



#R4E

Software Developer

Android

Primeiro Projeto - Material Design

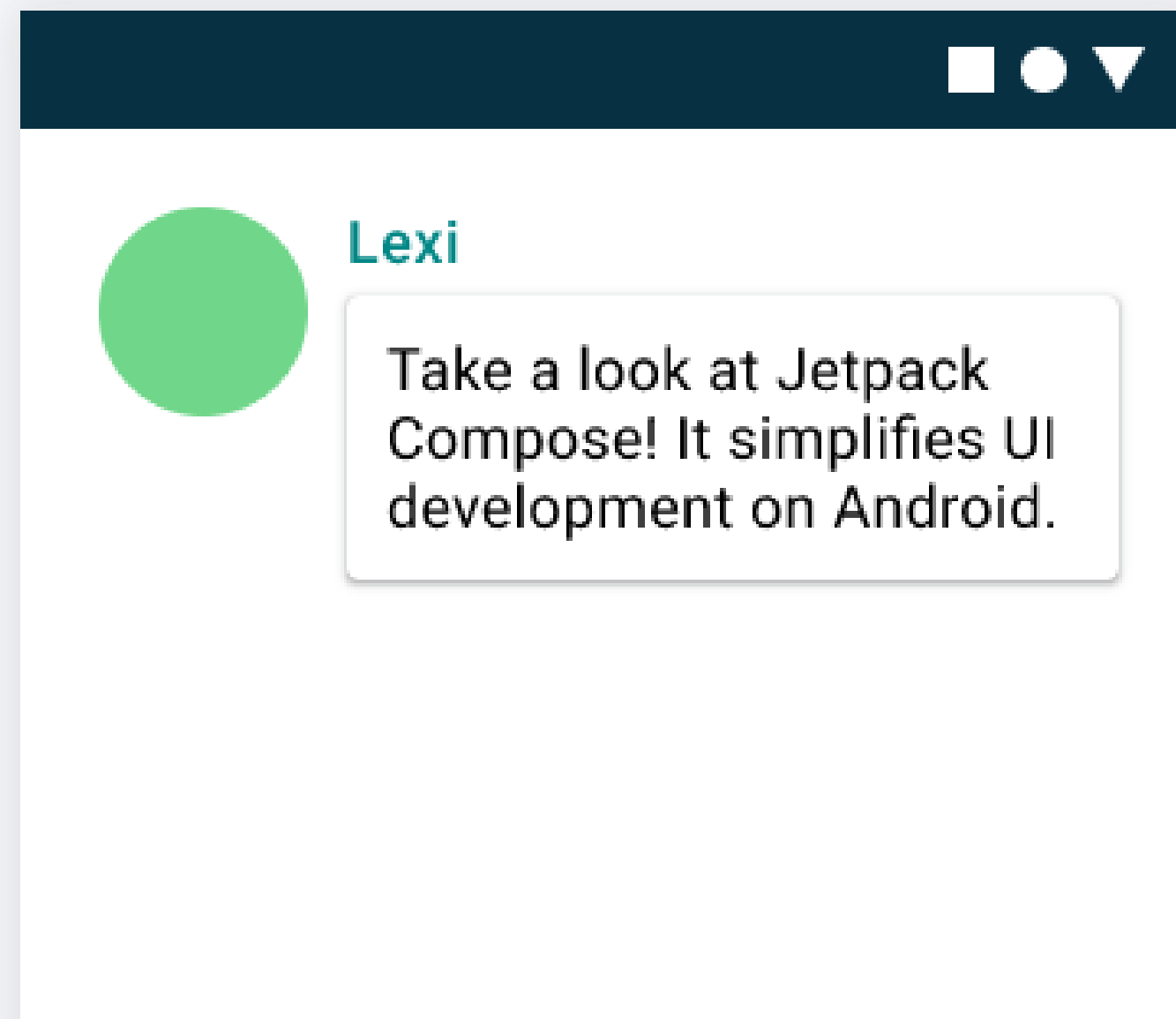


Conteúdo

- Elaboração da Primeira App
- Jetpack Compose
- Material Design

Layouts

- O Compose foi criado para oferecer suporte aos princípios do Material Design. Muitos dos elementos de IU dele implementam o Material Design por padrão. Nesta lição, você vai estilizar seu app com widgets do Material Design.



