



instituto
superior de
engenharia
de lisboa



POLITÉCNICO
DE LISBOA

POLYTECHNIC
UNIVERSITY OF LISBON

Compose - Estado

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt

PRO
DIGI

Modelo simplificado



- UI em função dos dados
- Recomposição automática quando os dados mudam

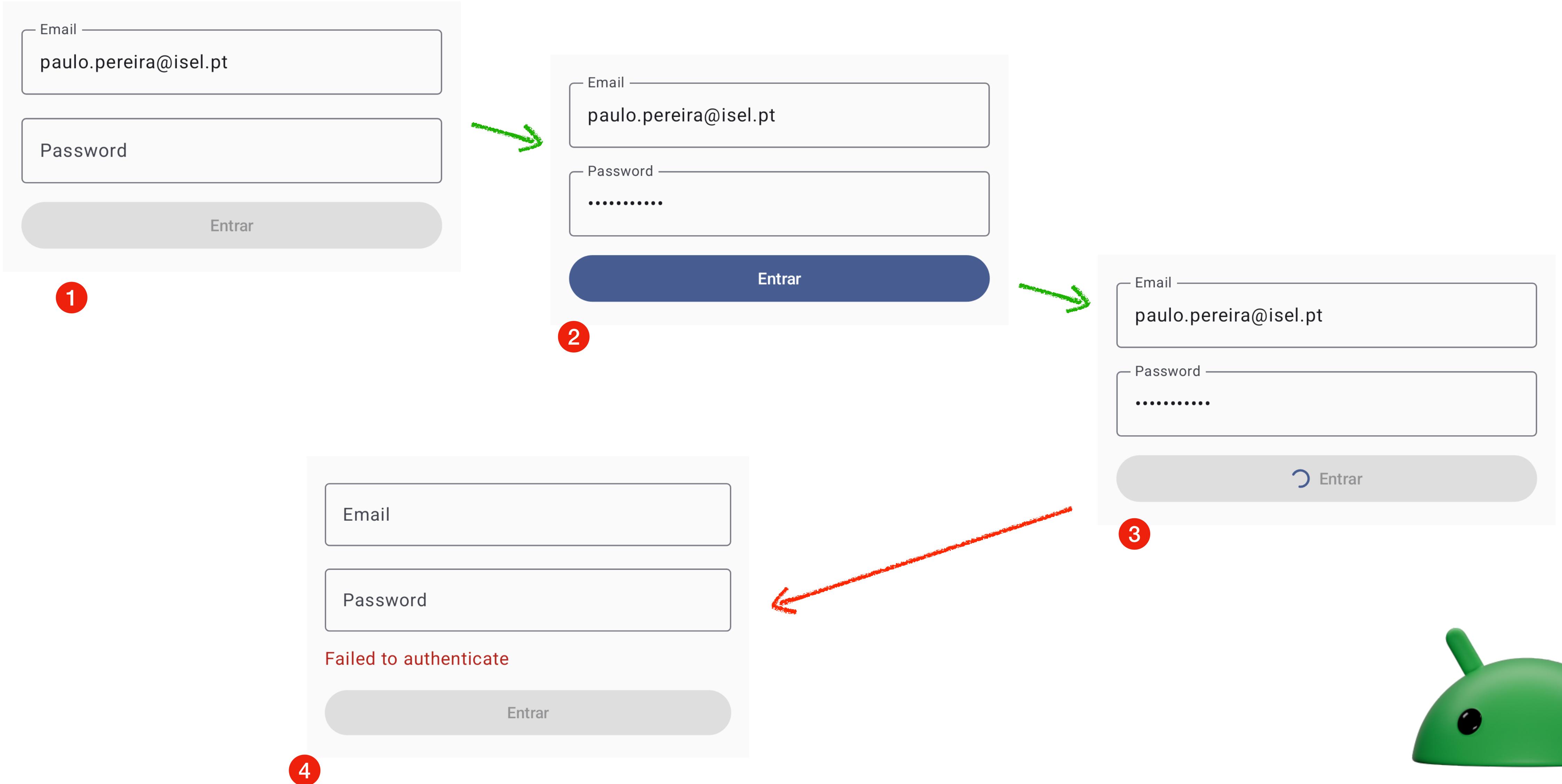


Gestão de Estado em Compose

- É acerca de decidir onde colocamos os dados (qual a autoridade)
 - A que se referem?
 - Qual a sua origem?
 - O que pretendemos fazer-lhes?
- Resumidamente:
 - Estado de apresentação 
 - Estado de aplicação 



