

# Introducción al Desarrollo de Aplicaciones en Android

- Ing. Skrauba Axel

# KivyMD

KivyMD, que significa Kivy Material Design, es una biblioteca de interfaz de usuario (UI) de código abierto diseñada para facilitar el desarrollo de aplicaciones móviles con una interfaz atractiva y moderna en el framework Kivy.

Se enfoca específicamente en la implementación de los [principios de diseño de \*Material Design\*](#), un conjunto de pautas de diseño desarrollado por Google para crear aplicaciones móviles y web consistentes, visualmente atractivas y funcionales.



# Características

**Diseño Material Design:** KivyMD se basa en los principios de diseño de Material Design, lo que significa que las aplicaciones desarrolladas con KivyMD tienen una apariencia y comportamiento modernos y coherentes.

**Amplia variedad de componentes:** KivyMD proporciona una amplia gama de componentes de interfaz de usuario listos para usar, como botones, etiquetas, barras de navegación, tarjetas, diálogos, y más. Estos componentes facilitan la creación de interfaces de usuario sofisticadas y funcionales.

# Características

**Flexibilidad y personalización:** Aunque KivyMD ofrece componentes predefinidos que siguen las pautas de Material Design, también permite una gran flexibilidad y personalización. Puedes ajustar la apariencia y el comportamiento de los componentes para que se adapten a las necesidades de tu aplicación.

**Compatibilidad multiplataforma:** Al estar basado en el framework Kivy, KivyMD es compatible con una amplia variedad de plataformas, incluyendo Android, iOS, y sistemas de escritorio. Esto significa que puedes desarrollar aplicaciones una vez y ejecutarlas en múltiples plataformas sin tener que reescribir gran parte del código.

# *Instalación*

Una vez que tu entorno virtual esté activo, puedes instalar KivyMD y sus dependencias. KivyMD se basa en Kivy, por lo que debes instalar ambos.

```
$ pip install "kivy[base]" kivymd
```

La opción **[base]** asegura que se instale la versión básica de Kivy junto con KivyMD. Para características específicas de Kivy, como soporte para OpenGL o audio, consulta la documentación oficial de Kivy para instalar las opciones adecuadas.

# *Primera Aplicación*

Si todo está correcto, debería de correr el “Hola Mundo”:



```
1  from kivymd.app import MDApp
2  from kivymd.uix.label import MDLabel
3
4
5  class MainApp(MDApp):
6      def build(self):
7          return MDLabel(text="Hello, World", halign="center")
8
9
10 MainApp().run()
```

# *Equivalencia Kivy*



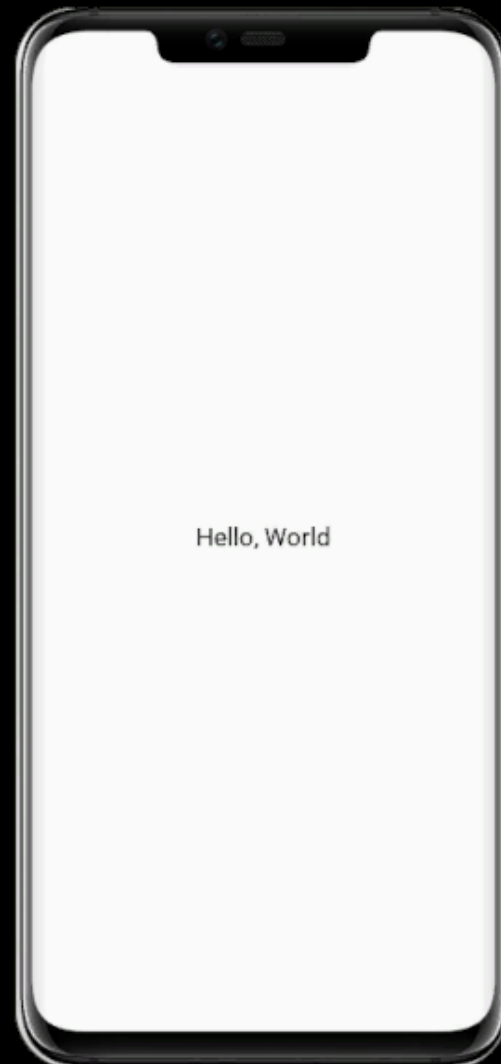
```
1  from kivymd.app import MDApp
2  from kivymd.uix.label import MDLabel
3
4
5  class MainApp(MDApp):
6      def build(self):
7          return MDLabel(text="Hello, World", halign="center")
8
9
10 MainApp().run()
```



```
1  from kivy.app import App
2  from kivy.uix.label import Label
3
4
5  class MainApp(App):
6      def build(self):
7          return Label(text="Hello, World")
8
9
10 MainApp().run()
```

A primera vista, el ejemplo de KivyMD contiene más código...

