

## Instagram Layout Clone

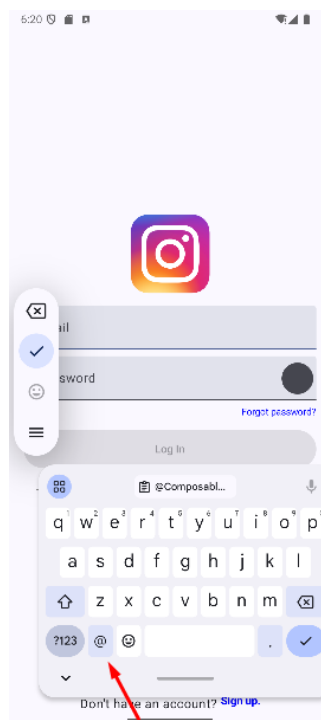
Replicamos el diseño de la pantalla de inicio de sesión de Instagram.

### Teclado tipo email

El primer paso es poner el teclado de tipo email, para ello añadiremos esta última línea a la función Email que muestra el campo donde se introduce el email:

```
@Composable
fun Email(email: String, onTextChanged: (String) → Unit) {
    TextField(
        value = email,
        onValueChange = onTextChanged,
        placeholder = { Text("Email") },
        modifier = Modifier.fillMaxWidth(),
        singleline = true,
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Email)
    )
}
```

Y como podemos observar, al compilar ya se nos muestra el teclado de esta forma:

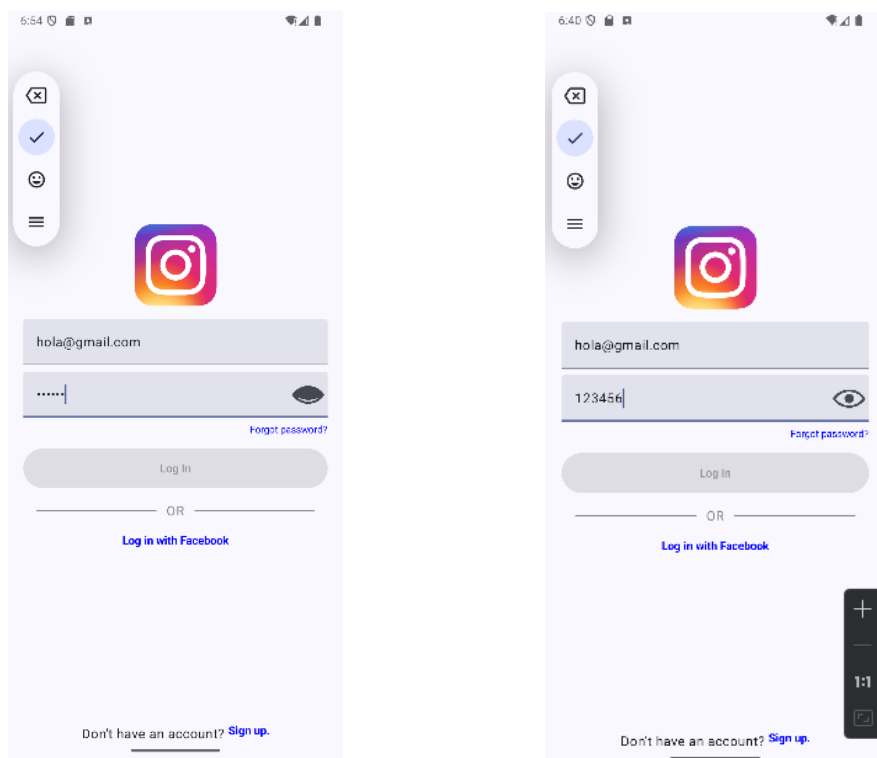


## Contraseña ofuscada

La contraseña ofuscada con opción de mostrarla lo permite el `visualTransformation` junto con el `trailingIcon`, que también nos permite cambiar el icono del ojo si lo deseamos, como he hecho.

```
@Composable
fun Password(password: String, onTextChanged: (String) → Unit) {
    var isVisible by remember { mutableStateOf(false) }
    TextField(
        value = password,
        onValueChange = onTextChanged,
        placeholder = { Text("Password") },
        modifier = Modifier.fillMaxWidth(),
        singleLine = true,
        visualTransformation = if (isVisible) VisualTransformation.None else
        PasswordVisualTransformation(),
        trailingIcon = {
            val eye_con = if (isVisible) R.drawable.eye_icon else R.drawable.closed_eye_icon
            IconButton(onClick = { isVisible = !isVisible }) {
                Icon(painter = painterResource(id = eye_con), contentDescription = null)
            }
        }
    )
}
```

El password se mostrará o dejará de mostrarse cuando clickemos el icono del ojo como se puede ver en la imagen. También se cambia el icono del ojo de abierto a cerrado.



## Botón de inicio de sesión

La función `EnableLogin` nos permite mantener el botón de login deshabilitado hasta que se cumplan sus condiciones, en este caso que el email contenga un '@' y que la longitud de la contraseña sea mayor a 6 caracteres.

```
fun EnableLogin(email: String, password: String): Boolean {  
    return email.contains("@") && password.length > 6  
}
```

Como podemos ver, con 6 está deshabilitado el botón, y con 7 se activa.