Exemplo [1]



Exemplo [2]

Email
paulo.pereira@isel.pt

Password

Entrar

```
graph LR; Email[Email] --- PF1[Password]; PF1 --- Entrar1[Entrar]; Entrar1 --- Column1[Column]; Column1 --- TF1[Text Field]; Column1 --- TF2[Text Field]; Column1 --- Button1[Button]; Button1 --- Text1[Text];
```

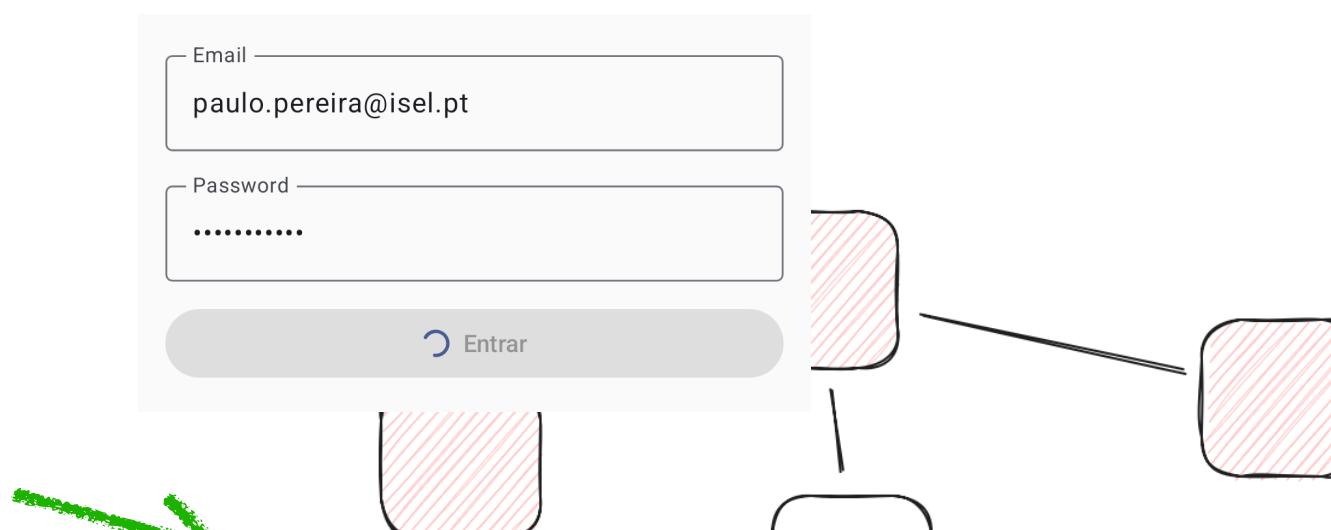
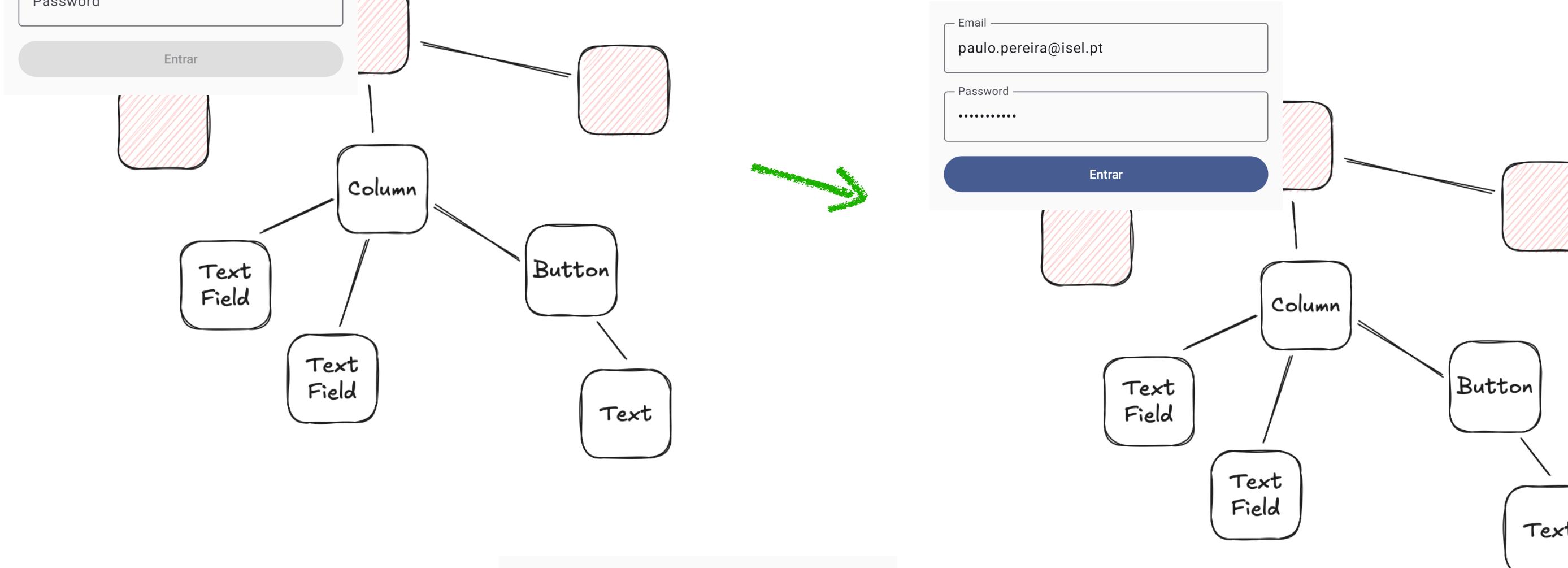
Email

