#R4E
Software Developer

Android

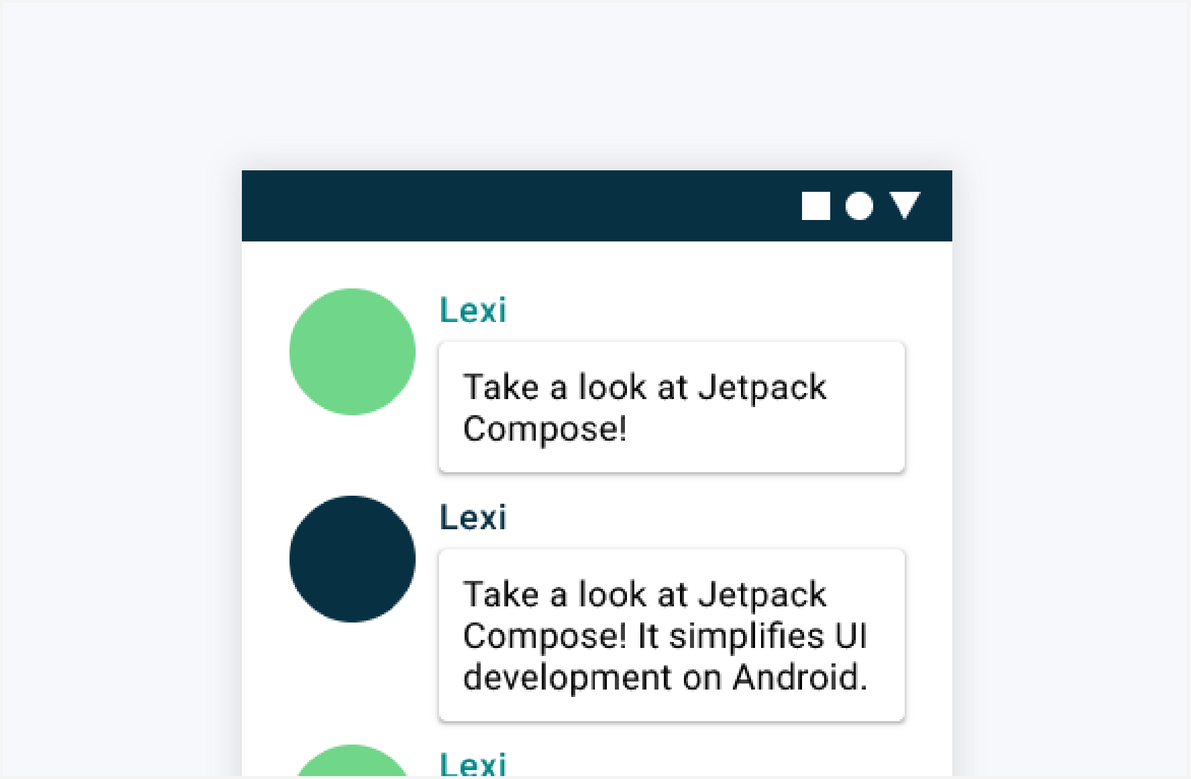
Primeiro Projeto - Listas e Animações

Conteúdo

- Elaboração da Primeira App
- Jetpack Compose
- Listas e Animações

Layouts

- Listas e animações estão por toda parte nos apps. Nesta lição, você vai aprender como o Compose facilita a criação de listas e torna divertido o acréscimo de animações.



Criar uma lista de mensagens

- Um chat com apenas uma mensagem parece um pouco solitário, então mude a conversa para que ela tenha mais de uma mensagem. Você vai precisar criar uma função Conversation que mostrará várias mensagens. Para este caso de uso, use a LazyColumn e a LazyRow do Compose. Essas funções de composição processam apenas os elementos visíveis na tela. Portanto, elas são muito eficientes para listas longas.
- Neste snippet de código, é possível ver que a LazyColumn tem um elemento items filho. Uma List é usada como parâmetro, e o lambda recebe um parâmetro que chamamos de message, mas poderíamos ter dado qualquer outro nome, que é uma instância da Message. Em resumo, esse lambda é chamado para cada item da List fornecida. Importe este exemplo de conjunto de dados (https://gist.github.com/yrezgui/26a1060d67bf0ec2a73fa12695166436) para seu projeto e ajude a conversa a ser inicializada mais rapidamente.