Usar o Material Design

- O design da mensagem agora tem um layout, mas ainda não está bom.
- O Jetpack Compose oferece uma implementação do Material Design e elementos de IU prontos para uso. Você vai melhorar a aparência da nossa função de composição `MessageCard` usando os estilos do Material Design.
- Para começar, una a função `MessageCard` ao tema do Material Design criado no projeto, `ComposeTutorialTheme`, bem como a um `Surface`. Faça isso nas funções `@Preview` e `setContent`. Isso vai permitir que suas funções de composição herdem os estilos definidos no tema do app, garantindo a consistência dentro dele.
- O Material Design foi criado com base em três pilares: Color (cor), Typography (tipografia) e Shape (forma). Eles vão ser adicionados um por um.

Observação

- Observação: o modelo Empty Compose Activity gera um tema padrão para o projeto, que permite personalizar o MaterialTheme. Se você deu ao seu projeto um nome diferente de ComposeTutorial, pode encontrar o tema personalizado no arquivo Theme.kt do subpacote ui.theme.

Using Material Design

```
// ...

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MyLittleAppTheme {
                Surface(modifier = Modifier.fillMaxSize()) {
                    MessageCard(Message("Android", "Jetpack Compose"))
                }
            }
        }
    }
}
```

```
@Preview
@Composable
fun PreviewMessageCard() {
    ComposeTutorialTheme {
        Surface {
            MessageCard(
                msg = Message("Colleague", "Take a look at Jetpack Compose, it's great!")
            )
        }
    }
}
```

