instituto
superior de
engenharia
de lisboa

POLITÉCNICO
DE LISBOA
POLYTECHNIC
UNIVERSITY OF LISBON

Compose - Introdução

PDM - Programação para Dispositivos Móveis

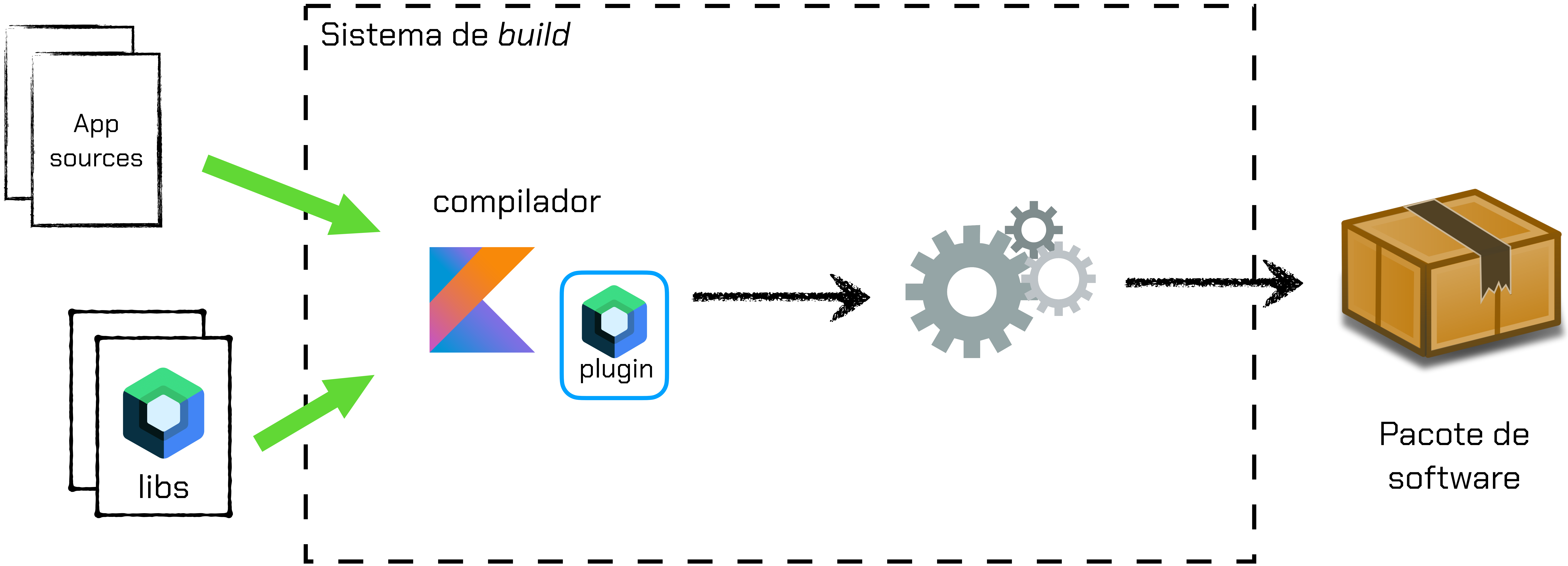
Paulo Pereira
paulo.pereira@isel.pt

PRO
DIGI

Compose - o que é?

- Framework para criação de interfaces gráficas
- Composta por
    - Extensão ao compilador de Kotlin
    - Bibliotecas
    - Runtime