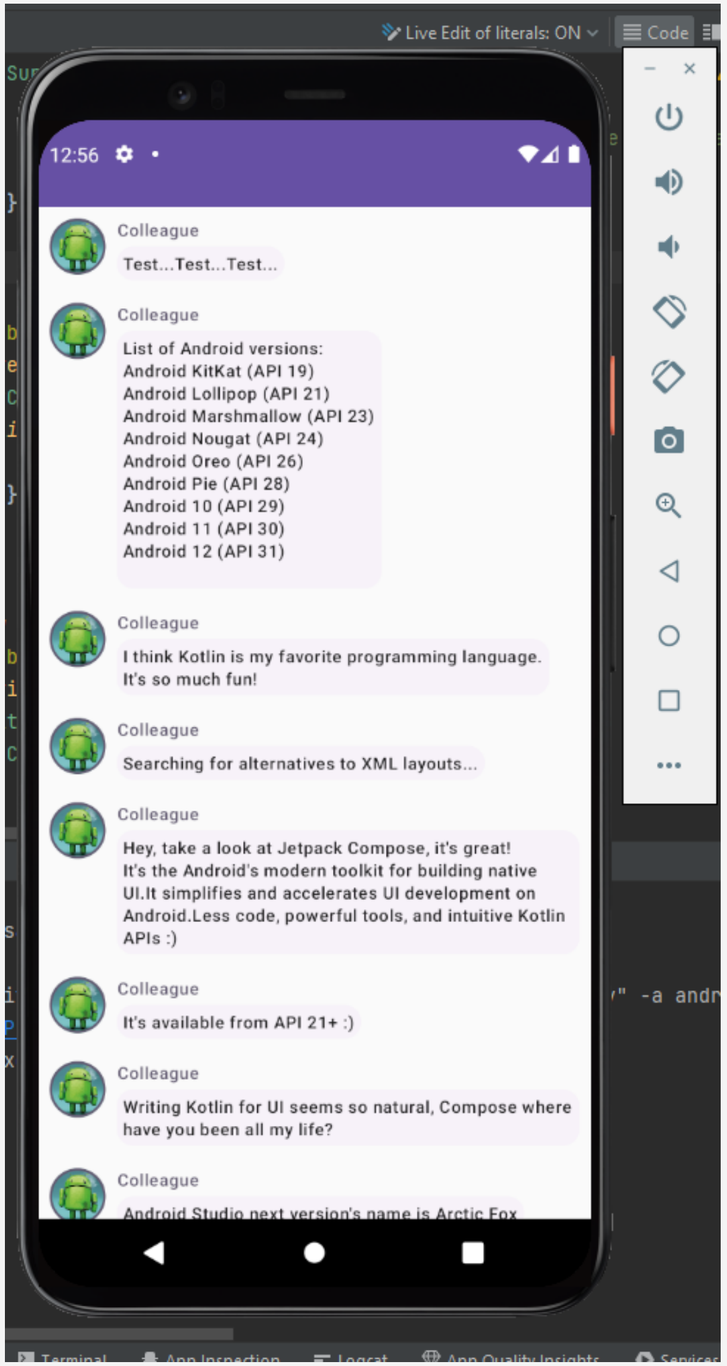
Criar uma lista de mensagens

```
@Composable
fun Conversation(messages: List<Message>) {
    LazyColumn {
        items(messages) { message ->
            MessageCard(message)
        }
    }
}

@Preview
@Composable
fun PreviewConversation() {
    MyLittleAppTheme() {
        Conversation(SampleData.conversationSample)
    }
}
```

Criar uma lista de mensagens

```
@Composable
fun Conversation(messages: List<Message>) {
    LazyColumn { this: LazyListScope
        items(messages) { this: LazyItemScope message ->
            MessageCard(message)
        }
    }
}

@Preview
@Composable
fun PreviewConversation() {
    MyLittleAppTheme() {
        Conversation(SampleData.conversationSample)
    }
}

object SampleData {
    // Sample conversation data
    val conversationSample = listOf(
        Message(
            author: "Colleague",
            body: "Test...Test...Test..."
        ),
        Message(
            author: "Colleague",
            body: "List of Android versions:\n" +
                    "Android KitKat (API 19)\n" +
                    "Android Lollipop (API 21)\n" +
                    "Android Marshmallow (API 23)\n" +
                    "Android Nougat (API 24)\n" +
```

Criar uma lista de mensagens



Animar mensagens ao abrir

- A conversa está ficando mais interessante. Chegou a hora de brincar com animações. Você vai adicionar a capacidade de abrir uma mensagem para mostrar uma parte maior dela, animando o tamanho do conteúdo e a cor do plano de fundo. Para armazenar esse estado da IU local, precisamos conferir se uma mensagem foi aberta ou não. Para monitorar essa mudança de estado, é preciso usar as funções remember e mutableStateOf.
- As funções de composição podem armazenar o estado local na memória usando remember e monitorar as mudanças no valor transmitido para mutableStateOf. As funções de composição (e os filhos delas) que usam esse estado serão redesenhadas automaticamente quando o valor for atualizado. Isso é chamado de recomposição.
- Com o uso das APIs de estado do Compose, como remember e mutableStateOf, qualquer mudança no estado atualiza automaticamente a IU.

