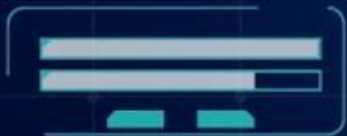


Curso de Desarrollo Móvil en Android





Bienvenido alumno

Sesiones



Completadas



3

Exámenes

2

Tareas

2



José María Cruz Parada
chema.cruz.p@gmail.com



Resumen de curso



Reglamento



Temario



Sesiones

0



Exámenes

2

Tareas

2



Sesión 1



Fundamentos de Java

Sesión 3



Ciclo de vida del Activity

Sesión 5



Fragments

Sesión 2



Fundamentos de Android

Sesión 4



Componentes gráficos

Sesión 6



Servicios Web

Sesión 7

Sesión 8



TextView





TextView.xml

```
<TextView
    android:id="@+id/text_view_id"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="Curso Sesión 4" />
```





TextView.kt

```
// Obtén una referencia al TextView utilizando su ID
val textView = findViewById<TextView>(R.id.text_view_id)

// Establece el texto en el TextView
textView.text = "Curso Sesión 4"

// Agregar un oyente de evento Click
textView.setOnClickListener {
    // Realiza alguna acción cuando se hace clic en el TextView
    Toast.makeText(this, "¡Texto modificado!", Toast.LENGTH_SHORT).show()
}
```



Componentes gráficos

Widgets



EditText





EditText.xml

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





```
EditText.kt

// Obtén una referencia al EditText utilizando su ID
val editText = findViewById<EditText>(R.id.editText)

// Agrega un TextChangeListener para capturar los cambios de texto en el EditText
editText.addTextChangedListener(object : TextWatcher {
    override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        // No se utiliza en este ejemplo
    }

    override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        // Realiza alguna acción cuando el texto cambia en el EditText
        val texto = editText.text.toString()
        if (texto.isNotEmpty()) {
            // Realiza alguna acción con el texto ingresado
            // Por ejemplo, puedes mostrarlo en un TextView
            textView.text = texto
        }
    }

    override fun afterTextChanged(s: Editable?) {
        // No se utiliza en este ejemplo
    }
})
```



Button





Button.xml

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





Button.kt

```
// Encuentra el botón por su ID
val button = findViewById<Button>(R.id.button)

// Agrega un OnClickListener al botón
button.setOnClickListener {
    // Realiza alguna acción cuando se hace clic en el botón
    textView.text = "¡Botón presionado!"
}
```



ImageView





ImageView.xml

```
<ImageView  
    android:id="@+id/imageView"  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:src="@drawable/my_image"  
    android:scaleType="centerCrop"  
    android:padding="8dp" />
```





ImageView.kt



```
// Obtén una referencia al ImageView utilizando su ID
val imageView = findViewById<ImageView>(R.id.imageView)

// Establece la imagen en el ImageView
imageView.setImageResource(R.drawable.my_image)
```




ImageButton





ImageButton.xml

```
<ImageButton  
    android:id="@+id/imageButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/my_image"  
    android:background="@null"  
    android:onClick="onImageButtonClick" />
```



ImageButton.kt

```
override fun onCreate(savedInstanceState: Bundle?) {  
    ...  
  
    // Encuentra el ImageButton por su ID  
    val imageButton = findViewById<ImageButton>(R.id.imageButton)  
}  
  
// Método llamado cuando se hace clic en el ImageButton  
fun onImageButtonClick(view: View) {  
    // Realiza alguna acción cuando se hace clic en el ImageButton  
    // Este método se define en el atributo android:onClick del ImageButton en XML  
    Toast.makeText(this, "ImageButton clickeado", Toast.LENGTH_SHORT).show()  
}
```



Componentes gráficos

Widgets



Checkbox





CheckBox.xml

```
<CheckBox
    android:id="@+id/checkBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Aceptar los términos y condiciones"
    android:checked="false" />
```





CheckBox.kt

```
val checkBox = findViewById<CheckBox>(R.id.checkBox)

checkBox.setOnCheckedChangeListener { buttonView, isChecked ->
    // Realiza alguna acción cuando el estado del CheckBox cambia
    if (isChecked) {
        // El CheckBox está marcado
        // Realiza alguna acción
    } else {
        // El CheckBox está desmarcado
        // Realiza alguna otra acción
    }
}
```