Build

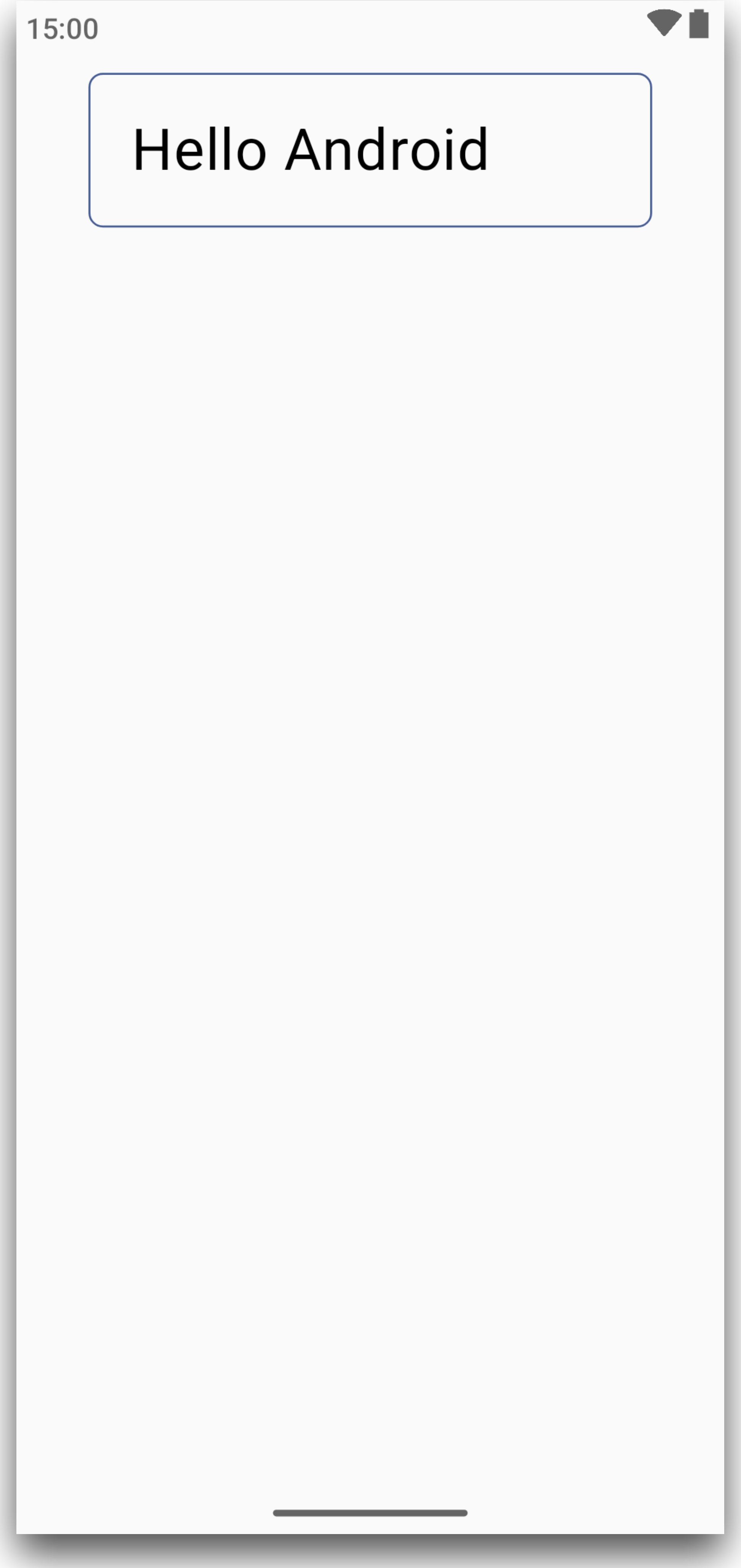

App sources
libs
Sistema de *build*
compilador
plugin
Pacote de software

Modelo de programação

- Definição de componentes usando uma DSL (Domain Specific Language)
  - abordagem declarativa
  - composição (composables)
  - valorização da imutabilidade
- Em claro contraste com a abordagem Object Oriented clássica

Exemplo

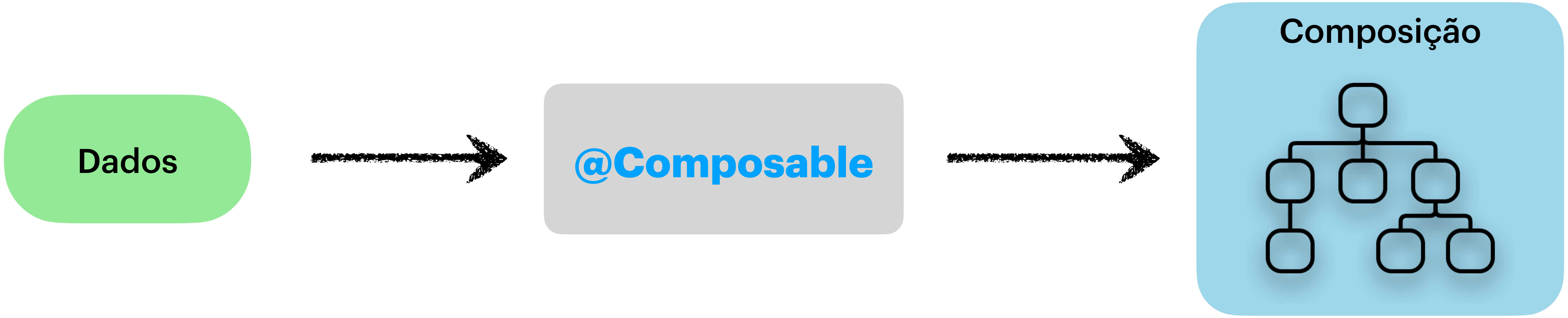
```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Box(
        modifier = modifier
            .fillMaxWidth()
            .padding(all = 8.dp)
            .border(
                width = 1.dp,
                color = MaterialTheme.colorScheme.primary,
                shape = MaterialTheme.shapes.small
            )
    ) {
        Text(
            text = "Hello $name",
            fontSize = 32.sp,
            modifier = Modifier.padding(24.dp)
        )
    }
}
```



15:00

Hello Android

Modelo simplificado



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

@Composables

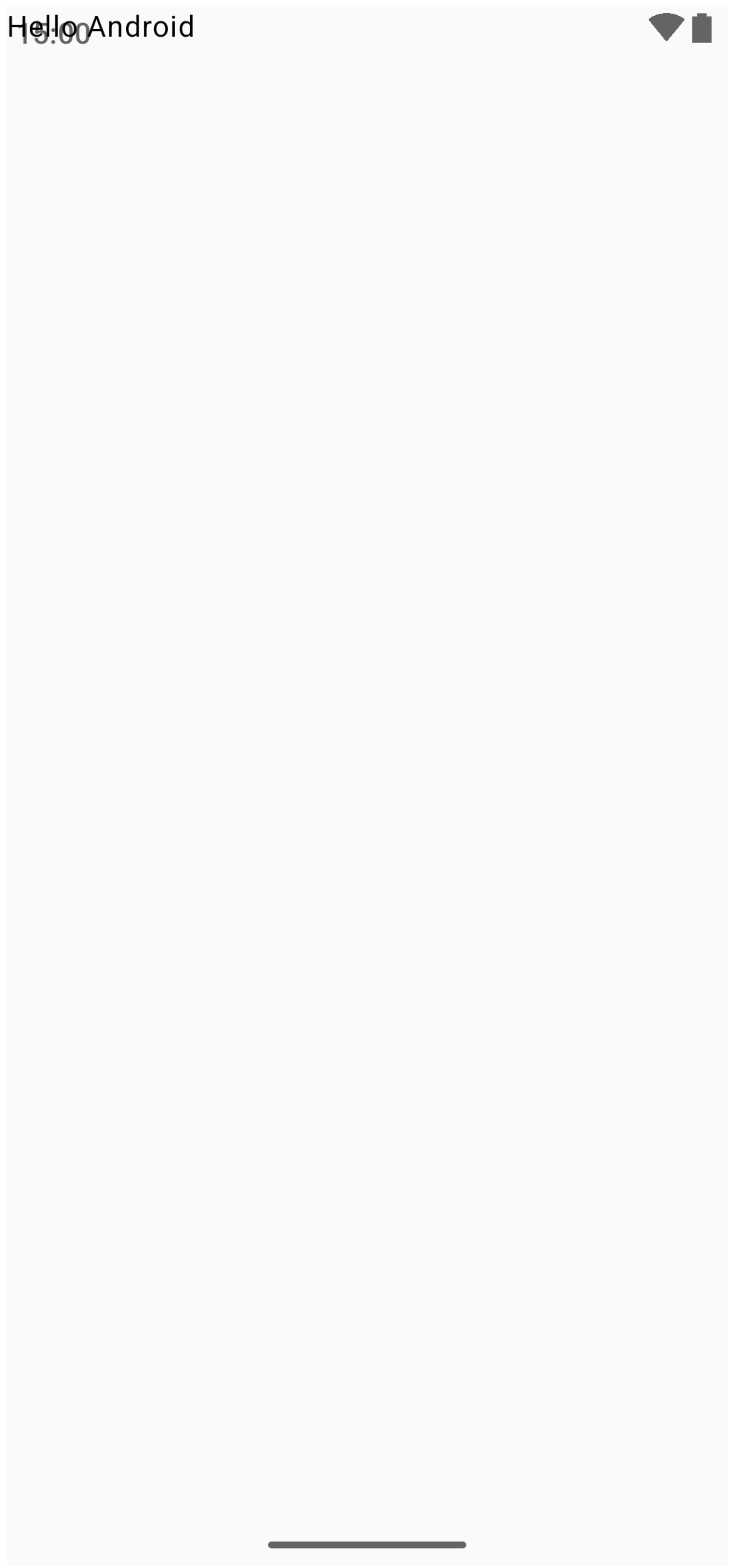
- Descrevem a UI em termos de
    - Estrutura
    - Aspecto
    - Zonas sensíveis a inputs do utilizador

