Investiga y Crea Actividades por Tipo de Vista

1. TextView
   •Para qué sirve: TextView es una vista fundamental que se utiliza para mostrar texto al usuario. Puede mostrar texto estático, texto que cambia dinámicamente, e incluso texto con formato (como negrita, cursiva, diferentes colores, etc.).
   •Funcionamiento: Simplemente estableces el texto que quieres mostrar, ya sea directamente en el archivo XML de diseño o mediante programación en tu código Kotlin o Java. Puedes personalizar su apariencia a través de atributos XML (como android:text, android:textSize, android:textColor, android:textStyle) o métodos en el código.
   •Ejemplo de uso XML:

```
Layout
<TextView
    android:id="@+id/myTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hola, Mundo!"
    android:textSize="18sp"
    android:textColor="@color/black" />
```

   •Ejemplo de uso Kotlin:

```
Kotlin
val myTextView = findViewById<TextView>(R.id.my
myTextView.text = "Texto actualizado"
myTextView.setTextColor(ContextCompat.getColor(
```

2. Button
   •Para qué sirve: Button es una vista interactiva que permite al usuario realizar una acción al presionarlo. Es uno de los componentes más comunes para la interacción del usuario.
   •Funcionamiento: Un Button muestra un texto (o un icono) que indica su propósito. Se le asocia un "listener" (generalmente un OnClickListener) que define qué código se ejecutará cuando el usuario toque el botón.
   •Ejemplo de uso XML:

```
Layout
<Button
    android:id="@+id/myButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Haz clic aquí" />
```

•Ejemplo de uso Kotlin:

```kotlin
val myButton = findViewById<Button>(R.id.myButt
myButton.setOnClickListener {
    // Código a ejecutar cuando se presiona el
    Log.d("MiApp", "Botón presionado")
}
```

3. ImageView
   •Para qué sirve: ImageView se utiliza para mostrar imágenes. Puede mostrar imágenes desde los recursos de la aplicación (drawable), desde archivos en el dispositivo o desde una URL de internet (requiere librerías adicionales para esto último, como Glide o Picasso).
   •Funcionamiento: Especificas la fuente de la imagen (el "source") a través del atributo android:src en XML o mediante métodos en el código. ImageView ofrece varios atributos para controlar cómo se escala y se muestra la imagen (por ejemplo, android:scaleType).
   •Ejemplo de uso XML (desde drawable):

```xml
<ImageView
    android:id="@+id/myImageView"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:src="@drawable/mi_imagen"
    android:scaleType="centerCrop" />
```

   •Ejemplo de uso Kotlin (estableciendo un drawable):

```kotlin
val myImageView = findViewById<ImageView>(R.id.
myImageView.setImageResource(R.drawable.otra_im
```

4. EditText
   •Para qué sirve: EditText es una subclase de TextView que permite al usuario ingresar y editar texto. Es el componente principal para obtener entradas de texto del usuario, como nombres, contraseñas, mensajes, etc.
   •Funcionamiento: Proporciona un campo donde el usuario puede escribir. Puedes configurar el tipo de entrada que se espera (números, texto, contraseña, email, etc.) usando el atributo android:inputType. Puedes obtener el texto ingresado por el usuario y escuchar cambios en el texto.
   •Ejemplo de uso XML:

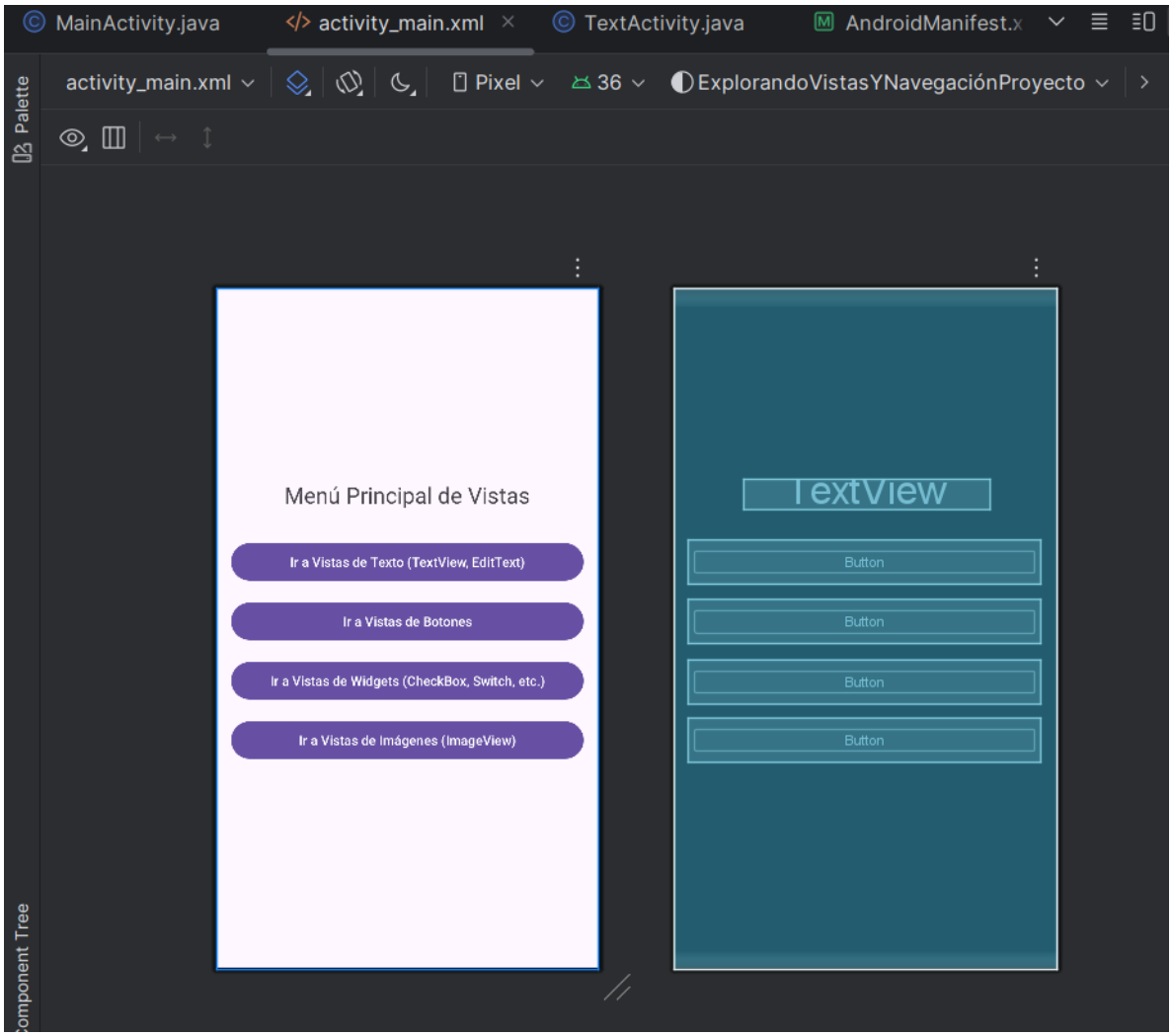
Layout

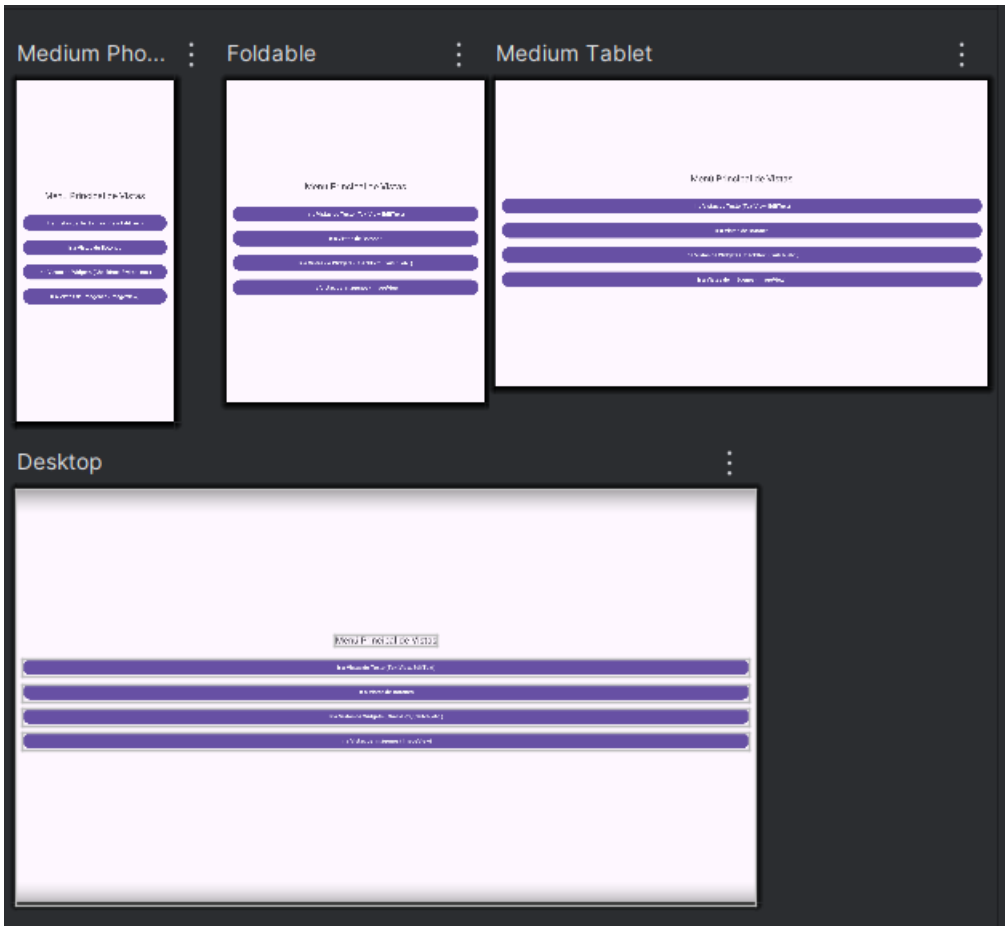
```
<EditText
    android:id="@+id/myEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Ingresa tu nombre"
    android:inputType="textPersonName"/>
```