Password

Failed to authenticate

Entrar

```
graph LR; Email[Email] --- PF1[Password]; PF1 --- Entrar1[Entrar]; Entrar1 --- Column1[Column]; Column1 --- TF1[Text Field]; Column1 --- TF2[Text Field]; Column1 --- Button1[Button]; Button1 --- Text1[Text];
```



2



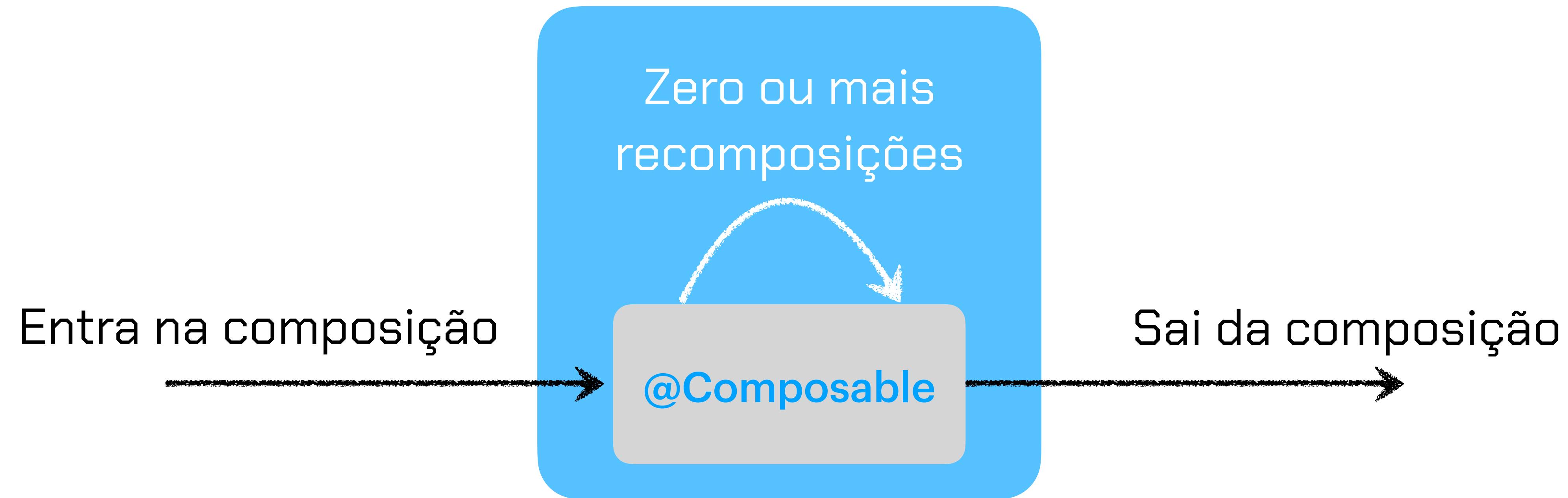
1

Exemplo [3]

```
@Composable
fun LoginFormStateless(
    loading: Boolean,
    error: String?,
    email: String,
    password: String,
    isDataValid: Boolean,
    onEmailChange: (String) -> Unit,
    onPasswordChange: (String) -> Unit,
    onLogin: (email: String, password: String) -> Unit
) {
    Column(...) {
        OutlinedTextField(
            value = email,
            onValueChange = onEmailChange,
        )
        OutlinedTextField(
            value = password,
            onValueChange = onPasswordChange,
        )
        if (!error.isNullOrEmpty()) {
            1 Text(error, color = MaterialTheme.colorScheme.error)
        }
        Button(
            onClick = { onLogin(email, password) },
            enabled = isDataValid && !loading,
        ) {
            1 if (loading) {
                CircularProgressIndicator(Modifier.size(16.dp), strokeWidth = 2.dp)
            }
            Text("Entrar")
        }
    }
}
```



