instituto
superior de
engenharia
de lisboa

POLITÉCNICO
DE LISBOA
POLYTECHNIC
UNIVERSITY OF LISBON

Compose - Layouts

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt

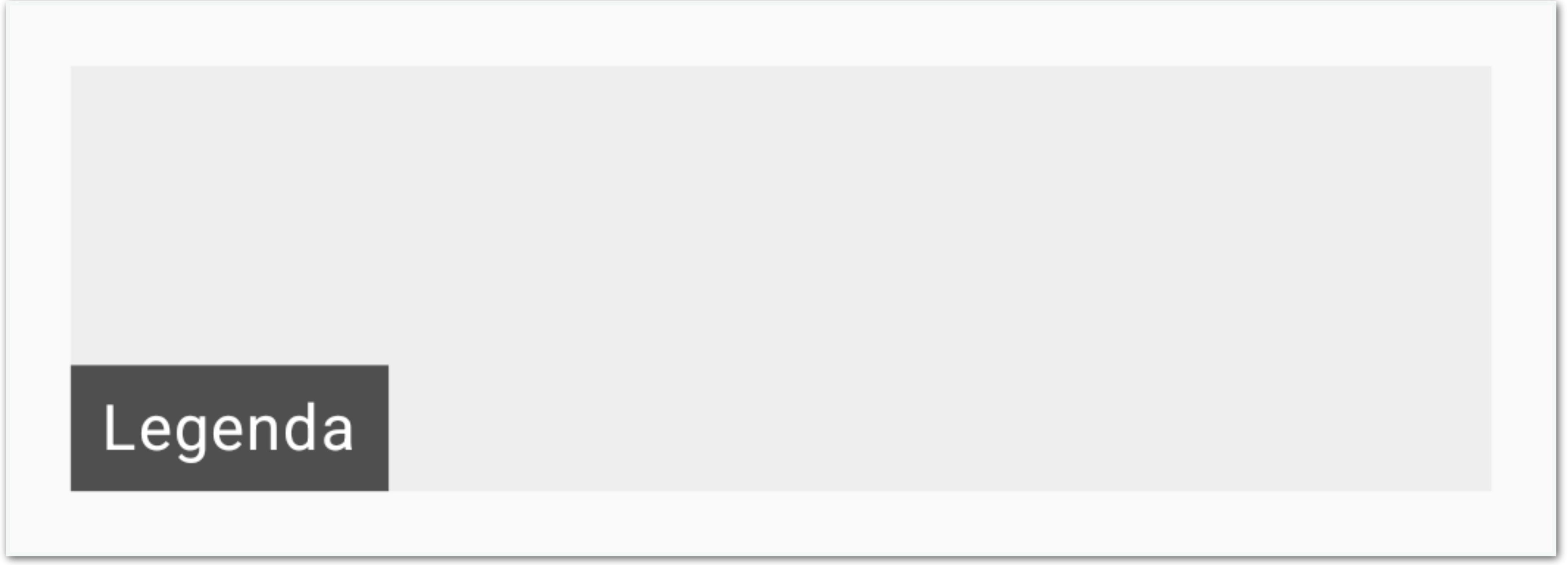
PRO
DIGI

Layouts - o que são?

- Composables para **disposição de elementos** no ecrã
  - Medir —> medir filhos (com restrições) —> colocação (filhos primeiro)
- Layouts essenciais:
  - Box, Column, Row
  - LazyColumn, LazyRow e LazyVerticalGrid
  - Scaffold, Navigation Drawer, Bottom Sheet e TabRow

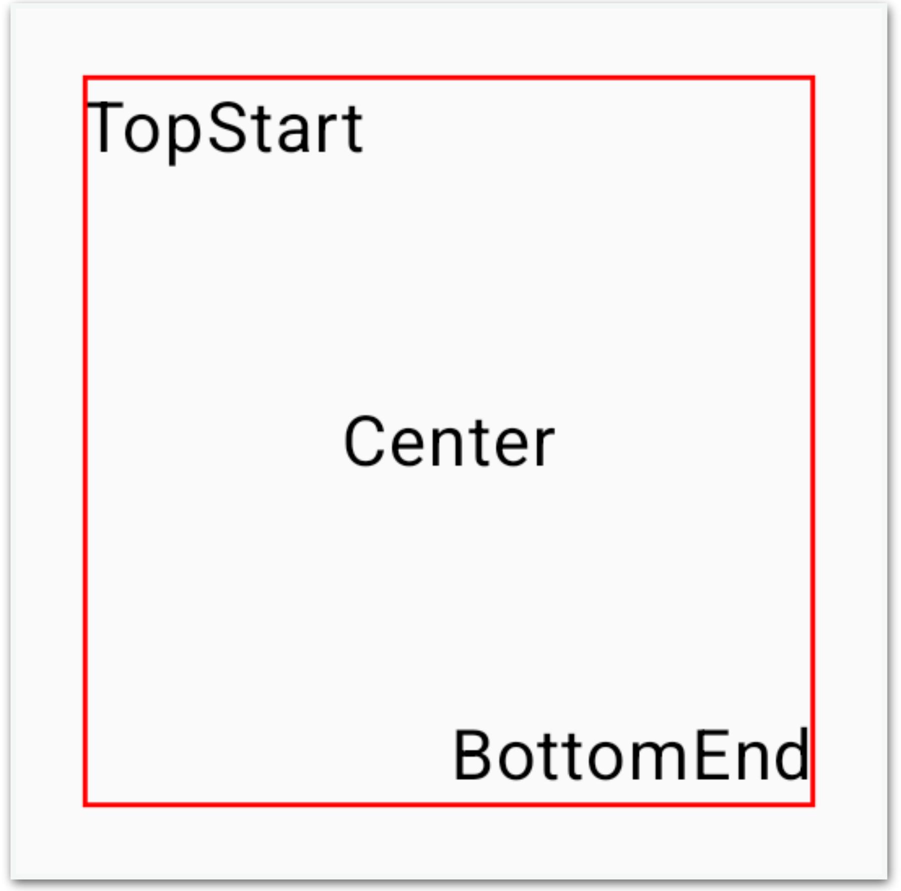
@Composable Box (1)

```
@Composable
fun BoxOverlayDemo() {
    Box(modifier = Modifier.fillMaxWidth().height(140.dp).padding(all = 16.dp)) {
      1 Box(Modifier.matchParentSize().background(Color(0xFFEEEEEE)))
        Text(
        2   text = "Legenda",
            modifier = Modifier
                .align(Alignment.BottomStart)
                .background(Color(0xAA000000))
                .padding(horizontal = 8.dp, vertical = 4.dp),
            color = Color.White
        )
    }
}
```



