

## Exercício Prático: Tela de Perfil de Viajante

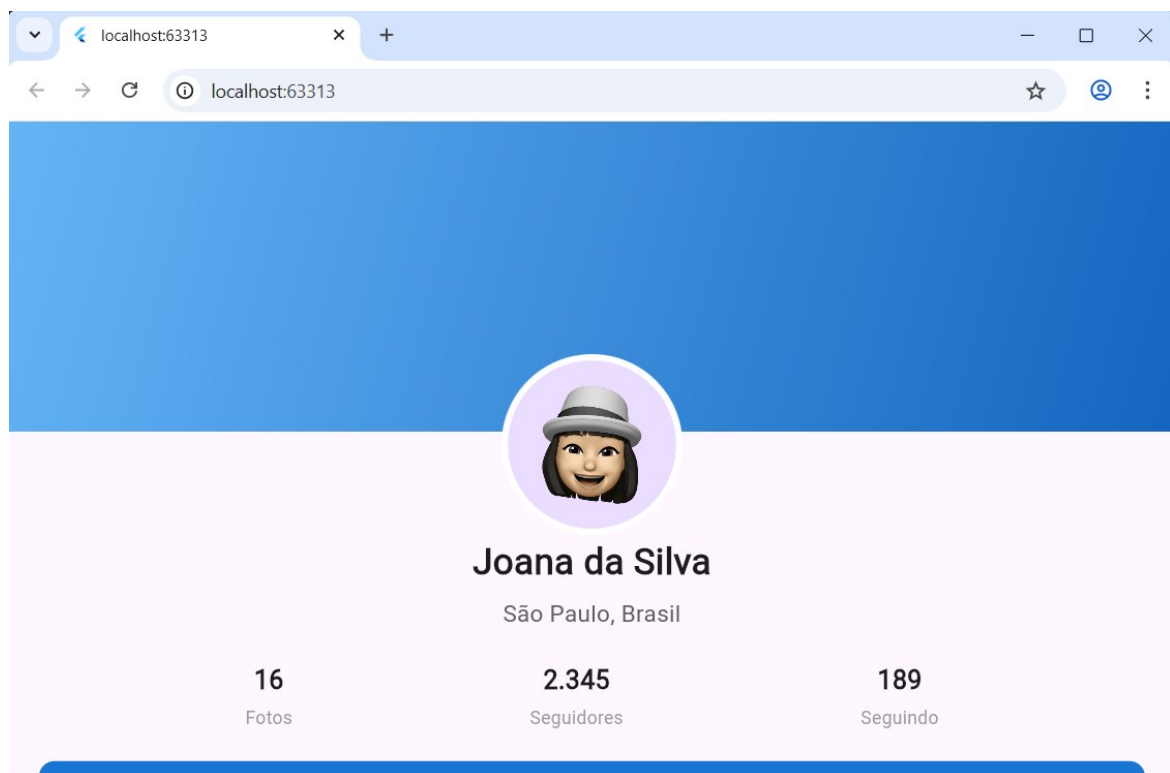
**Objetivo:** Criar uma tela de perfil estática para um aplicativo de viagens. Os alunos deverão aplicar os conhecimentos sobre widgets de layout (Stack, Column, Row), widgets de conteúdo (Container, CircleAvatar, Icon, Text) e estilização para recriar um layout a partir de uma especificação.

### Descrição do Layout a ser Construído:

A tela de perfil deve conter os seguintes elementos, organizados de cima para baixo:

1. **Banner Superior:** Uma imagem de fundo que ocupa a parte superior da tela (aproximadamente 30% da altura).
2. **Foto de Perfil:** Um avatar circular (CircleAvatar) que fica **sobreposto** entre o banner e o conteúdo abaixo. Metade dele fica sobre a imagem do banner e metade sobre a área de conteúdo branca.
3. **Informações do Usuário:** Abaixo da foto de perfil, deve haver o nome do usuário e sua localização (cidade/país), ambos centralizados.
4. **Estatísticas:** Uma linha (Row) contendo três estatísticas: "Fotos", "Seguidores" e "Seguindo". Cada estatística deve ter o número em destaque e o rótulo logo abaixo.
5. **Botão de Ação:** Um botão "Seguir" abaixo das estatísticas.
6. **Galeria de Fotos:** Uma grade simples de imagens quadradas para representar as fotos postadas pelo usuário.

### Visual de Referência:



## Requisitos e Dicas (Guia para os Alunos):

### Passo 1: Estrutura Inicial

- Crie um novo arquivo (ex: tela\_perfil.dart) ou modifique seu main.dart.
- A tela deve ser um StatelessWidget ou StatefulWidget.
- Utilize um Scaffold como base, mas desta vez, **não adicione uma AppBar** para que o conteúdo possa começar do topo da tela.