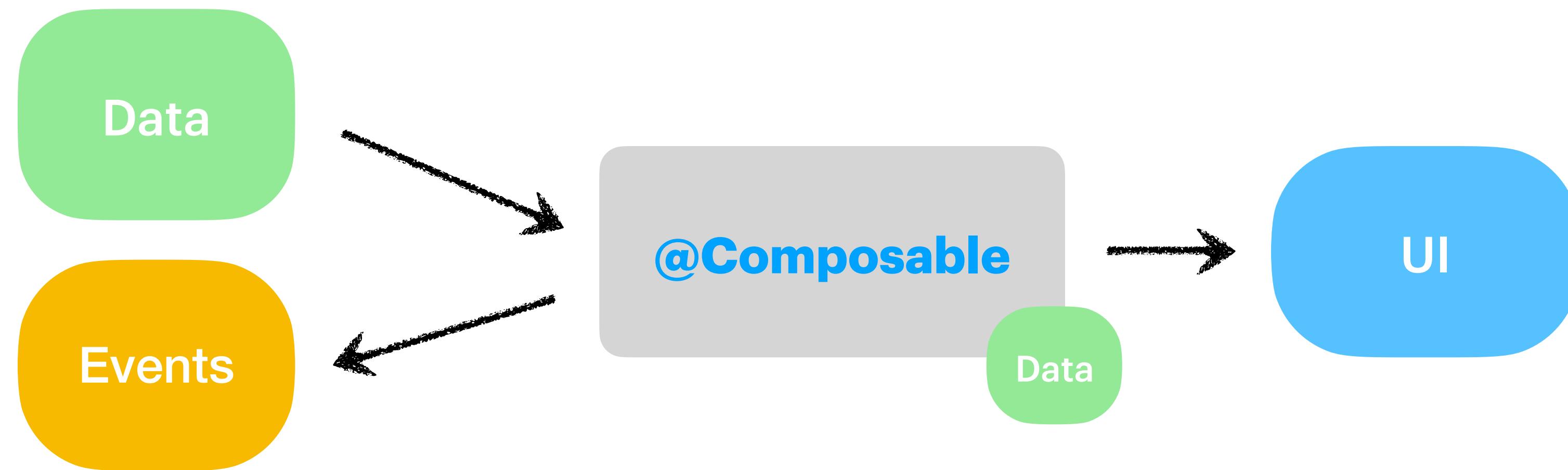
Ciclo de vida



- As recomposições ocorrem quando os dados mudam
- Desencadeadas por alterações em instâncias de State<T>