https://developer.android.com/develop/ui/compose/lifecycle#composition-anatomy

@Composable elementar (1)

```
@Composable
fun Greeting(name: String) {
    Text(text = "Hello $name")
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Android")
    }
}
```



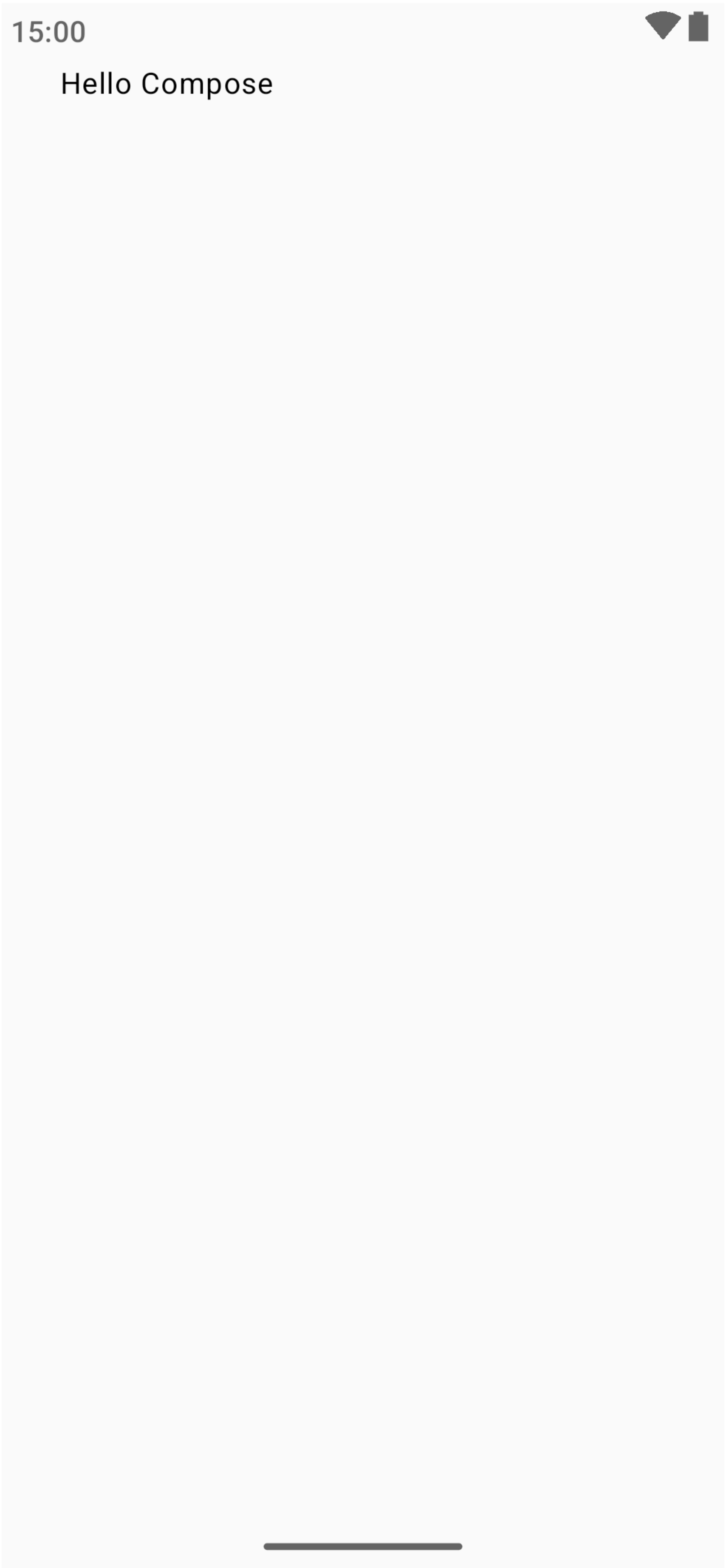
Hello Android

15:00

@Composable elementar (2)

```
@Composable
fun Greeting(name: String) {
    Text(
        text = "Hello $name",
        modifier = Modifier.padding(all = 32.dp),
    )
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Compose")
    }
}
```

