

# MainActivity

```
MainActivity.kt × Color.kt
1 package com.example.appcadastro
2
3 import android.os.Bundle
4 import androidx.activity.ComponentActivity
5 import androidx.activity.compose.setContent
6 import androidx.activity.enableEdgeToEdge
7 import androidx.compose.foundation.Image
8 import androidx.compose.foundation.background
9 import androidx.compose.foundation.layout.*
10 import androidx.compose.foundation.rememberScrollState
11 import androidx.compose.foundation.shape.CircleShape
12 import androidx.compose.foundation.text.KeyboardOptions
13 import androidx.compose.foundation.verticalScroll
14 import androidx.compose.material3.*
15 import androidx.compose.runtime.*
16 import androidx.compose.ui.Alignment
17 import androidx.compose.ui.Modifier
18 import androidx.compose.ui.draw.clip
19 import androidx.compose.ui.graphics.Brush
20 import androidx.compose.ui.layout.ContentScale
21 import androidx.compose.ui.res.painterResource
22 import androidx.compose.ui.text.font.FontFamily
23 import androidx.compose.ui.text.font.FontWeight
24 import androidx.compose.ui.text.input.KeyboardType
25 import androidx.compose.ui.unit.dp
26 import androidx.compose.ui.unit.sp
27 import androidx.compose.ui.tooling.preview.Preview
28 import androidx.compose.ui.platform.LocalContext
29 import androidx.compose.ui.platform.LocalDensity
30 import androidx.compose.ui.platform.LocalView
31 import androidx.compose.ui.platform.ViewCompositionStrategy
32 import androidx.compose.ui.platform.LocalSoftwareKeyboardController
33 import androidx.compose.ui.text.input.PasswordVisualTransformation
34 import androidx.compose.ui.unit.Dp
35 import androidx.core.view.WindowCompat
36 import com.example.appcadastro.ui.theme.AppCadastroTheme
37 import com.example.appcadastro.ui.theme.Darkblue
38 import com.example.appcadastro.ui.theme.Lightblue
39
```

- **package:** É o "endereço" do arquivo no projeto.
- **imports:** São as bibliotecas/funcionalidades que o código usa.

```

39
40 class MainActivity : ComponentActivity() {
41     override fun onCreate(savedInstanceState: Bundle?) {
42         super.onCreate(savedInstanceState)
43
44         // Ativa um modo onde o conteúdo pode usar a tela inteira, até mesmo por baixo da barra de status
45         enableEdgeToEdge()
46
47         // Define o conteúdo da tela
48         setContent {
49             AppCadastroTheme {
50                 // Scaffold é tipo uma estrutura padrão com área segura pra conteúdo
51                 Scaffold(modifier = Modifier.fillMaxSize()) {
52                     appPreview()
53                 }
54             }
55         }
56     }
57 }
58

```

- **ComponentActivity:** Classe base para telas no Android (herdamos tudo que uma tela precisa).
- **onCreate:** Método que roda quando a tela é criada.
- **enableEdgeToEdge():** Faz o app usar até as bordas da tela (debaixo do furo da câmera, por exemplo).
- **setContent:** Define o layout.
- **Scaffold:** Layout básico com áreas seguras (evita que textos fiquem atrás da barra de status).

```

58
59 @Composable
60 fun ProdutoItem() {
61     // Variáveis que vão guardar o que o usuário digita nos campos
62     var nome by remember { mutableStateOf("") }
63     var telefone by remember { mutableStateOf("") }
64     var curso by remember { mutableStateOf("") }
65     var serie by remember { mutableStateOf("") }
66

```

- **remember:** Guarda o valor mesmo se a tela for redesenhada (como uma "memória" temporária).
- **mutableStateOf:** Cria um valor que, quando muda, atualiza automaticamente a tela.

```

66
67
68
69
70
71
72
    Column(
        modifier = Modifier
            .fillMaxSize() // ocupa toda a altura possível
            .verticalScroll(rememberScrollState()) // permite rolar a tela
            .imePadding() // 🐞 ESSA LINHA FAZ A MÁGICA: dá espaço quando o teclado aparece
    ) {

```

- **Column:** Empilha elementos verticalmente.
- **verticalScroll:** Permite rolagem se o conteúdo for grande.
- **imePadding():** Ajusta o espaço quando o teclado virtual aparece (pra não tampar os campos).

```

73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
    // Topo com um fundo colorido em gradiente
    Box(
        modifier = Modifier
            .height(180.dp)
            .background(
                brush = Brush.horizontalGradient(
                    colors = listOf(Darkblue, Lightblue)
                )
            )
            .fillMaxWidth()
    ) {
        // Avatar centralizado no topo
        Image(
            painter = painterResource(R.drawable.avatr_woman),
            contentDescription = "avatar image",
            contentScale = ContentScale.Crop,
            modifier = Modifier
                .offset(y = 70.dp) // desce um pouco o avatar
                .size(150.dp) // tamanho da imagem
                .clip(CircleShape) // deixa redondinho
                .align(Alignment.Center) // centraliza dentro da Box
        )
    }