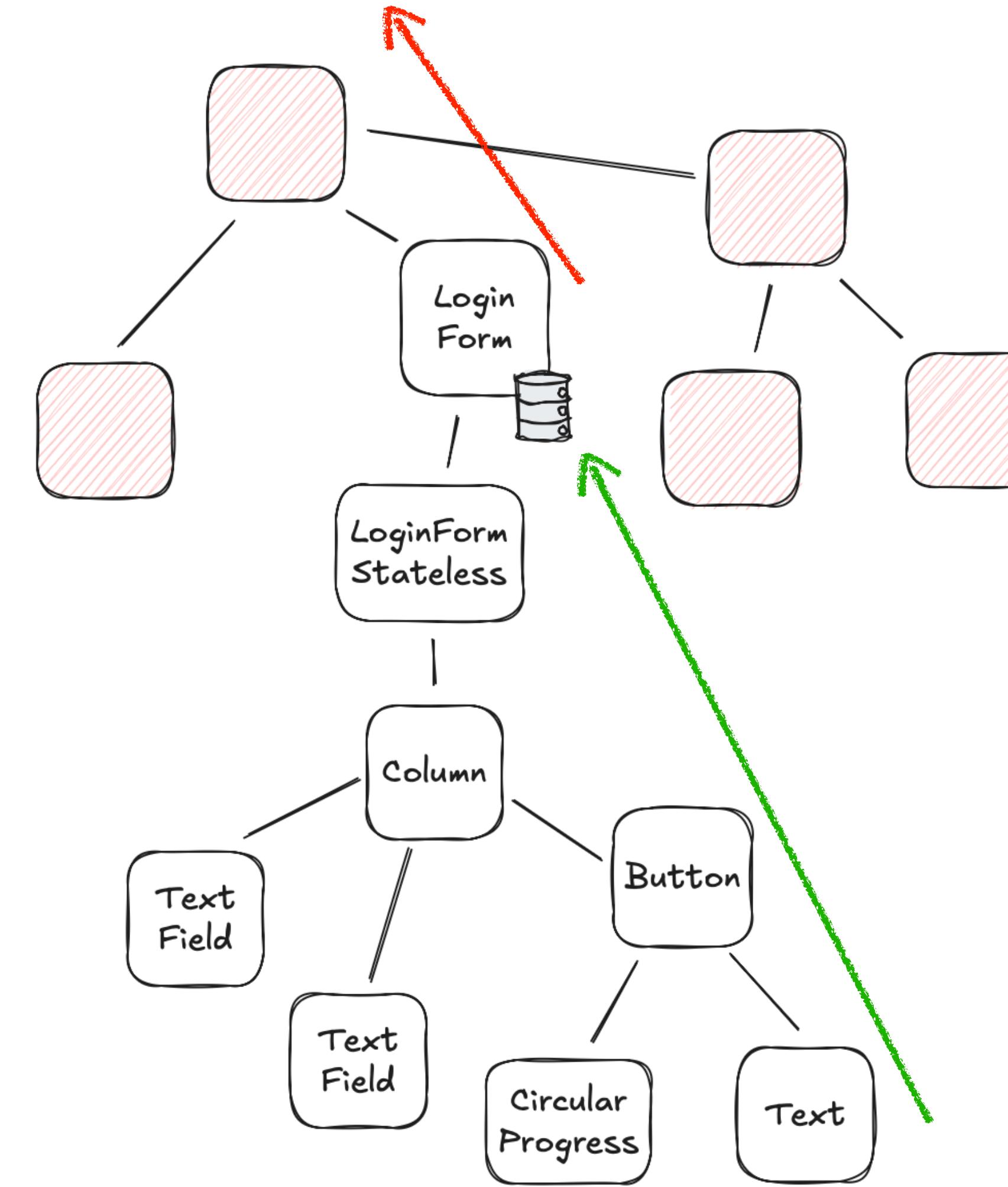
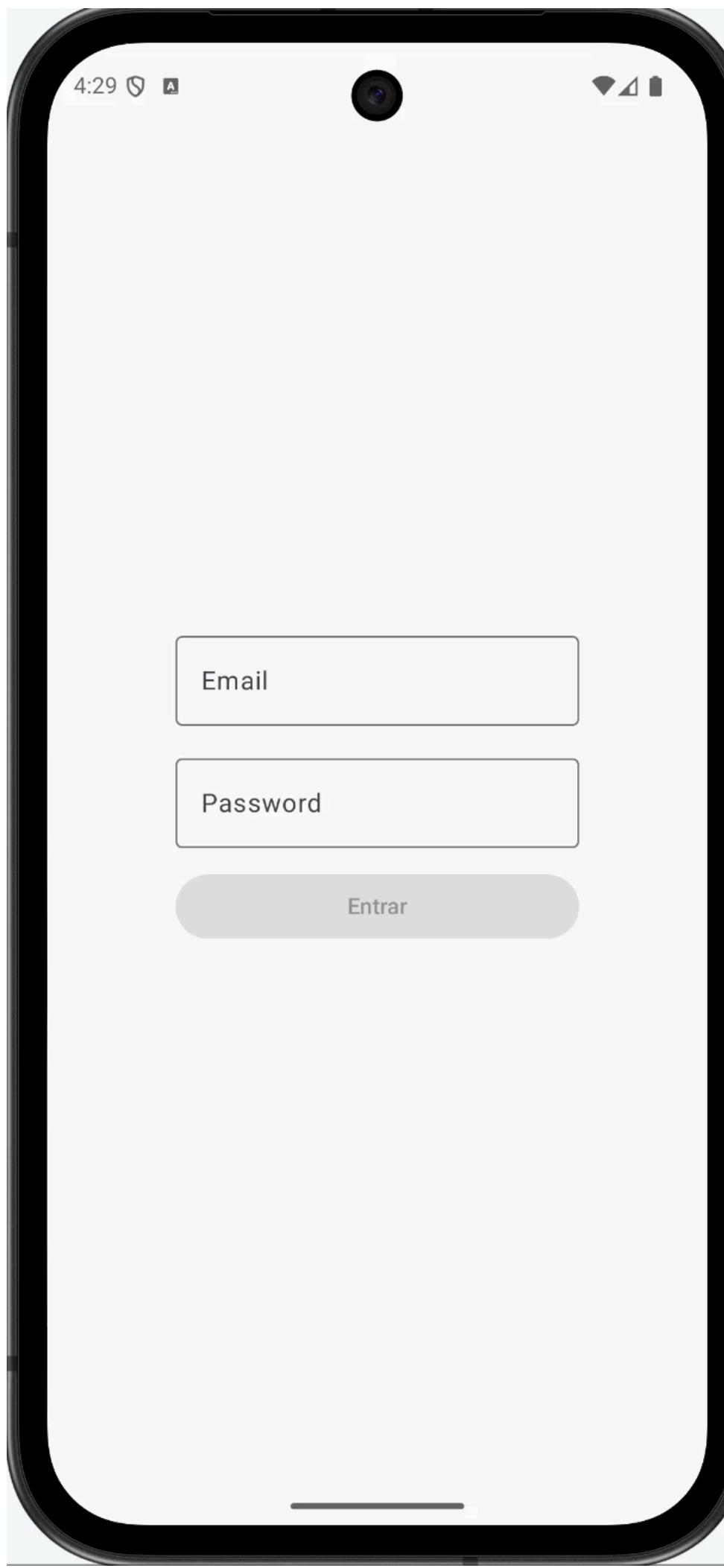
Modelo, revisitado