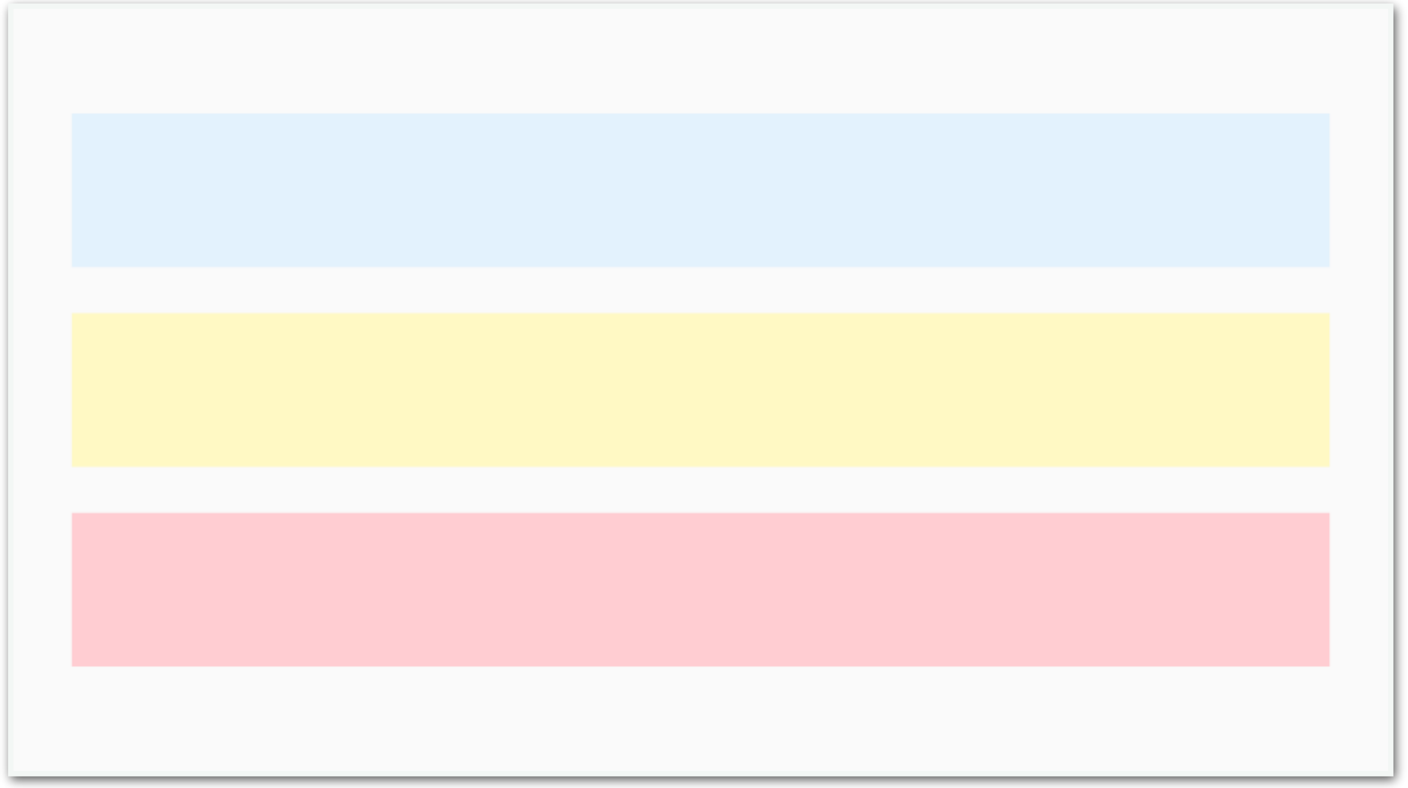
@Composable Box (2)

```
@Composable
fun BoxAlignDemo() {
    Box(
        modifier = Modifier
            .size(200.dp)
          1 .padding(all = 16.dp)
            .border(width = 1.dp, color = Color.Red)
    ) {
        Text(text = "TopStart", modifier = Modifier.align(Alignment.TopStart))
        Text(text = "Center", modifier = Modifier.align(Alignment.Center))
        Text(text = "BottomEnd", modifier = Modifier.align(Alignment.BottomEnd))
    }
}

@Preview(showBackground = true)
@Composable fun BoxAlignDemoPreview() {
    ComposeExamplesTheme { BoxAlignDemo() }
}
```



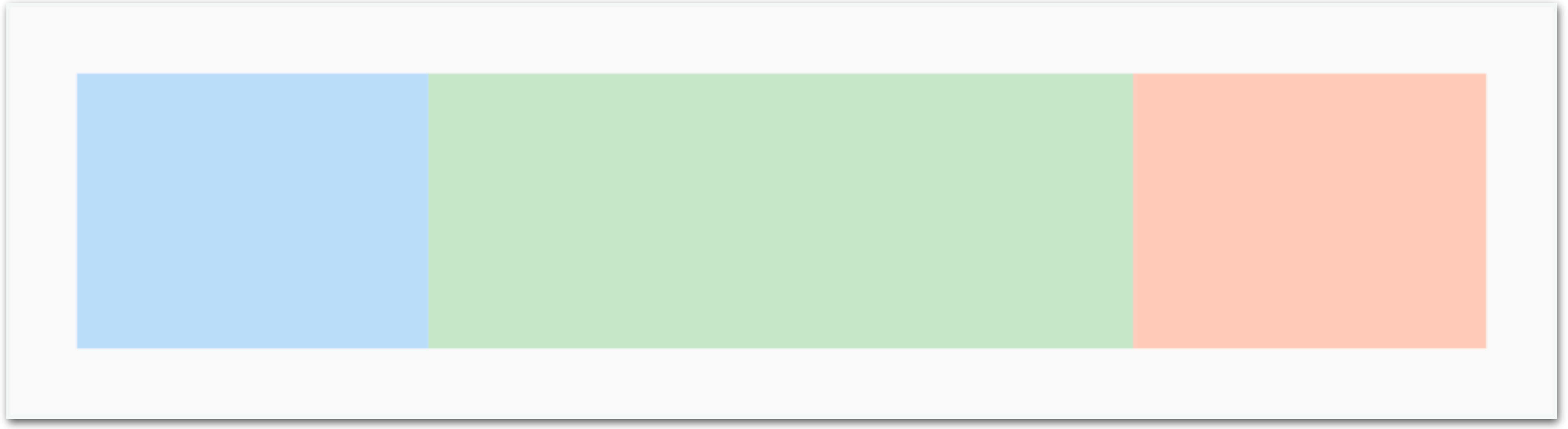
@Composable Column

```kotlin
@Composable
fun ColumnDemo() {
    Column(
        verticalArrangement = Arrangement.SpaceEvenly,
        modifier = Modifier.fillMaxWidth().padding(16.dp)
    ) {
        Box(Modifier.fillMaxWidth().height(40.dp).background(Color(0xFFE3F2FD)))
        Box(Modifier.fillMaxWidth().height(40.dp).background(Color(0xFFFFF9C4)))
        Box(Modifier.fillMaxWidth().height(40.dp).background(Color(0xFFFFCDD2)))
    }
}

@Preview(showBackground = true, widthDp = 360, heightDp = 200)
@Composable fun ColumnDemoPreview() {
    ComposeExamplesTheme { ColumnDemo() }
}
```