RadioGroup/RadioButton





```
RadioGroupButton.xml

<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Opción 1" />

    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Opción 2" />

</RadioGroup>
```




```
RadioGroupButton.kt

val radioGroup = findViewById<RadioGroup>(R.id.radioGroup)
val radioButton1 = findViewById<RadioButton>(R.id.radioButton1)
val radioButton2 = findViewById<RadioButton>(R.id.radioButton2)

radioGroup.setOnCheckedChangeListener { _, checkedId ->
    when (checkedId) {
        R.id.radioButton1 -> {
            // Opción 1 seleccionada
            // Realiza alguna acción
        }
        R.id.radioButton2 -> {
            // Opción 2 seleccionada
            // Realiza alguna otra acción
        }
    }
}
```



ToggleButton





```
ToggleButton.xml

<ToggleButton
    android:id="@+id/toggleButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Activado"
    android:textOff="Desactivado"
    android:checked="true"
    android:background="@drawable/toggle_button_background"
    android:thumb="@drawable/toggle_button_thumb"
    android:textColor="@color/toggle_button_text_color"
    android:textSize="16sp"
    android:padding="8dp" />
```





ToggleButton.kt

```
val toggleButton = findViewById<ToggleButton>(R.id.toggleButton)

toggleButton.setOnCheckedChangeListener { buttonView, isChecked ->
    if (isChecked) {
        // El ToggleButton está activado
        // Realiza alguna acción
    } else {
        // El ToggleButton está desactivado
        // Realiza alguna otra acción
    }
}
```





Componentes gráficos

Widgets



Switch





```
Switch.xml

<Switch
    android:id="@+id/switchButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Activado"
    android:textOff="Apagado"
    android:thumb="@drawable/switch_thumb"
    android:track="@drawable/switch_track"
    android:showText="true"
    android:checked="true"
    android:thumbTint="@color/switch_thumb_tint"
    android:trackTint="@color/switch_track_tint"
    android:textColor="@color/switch_text_color"
    android:textSize="14sp" />
```



```
Switch.kt

val switch = findViewById<Switch>(R.id.switchButton)

switch.setOnCheckedChangeListener { _, isChecked ->
    if (isChecked) {
        // El Switch está encendido
        // Realiza alguna acción
    } else {
        // El Switch está apagado
        // Realiza alguna otra acción
    }
}
```





Componentes gráficos

Widgets



ProgressBar





ProgressBar.xml

```
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    style="@android:style/Widget.ProgressBar.Horizontal"
    android:progress="50" />
```





ProgressBar.kt

```
val progressBar = findViewById<ProgressBar>(R.id.progressBar)

// Establecer el progreso actual del ProgressBar
progressBar.progress = 50

// Mostrar u ocultar el ProgressBar
progressBar.visibility = ProgressBar.VISIBLE

// Realizar alguna tarea en segundo plano y actualizar el progreso
// progressBar.progress = ... (actualizar el progreso según la tarea)
```





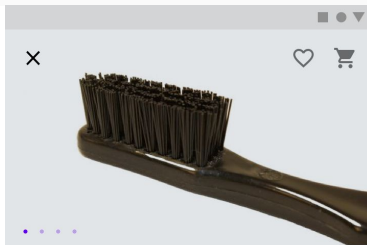
Horizontal Divider





HorizontalDivider.xml

```
<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/darker_gray" />
```



Toothbrush

\$4.50

This toothbrush has bristles infused with charcoal to detoxify and deodorize, making sure you're effectively brushing your teeth.

Select type

Extra Soft

Soft

Medium

Hard

ADD TO CART

Related Items



Vertical Divider





VerticalDivider.xml

```
<View
    android:layout_width="1dp"
    android:layout_height="match_parent"
    android:background="@android:color/darker_gray" />
```





Componentes gráficos

Widgets



by Selen Zeynep

What Buttons are Artists Pushing When They Perform Live

The general consensus is that artists using digital turntables are just pushing random buttons when they perform live. But is that true? In the DMC Championships it's obvious that some artists are putting their various honed skills on display.

Most of the scorn for solo performers comes from the many who headline huge festivals.

♡ Favourite

≡+ Add to playlist





Sesión 1



Fundamentos de Java

Sesión 2



Fundamentos de Android

Sesión 3



Ciclo de vida del Activity

Sesión 4



Componentes gráficos

Sesión 5



Fragments

Sesión 6



Servicios Web



José María Cruz Parada
chema.cruz.p@gmail.com



Resumen de curso



Reglamento



Temario



Sesiones

Sesión 2



Fundamentos de Android

Sesión 4



Componentes gráficos

Sesión 6



Servicios Web



Bienvenido alumno

Sesiones



Completadas



4

Exámenes

3

Tareas

3

Curso de Desarrollo Móvil en Android

