Nesse exemplo, vamos colocar um campo para o usuário selecionar qual nível ele se enquadra.

Vamos imaginar que essa informação está vindo do banco de dados.

Para isso vamos usar o repository que é um pacote que informa que são dados que vem do banco de dados.

Criamos uma classe que vai retornar uma lista

class NivelRepository {

  List<String> retornaNiveis() {

    return ["Iniciando", "Intermediario", "Avancado"];

  }

}

Agora vamos na nossa tela “dados\_cadastrais.dart”

E vamos iniciar uma lista vazia.

class \_DadosCadastraisState extends State<DadosCadastrais> {

  TextEditingController nomeController = TextEditingController(text: "");

  var dataNascimentoController = TextEditingController(text: "");

  DateTime? dataDeNascimento;

  var niveis = [];

Vamos usar também o initState

Aqui vamos simular como se esses dados estivessem vindo do banco de dados, mas sabemos que ele esta dentro da classe.

class \_DadosCadastraisState extends State<DadosCadastrais> {

  TextEditingController nomeController = TextEditingController(text: "");

  var dataNascimentoController = TextEditingController(text: "");

  DateTime? dataDeNascimento;

  var nivelRepository = NivelRepository();

  var niveis = [];

  @override

  void initState() {

    niveis = nivelRepository.retornaNiveis();

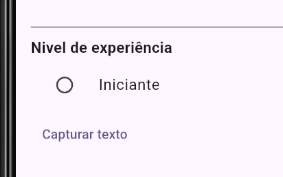
    super.initState();

  }

Aqui ele vai retornar os níveis que está dentro da minha lista, lá da classe.

Dentro da variável níveis, vai ter todos os níveis que estão disponíveis dentro da lista que adicionamos.

Agora vamos adicionar ele na tela.



            RadioListTile(

              title: Text("Iniciante"),

              value:

                  "Iniciante", // Aqui vai ser o valor do iniciante.

              groupValue: "nivel\_experiencia",

              onChanged: (value) {

                // value é o valor que o usuario clicou

                print(value);

              },

            ),



Só que aqui, estamos fazendo de forma estática, não estamos pegando do “banco de dados”

Lembrar que vamos ter 3 ou mais itens, para isso vamos criar um column

Explicando o código

1 criamos um column para ele adicionar um item a baixo do outro, e também criamos o column porque vamos precisar adicionar mais de um item na tela.

2 criamos um níveis.map

* Níveis é a variável que esta nossa lista
* Map é para transformar o map faz uma mudança, então ele vai pegar todos aqueles itens da lista e vai mudar para widgets

Atenção estamos colocando tudo isso em baixo do nível de experiência, que é um widget de texto.

Column(

              children: niveis.map((nivel) {

                return RadioListTile(

                  title: Text(nivel.toString()),

                  value: nivel.toString(), // valor único para cada opção

                  groupValue: nivelSelecionado, // valor atualmente selecionado

                  onChanged: (value) {

                    setState(() {

                      nivelSelecionado = value.toString();

                    });

                    print(value);

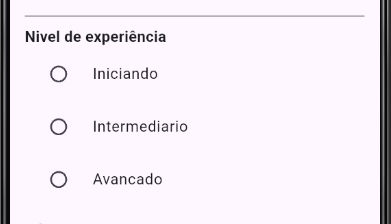
                  },

                );

              }).toList(),

            ),

Ficando dessa forma



Agora colocando o nível selecionado dentro de uma variável

class \_DadosCadastraisState extends State<DadosCadastrais> {

  TextEditingController nomeController = TextEditingController(text: "");

  var dataNascimentoController = TextEditingController(text: "");

  DateTime? dataDeNascimento;

  var nivelRepository = NivelRepository();

  var niveis = [];

  var nivelSelecionado = "";

Primeiro criamos a variável que vai receber esse item.

Depois vamos precisar fazer isso, para atualizar a pagina e inseri o valor la dentro da variável.

Column(

              children: niveis.map((nivel) {

                return RadioListTile(

                  title: Text(nivel.toString()),

                  value: nivel.toString(), // valor único para cada opção

                  groupValue: nivelSelecionado, // valor atualmente selecionado

                  onChanged: (value) {

                    setState(() {

                      nivelSelecionado = value.toString();

                    });

                    print(value);

                  },

                );

              }).toList(),

            ),

Atenção o selected é para ficar com o texto também em azul, igual da bolinha



E como esta sendo feito? Ele esta verificando se o valor(item da lista) é igual ao itemSelecionado se for ele deixa os 2 da mesma cor

O dense é para colocar tudo mais próximo, igual ai na foto, repara que diminuiu também a fonte.