•Ejemplo de uso Kotlin (obtener el texto):

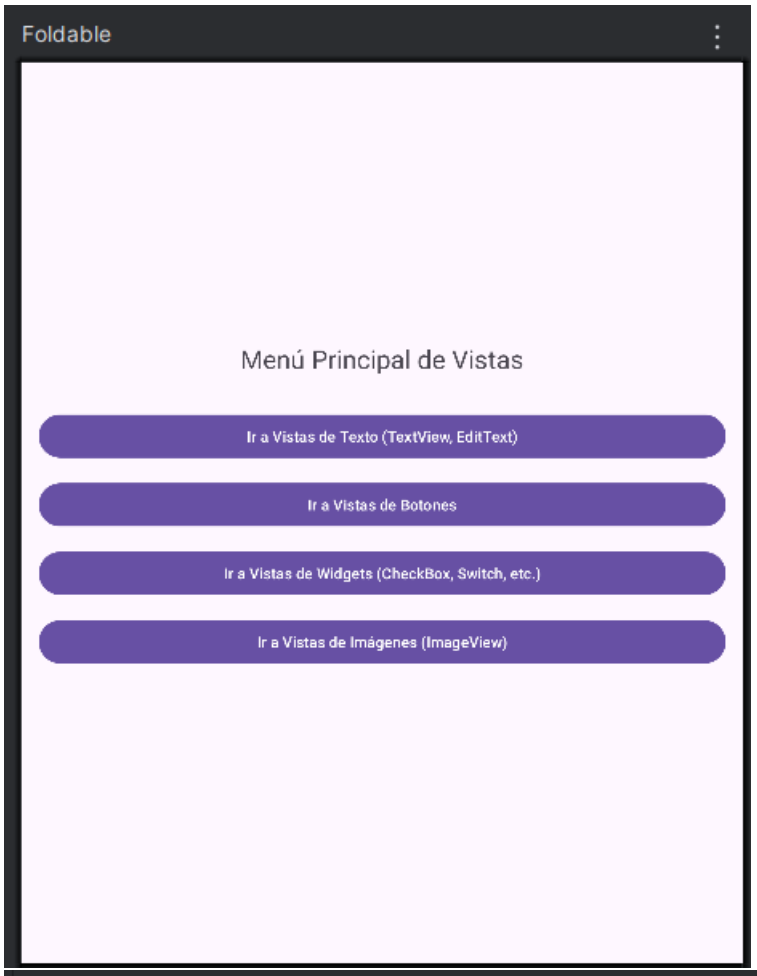
Kotlin

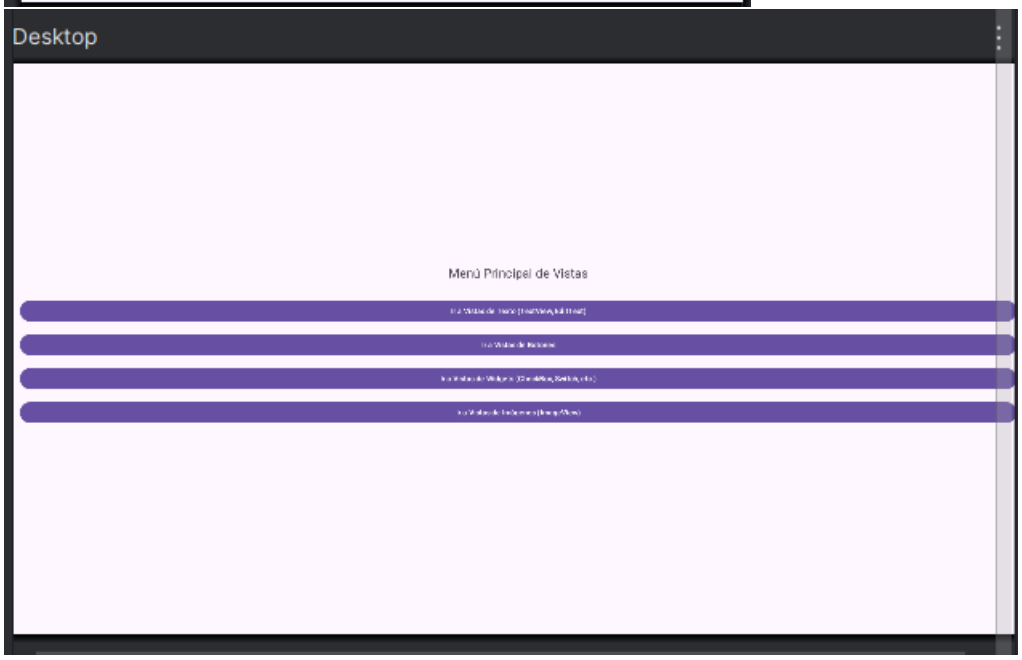
```
val myEditText = findViewById<EditText>(R.id.myI
// Para obtener el texto ingresado
val textoIngresado = myEditText.text.toString()

// Para escuchar cambios en el texto (opcional)
myEditText.addTextChangedListener(object : TextW
    override fun beforeTextChanged(s: CharSequer
    override fun onTextChanged(s: CharSequence?
        Log.d("MiApp", "Texto cambiado a: $s")
    }
    override fun afterTextChanged(s: Editable?)
})
```

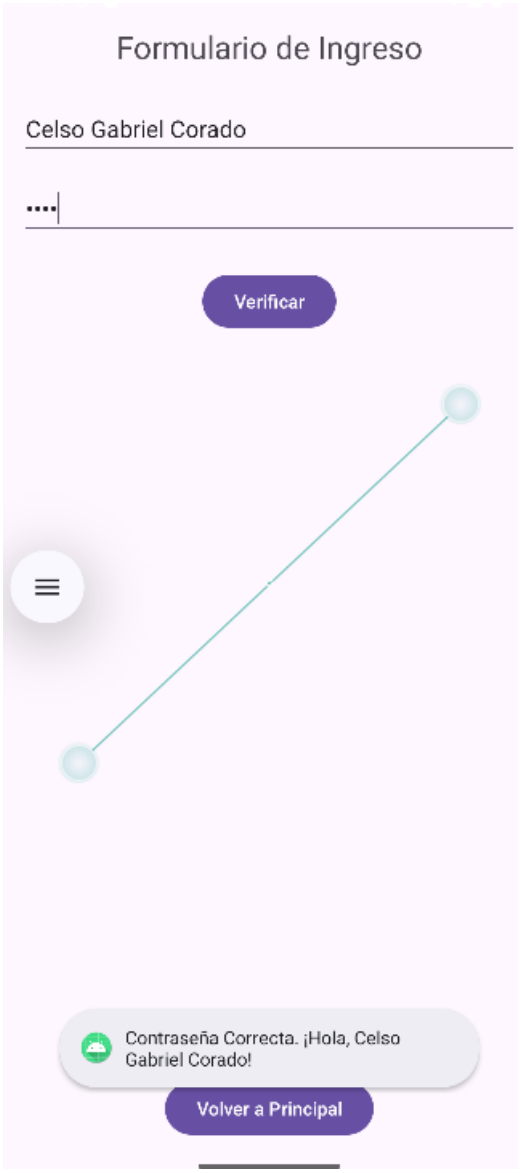

MainActivity.java
activity_main.xml
TextActivity.java
AndroidManifest.x
activity_main.xml
Pixel
36
ExplorandoVistasYNavegaciónProyecto
Palette
Component Tree
Menú Principal de Vistas
Ir a Vistas de Texto (TextView, EditText)
Ir a Vistas de Botones
Ir a Vistas de Widgets (CheckBox, Switch, etc.)
Ir a Vistas de Imágenes (ImageView)
TextView
Button
Button
Button
Button


Medium Pho...
Foldable
Medium Tablet
Desktop


Medium Phone
Menú Principal de Vistas
Ir a Vistas de Texto (TextView, EditText)
Ir a Vistas de Botones
Ir a Vistas de Widgets (CheckBox, Switch, etc.)
Ir a Vistas de Imágenes (ImageView)


Foldable
Menú Principal de Vistas
Ir a Vistas de Texto (TextView, EditText)
Ir a Vistas de Botones
Ir a Vistas de Widgets (CheckBox, Switch, etc.)
Ir a Vistas de Imágenes (ImageView)


Desktop
Menú Principal de Vistas


Formulario de Ingreso
Celso Gabriel Corado
Verificar
Contraseña Correcta. ¡Hola, Celso Gabriel Corado!
Volver a Principal