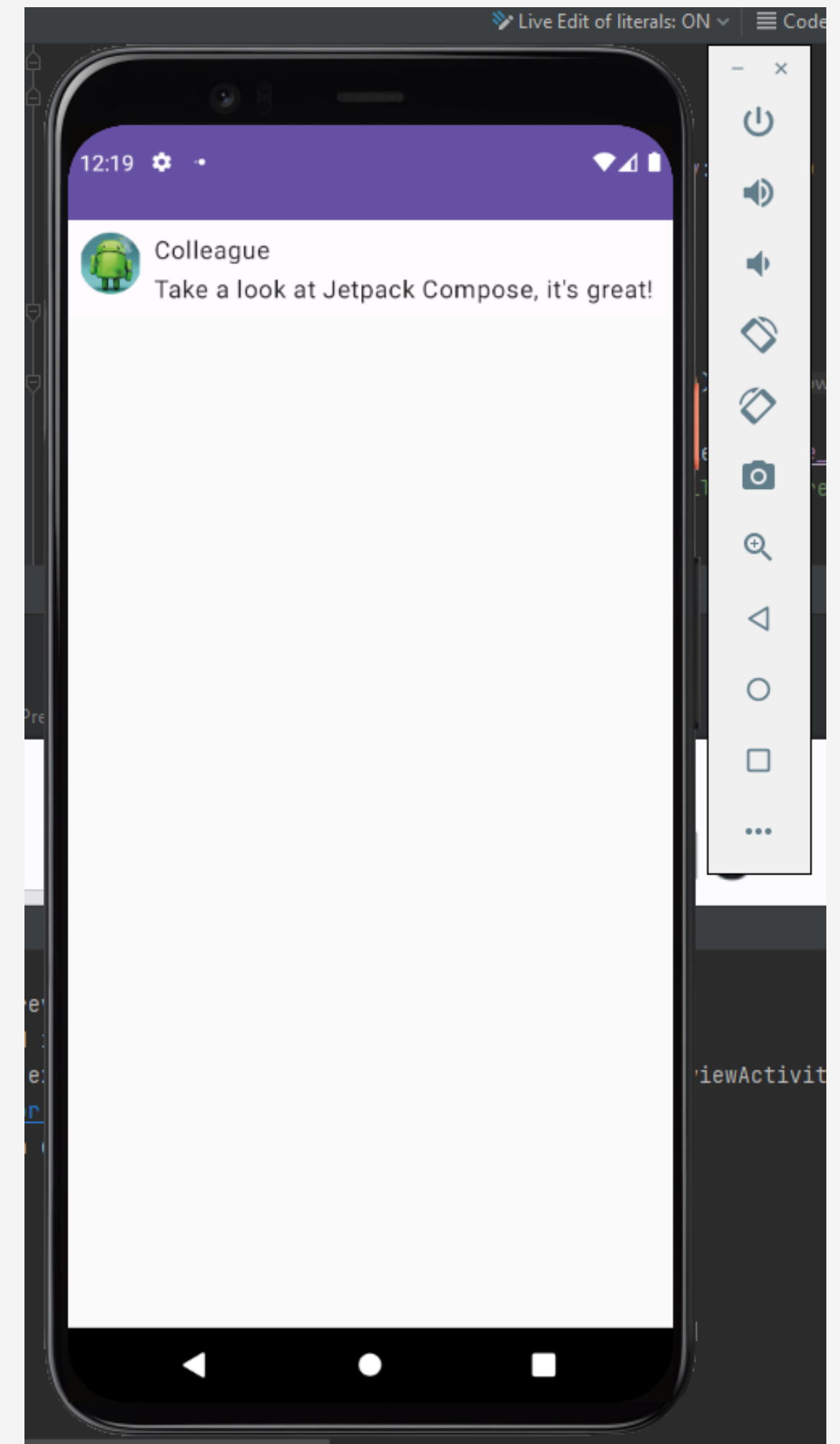
Using Material Design

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            MyLittleAppTheme() {  
                Surface(modifier = Modifier.fillMaxSize()) {  
                    MessageCard(Message(author: "Android", body: "Jetpack Compose"))  
                }  
            }  
        }  
    }  
}  
  
data class Message(val author: String, val body: String)
```

...

```
@Preview  
@Composable  
fun PreviewMessageCard() {  
    MyLittleAppTheme() {  
        Surface {  
            MessageCard(  
                msg = Message(author: "Colleague", body: "Take a look at Jetpack Compose, it's great!")  
            )  
        }  
    }  
}
```

Usar o Material Design



Cor

- Use `MaterialTheme.colors` para definir o estilo com as cores do tema envolvido. Você pode usar esses valores do tema em qualquer lugar em que uma cor seja necessária.
- Defina o estilo do título e adicione uma borda à imagem.

Cor

```
/// ...
import androidx.compose.foundation.border
import androidx.compose.material.MaterialTheme

@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = null,
            modifier = Modifier
                .size(40.dp)
                .clip(CircleShape)
                .border(1.5.dp, MaterialTheme.colorScheme.secondary, CircleShape)
        )
```

```
        Spacer(modifier = Modifier.width(8.dp))

        Column {
            Text(
                text = msg.author,
                color = MaterialTheme.colorScheme.secondary
            )