@Composable Row

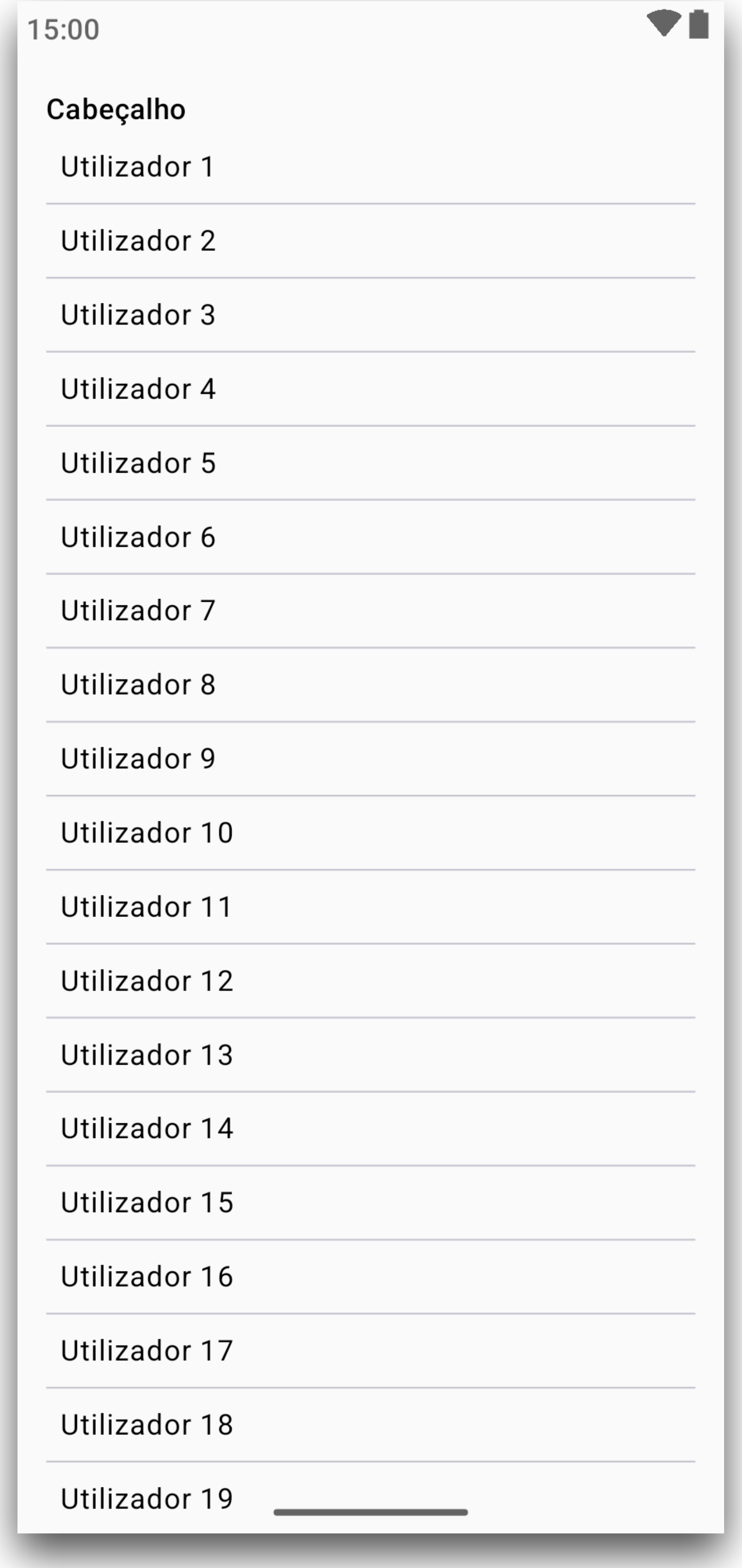
```
@Composable
fun RowWeightDemo() {
    Row(Modifier.fillMaxWidth().height(48.dp).padding(all = 16.dp)) {
        Box(Modifier.weight(1f).fillMaxHeight().background(Color(0xFFBBDEFB)))
        Box(Modifier.weight(2f).fillMaxHeight().background(Color(0xFFC8E6C9)))
        Box(Modifier.weight(1f).fillMaxHeight().background(Color(0xFFFFCCBC)))
    }
}

@Preview(showBackground = true, widthDp = 360, heightDp = 96)
@Composable fun RowWeightDemoPreview() {
    ComposeExamplesTheme { RowWeightDemo() }
}
```



Lazy Layouts

@Composable LazyColumn

```
@Composable
fun LazyColumnDemo(items: List<String>, modifier: Modifier = Modifier) {
    LazyColumn(
        modifier = modifier,
        contentPadding = PaddingValues(all = 16.dp)
    ) {
        item {
            Text(
                text = "Cabeçalho",
                style = MaterialTheme.typography.titleMedium
            )
        }
        items(items = items, key = { it }) { name ->
            Text(
                text = name,
                modifier = Modifier.padding(all = 8.dp)
            )
            HorizontalDivider()
        }
    }
}

@Preview(showBackground = true, showSystemUi = true)
@Composable fun LazyColumnDemoPreview() {
    ComposeExamplesTheme {
        val items = (1..50).map { "Utilizador $it" }
        LazyColumnDemo(items, modifier = Modifier.padding(top = 32.dp))
    }
}
```



