

FLUTTER FLOW

APP DOS SONHOS

Autora: Terezinha Márcia de Carvalho Lino

**JOÃO PESSOA
2024**

SUMÁRIO

INTRODUÇÃO	6
1.1 Principais Recursos do Flutter Flow	6
1.2 Benefícios de usar Flutter Flow	6
1.3 Conexão do FlutterFlow com o Firebase	7
2. OBJETIVO.....	8
3. CADASTRO NO FLUTTER FLOW.....	9
3.1 Criar Novo Projeto.....	9
4. FIREBASE – BANCO DE DADOS	10
4.1 Criar conta e configurar o Firebase.....	10
4.2 Banco de Dados - fazer integração entre Firebase e FlutterFlow	11
4.2.1 Authentication (para fazer login no app)	11
4.2.2 Storage (para guardar as mídias utilizadas no app: imagens, vídeos etc)	11
4.2.3 Configurações.....	12
4.2.4 Volte ao FlutterFlow para terminar a integração	12
4.2.5 Firestore no FlutterFlow – Criar Coleção	12
4.2.6 Na engrenagem do Firestore	13
4.3 Criação de Coleção no Firebase.....	13
4.4 Corrigindo Erros Firebase	14
5. FLUTTERFLOW – APRESENTAÇÃO DA INTERFACE	16
5.1 Menu do lado esquerdo – Menu dos Elementos.....	16
5.2 Menu do lado esquerdo - Menu dos Componentes	16

5.2.1 Templates - terceiro ícone da barra de menu horizontal do lado esquerdo da tela.....	16
5.2.2 Widget Tree	16
5.2.3 Principais Widgets - Ferramentas	17
5.2.4 Storyboard	18
5.2.5 Firestore.....	18
5.2.6 DataTypes - App State.....	18
5.2.7 API Calls	19
5.2.8 Media Assets	19
5.2.9 Custom Code	19
5.2.10 Cloud Functions	19
5.2.11 Automated Tests.....	19
5.2.12 Theme Settings.....	19
5.2.13 Configurações.....	20
5.3 Menu Superior.....	20
5.3.1 FlutterFlow IA	20
5.4 Menu do lado direito.....	21
5.4.1 Primeiro Ícone: Propriedades	21
5.4.2 Segundo ícone: Actions	21
5.4.3 Terceiro Ícone: Backend Query	21
5.4.4 Quarto ícone - Animações	21
5.4.5 Quinto ícone: Documentação	21
6. CRIAÇÃO DE UM PROJETO.....	23
6.1 Elementos de página Fundamentos Linhas e Colunas	23
6.2 Criando Projeto do Zero.....	23
7. HOME PAGE.....	24

7.1 Home Page Primeiro Elemento da Tela - Propriedade	24
7.2 Visualizar dados em listas - Sonhos Atuais	26
7.2.1 Backend Query	26
7.2.2 Colocando os elementos da ListView de forma dinâmica - Sonhos Atuais	27
7.3 Action Navigation Back e Navigation To	27
7.4 Ver detalhes dos sonhos - Ir da HomePage para DetalhesSonhos	28
7.5 User - tornando o app multi usuário	28
7.5.1 Campo userRef na Coleção Sonhos.....	28
7.5.2 Atualizar a ListView co userRef - Entrar na Home Page.....	29
7.5.3 Dados Dinâmicos – Menu superior HomePage	29
OBS: Os sonhos criados antes de tornar o App multiusuário, não terão o campo userRef, no Firebase. Mas, você pode fazer isso manualmente:	29
7.5.4 Associar um Novo Sonho ao Usuário Logado	30
7.6 Passar dados da Home Page para DetalhesSonhos	30
8. PÁGINA DETALHES SONHO	32
8.1 Criar um PopUp	32
8.2 Propriedade do Componente DetalhesSonho.....	32
8.3 Colocando os elementos no PopUp.....	32
8.4 Criar uma AppState.....	33
8.5 Orçamento	37
8.6 Comentário	37
8.7 Sonho Realizado.....	38
8.7.1 Criar os Botões Sonho Realizado, Editar/Deletar	38
8.7.2 Botões Editar Deletar e Sonho Realizado – Visibilidade Condicional ...	38
8.7.3 UpDate.....	39