            Spacer(modifier = Modifier.height(4.dp))
            Text(text = msg.body)
        }
    }
}
```

Cor

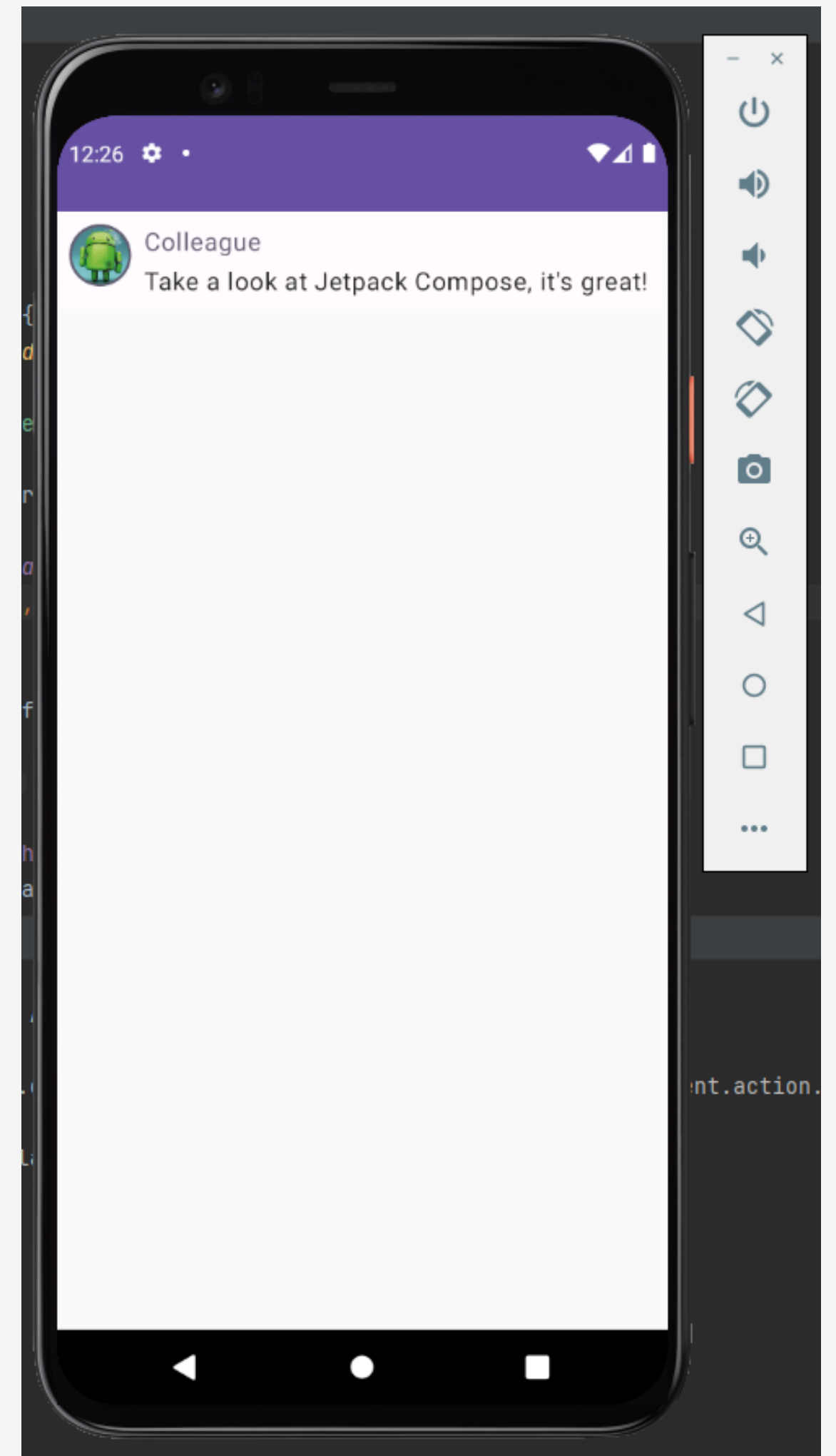
```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) { this: RowScope
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = null,
            modifier = Modifier
                .size(40.dp)
                .clip(CircleShape)
                .border(1.5.dp, MaterialTheme.colorScheme.secondary, CircleShape)
        )

        Spacer(modifier = Modifier.width(8.dp))

        Column { this: ColumnScope
            Text(
                text = msg.author,
                color = MaterialTheme.colorScheme.secondary
            )

            Spacer(modifier = Modifier.height(4.dp))
            Text(text = msg.body)
        }
    }
}
```

Cor



Tipografia

- Os estilos de tipografia do Material Design ficam disponíveis no `MaterialTheme`, eles só precisam ser adicionados à função de composição `Text`.

Tipografia

```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = null,
            modifier = Modifier
                .size(40.dp)
                .clip(CircleShape)
                .border(1.5.dp, MaterialTheme.colorScheme.secondary, CircleShape)
        )
        Spacer(modifier = Modifier.width(8.dp))