# *Equivalencia Kivy*





# Botón customizado en Kivy

```
1 from kivy.app import App
2 from kivy.metrics import dp
3 from kivy.uix.behaviors import TouchRippleBehavior
4 from kivy.uix.button import Button
5 from kivy.lang import Builder
6 from kivy.utils import get_color_from_hex
7
8 KV = """
9 #:import get_color_from_hex kivy.utils.get_color_from_hex
10
11
12 <RectangleFlatButton>:
13     ripple_color: 0, 0, 0, .2
14     background_color: 0, 0, 0, 0
15     color: root.primary_color
16
17     canvas.before:
18         Color:
19             rgba: root.primary_color
20         Line:
21             width: 1
22             rectangle: (self.x, self.y, self.width, self.height)
23
24 Screen:
25     canvas:
26         Color:
27             rgba: get_color_from_hex("#0F0F0F")
28         Rectangle:
29             pos: self.pos
30             size: self.size
31 """
32
33
34 class RectangleFlatButton(TouchRippleBehavior, Button):
35     primary_color = get_color_from_hex("#EB8933")
36
37     def on_touch_down(self, touch):
38         collide_point = self.collide_point(touch.x, touch.y)
39         if collide_point:
40             touch.grab(self)
41             self.ripple_show(touch)
42             return True
43         return False
44
45     def on_touch_up(self, touch):
46         if touch.grab_current is self:
47             touch.ungrab(self)
48             self.ripple_fade()
49             return True
50         return False
51
52
53 class MainApp(App):
54     def build(self):
55         screen = Builder.load_string(KV)
56         screen.add_widget(
57             RectangleFlatButton(
58                 text="Hello, World",
59                 pos_hint={"center_x": 0.5, "center_y": 0.5},
60                 size_hint=(None, None),
61                 size=(dp(110), dp(35)),
62                 ripple_color=(0.8, 0.8, 0.8, 0.5),
63             )
64         )
65         return screen
66
67
68 MainApp().run()
```



<https://kivymd.readthedocs.io/en/1.1.1/getting-started/#first-kivymd-application>

# *Botón customizado en KivyMD*

```
1 from kivymd.app import MDApp
2 from kivymd.uix.screen import MDScreen
3 from kivymd.uix.button import MDRectangleFlatButton
4
5
6 class MainApp(MDApp):
7     def build(self):
8         self.theme_cls.theme_style = "Dark"
9         self.theme_cls.primary_palette = "Orange"
10
11     return (
12         MDScreen(
13             MDRectangleFlatButton(
14                 text="Hello, World",
15                 pos_hint={"center_x": 0.5, "center_y": 0.5},
16             )
17         )
18     )
19
20
21 MainApp().run()
```



Hello, World

Contraste Kivy:



Hello, World

# *Arquitectura de KivyMD*

La arquitectura de KivyMD se basa en el framework Kivy, pero se centra específicamente en implementar las pautas de diseño de Material Design de Google.

## **1. Aplicación KivyMD:**

En el centro de la arquitectura se encuentra la aplicación KivyMD. Esta es la clase principal de la aplicación, que hereda de ***kivymd.app.MDApp***. Aquí se define el comportamiento global de la aplicación y se configuran sus propiedades, como el título, el ícono y el tema.

# *Arquitectura de KivyMD*

## **2. Interfaz de usuario (UI):**

Se compone de widgets y componentes de interfaz de usuario que siguen las pautas de Material Design. KivyMD proporciona una amplia variedad de widgets predefinidos, como botones, etiquetas, tarjetas, barras de navegación, diálogos y más.

## **3. Lenguaje de marcado (KV):**

KivyMD utiliza el lenguaje de marcado KV para definir la estructura de la interfaz de usuario. Este lenguaje permite crear una descripción declarativa de la interfaz de usuario, separando la lógica de presentación del código Python. Los archivos KV se utilizan para definir la disposición de los widgets y sus propiedades.

# *Arquitectura de KivyMD*

## **4. Controladores y lógica de la aplicación:**

En el código Python de tu aplicación, puedes definir controladores y lógica para manejar eventos, acciones del usuario y la funcionalidad general de la aplicación. Esto incluye la gestión de datos, la navegación entre pantallas y la interacción con el sistema operativo.

## **5. Temas y estilos:**

KivyMD ofrece una capacidad robusta para personalizar la apariencia de la aplicación a través de temas y estilos. Puedes definir colores, fuentes y otros aspectos visuales para que coincidan con el diseño de tu aplicación.

# *Arquitectura de KivyMD*

## **6. Comportamiento de Material Design:**

La implementación de KivyMD se adhiere a los principios de Material Design, lo que significa que los widgets y componentes tienen un comportamiento coherente y predecible. Esto incluye animaciones suaves, efectos de sombreado y transiciones fluidas.

## **7. Integración con Kivy:**

KivyMD se integra estrechamente con el framework Kivy. Esto significa que puedes combinar widgets de KivyMD con widgets de Kivy estándar para crear interfaces de usuario personalizadas y avanzadas.

# *Diseño de la Interfaz de Usuario*

Con KivyMD, puedes crear interfaces de usuario atractivas y funcionales siguiendo las pautas de Material Design de Google.

