

ACTIVIDAD 05

- ❖ Empezamos creando el archivo `main.py`.
- ❖ luego creamos los archivos `home.kv`, `login.kv`, `registro.kv`.
- ❖ también es importante crear la base de datos local en xampp, la base de datos lo llamare dbkivy con una tabla usuario con el cual se comunicara de forma directa nuestra aplicación móvil.

dbkivy usuario	
id	int(11)
usuario	varchar(50)
email	varchar(65)
contraseña	varchar(30)

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0002 segundos.)

```
SELECT * FROM `usuario`
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

	id	usuario	email	contraseña
<input type="checkbox"/>		1 mullisaca	mullisaca@kivy.com	123

☐ [Editar](#) [Copiar](#) [Borrar](#) [Seleccionar todo](#) Para los elementos que están marcados: [Editar](#) [Copiar](#) [Borrar](#) [Exportar](#)

Home.kv

```

1 MDScreen:
2     name: 'home'
3     MDFloatLayout:
4         md_bg_color: ('#00d8ff')
5         Image:
6             source: ('pruebas/lobo.png')
7             size_hint: (.2, .2)
8             pos_hint: {'center_x':.15,'center_y':.9}
9         Image:
10            source: ("assets/developer.png")
11            pos_hint: {'center_x':.5,'center_y':.8}
12            size_hint: .6,.6
13        MDLabel:
14            text: "MC Danny.Dev"
15            font_name: 'assets/B_popins'
16            font_size: '28sp'
17            pos_hint: {'center_y':.4}
18            halign: 'center'
19            color: ('#ffffff')
20        MDLabel:
21            text: "SENATI el mejor lugar para escribir codigo y compartir experiencias"
22            font_name: 'assets/Lpopins'
23            font_size: '12sp'
24            size_hint_x: .85
25            pos_hint: {'center_y':.33,'center_x':.5}
26            halign: 'center'
27            color: ("#F04924")
28        Button:
29            text: "LOGIN"
30            size_hint: (.66, .065)
31            pos_hint: {'center_x':.5,'center_y':.18}
32            background_color: (1,0,0,1)
33            font_name: 'assets/B_popins'

```

Este código es para la pantalla principal de mi aplicación. Tiene un fondo azul claro, dos imágenes (una del logo y otra de un desarrollador), y dos textos: uno con el nombre "MC Danny.Dev" y otro con una frase sobre SENATI. Abajo hay dos botones, uno para iniciar sesión ("LOGIN") y otro para registrarse ("SIGN UP"). Al hacer clic en los botones, la aplicación cambia de pantalla.

Login.kv

```

1 login.kv
2 MDScreen:
3     name: 'login'
4     MDFloatLayout:
5         md_bg_color: '#eeeeee'
6
7     MDIconButton:
8         icon: "assets/atras.png"
9         size_hint: (.05, .05)
10        pos_hint: {'center_x': .05, 'center_y': .975}
11        on_release:
12            root.manager.transition.direction = "right"
13            root.manager.current = 'home'
14
15        MDLabel:
16            text: '¡Bienvenido!'
17            pos_hint: {'center_x': .5, 'center_y': .9}
18            halign: 'center'
19            font_name: 'assets/B_popins'
20            font_size: '26sp'
21            color: (0, 0.85, 1, 1)
22
23        MDCard:
24            orientation: 'vertical'
25            size_hint: (0.9, None)
26            height: self.minimum_height
27            pos_hint: {'center_x': .5, 'center_y': .41}
28            padding: "24dp"
29            spacing: "16dp"
30            md_bg_color: '#ffffff'
31
32            MDLabel:
33                text: 'LOGIN'
34                halign: 'center'

```

Este código define la pantalla de inicio de sesión de mi aplicación. El fondo es gris claro, y en la parte superior hay un botón de flecha que te permite volver a la pantalla principal. La pantalla tiene un saludo de "¡Bienvenido!" y un formulario de inicio de sesión con campos para ingresar el correo electrónico y la contraseña.

También hay un botón para iniciar sesión, y debajo se muestra un mensaje de error si es necesario. Si olvidas tu contraseña, hay un enlace para recuperarla. Además, hay botones para iniciar sesión con Google, Facebook, GitHub y X. Al final, hay un enlace para registrarse si aún no tienes una cuenta.

Registro.kv