15:00

Cabeçalho

Utilizador 1

Utilizador 2

Utilizador 3

Utilizador 4

Utilizador 5

Utilizador 6

Utilizador 7

Utilizador 8

Utilizador 9

Utilizador 10

Utilizador 11

Utilizador 12

Utilizador 13

Utilizador 14

Utilizador 15

Utilizador 16

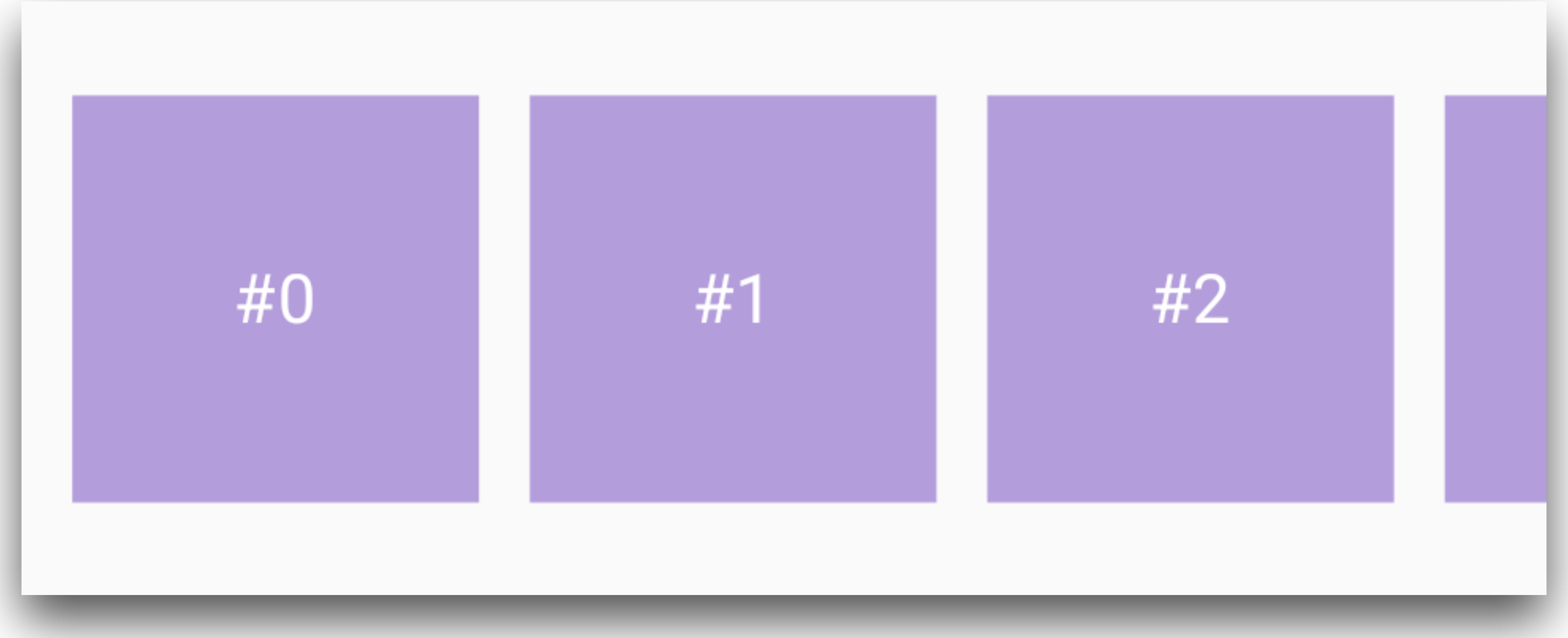
Utilizador 17

Utilizador 18

Utilizador 19

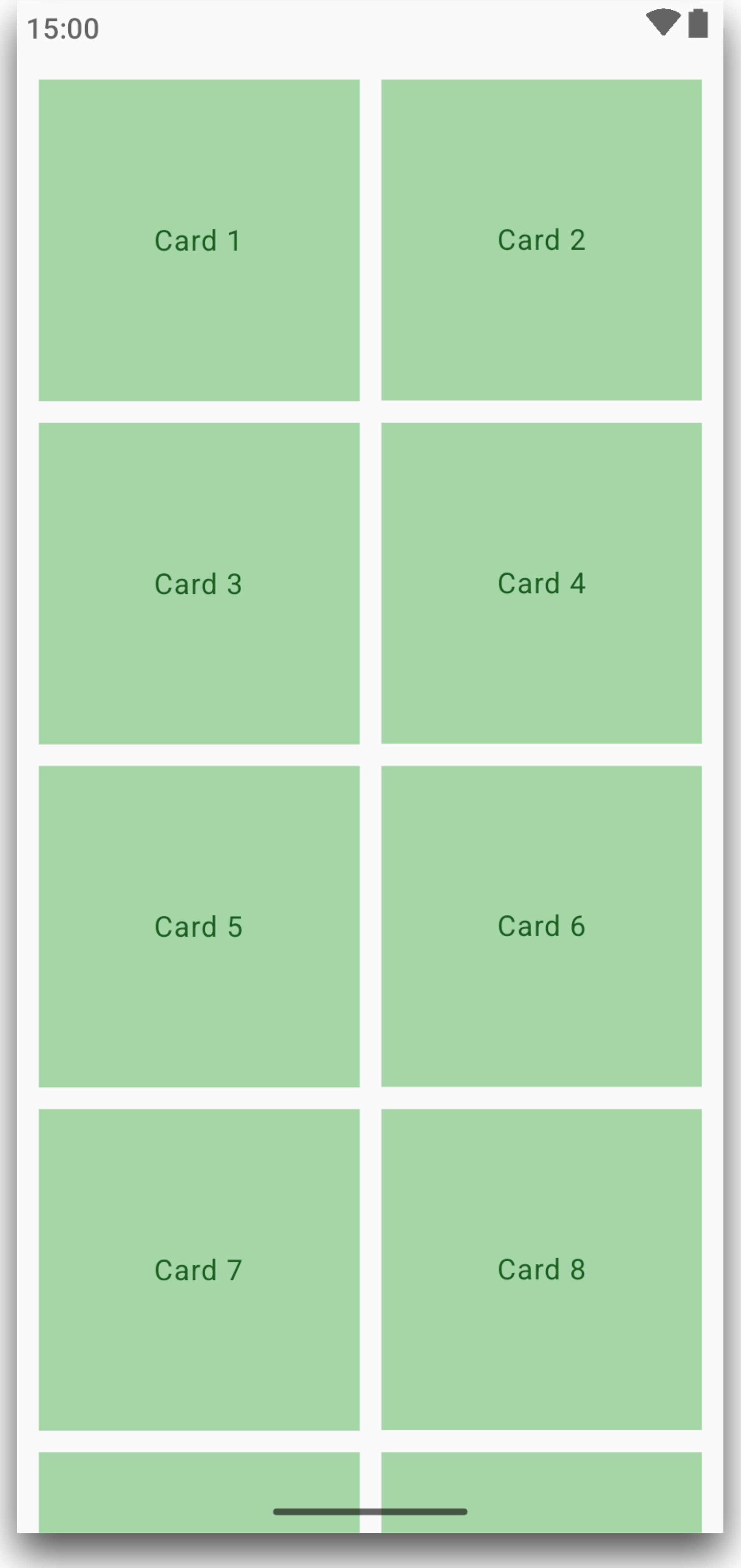
@Composable LazyRow

```kotlin
@Composable
fun LazyRowDemo(itemsCount: Int = 0) {
    LazyRow(
        contentPadding = PaddingValues(all = 12.dp),
        horizontalArrangement = Arrangement.spacedBy(space = 12.dp),
        verticalAlignment = Alignment.CenterVertically
    ) {
        items(itemsCount) { idx ->
            Box(
                modifier = Modifier
                    .size(96.dp)
                    .background(Color(0xFFB39DDB)),
                contentAlignment = Alignment.Center
            ) {
                Text("#$idx", color = Color.White)
            }
        }
    }
}

@Preview(showBackground = true, widthDp = 360, heightDp = 140)
@Composable fun LazyRowDemoPreview() {
    ComposeExamplesTheme { LazyRowDemo(itemsCount = 12) }
}
```