Observação

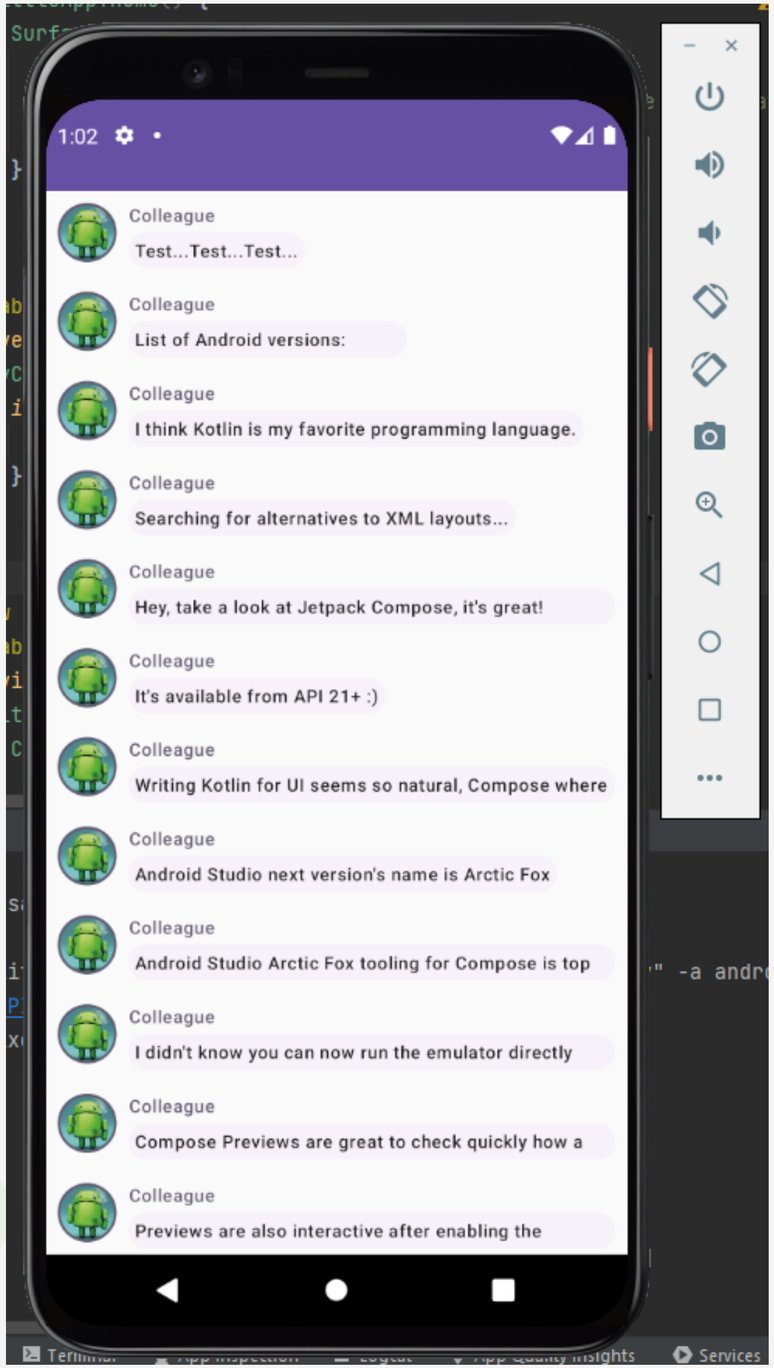
- Adicione as seguintes importações para usar o by corretamente. Pressione Alt + Enter ou Option + Enter para adicionar.
- import androidx.compose.runtime.getValue