## **1. Componentes de la Interfaz de Usuario:**

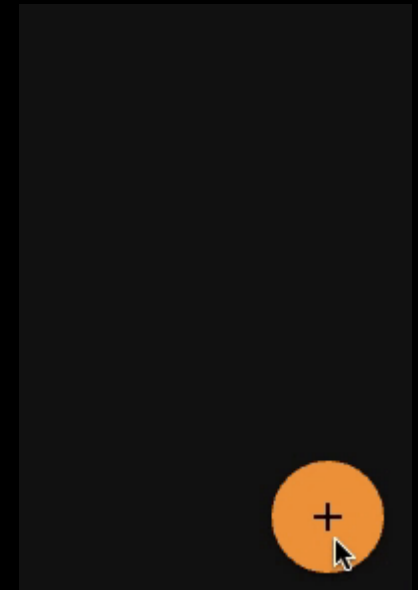
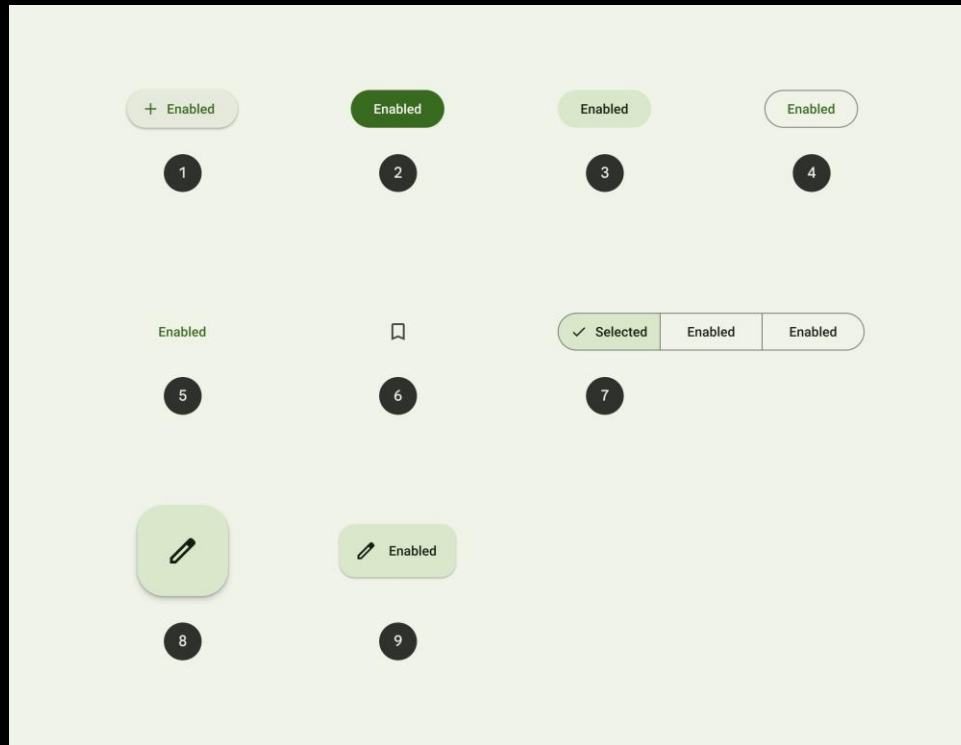
Algunos de los componentes más comunes incluyen: Botones, etiquetas, tarjetas, barras de navegación, listas, diálogos, etc.

Ver: <https://kivymd.readthedocs.io/en/1.1.1/components/>

# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

### ■ Ejemplo Botones



<https://kivymd.readthedocs.io/en/1.1.1/components/button/>



# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

### ■ Ejemplo Etiquetas

#### MDIcon with badge icon #

```
MDIcon:  
  icon: "gmail"  
  badge_icon: "numeric-10"  
  pos_hint: {"center_x": .5, "center_y": .5}
```