### Passo 2: Criando a Sobreposição com Stack

- O body do seu Scaffold deverá ter um Stack como widget principal.
- O primeiro filho do Stack será um Container para o banner. Dê a ele uma altura fixa (ex: 250.0) e uma cor ou, como desafio, uma imagem de fundo usando BoxDecoration.
- O segundo filho do Stack será a Column que conterá todo o resto do conteúdo.

**Dica:** Para que a Column com o conteúdo apareça na frente, ela deve vir *depois* do banner na lista de children do Stack.

### Passo 3: Posicionando a Foto de Perfil

- O primeiro item da sua Column de conteúdo será o CircleAvatar.
- Para que ele "suba" e fique sobre o banner, você pode envolvê-lo com um Padding e dar um padding **negativo** no topo (ex: EdgeInsets.only(top: -50.0)), ou usar um widget Transform.translate com um offset Y negativo. Uma abordagem mais simples é usar um Padding com um valor positivo no topo da Column inteira para "empurrar" todo o conteúdo para baixo, exceto o CircleAvatar que será o primeiro item.

### Passo 4: Adicionando as Informações e Estatísticas

- Dentro da Column (após o CircleAvatar), adicione os Text para nome e localização. Use SizedBox para criar espaçamentos verticais.
- Para as estatísticas, crie uma Row. O mainAxisAlignment dela deve ser MainAxisAlignment.spaceEvenly para espaçar os itens igualmente.
- Cada item dentro da Row será uma pequena Column com dois Text: o número e o rótulo.

### Passo 5: Botão e Galeria

- O botão "Seguir" pode ser um Container com altura, largura, cor, bordas arredondadas (BoxDecoration) e um Text centralizado dentro dele.
- Para a galeria, crie uma Row. Dentro dela, adicione 3 Containers quadrados. Para que eles se ajustem ao tamanho da tela, envolva cada um com um widget Expanded. Repita o processo para criar mais linhas de fotos.

### Dica:

O repositório alohe/avatars no GitHub é uma **biblioteca de avatares gratuitos e de código aberto**.

Em resumo, ele oferece uma coleção de imagens e ilustrações diversas que desenvolvedores e designers podem usar em seus projetos, principalmente como:

- **Fotos de perfil temporárias (placeholders):** Quando um novo usuário se cadastra em uma plataforma e ainda não enviou sua foto, o sistema pode usar um avatar dessa biblioteca.
- **Ilustrações para projetos:** As imagens podem ser usadas para dar mais personalidade e apelo visual a sites e aplicativos.

É uma maneira fácil e gratuita de adicionar um toque visual aos projetos sem precisar criar ou comprar imagens de avatares.

<https://github.com/alohe/avatars>

### Dart

```
// Importa o pacote principal do Material Design.
```

```
import 'package:flutter/material.dart';
```

```
// Ponto de entrada da aplicação.
```

```
void main() {
```

```
  runApp(const MyApp());
```

```
}
```

```
// Widget raiz da aplicação.
```

```
class MyApp extends StatelessWidget {
```

```
  const MyApp({Key? key}) : super(key: key);
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return MaterialApp(
```

```
      home: TelaPerfil(),
```

```
      debugShowCheckedModeBanner: false,
```

```
);  
}  
}
```

// Widget que constrói a nossa tela de perfil.

```
class TelaPerfil extends StatelessWidget {
```

```
  const TelaPerfil({Key? key}) : super(key: key);
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Scaffold(  
      body: ListView(  
        padding: EdgeInsets.zero,  
        children: <Widget>[  
          // TODO: ETAPA 1 - SOBREPOSIÇÃO  
          // Para sobrepor o banner e a foto de perfil, qual widget devemos usar aqui?  
          // Lembre-se que o primeiro filho será o fundo e os seguintes ficarão na frente.  
          // Dica: A aula 4 explicou sobre widgets que empilham.  
  
          /* Substitua este comentário pelo widget correto */(  
            clipBehavior: Clip.none,  
            alignment: Alignment.center,  
            children: <Widget>[  
              // 1. O BANNER SUPERIOR  
              Container(  
                height: 240,  
                decoration: BoxDecoration(  
                  gradient: LinearGradient(  
                    colors: [Colors.blue.shade300, Colors.blue.shade800],  
                    begin: Alignment.topLeft,  
                    end: Alignment.bottomRight,
```

```
),  
),  
),
```

```
// TODO: ETAPA 2 - POSICIONAMENTO DO AVATAR
```

```
// Precisamos de um widget que nos permita posicionar o CircleAvatar
```

```
// de forma precisa dentro do Stack. Qual widget faz isso?
```

```
// Ele deve ter uma propriedade para definir a distância do topo ('top').
```

```
// Calcule o valor do 'top' para que o avatar fique metade para fora do banner.
```

```
// Dica: alturaDoBanner - raioDoAvatar = 240 - 70 = 170. Tente um valor próximo.
```

```
/* Substitua este comentário pelo widget de posicionamento */(  
  top: 180, // Valor de exemplo, ajuste se necessário  
  child: CircleAvatar(  
    radius: 70,  
    backgroundColor: Colors.white,  
    child: CircleAvatar(  
      radius: 65,  
      backgroundImage: NetworkImage('https://i.pravatar.cc/150?img=5'),  
    ),  
  ),  
),  
],  
),  
  
  SizedBox(height: 80),  
  
// 3. INFORMAÇÕES DO USUÁRIO  
Text(  
  'Joana da Silva',  
  textAlign: TextAlign.center,
```

```
style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold),
),
SizedBox(height: 8),
Text(
  'São Paulo, Brasil',
  textAlign: TextAlign.center,
  style: TextStyle(fontSize: 18, color: Colors.grey.shade700),
),
SizedBox(height: 24),
// 4. ESTATÍSTICAS
Row(
  // TODO: ETAPA 3 - ALINHAMENTO DA ROW
  // Qual atributo usamos em uma Row para distribuir o espaço
  // horizontal igualmente entre seus filhos?
  // Dica: Vimos na aula 3 sobre alinhamento no eixo principal.
  /* Substitua este comentário pelo atributo de alinhamento correto */
  children: <Widget>[
    Column(
      children: [
        Text('16', style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold)),
        SizedBox(height: 4),
        Text('Fotos', style: TextStyle(color: Colors.grey)),
      ],
    ),
    Column(
      children: [
        Text('2.345', style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold)),
        SizedBox(height: 4),
        Text('Seguidores', style: TextStyle(color: Colors.grey)),
      ],
    ),
  ],
),
```

```
Column(  
  children: [  
    Text('189', style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold)),  
    SizedBox(height: 4),  
    Text('Seguindo', style: TextStyle(color: Colors.grey)),  
  ],  
,  
],  
,  
SizedBox(height: 24),
```

// 5. BOTÃO DE AÇÃO

```
Padding(  
  padding: const EdgeInsets.symmetric(horizontal: 24.0),  
  child: Container(  
    padding: EdgeInsets.all(16),  
    // TODO: ETAPA 4 - ESTILIZAÇÃO DO BOTÃO  
    // Como aplicamos cor de fundo e bordas arredondadas a um Container?  
    // Você precisará usar o atributo 'decoration' e um widget específico  
    // que vimos na aula 2.  
    // Dica: A cor deve ser `Colors.blue.shade700` e o raio da borda `12`.  
    /* Substitua este comentário pelo atributo de decoração correto */  
  
    child: Center(  
      child: Text(  
        'SEGUIR',  
        style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold, fontSize: 16),  
      ),  
    ),  
  ),  
,  
)
```

```
    SizedBox(height: 24),
```

```
// 6. GALERIA DE FOTOS
```

```
Padding(
```

```
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
```

```
  child: Row(
```

```
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```
    children: [
```

```
      // TODO: ETAPA 5 - LAYOUT FLEXÍVEL
```

```
      // Para fazer os 3 Containers da galeria dividirem o espaço da tela
```

```
      // igualmente, qual widget devemos usar para envolvê-los?
```

```
      // Dica: Vimos na aula 3 um widget que "expande" para preencher o espaço.
```

```
      /* Envolve o Container abaixo com o widget correto */
```

```
      Container(height: 120, color: Colors.grey.shade300),
```

```
      SizedBox(width: 8),
```

```
      /* Faça o mesmo para o segundo Container */
```

```
      Container(height: 120, color: Colors.grey.shade300),
```

```
      SizedBox(width: 8),
```

```
      /* E para o terceiro Container */
```

```
      Container(height: 120, color: Colors.grey.shade300),
```

```
    ],
```

```
  ),
```

```
);
```

```
],
```

```
),
```

```
);
```

```
}
```

```
}
```



### Desafio Extra (Para os alunos mais avançados):

1. **Imagens Reais:** Substitua as cores sólidas do banner e dos Containers da galeria por imagens da internet, usando Image.network.
2. **Botão Interativo:** Envolver o Container do botão "Seguir" com um widget GestureDetector e use a propriedade onTap para imprimir uma mensagem no console (ex: `print('Botão Seguir pressionado!');`).
3. **Componentização:** Crie um novo widget para o item de estatística (o número e o rótulo) e reutilize-o três vezes na Row de estatísticas, passando os valores como parâmetros.