

Basicamente é isso é um campo onde o usuário vai clicar e vai exibir um calendário para ele entrar com uma data.

Essa data podemos usar, para cadastro no servidor.

É um campo onde vamos colocar datas

Text("Data de nascimento"),

            TextField(readOnly: true),

Repara que aqui colocamos um texto pedindo a data de nascimento

E colocamos um textField para o usuário entrar com a data

(readOnly: true), >> Isso serve para o usuário não poder digitar nada no teclado do celular, na verdade o teclado nem aparece.

Agora mao no código

Primeiro criamos a variável que vai controlar a data de nascimento

class \_DadosCadastraisState extends State<DadosCadastrais> {

  TextEditingController nomeController = TextEditingController(text: "");

  var dataNascimentoController = TextEditingController(text: "");

Adicionamos ela la

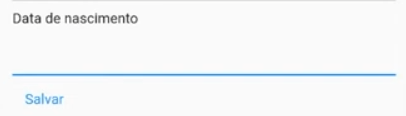
Text("Data de nascimento"),

            TextField(

              controller: dataNascimentoController,

              readOnly: true),

Como teste vamos pegar quando o usuário clicar aqui dentro.



Dessa caixa



Text("Data de nascimento"),

            TextField(

              controller: dataNascimentoController,

              readOnly: true,

              onTap: () {

                print("Foi clicado.");

              },

            ),

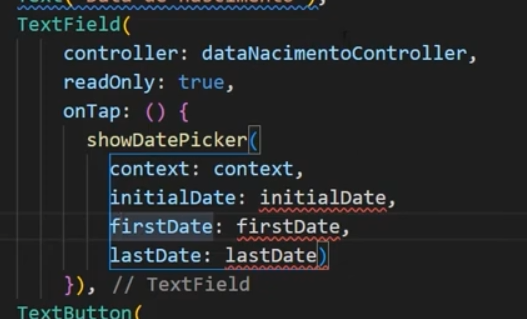
Agora quando o usuário clicar aqui, vamos exibir o componente de data.

Aqui vamos ter os parâmetros, que devemos deixar configurado

1 data inicial que vai ser aberto

2 Primeira data possível a ser selecionada.

3 a última data disponível



E vamos mudar e deve ficar dessa forma

Text("Data de nascimento"),

            TextField(

              controller: dataNascimentoController,

              readOnly: true,

              onTap: () {

                showDatePicker(

                  context: context,

                  initialDate: DateTime.now(),

                  firstDate: DateTime(1994, 5, 20),

                  lastDate: DateTime(2030, 10, 23),

                );

              },

            ),

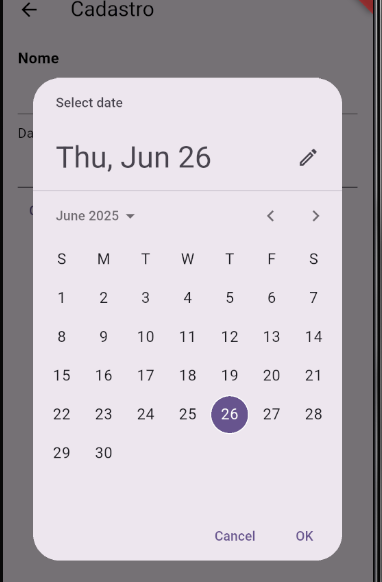
Pegando a data inicial

A primeira data que deve aparecer

E a última data disponível

\*\* Atenção a última data disponível não pode ser inferior a data inicial \*\*\*

Estava com esse erro, gerando uma exceção.



Agora atenção também devemos por um async e também o await

Para ele esperar a caixa fechar, ou seja quando a caixa de data fechar que você deve prosseguir.

Ficando dessa forma

Text("Data de nascimento"),

            TextField(

              controller: dataNascimentoController,

              readOnly: true,

              onTap: () async {

                await showDatePicker(

                  context: context,

                  initialDate: DateTime.now(),

                  firstDate: DateTime(1994, 5, 20),

                  lastDate: DateTime(2030, 10, 23),

                );

              },

            ),

Agora vamos recuperar a data que o usuário digitou e vamos exibir ela.