#### MDLabel

H1 style

H2 style

H3 style

H4 style

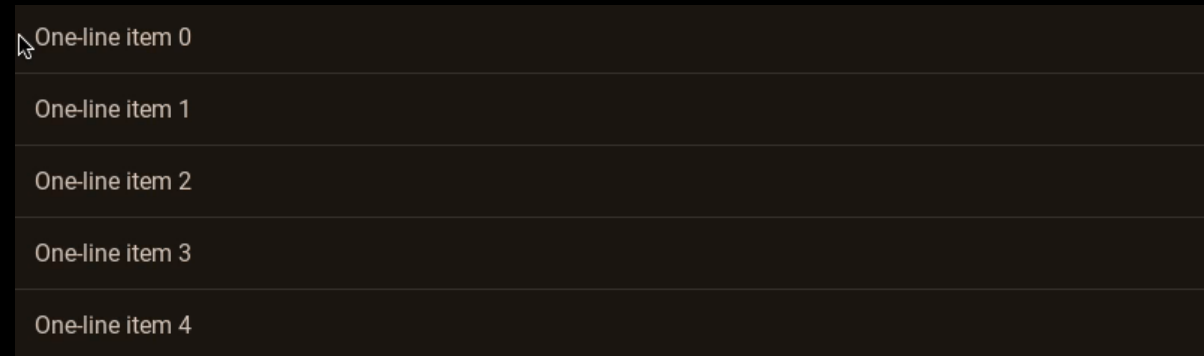
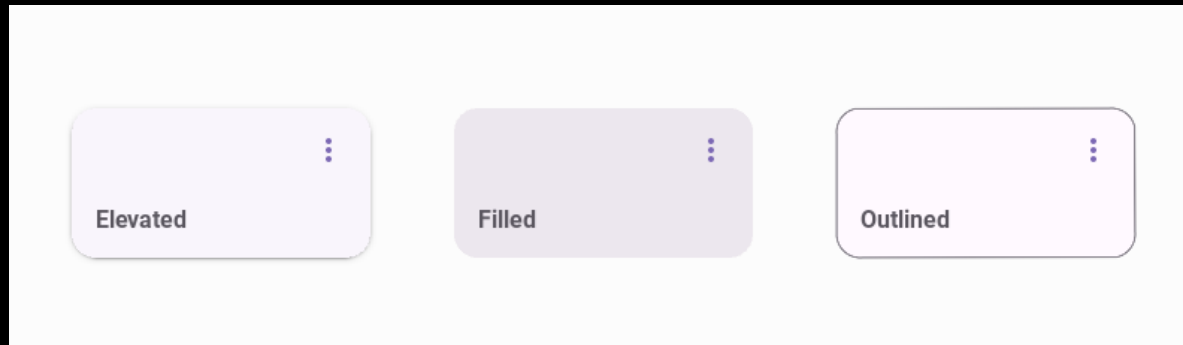
H5 style

H6 style

# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

### ■ Ejemplo Tarjetas

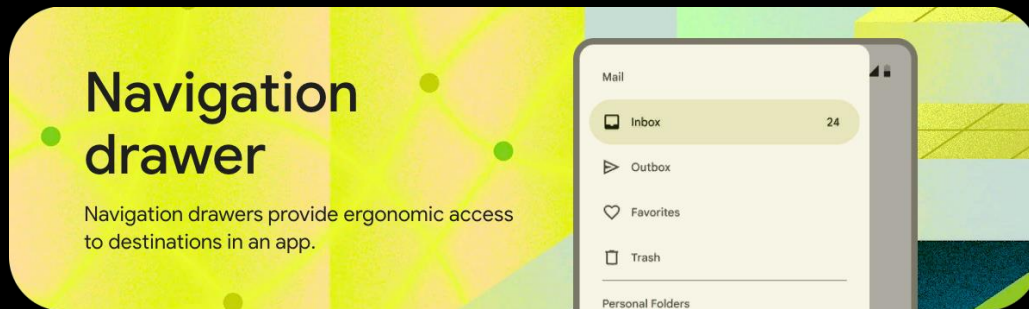


<https://kivymd.readthedocs.io/en/1.1.1/components/card/>

# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

### ■ Ejemplo Barras de Navegación







<https://kivymd.readthedocs.io/en/1.1.1/components/navigationdrawer/>

# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

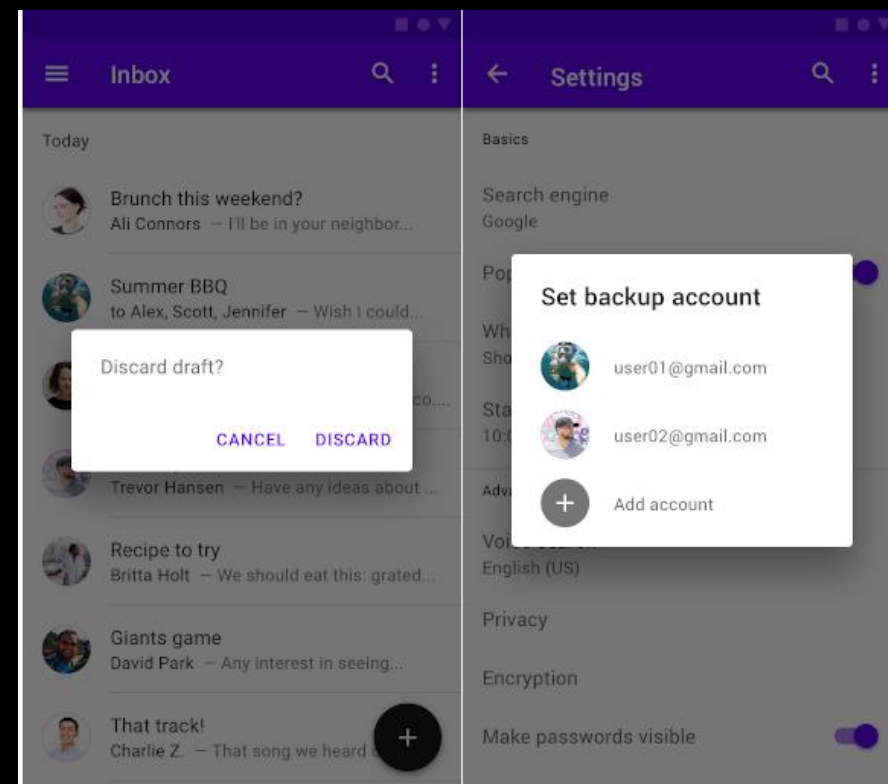
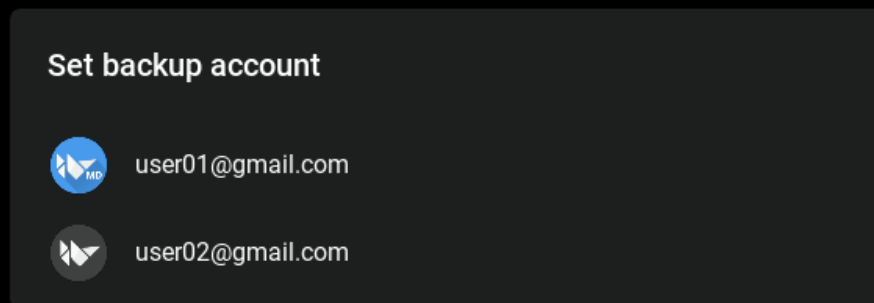
### ■ Ejemplo Listas

	Item 0	<input checked="" type="checkbox"/>
	Item 1	<input checked="" type="checkbox"/>
	Item 2	<input type="checkbox"/>
	Item 3	<input type="checkbox"/>

# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

### ■ Ejemplo Diálogos

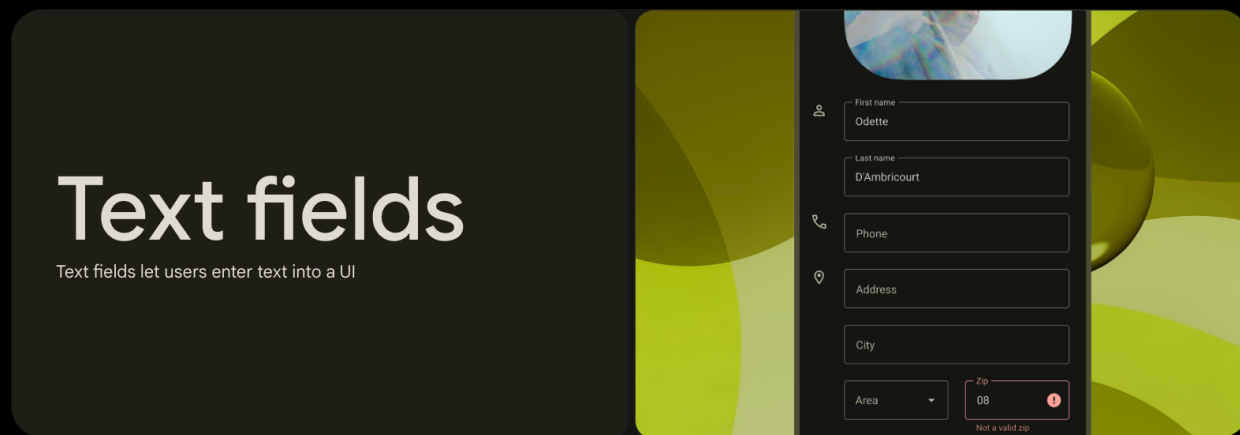


<https://kivymd.readthedocs.io/en/1.1.1/components/dialog/>

# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

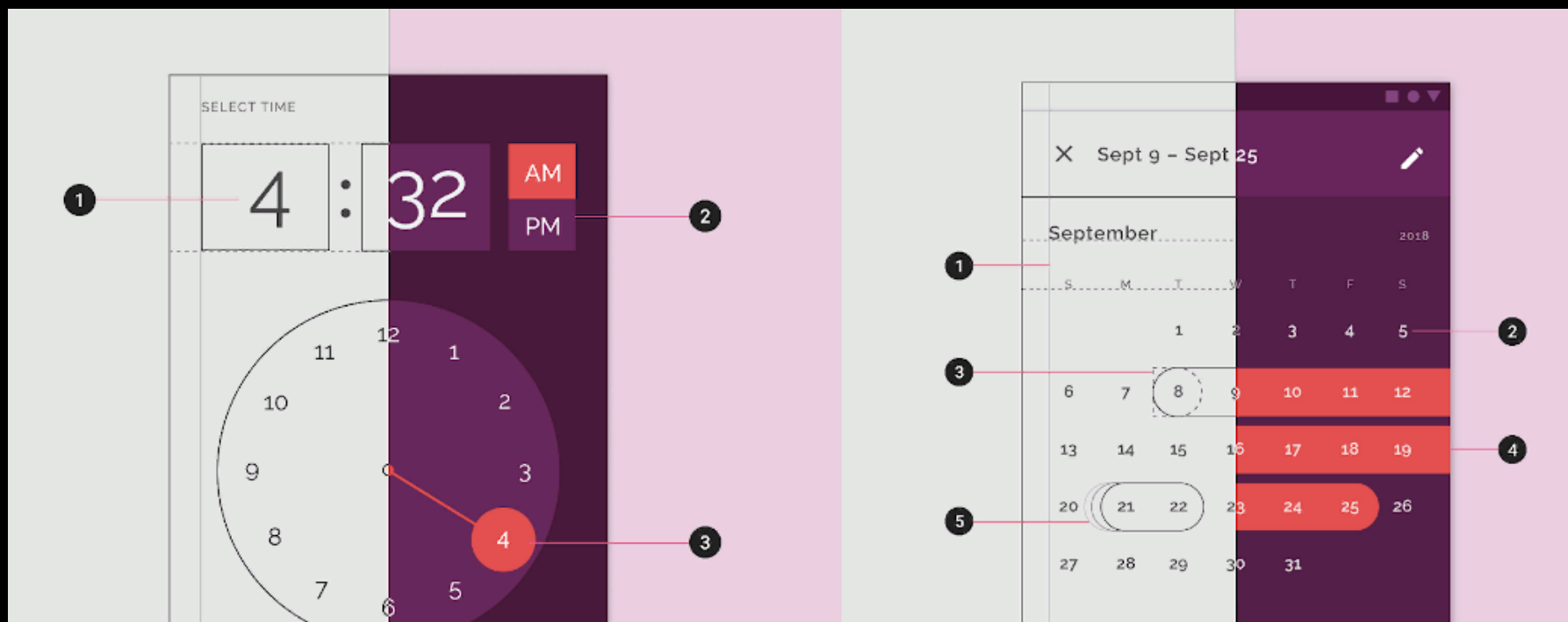
### ■ Ejemplo Campos de Texto



# *Diseño de la Interfaz de Usuario*

## 1. Componentes de la Interfaz de Usuario:

### ■ Ejemplo Selector de Fecha



<https://kivymd.readthedocs.io/en/1.1.1/components/datepicker/>

# *Diseño de la Interfaz de Usuario*

## **2. Diseño Responsivo:**

Es importante que el diseño de tu interfaz de usuario sea responsivo, lo que significa que debe adaptarse a diferentes tamaños de pantalla y orientaciones (vertical u horizontal). KivyMD facilita esto al proporcionar contenedores flexibles y reglas de diseño que se ajustan automáticamente a las dimensiones de la pantalla.

## **3. Colores y Temas:**

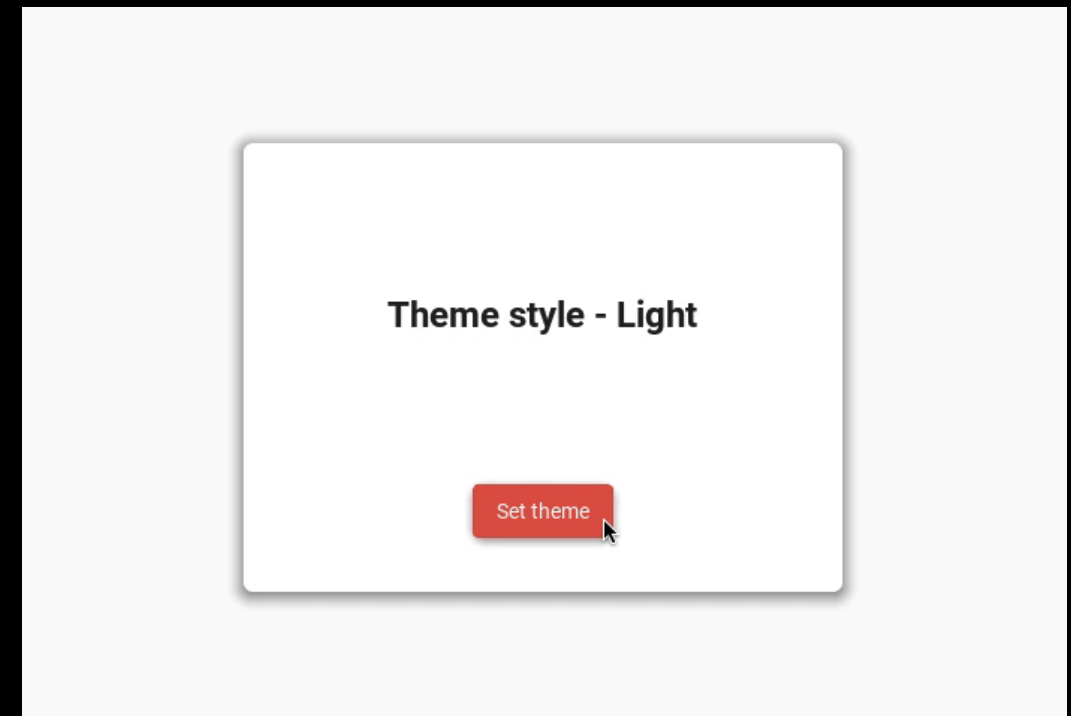
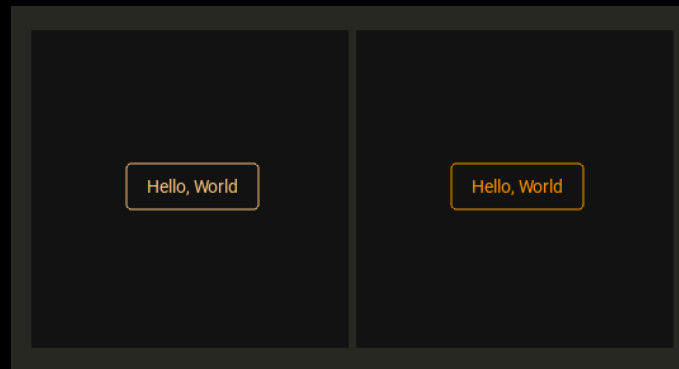
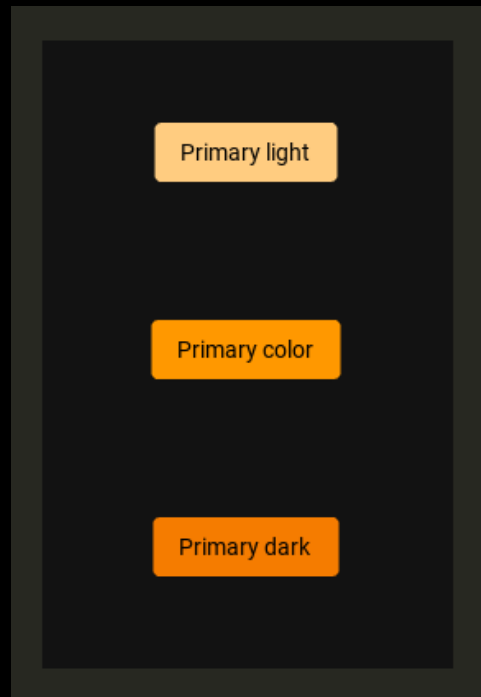
Permite personalizar la apariencia de tu aplicación a través de temas y estilos. Puedes definir colores, fuentes y otros aspectos visuales para que coincidan con el diseño de tu aplicación y seguir las pautas de Material Design.



# *Diseño de la Interfaz de Usuario*

## 3. Colores y Temas:

### ■ Ejemplo Temas



<https://kivymd.readthedocs.io/en/1.1.1/themes/theming/>

# *Diseño de la Interfaz de Usuario*

### 3. Colores y Temas:

- ## ■ Ejemplo Definición de Colores



# *Diseño de la Interfaz de Usuario*

## **4. Iconografía:**

Material Design también hace hincapié en el uso de íconos para mejorar la usabilidad y la identificación visual. KivyMD proporciona una colección de íconos incorporados que puedes utilizar, y también permite la incorporación de íconos personalizados.

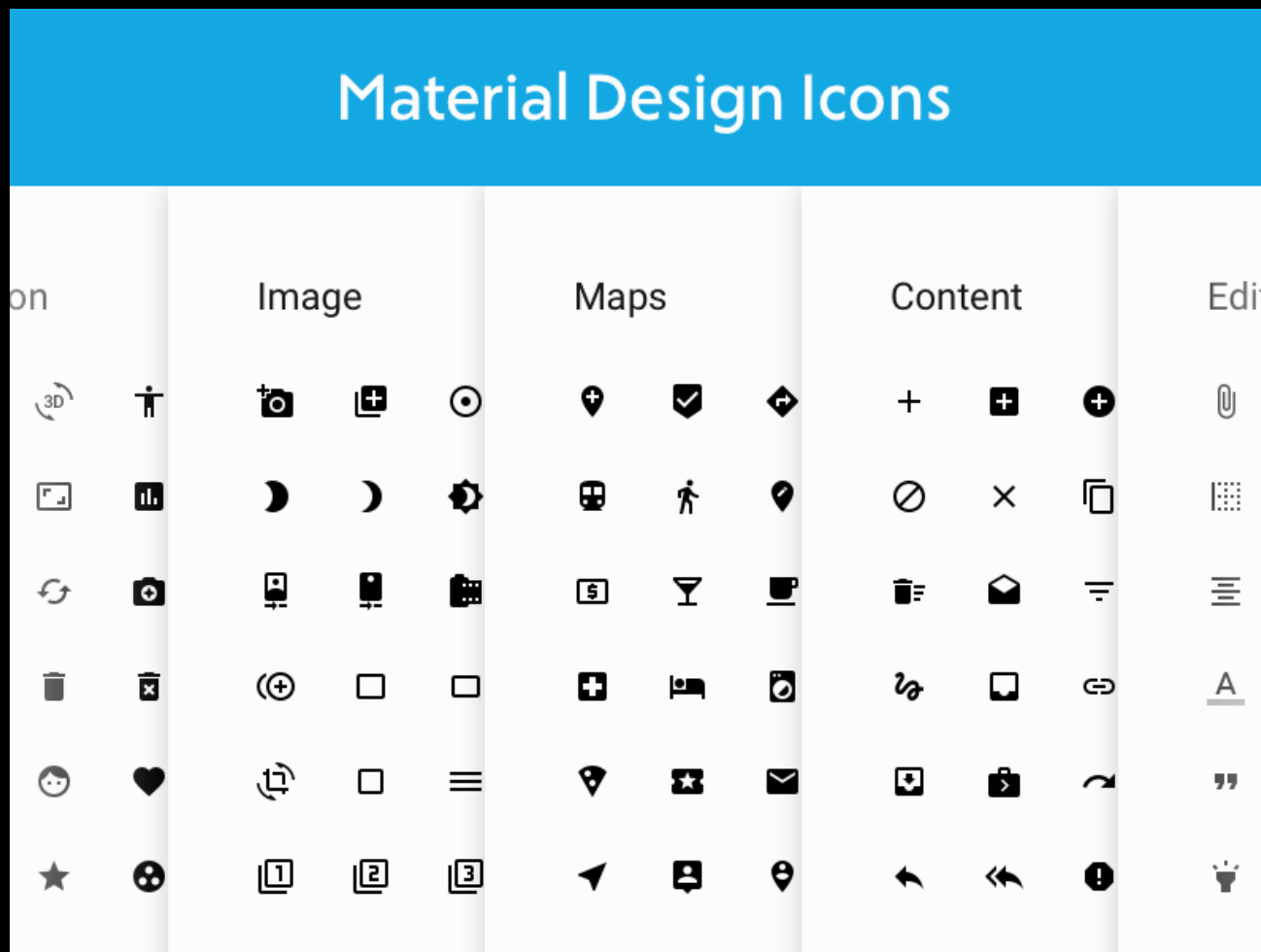
<https://kivymd.readthedocs.io/en/1.1.1/themes/icon-definitions/>