8.7.4 Botões Concluir Edição.....	40
8.7.5 Botão Deletar - UpDate.....	41
8.7.6 UpDate – Edição de Ano, Orçamento, Comentário	42
8.7.7 Botão Tornar um Sonho para este Ano - Update.....	43
9. PÁGINA NOVO SONHO	45
9.1.Criar a Página NovoSonho	45
9.2 Abrir Página NovoSonho, a partir da HomePage.....	46
9.3 Criar Novos Dados na Página NovosSonhos - Create.....	46
10. POPUP MEU PERFIL	47
10.1 UpLoad da Imagem do perfil.....	48
10.2 Colocar os dados da MeuPerfil dinâmicos	48
10.3 Botão de Logout – Ação.....	49
11. PÁGINA DE CADASTRO E LOGIN	50
12. AÇÃO DE CADASTRAR E DE FAZER LOGIN	51
12.1 Fazer o setup da coleção de usuário:	51
12.2 Ação de Cadastrar	51
13. PUBLICAR O APP NA WEB	53
REFERÊNCIA	54

INTRODUÇÃO

Flutter Flow é uma plataforma de desenvolvimento, sem código, que permite criar aplicativos móveis nativos para iOS e Android. Com o Flutter Flow, você não precisa saber codificar para criar aplicativos.

O Flutter Flow fornece uma interface visual e widgets de arrastar e soltar para projetar a interface do aplicativo, eliminando a necessidade de escrever o código você mesmo. O que torna o desenvolvimento de aplicativos mais rápido e acessível para não programadores.

1.1 Principais Recursos do Flutter Flow

- Ter biblioteca de widgets de arrastar e soltar para projetar a interface do aplicativo;
- Conectar aplicativos a bancos de dados, APIs, sem código;
- Permitir, por meio, da colaboração em tempo real, que as equipes construam juntas, os seus projetos;
- Ter planos gratuitos e pagos para atender diferentes necessidades;
- Publicar os App na Web e/ou em lojas oficiais de aplicativos;
- Desempenho nativo e UI com Flutter;
- Permitir personalizar os temas e design;
- Permitir inserir ações, lógica condicional, animações e muito mais sem código;

1.2 Benefícios de usar Flutter Flow

O Flutter Flow oferece vários benefícios importantes, que o tornam uma das melhores escolhas para a criação de APP.

- **Velocidade de Desenvolvimento:** uma das principais vantagens do Flutter Flow é a rapidez com que você pode construir um aplicativo. A interface intuitiva de arrastar e soltar permite criar aplicativos completos sem a

necessidade de escrever código. O que agiliza e acelera o processo de desenvolvimento.

- **Suporte multiplataforma:** Flutter Flow permite criar aplicativos compilados nativamente para iOS e Android a partir de uma única base de código.
- **Grande biblioteca de widgets:** Flutter Flow fornece acesso a toda a biblioteca de widgets Flutter enquanto usa uma interface visual, como gráficos interativos, mapas, animações, arrastando e soltando em vez de codificação.

1.3 Conexão do FlutterFlow com o Firebase

Para conectar sua UI (UI, ou User Interface: como serão feitas as interações entre pessoas e o aplicativo) a um banco de dados back-end pode ser feita por meio do Firebase, Supabase ou outros Banco de Dados.

As configurações para o uso do Firebase. No desenvolvimento deste trabalho as configurações para conectar ao Firebase serão descritas passo a passo.

Principais Ferramentas do Firebase:

- **Banco de Dados:** gestão de dados do App;
- **Autenticação:** Login, Cadastro e controle seguro dos usuários do App;
- **Armazenamento:** storage para upload e gestão das mídias do aplicativo;
- **Ações de Servidor:** Ações que serão executadas diretamente no servidor. Pode-se, por exemplo, agendar ações sem depender de ações do usuário;
- **Webhook:** um gatilho que dispara uma determinada ação entre as aplicações. Exemplo: pagamentos on line.

2. OBJETIVO

Será criado o APP dos Sonhos, seguindo as orientações do professor Matheus Castelo, da NoCode Start Up.