        Column {
            Text(
                text = msg.author,
                color = MaterialTheme.colorScheme.secondary,
                style = MaterialTheme.typography.labelMedium
            )
```

```
                Spacer(modifier = Modifier.height(4.dp))

                Text(
                    text = msg.body,
                    style = MaterialTheme.typography.labelMedium
                )
            }
        }
    }
}
```

Tipografia

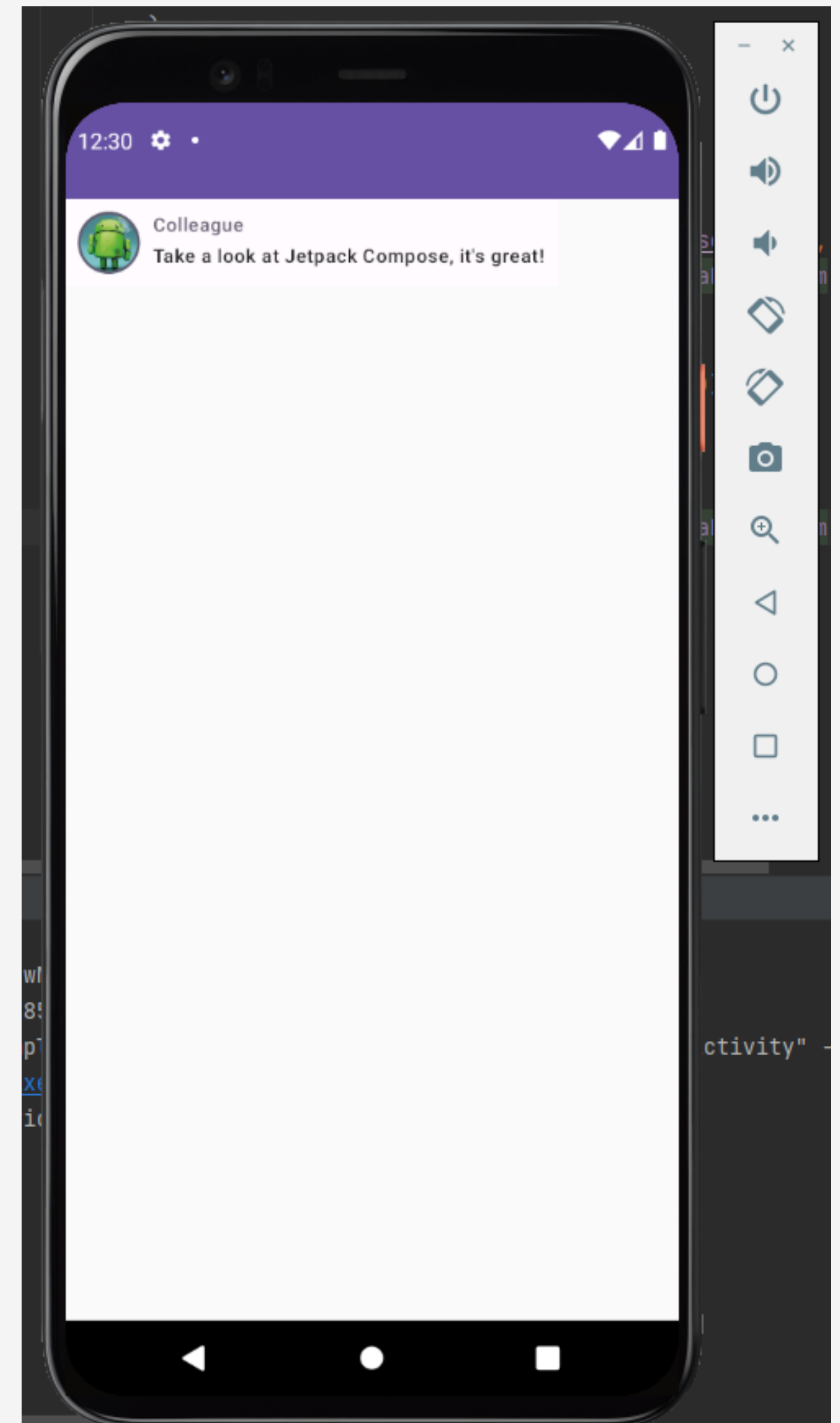
```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) { this: RowScope
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = null,
            modifier = Modifier
                .size(40.dp)
                .clip(CircleShape)
                .border(1.5.dp, MaterialTheme.colorScheme.secondary, CircleShape)
        )
        Spacer(modifier = Modifier.width(8.dp))