@Composable LazyVerticalGrid

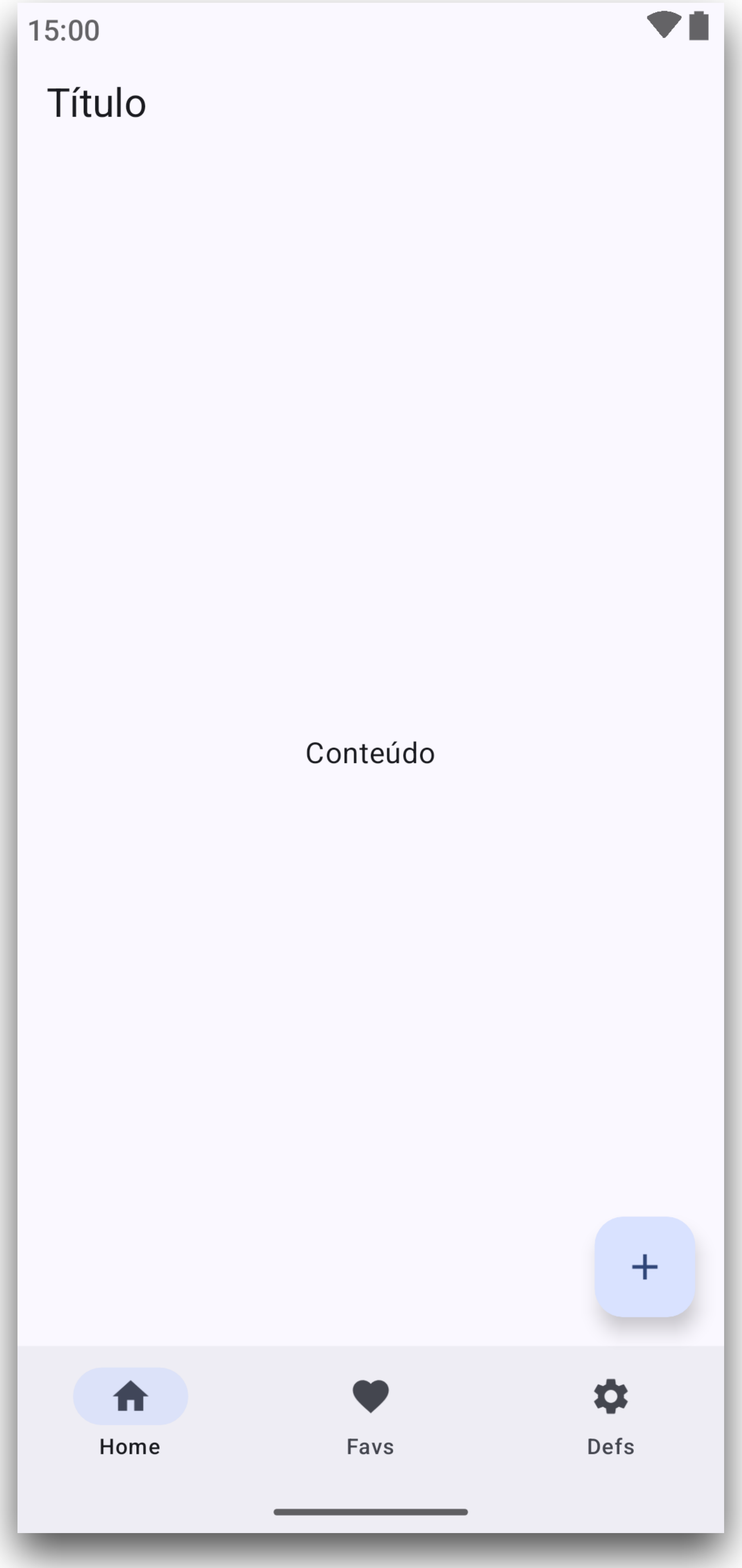
```
@OptIn(ExperimentalFoundationApi::class)
@Composable
fun GridDemo(items: List<String>, modifier: Modifier = Modifier) {
  LazyVerticalGrid(
    columns = GridCells.Adaptive(minSize = 140.dp),
    contentPadding = PaddingValues(all = 12.dp),
    horizontalArrangement = Arrangement.spacedBy(12.dp),
    verticalArrangement = Arrangement.spacedBy(12.dp),
    modifier = modifier
  ) {
    items(items = items, key = { it }) { item ->
      Box(
        modifier = Modifier.aspectRatio(1f)
          .background(Color(0xFFA5D6A7)),
        contentAlignment = Alignment.Center
      ) { Text(text = item, color = Color(0xFF1B5E20)) }
    }
  }
}

@Preview(showBackground = true, showSystemUi = true)
@Composable fun GridDemoPreview() {
  val items = (1..16).map { "Card $it" }
  ComposeExamplesTheme {
    GridDemo(items, modifier = Modifier.padding(top = 32.dp))
  }
}
```