O App foi idealizado para registrar os sonhos atuais, sonhos futuros e os sonhos realizados do usuário. Será um app multiusuário, isso significa que o usuário conectado poderá registrar os seus sonhos e apenas ele poderá ter acesso aos mesmos.

3. CADASTRO NO FLUTTER FLOW

- Entre no site do FlutterFlow: <https://flutterflow.io/>;
- Faça o seu cadastro, para criar a conta;
- Explorando o menu lateral do FlutterFlow:
 - Projects: área onde os seus projetos serão criados;
 - My Organization: a empresa pode inserir vários usuários na criação de um mesmo projeto. Mas, esse plano é pago;
 - Marketplace: são disponibilizados inúmeros templates. Alguns são pagos, outros não;
 - Resources: estão disponíveis documentação do FlutterFlow, links de ajuda, tutoriais e a Comunidade FlutterFlow;
 - Você pode desenvolver seus APPs na conta free, só quando precisar publicar nas lojas oficiais ou fazer download do código, será necessário migrar para conta Pro.

3.1 Criar Novo Projeto

- No FlutterFlow, clique em + Create New;
- Dar no nome do Projeto;
- Você escolher um Template;
- Ou Create Blank;
- Next Step;
- Colocar id do projeto criado no Firebase (crie o projeto no Firestone e depois coloque o id e faça a conexão);
- Start Building;
- O projeto está criado, mostrando sua primeira página.

4. FIREBASE – BANCO DE DADOS

Neste projeto o **Frontend** será desenvolvido utilizando os recursos do FlutterFlow. Já o **Backend** pode ser desenvolvido utilizando as ferramentas de Backend Low Code (exige conhecimento em programação) integradas nativamente com o FlutterFlow: Firebase e Supabase ou por Ferramentas Backend NoCode: Xano e Airtable. Neste projeto será utilizado o Firebase.

4.1 Criar conta e configurar o Firebase

- Entre no site, para criar sua conta: <https://firebase.google.com/>
- **Criar projeto:**
 - Dê o nome para o projeto, continuar;
 - Desative o Google Analytics, (só ativar, quando o projeto for publicado na lojas Oficiais e começar a vender), continuar;
- Depois que foi criado, continuar.
- **No menu lateral esquerdo, clique em criação;**
- Firestore Database;
- Criar um banco de dados;
- Escolher a região onde estará hospedado seu banco de dados, escolha South America East1(São Paulo);
- Iniciar no modo de teste. Nesse modo o tempo de atualização das regras do banco de dados é de 30 dias. Após isso, você precisará atualizar as regras, para ter acesso ao Banco de Dados;
- Avançar;
- Ativar;
- Depois que ativar aparece a tela da Cloud Firestore com as seguintes abas: Dados (onde serão criadas as coleções), Regras (aqui é que se deve atualizar as regras, depois de um mês, para garantir a segurança do seu Banco de Dados), Índice, Uso e Extensões;
- 1. Volte na engrenagem e clique em Usuários e Permissões;
- Você tem que colocar o Flutter Flow também seja membro;
- Coloque o email: firebase@flutterflow.io;
- Papéis: coloque editor;

- Adicionar membro;
- Clique na opção: configurações avançadas de permissão;
- Vai abrir uma nova janela;
- Procure: firebase@flutterflow.io, clicar para editar;
- Abre uma nova janela:
- Adicionar outro papel: procure por Usuário da Conta de Serviço e clique sobre ele, para ser adicionado;
- Adicione outro papel: procure por Administrador do Cloud Functions e clique sobre ele, para ser adicionado.
- Salvar;
- Feche a página.

4.2 Banco de Dados - fazer integração entre Firebase e FlutterFlow

4.2.1 Authentication (para fazer login no app)

- Entrar no Firebase;
- Escolher o seu projeto;
- Authentication;
- Vamos começar;
- O Firebase oferece vários tipos de provedores para fazer a autenticação. Vamos escolher email/senha;
- Ativar: email/senha;
- Salvar.

4.2.2 Storage (para guardar as mídias utilizadas no app: imagens, vídeos etc)

- Clicar em Storage;
- Vamos começar;
- Iniciar no modo teste;
- Próxima;
- Escolha o local do provedor: SouthAmerica-São Paulo;
- Concluir.