        Column { this: ColumnScope
            Text(
                text = msg.author,
                color = MaterialTheme.colorScheme.secondary,
                style = MaterialTheme.typography.labelMedium
            )

            Spacer(modifier = Modifier.height(4.dp))

            Text(
                text = msg.body,
                style = MaterialTheme.typography.labelMedium
            )
        }
    }
}
```

Tipografia



Forma

- Com o recurso Shape, você pode adicionar os toques finais. Primeiro, envolva o texto do corpo da mensagem em uma Surface que pode ser composta. Isso permite personalizar a forma e a elevação do corpo da mensagem. Um padding também é adicionado para melhorar o layout.

Forma

```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = null,
            modifier = Modifier
                .size(40.dp)
                .clip(CircleShape)
                .border(1.5.dp, MaterialTheme.colorScheme.secondary, CircleShape)
        )
        Spacer(modifier = Modifier.width(8.dp))

        Column {
            Text(
                text = msg.author,
                color = MaterialTheme.colorScheme.secondary,
                style = MaterialTheme.typography.labelMedium
            )
```

```
            Spacer(modifier = Modifier.height(4.dp))

            Surface(shape = MaterialTheme.shapes.medium, tonalElevation = 1.dp){
                Text(
                    text = msg.body,
                    modifier = Modifier.padding(all = 4.dp),
                    style = MaterialTheme.typography.labelMedium
                )
            }
        }
    }
}
```

Forma

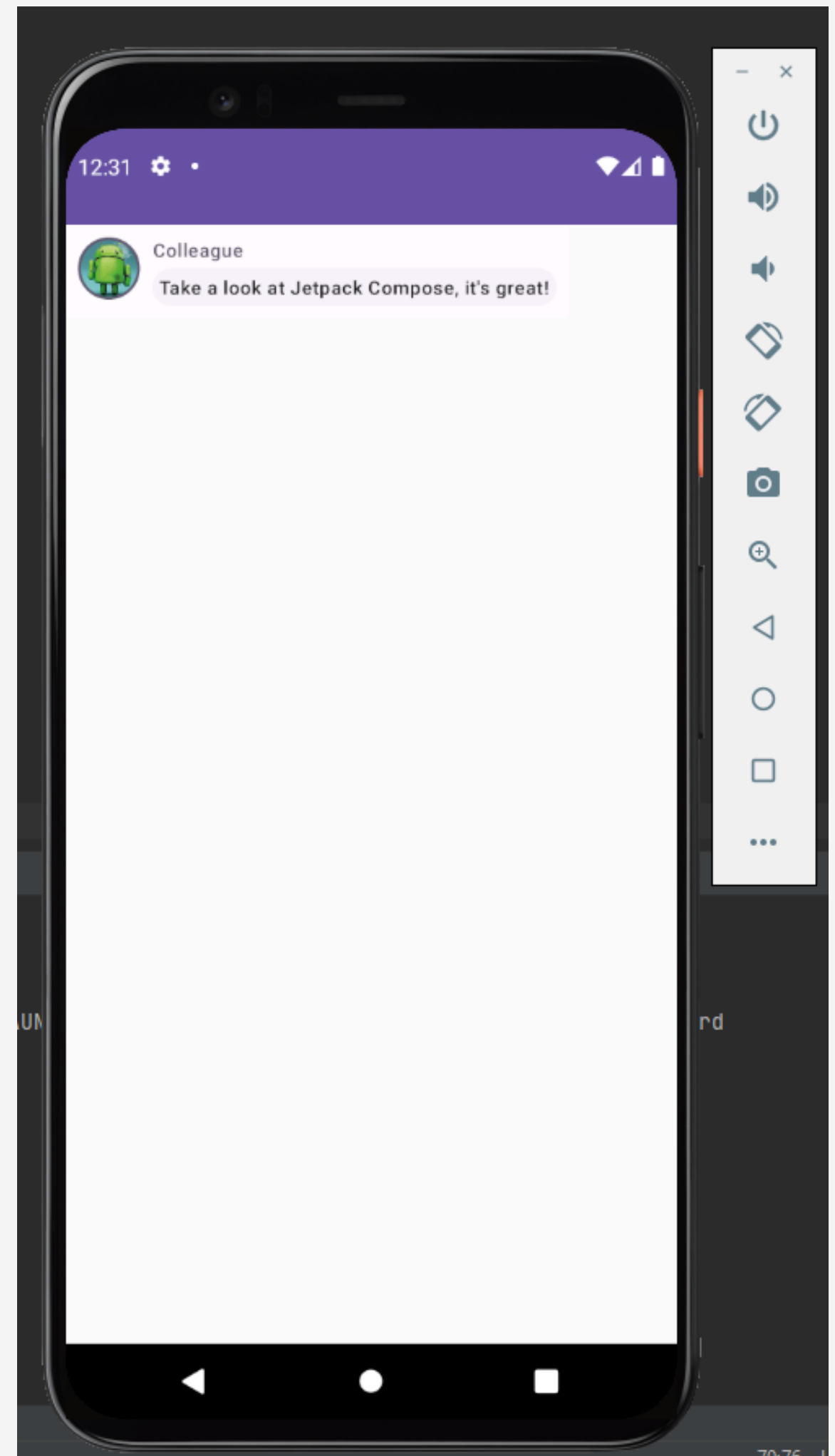
```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) { this: RowScope
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = null,
            modifier = Modifier
                .size(40.dp)
                .clip(CircleShape)
                .border(1.5.dp, MaterialTheme.colorScheme.secondary, CircleShape)
        )
        Spacer(modifier = Modifier.width(8.dp))

        Column { this: ColumnScope
            Text(
                text = msg.author,
                color = MaterialTheme.colorScheme.secondary,
                style = MaterialTheme.typography.labelMedium
            )

            Spacer(modifier = Modifier.height(4.dp))

            Surface(shape = MaterialTheme.shapes.medium, tonalElevation = 1.dp){
                Text(
                    text = msg.body,
                    modifier = Modifier.padding(all = 4.dp),
                    style = MaterialTheme.typography.labelMedium
                )
            }
        }
    }
}
```

Forma



Ativar tema escuro

- O tema escuro (ou modo noturno) pode ser ativado para evitar uma tela clara, especialmente à noite, ou apenas para economizar bateria do dispositivo. Graças ao suporte ao Material Design, o Jetpack Compose pode processar o tema escuro por padrão. Ao usar as cores do Material Design, o texto e os planos de fundo são adaptados automaticamente ao plano de fundo escuro.
- É possível criar diversas prévias no seu arquivo como funções separadas ou adicionar diversas anotações à mesma função.
- Adicione uma nova anotação de prévia e ative o modo noturno.

Ativar tema escuro

```
@Preview(name = "Light Mode")
@Preview(
    uiMode = Configuration.UI_MODE_NIGHT_YES,
    showBackground = true,
    name = "Dark Mode"
)