```
registro.kv
1  MDScreen:
2      name: 'registro'
3      MDFloatLayout:
4          md_bg_color: '#eeeeee' # Background color for the screen
5
6          MIconButton:
7              icon: "assets/atras.png"
8              size_hint: (.05, .05)
9              pos_hint: {'center_x': .05, 'center_y': .975}
10             on_release:
11                 root.manager.transition.direction = "right"
12                 root.manager.current = 'login'
13
14             MDLabel:
15                 text: '¡Registro!'
16                 pos_hint: {'center_x': .5, 'center_y': .9}
17                 halign: 'center'
18                 font_name: 'assets/B_popins'
19                 font_size: '26sp'
20                 color: (0, 0.85, 1, 1) # Light blue color
21
22             MDCard:
23                 orientation: 'vertical'
24                 size_hint: (0.9, None)
25                 height: self.minimum_height
26                 pos_hint: {'center_x': .5, 'center_y': .41}
27                 padding: "24dp"
28                 spacing: "16dp"
29                 md_bg_color: '#ffffff' # Card background color
30
31                 MDLabel:
32                     text: 'REGISTRARSE'
33                     halign: 'center'
```

Este código define la pantalla de registro de mi aplicación. El fondo es gris claro, y en la parte superior hay un botón de flecha para regresar a la pantalla de inicio de sesión. En el centro de la pantalla, hay una tarjeta blanca donde se encuentran los campos para ingresar un nombre de usuario, correo electrónico y una contraseña.

Hay un botón grande que dice "Registrarse", que al presionarlo llamará a la función para registrar al usuario. Si hay algún error, aparecerá un mensaje en rojo debajo del botón de registro.

También se incluyen iconos para opciones de registro con Google, Facebook, GitHub y X. Finalmente, si ya tienes una

cuenta, hay un enlace para volver a la pantalla de inicio de sesión.

Main.py

Configuración inicial :

```
main.py > MainApp > build
1 from kivy.app import MDApp
2 from kivy.lang import Builder
3 from kivy.uix.screenmanager import ScreenManager
4 from kivy.core.window import Window
5 from kivy.md.toast import toast
6 from database import Conectar, Cerrar_Conexion
7
8 Window.size = (310, 580)
```

Definimos el tamaño de la ventana para que sea pequeña, como el tamaño de un celular.

La aplicación se llama "App Login" y tiene un icono con una imagen de un lobo.

Pantallas:

```
10 class MainApp(MDApp):
11     def build(self):
12         self.ventana = ScreenManager()
13         self.icon = 'pruebas/Lobo.png'
14         self.title = "App Login"
15
16         # Cargar las pantallas desde archivos .kv
17         self.ventana.add_widget(Builder.load_file("home.kv"))
18         self.ventana.add_widget(Builder.load_file("login.kv"))
19         self.ventana.add_widget(Builder.load_file("registro.kv"))
20
21     return self.ventana
```

La aplicación tiene varias pantallas: una para la página principal, otra para el inicio de sesión y otra para el registro. Estas pantallas están en archivos separados llamados home.kv, login.kv, y registro.kv.

Inicio de sesión:

```
23 def IniciarSession(self):
24     app = MDApp.get_running_app()
25     correo = app.ventana.get_screen('login').ids.email.text
26     contra = app.ventana.get_screen('login').ids.contrasenha.text
27
28     # Conectar a la base de datos
29     conectar = Conectar()
30     c = conectar.cursor()
31     consulta = "SELECT COUNT(*) FROM usuario WHERE email = %s AND contrasenha = %s"
32     c.execute(consulta, (correo, contra))
33     data = c.fetchone()
34     count = data[0] if data else 0
35
36     if count == 0:
37         toast('Credenciales Incorrectas')
38     else:
39         toast('Credenciales Correctas')
40
41     Cerrar_Conexion(conectar)
```


Cuando alguien intenta iniciar sesión, obtendrá el correo electrónico y la contraseña que ingresó.

Luego, se conecta a la base de datos para verificar si esos datos coinciden con los registrados.

Si no coincide, muestra un mensaje que dice "Credenciales Incorrectas". Si coinciden, muestra "Credenciales Correctas".

Registro:

```
43 def register(self):
44     app = MDApp.get_running_app()
45     usuario = app.ventana.get_screen('registro').ids.usuario.text
46     correo = app.ventana.get_screen('registro').ids.correo.text
47     contrasena = app.ventana.get_screen('registro').ids.contrasenha.text
48
49     # Validar que los campos no estén vacíos
50     if not usuario or not correo or not contrasena:
51         toast('Por favor, complete todos los campos.')
52         return
53     if len(contrasena) < 3:
54         toast('La contraseña debe tener al menos 3 caracteres.')
55         return
56     conectar = Conectar()
57     c = conectar.cursor()
58     consulta = "SELECT COUNT(*) FROM usuario WHERE email = %s"
59     c.execute(consulta, (correo,))
60     if c.fetchone()[0] > 0:
61         toast('El correo ya está registrado.')
62         Cerrar_Conexion(conectar)
63         return
64     consulta_insertar = "INSERT INTO usuario (usuario, email, contrasenha) VALUES (%s, %s, %s)"
65     c.execute(consulta_insertar, (usuario, correo, contrasena))
66     conectar.commit()
67     Cerrar_Conexion(conectar)
68     toast('Registro exitoso. Ahora puede iniciar sesión.')
69     self.ventana.current = 'login'
```

En la pantalla de registro, se obtiene el nombre de usuario, correo electrónico y contraseña.

Primero, verifica que todos los campos estén llenos. Si falta algo, muestra un mensaje pidiendo que se complete.

También revisa si la contraseña es lo suficientemente larga (al menos 3 caracteres).

Luego, revisa si el correo ya está registrado en la base de datos. Si ya está registrado, muestra un mensaje que dice que ese correo ya fue usado.

Si todo está bien, guarda los datos en la base de datos y muestra un mensaje de éxito.

Después de registrarse, automáticamente lleva a la persona a la pantalla de inicio de sesión.

```
70
71 v if __name__ == "__main__":
72     MainApp().run()
```

Database.py

Conectar():