4.2.3 Configurações

Configurações do projeto. Já estão criados: Nome do projeto; Código do Projeto (este é o id que será colocado no FlutterFlow); Número do projeto.

Entre na Aba **Usuários e Permissões** para adicionar o FlutterFlow, assim ele poderá ter acesso ao Banco de Dados:

- Adicionar membro;
- Coloque o e-mail: firebase@flutterflow.io
- Papel: editor;
- Concluir;
- Adicionar membro
- OBS: cuidado, qualquer erro na digitação do e-mail, impedirá a integração do Banco de Dados.

4.2.4 Volte ao FlutterFlow para terminar a integração

- Escolha o seu projeto;
- Entrar em Configurações;
- Firebase (antes copie o Código do Projeto lá no site do Firebase)
- Cole no espaço Firebase Project Id o código do Projeto gerado pelo Firebase;
- Connect;
- Clique em Auto Generate Config Files, para o Firebase gerar os arquivos do app, automaticamente;
- Clicar em Deploy.

4.2.5 Firestore no FlutterFlow – Criar Coleção

Criar uma coleção: Sonhos

- Clique em Firestore (menu lateral);
- Create Collection;
- Name: Sonhos;

- Pergunta se é uma subcoleção (o FlutterFlow só permite trabalhar com uma coleção e uma subcoleção);
- Create;
- Na próxima tela, escolher: Choose template collection ou Start from scratch ou Generate with AI;
- Coloque os campos iniciando em minúsculo, sem acento;
 - nome: string;
 - comentario: string;
 - imgUrl: imagePath;
 - ano: interger;
 - orcamento: interger
 - status: string

4.2.6 Na engrenagem do Firestore

- Vá até as regras e coloque Everyone para todas as opções;
- Desmarque também os 3 últimos quadradinhos;
- Clique em deploy (em produção);
- Deploy now.

4.3 Criação de Coleção no Firebase

As coleções podem ser criadas no site do Firebase e também no Firebase, dentro do FlutterFlow.

Segue um exemplo de criação de coleção no Firebase:

- Clicar em iniciar coleção;
- Código da coleção: sonho (coloque o nome que identifique a atividade que será cadastrada);
- Id do Documento: é criado automático;
- Campo: coloque o nome do campo: ex: nome;
- Tipo: string;
- Valor: coloque o nome do seu sonho: ex: Casa de praia;
- Adicionar novos campos, até o necessário;

- Campo: status;
- Tipo: string;
- Valor: Sonho Atual;
- Campo: orcamento;
- Tipo: number;
- Valor: 100.000;
- Salvar.
- OBS: O Firebase permite colocar uma coleção dentro de uma coleção (subcoleção).
- Authentication
- Clicar em authentication;
- Vamos começar.

4.4 Corrigindo Erros Firebase

Depois de 30 dias de construção do APP, o Firebase começa a emitir mensagem para mudar as regras de acesso no Banco de Dados. Siga o procedimento abaixo para fazer isso:

- Entre no Firebase;
- Firestore Database;
- Regras;
- Percebe que aparece uma última regra imposta pelo Firebase. Basta atualizar. Você pode fazer pelo Firebase ou pelo Flutter Flow.

Atualizando regras pelo Flutter Flow

- Entre no FutlerFlow;
- Banco de dados (Firestore);
 - Firestore Setting;
 - Aparecem as regras, coloque: everyone para todas as colunas;
 - Desmarque todos os quadradinhos, de dados sensíveis;
 - Dar um deploy (para colocar as novas regras no Firebase).

Atualizando o Banco de Dados do Projeto

- No Flutter Flow, vá em engrenagem;
- Firebase;
- Clicar em Regenerated Config Files

5. FLUTTERFLOW – APRESENTAÇÃO DA INTERFACE

5.1 Menu do lado esquerdo – Menu dos Elementos

O menu de elementos é o primeiro ícone da barra de menu horizontal do lado esquerdo da tela. Os elementos estão agrupados da seguinte maneira:

- **Commonly Used Elements:** elementos que são bastante usados: text, column, row, container, image, button;
- **Layout Elements:** Container, Stack, Card, ListView, GridView, Divider, PageView, Carousel, Wrap etc;
- **Base Elements:** Text, Rich Text, Button, Ico, IconButton, ListTile, Calendar, VideoPlayer, Image, CircleImage, YouTuberPlayer.
- **Page Elements:** AppBar, FAB (botão, que coloca no canto da página), Drawer (menu que é usado nas laterais);
- **Form Elements:** elementos que são usados para construir formulários: TextField, DropDown, RadioButton etc.