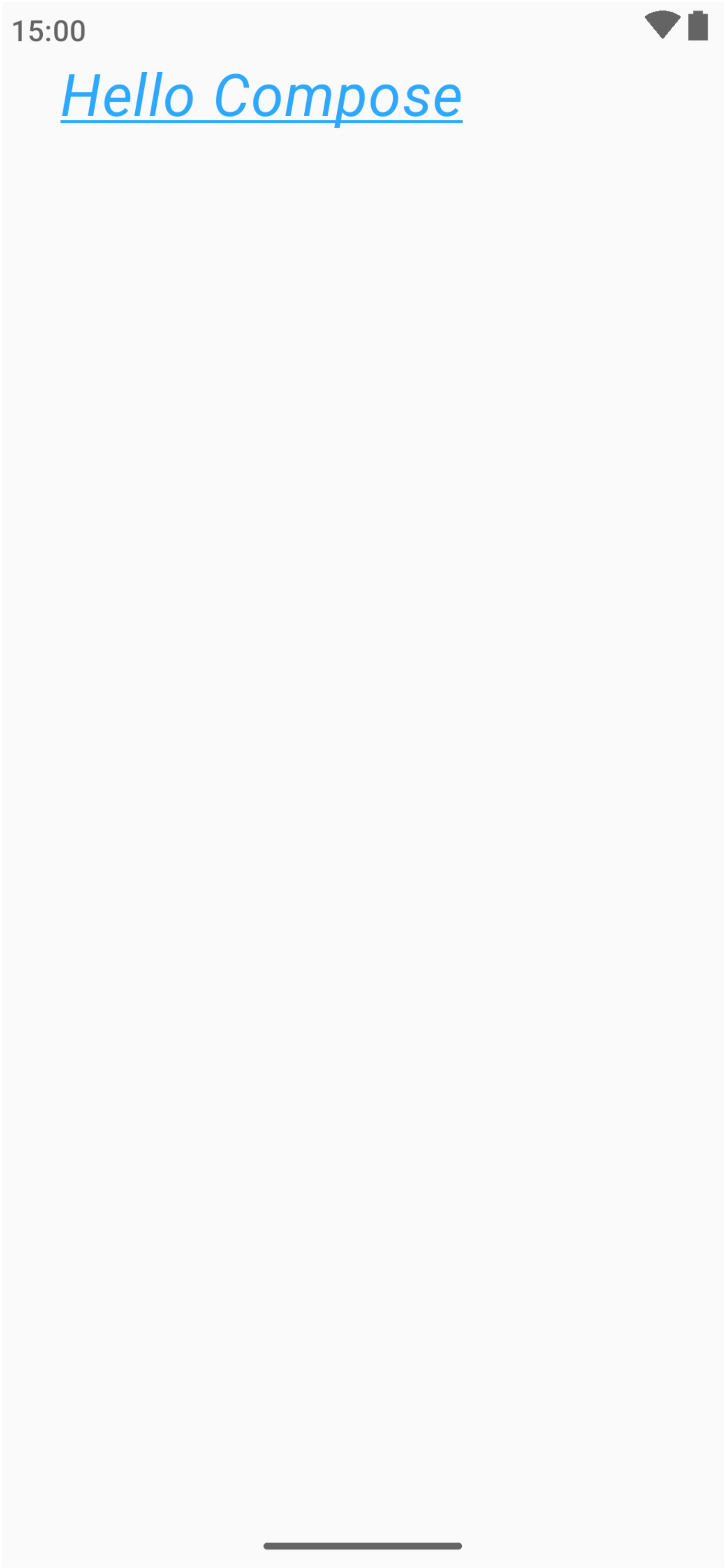


Hello Compose

@Composable elementar (3)

```
@Composable
fun Greeting(name: String) {
    Text(
        text = "Hello $name",
        fontSize = 32.sp,
        fontStyle = FontStyle.Italic,
        textDecoration = TextDecoration.Underline,
        color = Color(0xFF2DA8FC),
        modifier = Modifier.padding(all = 32.dp),
    )
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Compose")
    }
}
```