```
1  import mysql.connector
2
3  def Conectar():
4      """Conectar a la base de datos."""
5      return mysql.connector.connect(
6          host="localhost",
7          user="mullisaca",
8          password="123",
9          database="dbkivy"
10     )
```

Se utiliza para establecer una conexión con una base de datos MySQL. En este caso, se conecta a una base de datos de llamada dbkivy, que está alojada localmente (en su propia máquina), usando el nombre de usuario mullisacay la contraseña 123.

Esta función devuelve la conexión para poder interactuar con la base de datos.

Cerrar_Conexion(conexion):

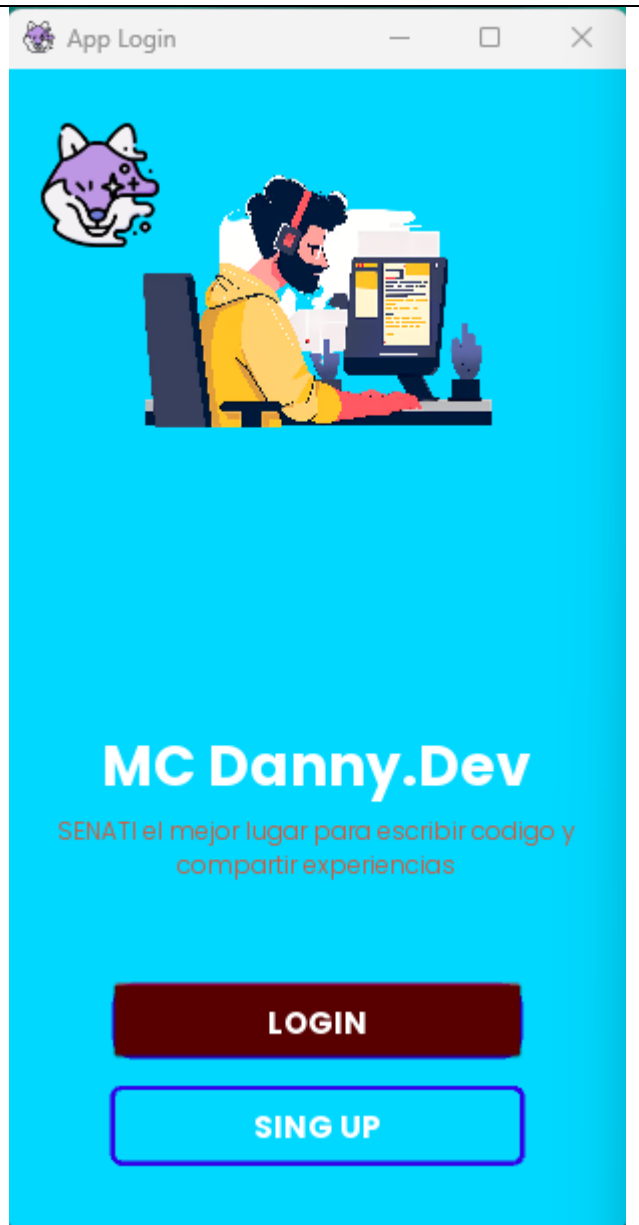
```
12  def Cerrar_Conexion(conexion):
13      """Cerrar la conexión a la base de datos."""
14      if conexion.is_connected():
15          conexion.close()
16
```

Sirve para cerrar la conexión a la base de datos que se creó anteriormente.

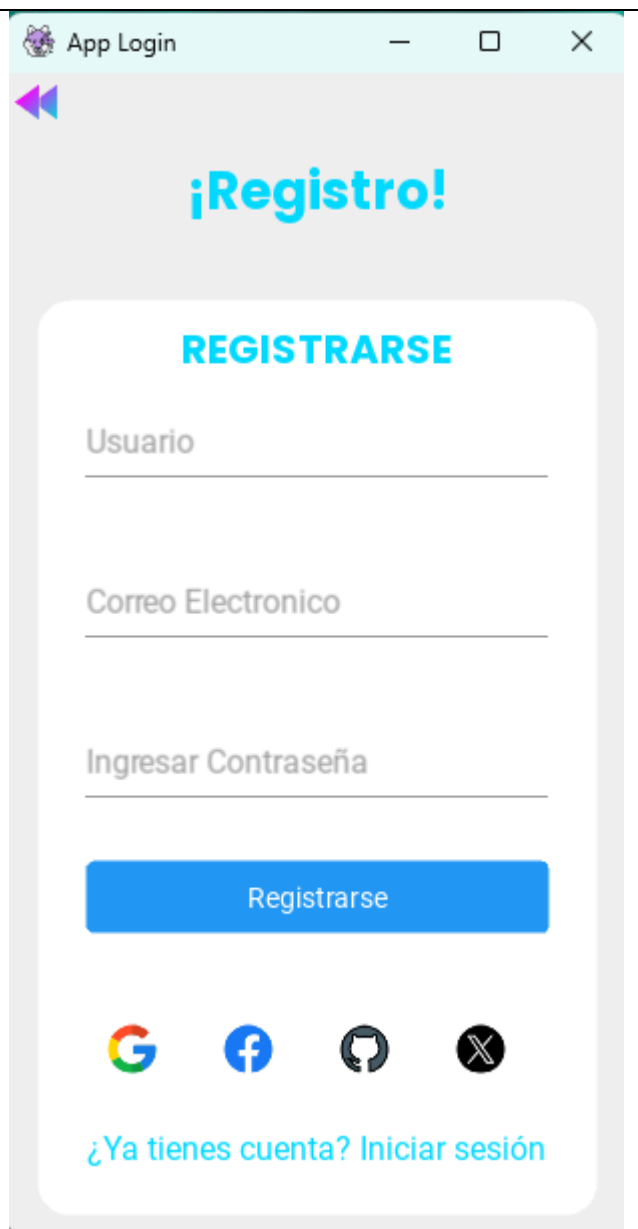
Verifique si la conexión está activa (is_connected()) y si es así, la cierra.

aplicación terminada

Pantalla de inicio o home donde nos muestra las opciones o botones para iniciar sesión o registrarse en la aplicación



**Vista para registrarse
en la aplicacion**



The image shows a mobile application interface for registration. At the top, there is a light blue header bar with a small icon and the text "App Login". Below the header, the main background is light gray. A large, bold, blue text "¡Registro!" is centered at the top. Below this, a white rounded rectangle contains the registration form. The form has a title "REGISTRARSE" in bold blue text. It includes three input fields: "Usuario", "Correo Electronico", and "Ingresar Contraseña", each with a light gray placeholder text and a horizontal line for input. Below the input fields is a blue button with the text "Registrarse" in white. Under the button are four social media icons: Google, Facebook, GitHub, and X. At the bottom of the white rounded rectangle, there is a link in blue text that says "¿Ya tienes cuenta? Iniciar sesión".

App Login

¡Registro!





REGISTRARSE

Usuario

Correo Electronico

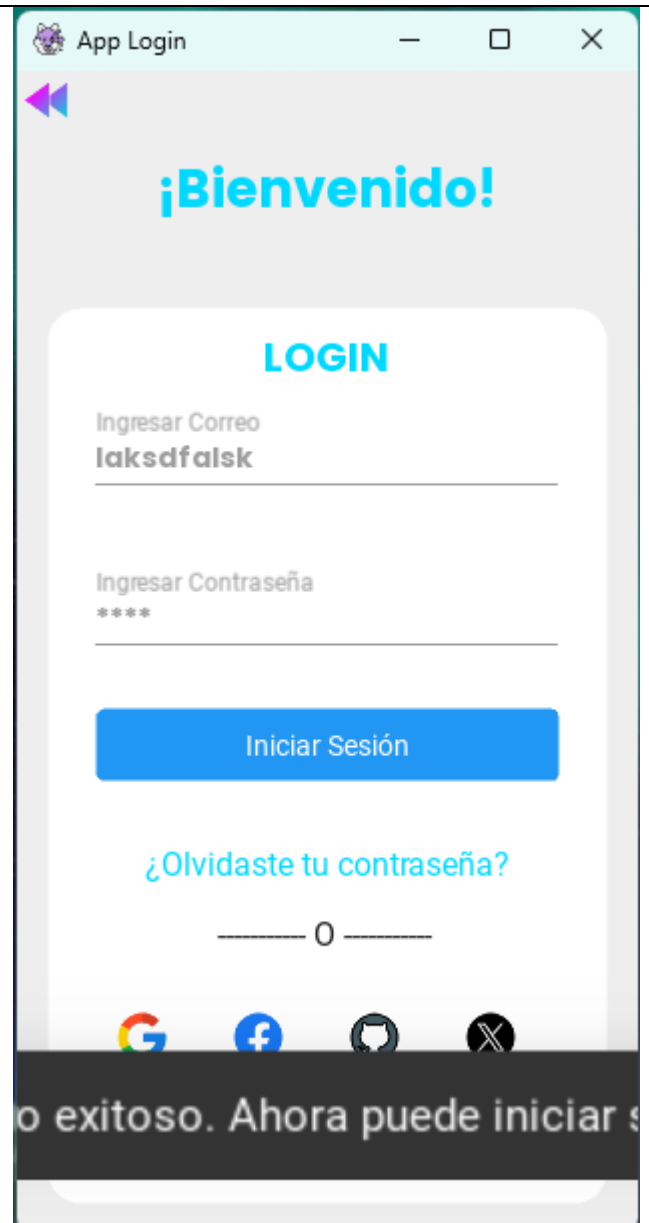
Ingresar Contraseña

Registrarse

[¿Ya tienes cuenta? Iniciar sesión](#)

Después el registro exitoso nos manda a la pagina de inicio de sesión para su respectivo inicio.



The screenshot shows a mobile application interface for login. At the top, there's a header bar with a purple cat icon, the text 'App Login', and standard mobile window controls (back, close, and a third icon). Below the header, a large blue '¡Bienvenido!' (Welcome!) message is displayed. The main content area is a white rounded rectangle with the title 'LOGIN' in blue. It contains two input fields: 'Ingresar Correo' (Email) with the placeholder text 'laksdfalsk' and 'Ingresar Contraseña' (Password) with four asterisks. A blue 'Iniciar Sesión' (Login) button is positioned below the fields. Underneath the button, there's a link '¿Olvidaste tu contraseña?' (Forgot your password?) and a progress indicator showing '0' between two horizontal lines. At the bottom of the white area, there are four social media icons: Google, Facebook, GitHub, and X. A dark grey banner at the very bottom of the screen contains the text 'o exitoso. Ahora puede iniciar s' (or successful. Now you can start s).

**Alerta de que los datos
son incorrectos**

App Login

¡Bienvenido!

LOGIN

Ingresar Correo





laksdfalskasdf

Ingresar Contraseña

Iniciar Sesión

¿Olvidaste tu contraseña?

0



Credenciales Incorrectas

**Inicio de sesión
exitoso**

