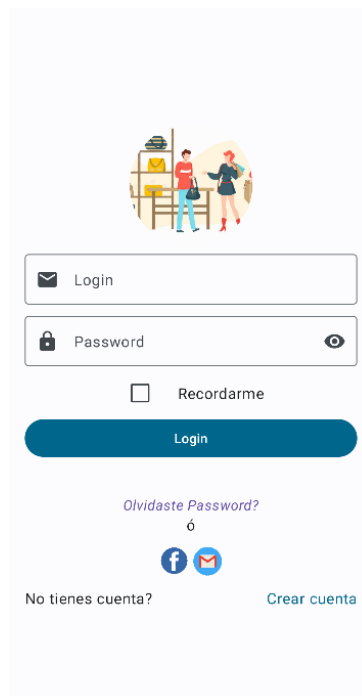


Primera versión de Login

[Descargar estos apuntes](#)

Ejercicio

Vamos a crear una aplicación con aspecto similar al siguiente, usaremos los **TextField** explicados en el tema con el tipo **Validacion**, será necesario crear un TextField propio para el password que permita visualizar o no la contraseña (será sencillo encontrarlo en la web). La pantalla de la imagen, la crearemos a partir de la función **LoginScreen** que deberemos situar en el lugar correcto, según la arquitectura explicada en temas anteriores.



⚠ Aviso: Como ya estamos usando varios componentes TextField, vamos a situarlos en el **paquete composables** del que ya hemos hablado en el tema de la arquitectura, podrías meterlos en un fichero **FieldTextsCommon.kt**. La interfaz Validacion la pondremos dentro del **paquete utilities.validacion**

Habrás que validar las **entradas incorrectas** de los TextField, de forma que se mostrarán los mensajes de error: indicando que no pueden estar vacíos, para el correo que debe cumplir el formato de correo y para el password que debe ser mayor o igual a 8.

Login*

✉ ejemplo@correo.com

No puede estar vacío

Password*

No puede estar vacío

Login*

✉ x

Correo no válido

Password

Login

✉ x@x.com

Password*

El password debe tener como mínimo 8 caracteres

Si se pulsa el botón Login con datos que tengan el formato correcto, se mostrará un Toast con el mensaje de que se está entrando en la App. Mientras si alguno de los formatos o ambos son incorrecto, el mensaje será de aviso.

Login

✉ x@x.com

Password*

No puede estar vacío

☐ Recordarme

Login

Olvidaste Password?

o

f e

No tienes cuenta? Crear cuenta

Comproueba los datos

Login

✉ x@x.com

Password

☐ Recordarme

Login

Olvidaste Password?

o

f e

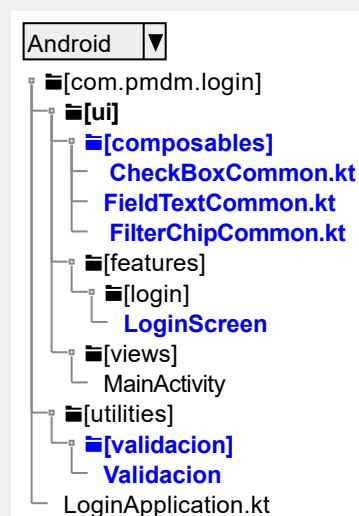
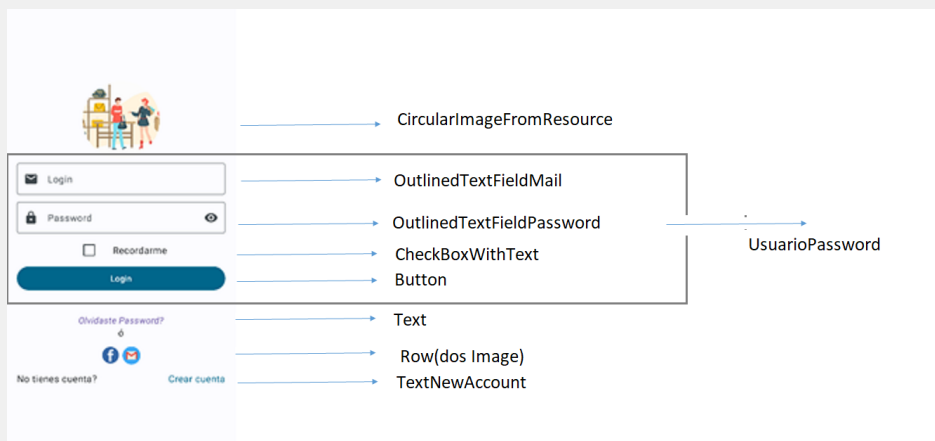
No tienes cuenta? Crear cuenta

Entrando a la APP

⚠ Aviso: como en ejercicios anteriores, la lógica de la app se manejará fuera de los Composables que forman la screen (Preview o Activity), por lo que toda la parte de las validaciones tendrán que ir fuera de la función LoginScreen, necesitando pasar como variables de tipo State **validacionLogin** y **validacionPassword** además de las necesarias para los TextField. Para definir una interfaz como estado deberemos de concretarle el tipo con `as`, por ejemplo:

```
var validacionLogin by remember { mutableStateOf(object : Validacion {} as Validacion) }
```

💡 Tips: Los componentes que debemos definir (algunos ya están en el tema) son:



- Definidos en **FieldTextCommon.kt**

OutlinedTextFieldMail se basa en un **OutlinedTextFieldWithErrorState**

OutlinedTextFilePassword basado en **OutlinedTextField**

- Definidos en **TextCommon.kt**

TextNewAccount que lo crearemos con la siguiente función:

```
fun TextNewAccount( onClick: () -> Unit,  
                    color: Color = MaterialTheme.colorScheme.primary  
                    )
```

Que incluirá un Row con dos Text.

- Definidos en **ImageCommon.kt**

CircularImageFromResource que incluirá un **Image**