<https://materialdesignicons.com/>

# *Diseño de la Interfaz de Usuario*

## 4. Iconografía:

- ## ■ Ejemplo Íconos



# *Diseño de la Interfaz de Usuario*

## 5. Tipografía:

### ■ Ejemplo Definición de Fuentes

Scale Category	Typeface	Font	Size	Case	Letter spacing
H1	Roboto	Light	96	Sentence	-1.5
H2	Roboto	Light	60	Sentence	-0.5
H3	Roboto	Regular	48	Sentence	0
H4	Roboto	Regular	34	Sentence	0.25
H5	Roboto	Regular	24	Sentence	0
H6	Roboto	Medium	20	Sentence	0.15

H5	Roboto	Regular	24	Sentence	0
H6	Roboto	Medium	20	Sentence	0.15
Subtitle 1	Roboto	Regular	16	Sentence	0.15
Subtitle 2	Roboto	Medium	14	Sentence	0.1
Body 1	Roboto	Regular	16	Sentence	0.5
Body 2	Roboto	Regular	14	Sentence	0.25
BUTTON	Roboto	Medium	14	All caps	1.25
Caption	Roboto	Regular	12	Sentence	0.4
OVERLINE	Roboto	Regular	10	All caps	1.5

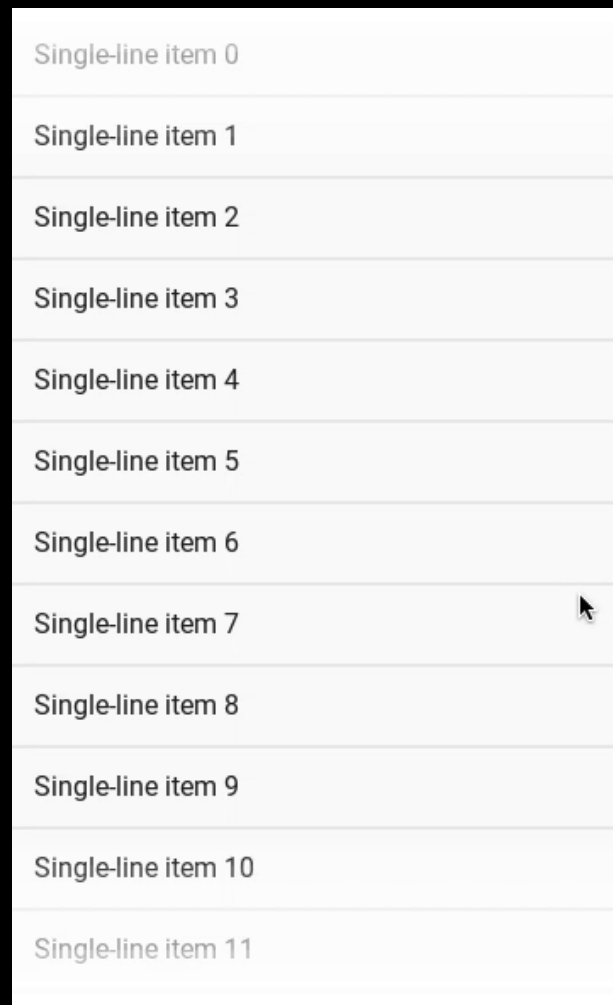
<https://kivymd.readthedocs.io/en/1.1.1/themes/font-definitions/>

<https://m2.material.io/design/typography/the-type-system.html#type-scale>

# *Diseño de la Interfaz de Usuario*

## 6. Animaciones y Transiciones:

KivyMD facilita la implementación de animaciones suaves y transiciones entre diferentes pantallas o estados de la aplicación. Las animaciones pueden mejorar significativamente la experiencia del usuario y hacer que la aplicación se sienta más atractiva y dinámica.



# Preguntas...

- Dudas
- Sugerencias
- Cuestiones
- Vamos al código...