- UI em função dos dados (externos e internos)
- Recomposição automática quando os dados mudam
- Classificam-se como: **Stateless** e **Stateful**

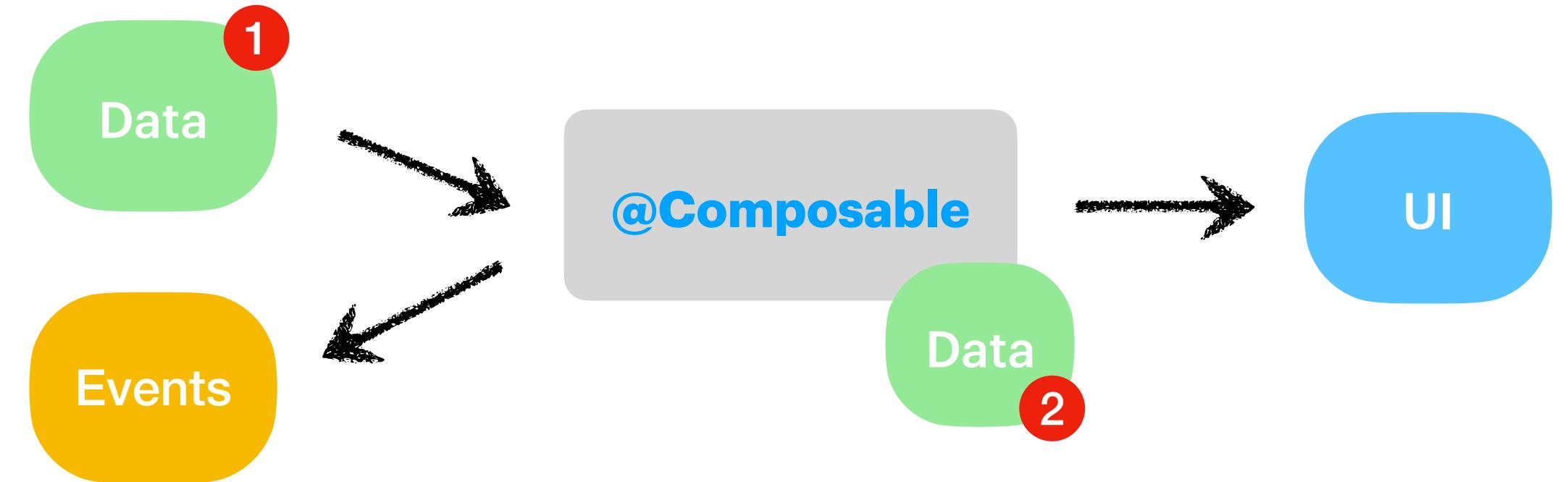


Exemplo [4]



Exemplo [5]

```
@Composable
fun LoginForm(
    ① loading: Boolean,
    error: String?,
    onLogin: (email: String, password: String) -> Unit,
    modifier: Modifier = Modifier
) {
    ② var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    val isDataValid = email.isValidEmail() && password.isNotBlank()
    LoginFormStateless(
        loading = loading,
        error = error,
        email = email,
        password = password,
        isDataValid = isDataValid,
        onEmailChange = { email = it },
        onPasswordChange = { password = it },
        onLogin = onLogin,
        modifier
    )
}
```