Text("Data de nascimento"),

            TextField(

              controller: dataNascimentoController,

              readOnly: true,

              onTap: () async {

                var dataNascimentoUsuario = await showDatePicker(

                  context: context,

                  initialDate: DateTime.now(),

                  firstDate: DateTime(1994, 5, 20),

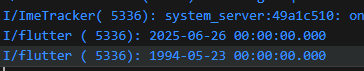
                  lastDate: DateTime(2030, 10, 23),

                );

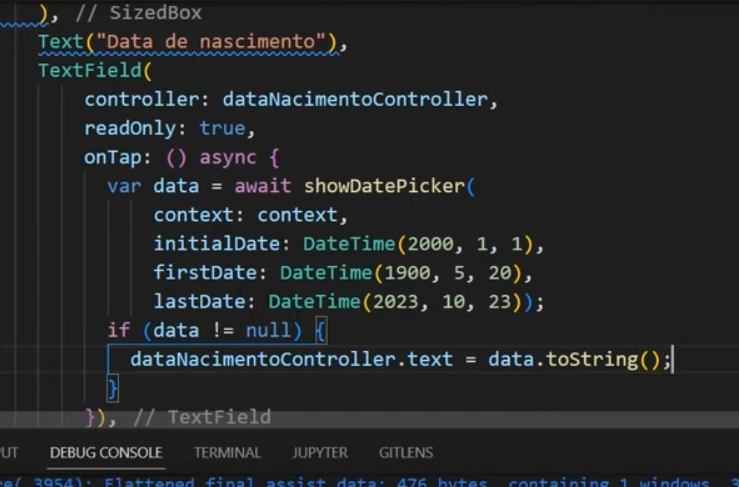
                print(dataNascimentoUsuario);

              },

            ),

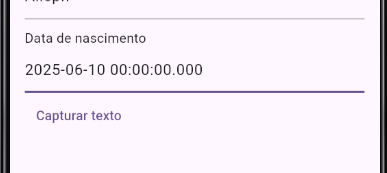


Agora fazendo teste, se por acaso o usuário selecionou uma data, essa data não vai ser nula



E ai o dataNascimentoController.text vai receber o data que o usuário selecionou.

Assim podemos printar direto no campo



Se for nula, ai fazemos dessa forma

Text("Data de nascimento"),

            TextField(

              controller: dataNascimentoController,

              readOnly: true,

              onTap: () async {

                var dataNascimentoUsuario = await showDatePicker(

                  context: context,

                  initialDate: DateTime.now(),

                  firstDate: DateTime(1994, 5, 20),

                  lastDate: DateTime(2030, 10, 23),

                );

                if (dataNascimentoUsuario != null) {

                  dataNascimentoController.text = dataNascimentoUsuario

                      .toString();

                } else {

                  dataNascimentoController.text = "Usuario nao informou a data";

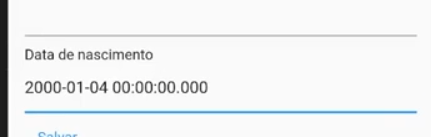
                }

              },

            ),

Repara que o controller ele serve para controlar esses campos.

Parei em 7 minutos.



Observa como esta a data.

Agora vamos armazenar ela em uma variável para poder exibir melhor

Primeiro criamos uma variável

DateTime? dataDeNascimento;

Colocamos o resultado do usuário dentro dela.

if (dataNascimentoUsuario != null) {

                  dataDeNascimento = dataNascimentoUsuario;

Agora vamos exibir ela.

if (dataNascimentoUsuario != null) {

                  dataDeNascimento = dataNascimentoUsuario;

                  dataNascimentoController.text = dataDeNascimento!.day

                      .toString();

                } else {

                  dataNascimentoController.text = "Usuario nao informou a data";

                }

              },

            ),

Nesse exemplo estamos exibindo apenas o dia

Mas podemos exibir também dia, mês e ano de forma separada.

Uma maneira de fazer é assim.

);

                if (dataNascimentoUsuario != null) {

                  dataDeNascimento = dataNascimentoUsuario;

                  var dia = dataDeNascimento!.day.toString();

                  var mes = dataDeNascimento!.month.toString();

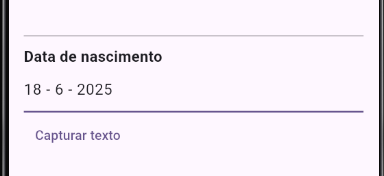
                  var ano = dataDeNascimento!.year.toString();

                  dataNascimentoController.text = "$dia - $mes - $ano";

                } else {

                  dataNascimentoController.text = "Usuario nao informou a data";

                }



Criando função de widgets para deixar alguns textos como padrão.

Primeira opção, criar uma função que retornar um widget

Text returnText(String texto) {

    return Text(

      texto,

      style: TextStyle(fontSize: 16, fontWeight: FontWeight.w700),

    );

  }

Repara que o nome da nossa função é returnText

Oque vamos retornar nela?

O próprio texto, mas com um stilo diferente.

Agora usando essa função.

Agora onde temos o

Text("Data de nascimento"),

Podemos fazer dessa forma.

antes



Depois



E assim podemos deixar padrão os nossos widgets de Textos

Essa é a vantagem de fazer isso, tudo fica padronizado.