```

- **Box:** Container que permite sobrepor elementos (tipo uma mesa onde você coloca coisas).
- **Brush.horizontalGradient:** Cria um degrade azul horizontal.
- **clip(CircleShape):** Corta a imagem em formato circular.
- **align(Alignment.Center):** Centraliza o avatar na Box.

```

96
97     Spacer(Modifier.height(55.dp)) // espaço entre avatar e formulário
98
99     // Campos do formulário
100     Column(Modifier.padding(16.dp)) {
101         Text(
102             text = "Cadastre-se",
103             fontFamily = FontFamily.SansSerif,
104             fontWeight = FontWeight(500),
105             fontSize = 35.sp,
106         )

```

```

108         Spacer(Modifier.height(25.dp))
109
110         // Campo Nome
111         Text("Nome:", fontSize = 25.sp, fontWeight = FontWeight(250))
112         Spacer(Modifier.height(15.dp))
113         TextField(
114             value = nome,
115             onValueChange = { nome = it },
116             label = { Text("Digite seu nome completo") },
117             maxLines = 2
118         )
119
120         Spacer(Modifier.height(20.dp))
121
122         // Campo Telefone
123         Text("Telefone:", fontSize = 25.sp, fontWeight = FontWeight(250))
124         Spacer(Modifier.height(15.dp))
125         TextField(
126             value = telefone,
127             onValueChange = { telefone = it },
128             label = { Text("Digite o telefone...") },
129             keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number)
130         )
131
132         Spacer(Modifier.height(20.dp))
133
134         // Campo Curso
135         Text("Curso:", fontSize = 25.sp, fontWeight = FontWeight(250))
136         Spacer(Modifier.height(15.dp))
137         TextField(
138             value = curso,
139             onValueChange = { curso = it },
140             label = { Text("Digite o nome do curso") },
141             maxLines = 2
142         )
143
144         Spacer(Modifier.height(20.dp))
145

```

```

145
146 // Campo Série
147 Text("Série:", fontSize = 25.sp, fontWeight = FontWeight(250))
148 Spacer(Modifier.height(15.dp))
149 TextField(
150     value = serie,
151     onValueChange = { serie = it },
152     label = { Text("Digite o número da sua série") },
153     keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),
154     maxLines = 2
155 )
156 }

```

- **Spacer(Modifier.height(55.dp))**: espaço entre avatar e formulário.
- **Column(Modifier.padding(16.dp))**: Agora, todos os campos do formulário vão ficar dentro de uma 'caixa' (a `Column`) que tem um espaçamento interno de 16 unidades em todas as bordas.
- **keyboardType.Number**: Força o teclado numérico a aparecer (útil para telefone/série).

```

157
158 // Botão de cadastro
159 Column(Modifier.padding(16.dp)) {
160     Button(
161         onClick = { /* Ação do botão */ }
162     ) {
163         Text(text = "Cadastrar")
164     }
165 }
166 }
167 }

```

- **onClick**: O que acontece quando o botão é pressionado (atualmente vazio/"TODO").

```

168
169 @Preview
170 @Composable
171 fun appPreview() {
172     AppCadastroTheme {
173         Surface(
174             modifier = Modifier.fillMaxSize(),
175             color = MaterialTheme.colorScheme.background
176         ) {
177             ProdutoItem()
178         }
179     }
180 }
181

```

- **@Preview:** Faz o Android Studio mostrar uma miniatura do layout (sem precisar rodar o app no celular).
- **Surface:** Container que aplica o tema (cor de fundo, etc.).

## Color

```

MainActivity.kt  Color.kt x
1 package com.example.appcadastro.ui.theme
2
3 import androidx.compose.ui.graphics.Color
4

```

- **package:** O "endereço" desse arquivo (aqui guardamos tudo relacionado ao tema visual do app).
- **import:** Pega a classe `Color` do Compose, que a gente usa pra definir cores.

```

5 // Essas cores aqui são tons mais clarinhos (mais "80") que geralmente são usadas no tema claro
6 val Purple80 = Color(0xFFD0BCFF)
7 val PurpleGrey80 = Color(0xFFCCC2DC)
8 val Pink80 = Color(0xFFE8BFD0)

```

- **O que são?** Cores pré-definidas pelo Material Design pra temas claros.

- **Por que "80"?** É uma convenção pra indicar tons mais claros (usados em fundos claros).

```
10 // Essas são as cores mais escuras (mais "40"), que podem ser usadas no tema escuro
11 ■ val Purple40 = Color(0xFF6650a4)
12 ■ val PurpleGrey40 = Color(0xFF625b71)
13 ■ val Pink40 = Color(0xFF7D5260)
14
```

- **O que são?** Versões mais escuras das cores acima, pro tema escuro.
- **Por que "40"?** Tons mais escuros (melhor contraste em fundos escuros).

```
15 // Cores personalizadas criadas por você - provavelmente pra dar uma identidade única pro app
16 ■ val Darkblue = Color(0xff2e4566) // azul escuro, tipo um fundo de cabeçalho
17 ■ val Lightblue = Color(0xff53b0aa) // azul mais claro, talvez pra um gradiente
18
```

- **Pra que servem?**
  - **Darkblue**: Ótimo pra cabeçalhos ou textos importantes.
  - **Lightblue**: Combina com o **Darkblue** em gradientes ou botões.
- **Notação 0xff...**: Código hexadecimal da cor (o **ff** no começo significa opacidade total).