Material Layouts

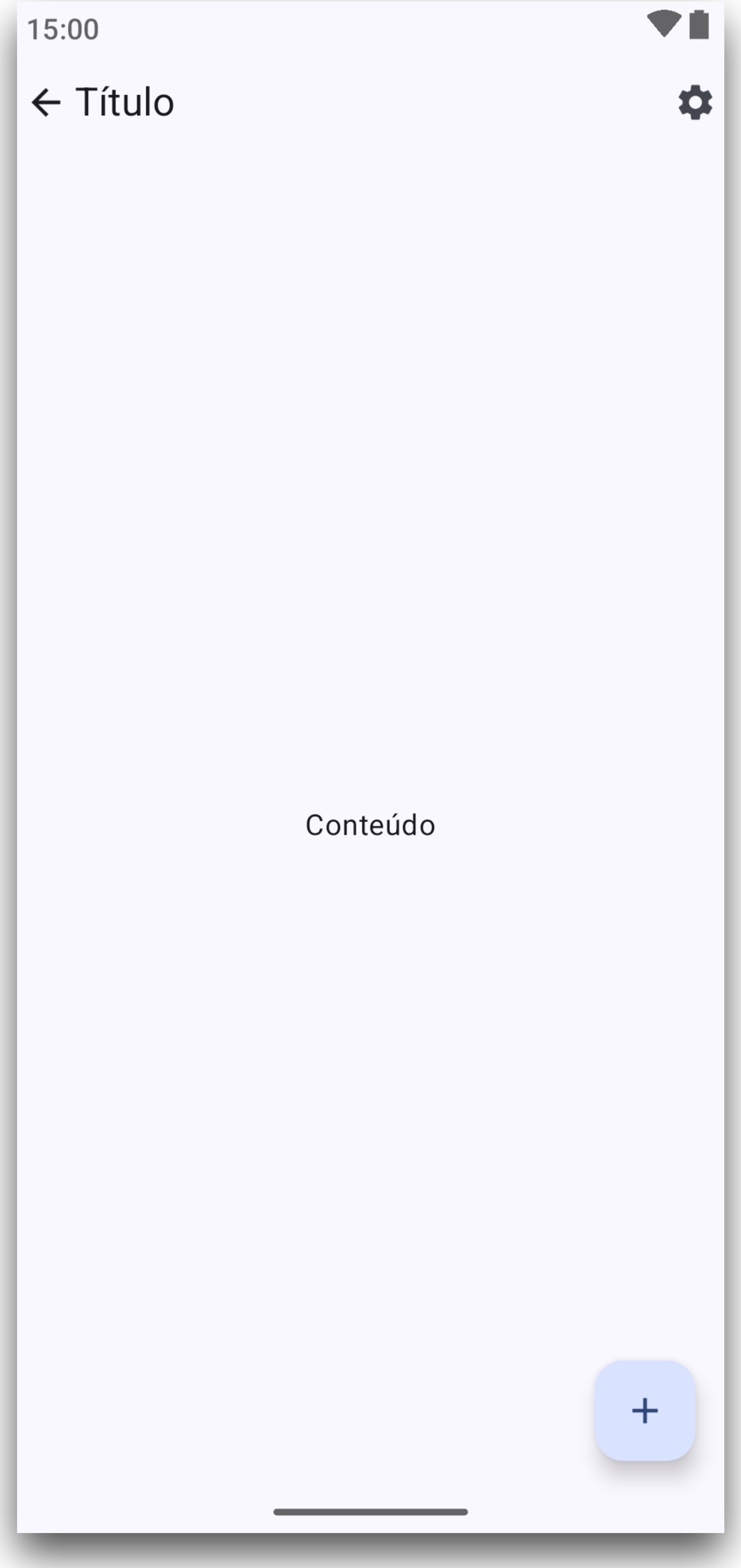
Scaffold com Bottom Bar

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScaffoldDemoBottomBar() {
    Scaffold(
        topBar = { TopAppBar(title = { Text("Título") }) },
        floatingActionButton = {
            FloatingActionButton(onClick = {}) { Icon(Icons.Default.Add, null) }
        },
        bottomBar = {
            NavigationBar {
                NavigationBarItem(
                    selected = true,  onClick = {},
                    icon = { Icon(Icons.Default.Home, null) },
                    label = { Text("Home") }
                )
                NavigationBarItem(
                    selected = false, onClick = {},
                    icon = { Icon(Icons.Default.Favorite, null) },
                    label = { Text("Favs") }
                )
                NavigationBarItem(
                    selected = false, onClick = {},
                    icon = { Icon(Icons.Default.Settings, null) },
                    label = { Text("Defs") }
                )
            }
        }
    ) { inner ->
        Box(Modifier.fillMaxSize().padding(inner), contentAlignment = Alignment.Center) {
            Text("Conteúdo")
        }
    }
}
```



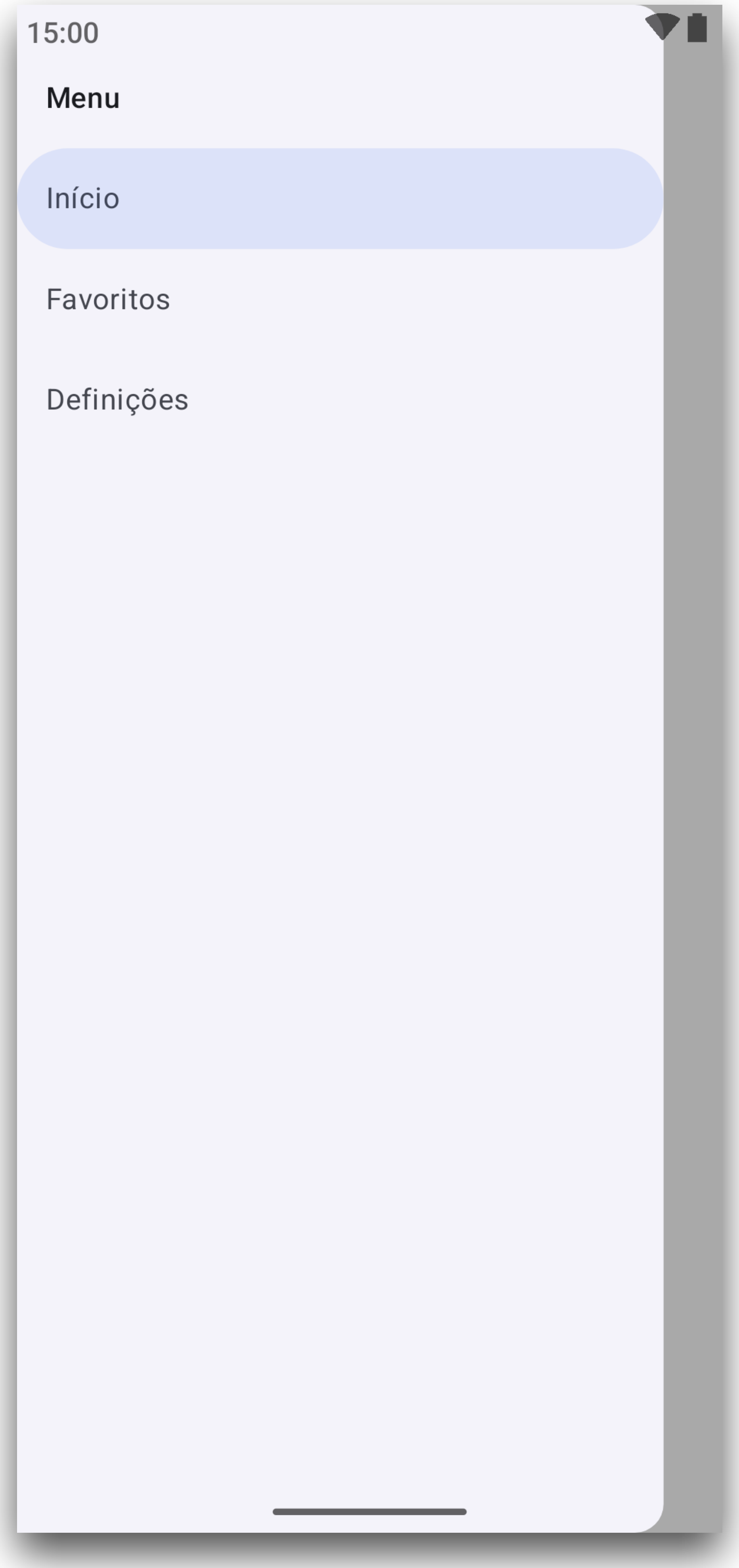
Scaffold com Top Bar

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScaffoldDemoTopBar() {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text("Título") },