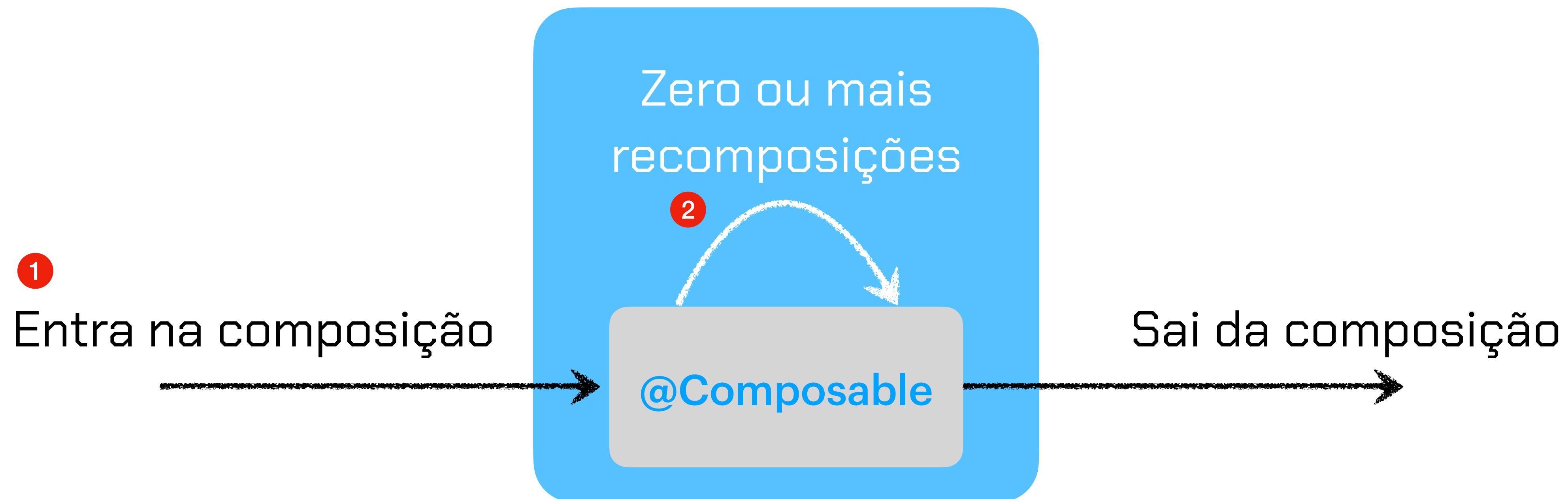


Estado de Apresentação

- Suportado por duas primitivas:
 - `mutableStateOf` (para intercepção das mutações)
 - `remember` (*para memoization*)
- Por conveniência, existem outras variantes construídas à custa destas



Ciclo de vida [2]



```
var email by remember { mutableStateOf("") }  
    2           1
```

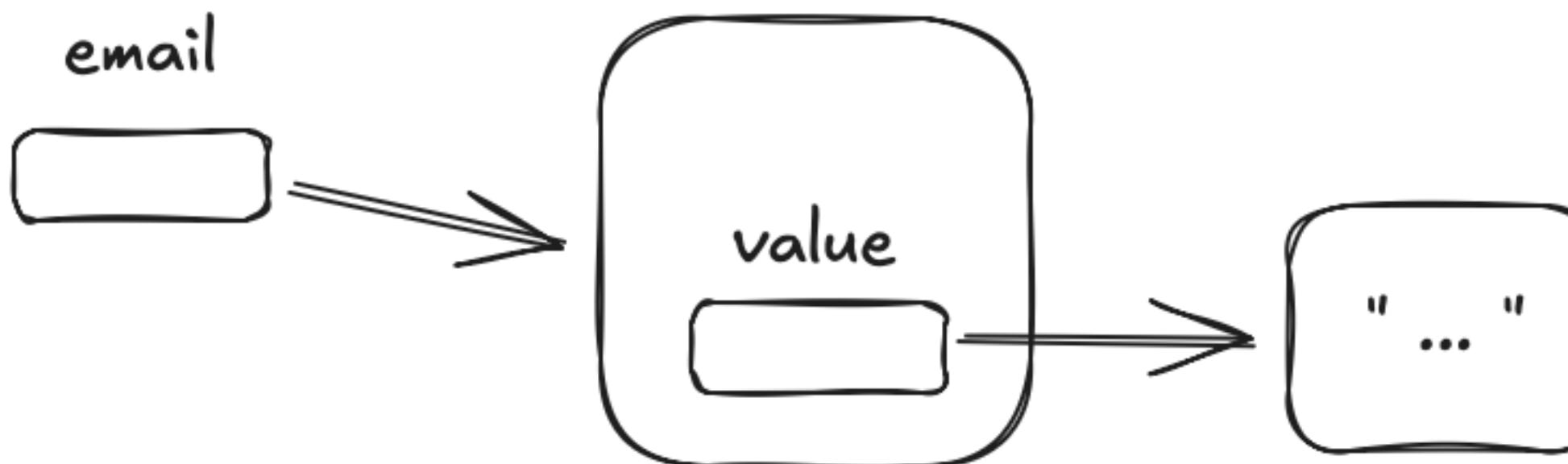
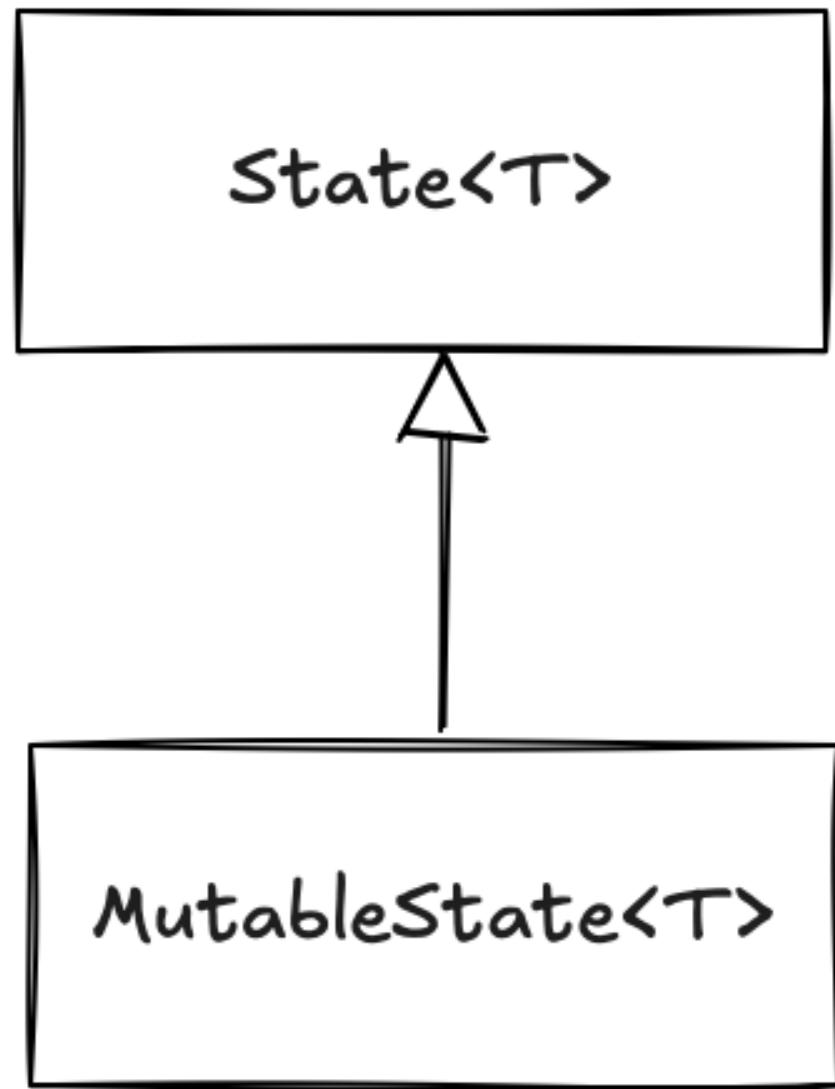


MutableStateOf<T>

```
var email = remember { mutableStateOf("") }
```

2

1



```
var email by remember { mutableStateOf("") }
```

2



Documentação oficial

- Ciclo de vida de Composables
 - <https://developer.android.com/develop/ui/compose/lifecycle>
- Estado de apresentação em Composables
 - <https://developer.android.com/develop/ui/compose/state>
 - <https://developer.android.com/develop/ui/compose/state-hoisting>





instituto
superior de
engenharia
de lisboa



POLITÉCNICO
DE LISBOA

POLYTECHNIC
UNIVERSITY OF LISBON

Compose - Estado

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt

PRO
DIGI