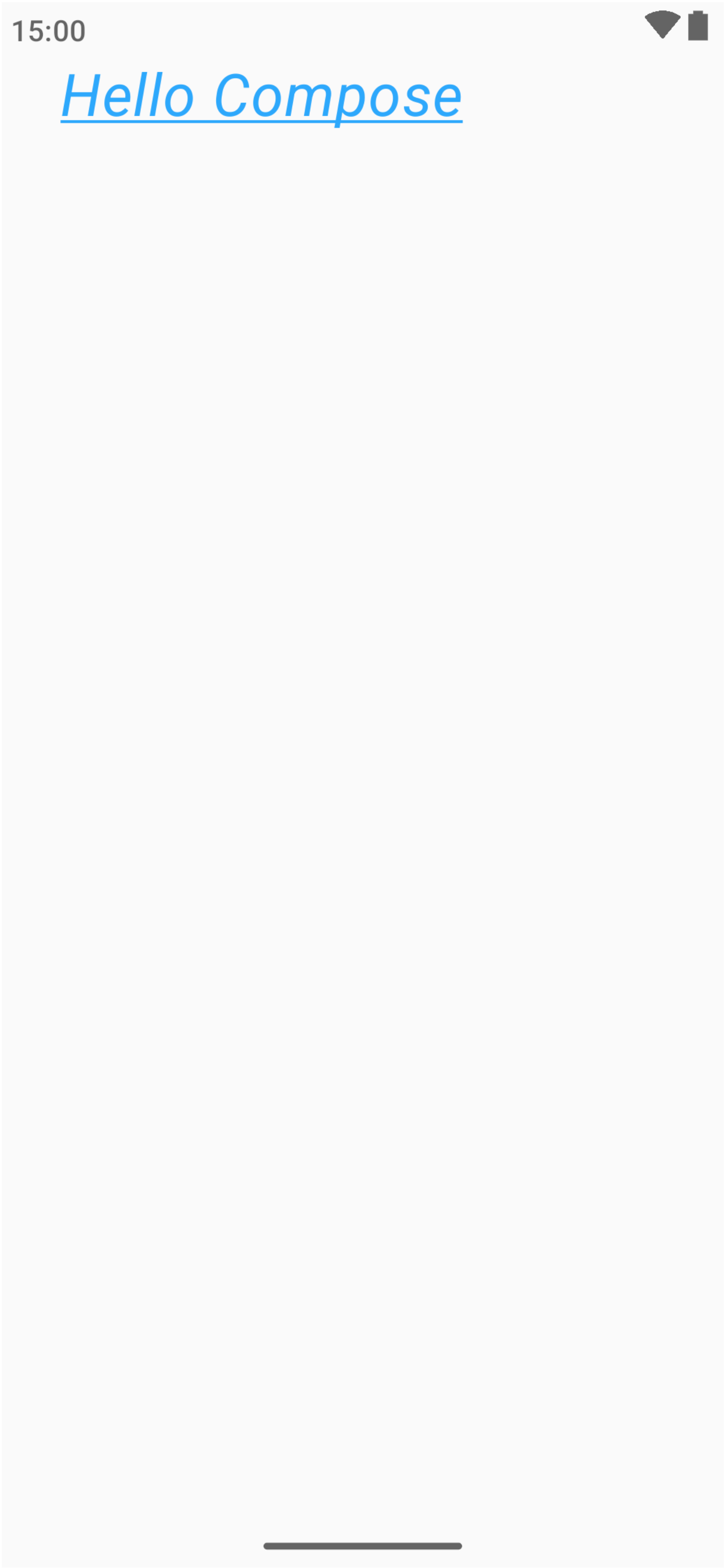
15:00

Hello Compose

@Composable elementar (4)

```
@Composable
fun Greeting(name: String) {
    Text(
        text = "Hello $name",
        style = TextStyle(
            fontSize = 32.sp,
            fontStyle = FontStyle.Italic,
            textDecoration = TextDecoration.Underline,
            color = Color(0xFF2DA8FC),
        ),
        modifier = Modifier.padding(all = 32.dp),
    )
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Compose")
    }
}
```

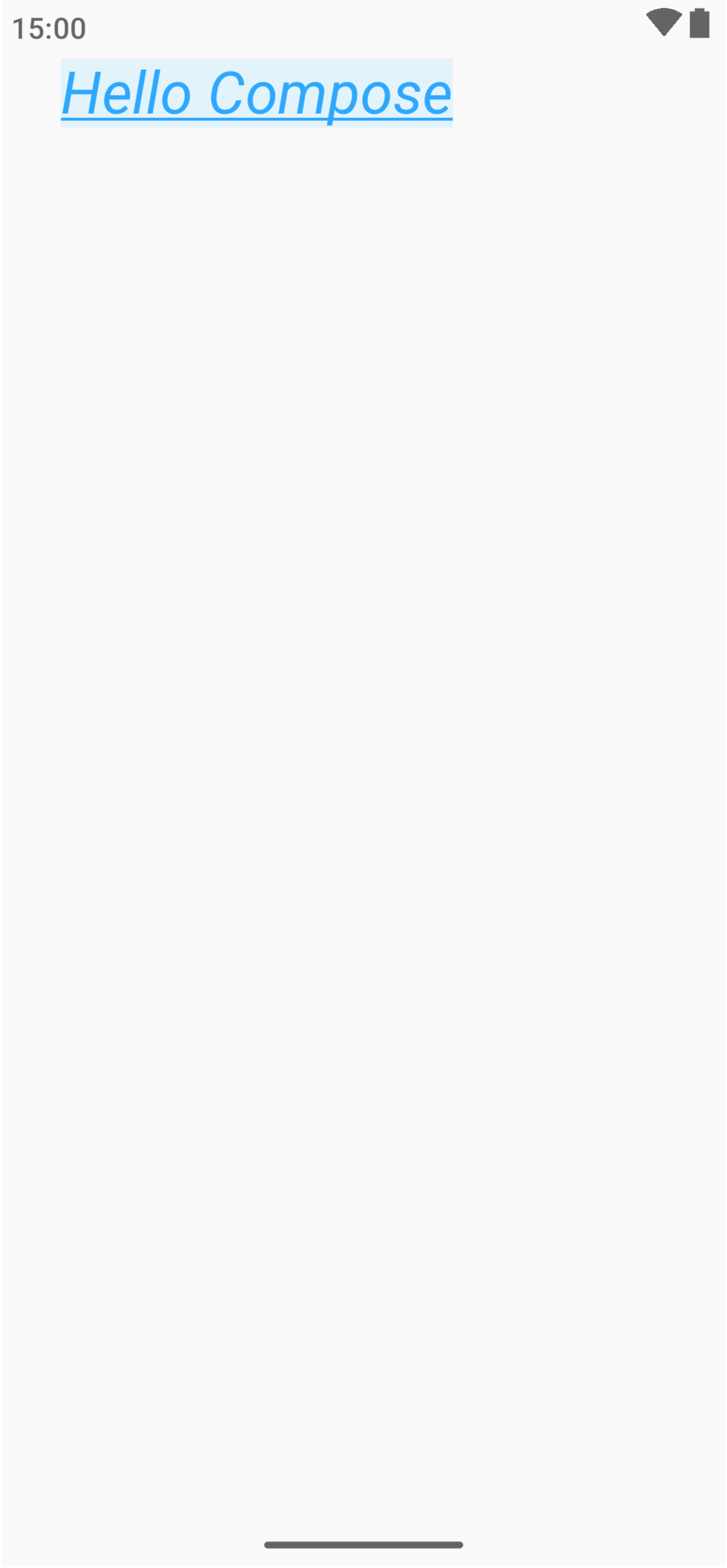


Hello Compose

@Composable elementar (4)

```
@Composable
fun Greeting(name: String) {
    Text(
        text = "Hello $name",
        style = TextStyle(
            fontSize = 32.sp,
            fontStyle = FontStyle.Italic,
            textDecoration = TextDecoration.Underline,
            color = Color(0xFF2DA8FC),
        ),
        modifier = Modifier
            .padding(all = 32.dp)
            .background(color = Color(0x88CFEDFE))
        ),
    )
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Compose")
    }
}
```