                navigationIcon = { Icon(Icons.AutoMirrored.Filled.ArrowBack, null) },
                actions = { Icon(Icons.Default.Settings, null) }
            )
        },
        floatingActionButton = {
            FloatingActionButton(onClick = {}) { Icon(Icons.Default.Add, null) }
        },
    ) { inner ->
        Box(Modifier.fillMaxSize().padding(inner), contentAlignment = Alignment.Center) {
            Text("Conteúdo")
        }
    }
}
```



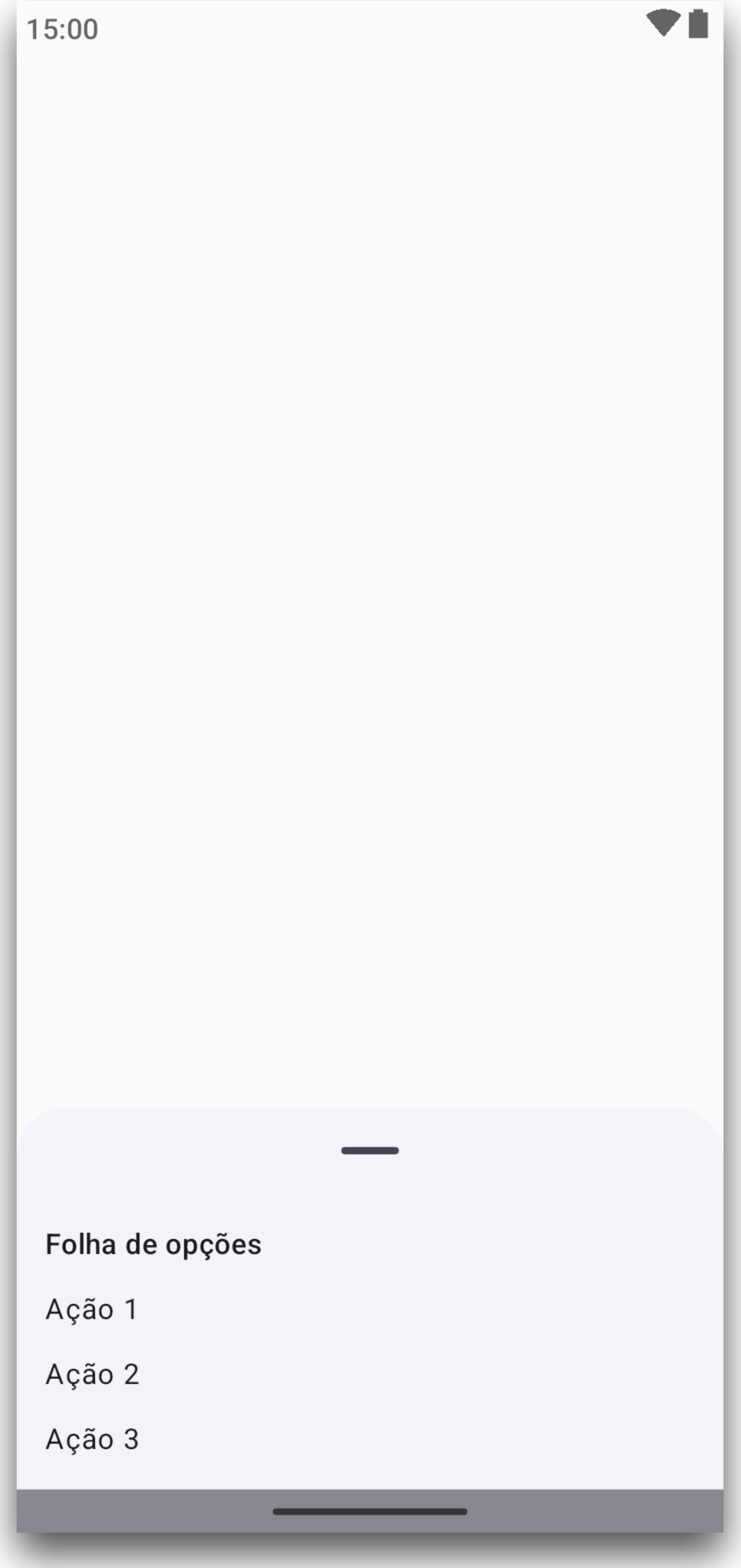
Modal Drawer

```
@Composable
fun ModalDrawerDemo() {
    val drawerState = rememberDrawerState(initialValue = DrawerValue.Open)
    ModalNavigationDrawer(
        drawerState = drawerState,
        drawerContent = {
            ModalDrawerSheet {
                Text(
                    text = "Menu",
                    modifier = Modifier.padding(16.dp),
                    style = MaterialTheme.typography.titleMedium
                )
                NavigationDrawerItem(label = { Text("Início") },
                    selected = true, onClick = {}
                )
                NavigationDrawerItem(label = { Text("Favoritos") },
                    selected = false, onClick = {}
                )
                NavigationDrawerItem(label = { Text("Definições") },
                    selected = false, onClick = {}
                )
            }
        },
    ) {
        Box(Modifier.fillMaxSize(), contentAlignment = Alignment.Center) {
            Text("Conteúdo")
        }
    }
}
```



