}
diviBoton.setOnClickListener { operacionPresionada(DIVISION)}

borrarBoton.setOnClickListener { it:View!
    num1 = 0.0
    num2 = 0.0
    resultadoTextView.text = "0"
    operación = NO_OPERACION
}

igualBoton.setOnClickListener(){ it:View
    var resultado = when(operación){
        SUMA -> num1 + num2
        RESTA -> num1 - num2
        MULTIPLICACION -> num1 * num2
        DIVISION -> num1 / num2
        else -> 0
    }
    resultadoTextView.text = resultado.toString( )
}
}

Private fun numeroPresionado(digito: String) {

```



```

If (operación == NO_OPERACION) {
    num1 = resultadoTextView.text.toString( ). toDouble( )
} else {
    num2 = resultadoTextView.text.toString( ). toDouble( )
}
}

resultadoTextView.text = “${resultadoTextView.text}$digito”
}

Private fun operacionPresionada(digito: String){
    this.operacion = operación
    //num1 = resultadoTextView.text.toString().toDouble()
    resultadoTextView.text = “0”
}

Companion object {
Const val SUMA = 1
Const val RESTA = 2
Const val MULTIPLICACION = 3
Const val DIVISION = 4
Const val NO_OPERACION = 0
}

```