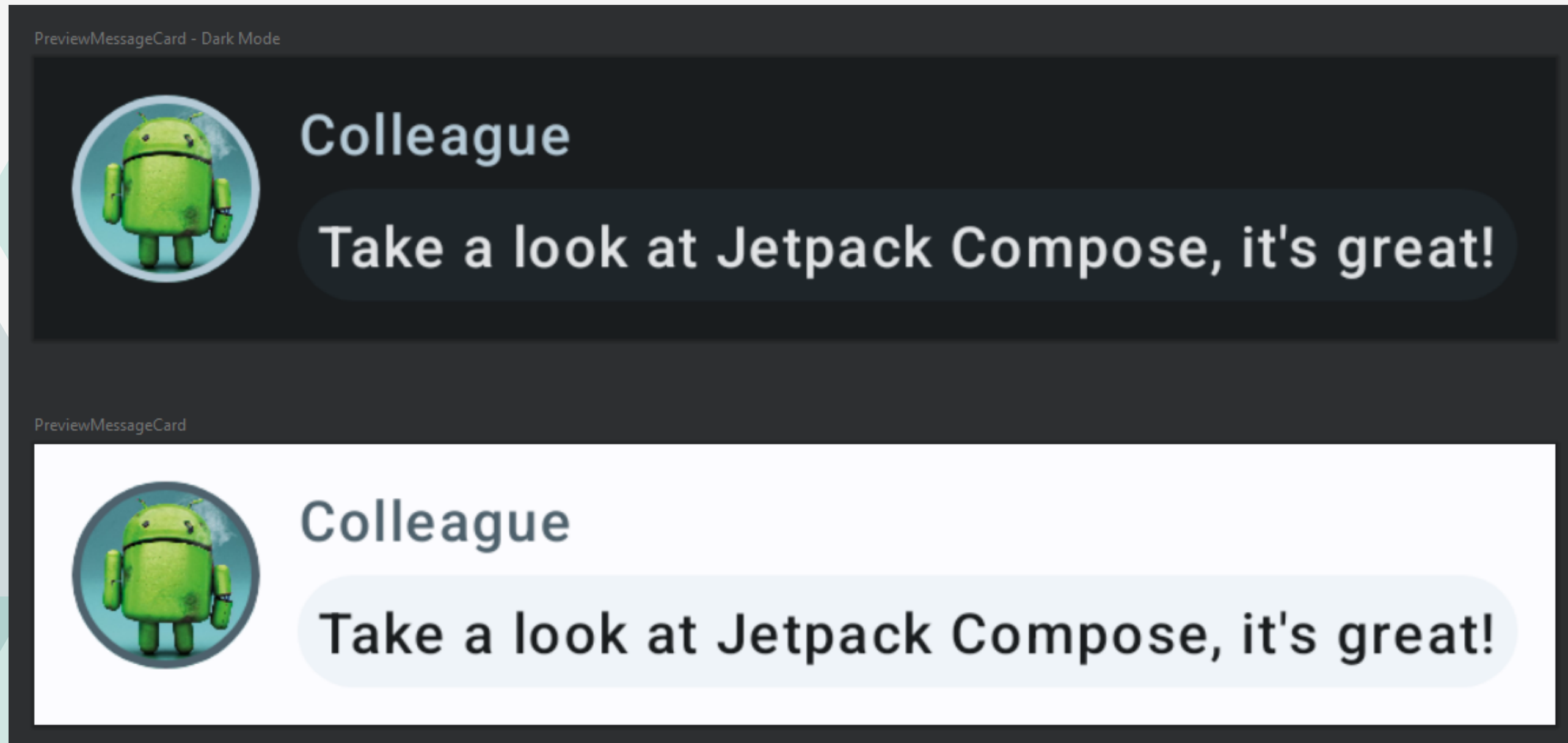
@Preview
@Composable
fun PreviewMessageCard() {
    MyLittleAppTheme() {
        Surface {
            MessageCard(
                msg = Message("Colleague", "Take a look at Jetpack Compose, it's great!")
            )
        }
    }
}
```

Ativar tema escuro

```
@Preview(name = "Light Mode")
@Preview(
    uiMode = Configuration.UI_MODE_NIGHT_YES,
    showBackground = true,
    name = "Dark Mode"
)

@Preview
@Composable
fun PreviewMessageCard() {
    MyLittleAppTheme() {
        Surface {
            MessageCard(
                msg = Message(author: "Colleague", body: "Take a look at Jetpack Compose, it's great!")
            )
        }
    }
}
```

Ativar tema escuro





#R4E

Software Developer

Android

Primeiro Projeto - Material Design