Bottom Sheet

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ModalBottomSheetDemo() {
    ModalBottomSheet(onDismissRequest = {}) {
        Column(
            Modifier.fillMaxWidth().padding(16.dp),
            verticalArrangement = Arrangement.spacedBy(12.dp)
        ) {
            Text(
                text = "Folha de opções",
                style = MaterialTheme.typography.titleMedium
            )
            Text("Ação 1")
            Text("Ação 2")
            Text("Ação 3")
        }
    }
}
```


15:00

Folha de opções

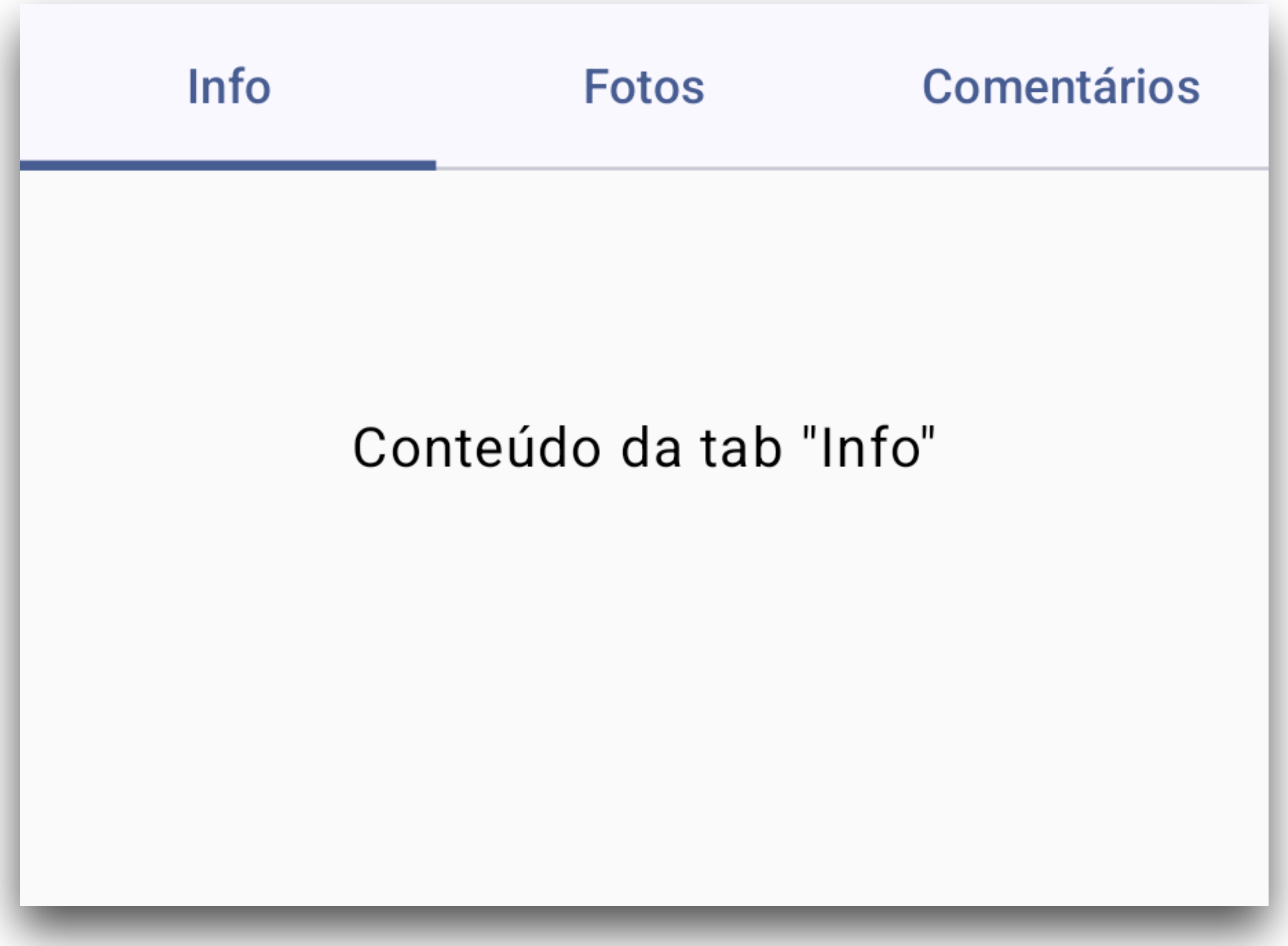
Ação 1

Ação 2

Ação 3

Tab Row

```kotlin
@Composable
fun TabRowDemo() {
    val tabs = listOf("Info", "Fotos", "Comentários")
    Column {
        TabRow(selectedTabIndex = 0) {
            tabs.forEachIndexed { idx, title ->
                Tab(
                    selected = idx == 0,
                    onClick = {},
                    text = { Text(title) }
                )
            }
        }
        Box(
            modifier = Modifier.fillMaxWidth().height(160.dp),
            contentAlignment = Alignment.Center
        ) {
            Text("Conteúdo da tab \"Info\"")
        }
    }
}

@Preview(showBackground = true, widthDp = 360, heightDp = 260)
@Composable fun TabRowDemoPreview() {
    ComposeExamplesTheme { TabRowDemo() }
}
```



Conteúdo da tab "Info"

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

Compose - Layouts

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt