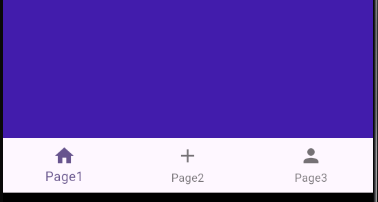
Temos a pagina que já fizemos, agora vamos navegar entre elas com botões



Vamos criar os botoes para nossa pagina, mas como criamos 3 paginas, devemos criar 3 botoes

body: Column(

          children: [

            Expanded(

              child: PageView(

                scrollDirection: Axis.vertical,

                children: const [Pagina1Page(), Pagina2Page(), Pagina3Page()],

              ),

            ),

            BottomNavigationBar(currentIndex: 0, items: [

              BottomNavigationBarItem(label: "Page1", icon: Icon(Icons.home)),

              BottomNavigationBarItem(label: "Page2", icon: Icon(Icons.home)),

              BottomNavigationBarItem(label: "Page3", icon: Icon(Icons.home))

            ], )

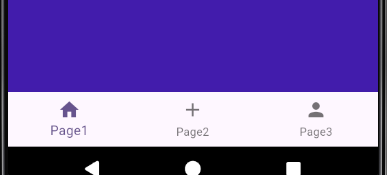
          ],

        ),

Aqui estamos passando o index da lista, e uma lista de items.

Que no caso essa lista de itens, vai ser uma lista de ícones para cada pagina.

Label é o nome de cada ícone que vai aparecer.

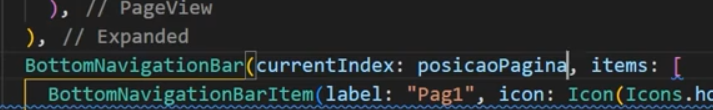


Agora para funcionar, devemos criar uma variável que vai guarda a posição de cada pagina.

class \_MainPageState extends State<MainPage> {

  int posicaoPagina = 0;

Porque sabemos que o currentIndex, ele vai abrir de acordo com a posição e para isso vamos passar a posição para ele.



Quando passamos a pagina, podemos capturar a posição da pagina

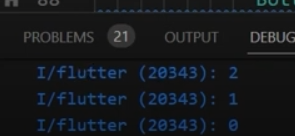
child: PageView(

                onPageChanged: (value) {

                  print(value);

                },

Dessa maneira



E com isso podemos alterar, dessa forma agora.

onPageChanged: (value) {

                  setState(() {

                    posicaoPagina = value;

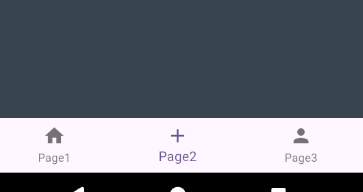
                  });

                },

Fazendo dessa forma e não podemos esquecer do setState que faz a pagina recarregar.

Agora vamos ver que o buttonNavigation vai entender em qual pagina estamos.

Repara que estamos na pagina 2



Repara o ícone

Agora se quiser mudar pelo botão?

Ele também trabalha com um onTap de inteiro

BottomNavigationBar(

              onTap: (value) {

                print(value);

              },

              currentIndex: posicaoPagina,

              items: [

                BottomNavigationBarItem(label: "Page1", icon: Icon(Icons.home)),

                BottomNavigationBarItem(label: "Page2", icon: Icon(Icons.add)),

                BottomNavigationBarItem(

                  label: "Page3",

                  icon: Icon(Icons.person),

                ),

              ],

            ),

Criamos também o para dizer em qual posição a pagina esta.

Colocamos a pagina inicial como 0

class \_MainPageState extends State<MainPage> {

  PageController controller = PageController(initialPage: 0);

Agora vamos colocar dentro da nossa pageView para dizer que quem esta controlando ela é nosso controller.

child: PageView(

                controller: controller,

Agora na parte do clique, vamos fazer uma alteração

Na parte do clique do bottomNavigation

BottomNavigationBar(

              onTap: (value) {

                controller.jumpToPage(value);

              },

              curre

Lembrar que no value, temos o valor da pagina.

E estamos mandando ele “Pular” para essa pagina.

Lembrar também que estamos usando o controller que esta controlando a pageView