15:00

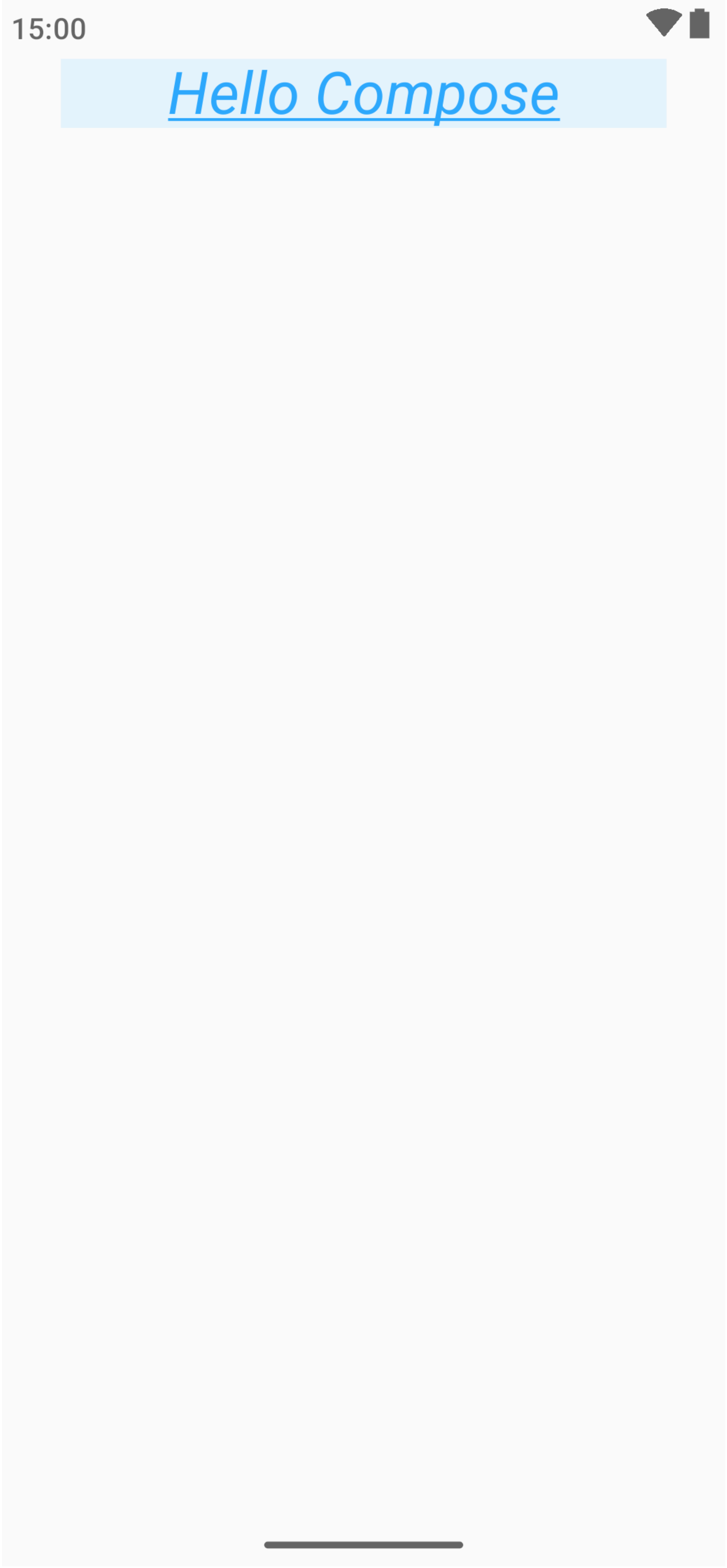
Hello Compose

@Composable elementar (5)

```
@Composable
fun Greeting(name: String) {
    Text(
        text = "Hello $name",
        style = TextStyle(
            fontSize = 32.sp,
            fontStyle = FontStyle.Italic,
            textDecoration = TextDecoration.Underline,
            color = Color(0xFF2DA8FC),
            textAlign = TextAlign.Center
        ),
        modifier = Modifier
            .padding(all = 32.dp)
            .background(color = Color(0x88CFEDFE))
            .fillMaxWidth()
    )
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Compose")
    }
}
```

1

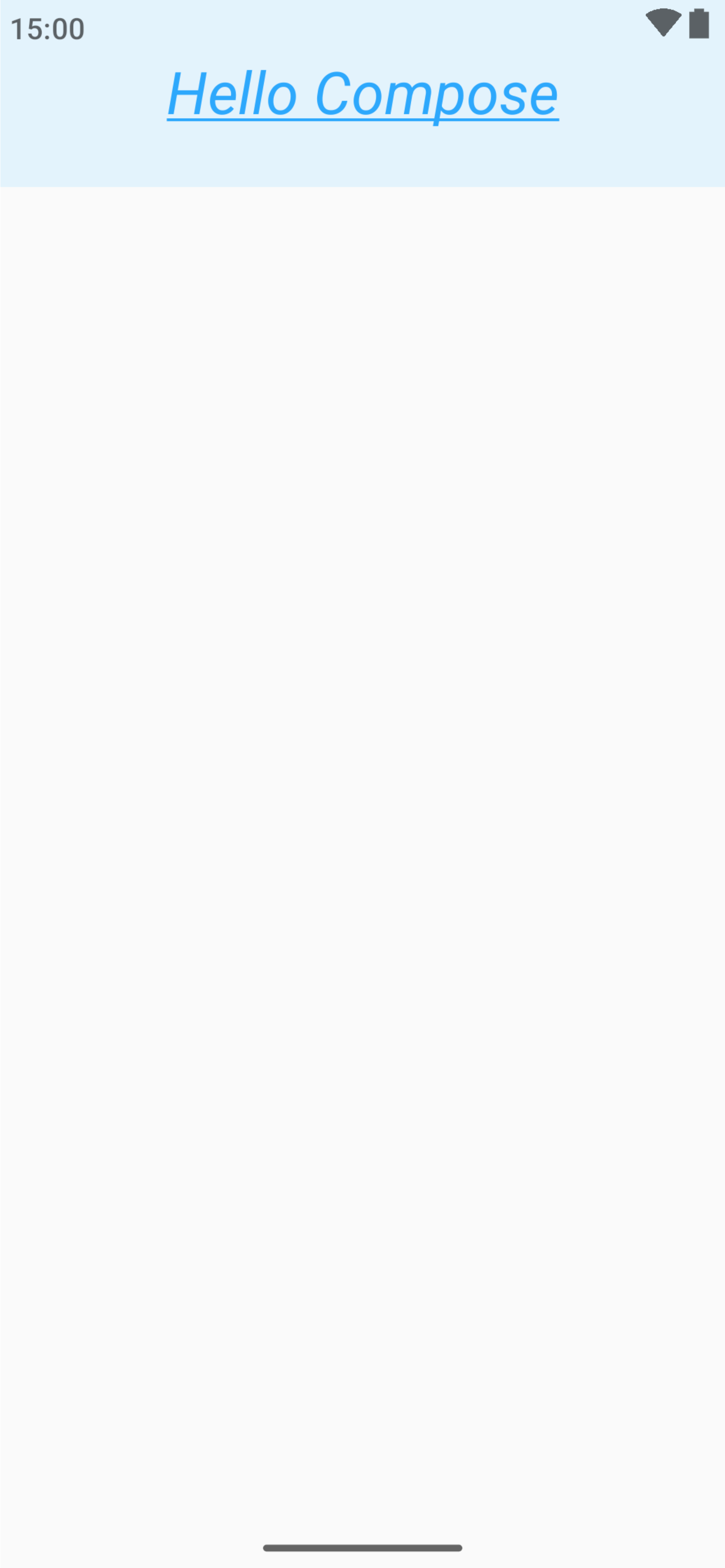

15:00

Hello Compose

@Composable elementar (5)

```kotlin
@Composable
fun Greeting(name: String) {
    Text(
        text = "Hello $name",
        style = TextStyle(
            fontSize = 32.sp,
            fontStyle = FontStyle.Italic,
            textDecoration = TextDecoration.Underline,
            color = Color(0xFF2DA8FC),
            textAlign = TextAlign.Center
        ),
        modifier = Modifier
      1   .background(color = Color(0x88CFEDFE))
            .padding(all = 32.dp)
            .fillMaxWidth()
    )
}

@Preview(showSystemUi = true, showBackground = true)
@Composable
fun GreetingPreview() {
    ComposeExamplesTheme {
        Greeting(name = "Compose")
    }
}
```

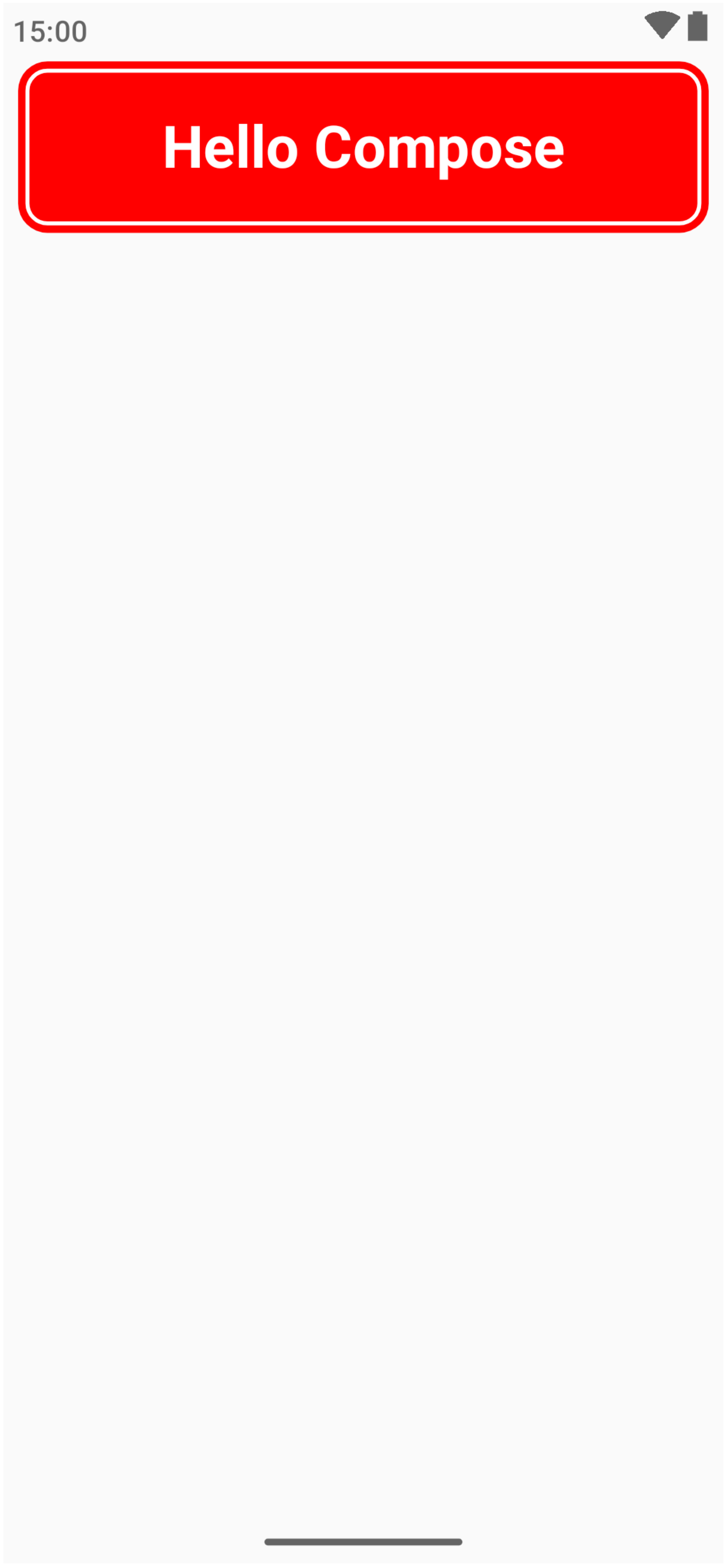


15:00

Hello Compose

@Composable elementar (6)