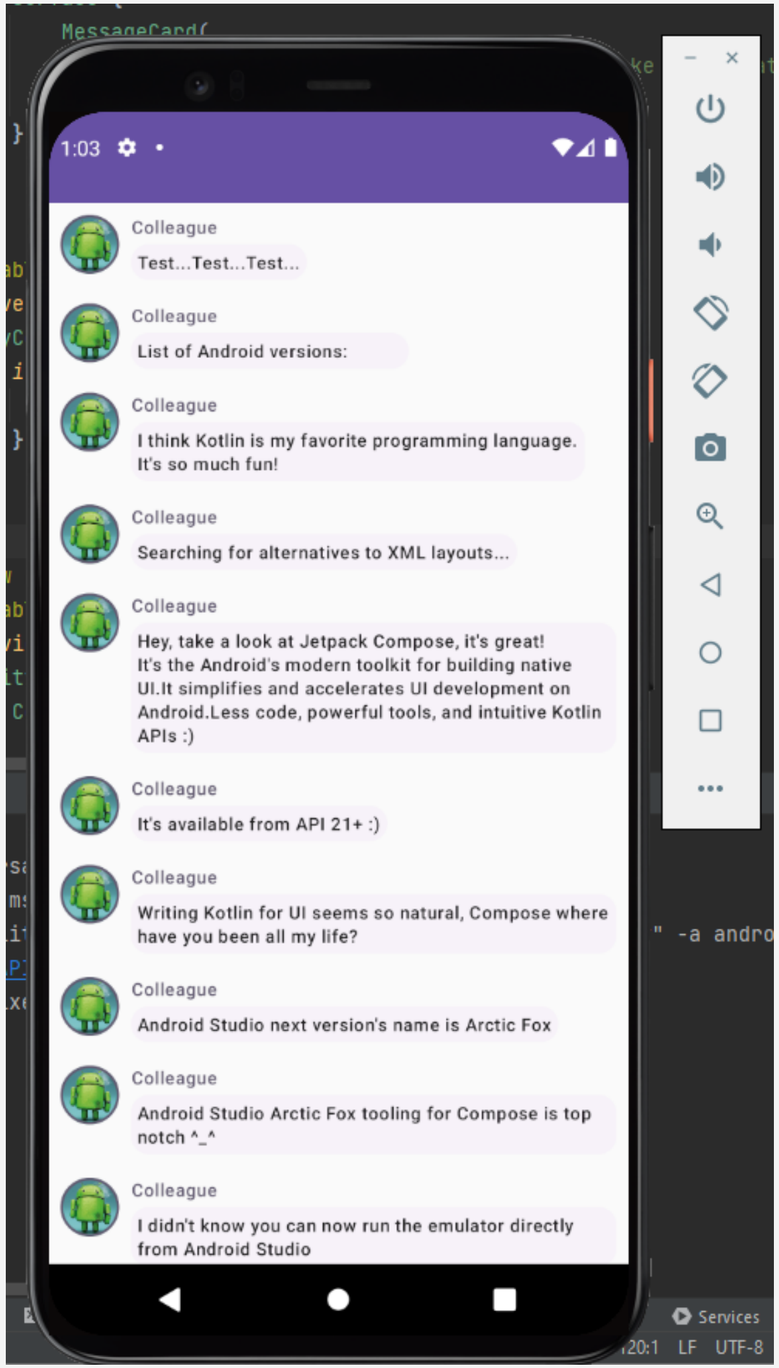
Animar mensagens ao abrir

```
// ...
import androidx.compose.foundation.clickable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue

class MainActivity : ComponentActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
      MyLittleAppTheme {
        Conversation(SampleData.conversationSample)
      }
    }
  }
}
@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))
```

```
    // We keep track if the message is expanded or not in this
    // variable
    var isExpanded by remember { mutableStateOf(false) }

    // We toggle the isExpanded variable when we click on this Column
    Column(modifier = Modifier.clickable { isExpanded = !isExpanded }) {
      Text(
        text = msg.author,
        color = MaterialTheme.colorScheme.secondary,
        style = MaterialTheme.typography.subtitle2
      )

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

      Surface(
        shape = MaterialTheme.shapes.medium,
        elevation = 1.dp,
      ) {
        Text(
          text = msg.body,
          modifier = Modifier.padding(all = 4.dp),
          // If the message is expanded, we display all its content
          // otherwise we only display the first line
          maxLines = if (isExpanded) Int.MAX_VALUE else 1,
          style = MaterialTheme.typography.body2
        )
      }
    }
  }
}
```

Animar mensagens ao abrir





#R4E
Software Developer

Android

Primeiro Projeto - Listas e Animações