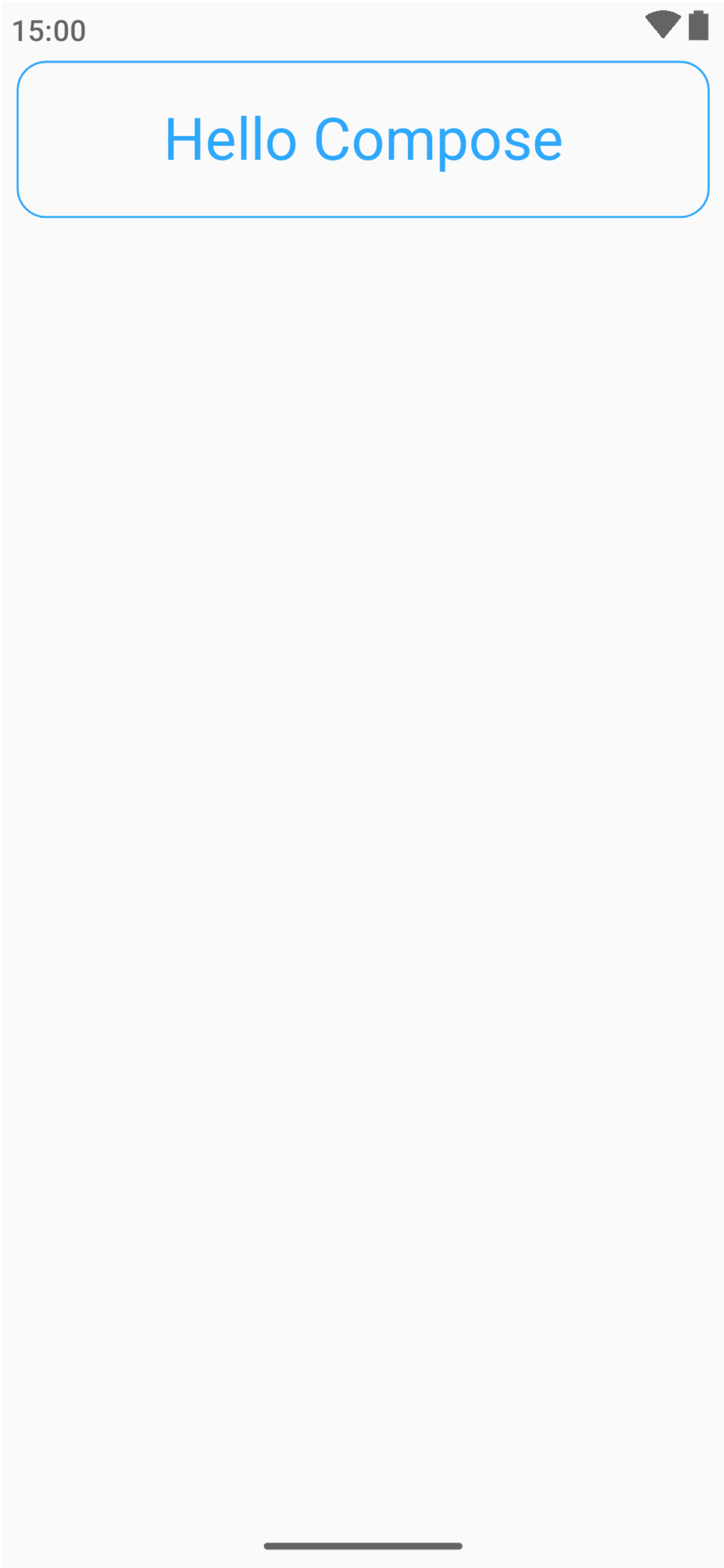
```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name",
        style = TextStyle(
            fontSize = 32.sp,
            color = Color.White,
            fontWeight = FontWeight.Bold,
            textAlign = TextAlign.Center
        ),
        modifier = modifier
            .padding(all = 8.dp)
            .background(
                brush = SolidColor(value = Color.Red),
                shape = RoundedCornerShape(size = 16.dp)
            )
            .padding(all = 4.dp)
            .border(
                width = 2.dp,
                brush = SolidColor(value = Color.White),
                shape = RoundedCornerShape(size = 12.dp)
            )
            .padding(all = 24.dp)
            .fillMaxWidth()
    )
}
```


15:00

Hello Compose

@Composable elementar (7)

```
@Composable
fun Greeting(
    name: String,
    onClick: () -> Unit,
    modifier: Modifier = Modifier
) {
    val foregroundColor = Color(0xFF2DA8FC)
    Text(
        text = "Hello $name",
        style = TextStyle(
            fontSize = 32.sp,
            color = foregroundColor,
            textAlign = TextAlign.Center
        ),
        modifier = modifier
            .padding(all = 8.dp)
            .border(
                width = 1.dp,
                brush = SolidColor(foregroundColor),
                shape = RoundedCornerShape(size = 16.dp)
            )
            .padding(all = 24.dp)
            .fillMaxWidth()
            .clickable(onClick = onClick)
    )
}
```



Catálogo

- Catálogo da biblioteca standard (i.e. Material Design 3)
  - https://developer.android.com/develop/ui/compose/components

- Outros catálogos online:
  - https://composables.com/docs/androidx.compose.material3/material3
  - https://foso.github.io/Jetpack-Compose-Playground/
  - https://foso.github.io/Jetpack-Compose-Playground/compose_projects/#libraries

instituto superior de engenharia de lisboa

POLITÉCNICO DE LISBOA
POLYTECHNIC UNIVERSITY OF LISBON

Compose - Introdução

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt

PRO
DIGI