5.2 Menu do lado esquerdo - Menu dos Componentes

O Menu dos Componentes pode ser encontrado no segundo ícone da barra de menu horizontal do lado esquerdo da tela. Você consegue criar componentes do zero

5.2.1 Templates - terceiro ícone da barra de menu horizontal do lado esquerdo da tela.

- Tem alguns templates que podem ser usados para acelerar o processo de criação do projeto;
- Templates em dark mode, light mode e outros.

5.2.2 Widget Tree

- Mostra a organização em pastas, dos elementos que estão sendo utilizados;
- Para adicionar um novo elemento, basta clicar no ícone e escolher o elemento. Cada página tem a sua Widget Tree.

5.2.3 Principais Widgets - Ferramentas

São várias as opções. Quando tiver dúvida sobre a utilização de cada uma, consulte a documentação do FlutterFlow. Alguns serão evidenciados aqui:

- **Linhas e Colunas: Colunas** Quando as informações são organizadas uma embaixo da outra. Sempre comece a organizar uma página usando uma coluna. **Linhas:** quando as informações são organizadas uma ao lado da outra.
- **Container:** elemento que serve para organizar outros elementos dentro do APP. Definir a altura e largura em pixel ou porcentagem em relação ao espaço total. Pode deixar com a dimensão infinita, mas deve-se tomar cuidado, pois as linhas e colunas que estarão dentro dele vão tentar se adaptar a essa dimensão e provavelmente vai dar erro. Colocar cor, bordas, bordas arredondadas. Definir os espaçamentos. Definir a posição do container depende do elemento pai. Se o contêiner estiver dentro de uma coluna ou outro container, procure o elemento pai e defina a posição do elemento child. Colocar Elevation para evidenciar as bordas. Usar Box Shadow para sombreamento das bordas. Se preferir duas cores use o Gradient. Pode-se colocar também imagem no Background. Alinhar Child: se colocar um elemento dentro do elemento, é por aqui que se controla o seu alinhamento.
- **ListView:** é muito usado para colocar informações do Banco de Dados, quando precisar repetir os dados, de acordo com os que estão no Banco de Dados. Basta colocar um container dentro da ListView que os dados serão replicados, como uma lista.
- **GridView (visualização em grades):** colocando container dentro do gridview, dá para organizar cards um ao lado do outro.
- **TabBar:** barra de abas: serve para dividir a tela em 3 abas. Cada aba terá uma visualização diferente.
- **PageView:** para adicionar uma imagem na página.
- **Carousel:** para adicionar uma coleção de imagens.
- **Form Validation:** para validar formulários, basta colocar os elementos do formulário dentro dele.

- **Wrap:** para os container se acomodarem dentro das colunas.
- **CheckBox:** caixa para mostrar que algum item foi checado;
- **Switch:** caixa para trocar uma ação. Por exemplo: visualizar ou não.
- **Timer:** contador de tempo.
- **PDF View:** para visualizar um texto em pdf;
- **AppBar:** para colocar uma barra de navegação no App. Mas, o seu uso deixa o layout um pouco engessado, pois não permite inserir outros elementos.
- **FAB:** Botão que fica no final da página. Para ter acesso a este elemento, tem que clicar no nome da página, assim ele fica visível para colocar. Não aceita colocar dentro de uma coluna, por exemplo.
- **Drawer:** serve para ter um menu que recolhe e se torna visível quando clicado;
- **Stack:** para organizar um elemento sobre o outro:
 - Criar página;
 - Acrescentar uma coluna;
 - Acrescentar dentro da coluna o stack;
 - Colocar dentro do stack, 3 container. Eles ficarão sobrepostos. Para mudar a propriedade de cada um, clique neles para identificar. Para mudar a posição, clique no container e arraste para o local desejado.

5.2.4 Storyboard

Aqui você consegue visualizar quais páginas se conectam com outras páginas.

5.2.5 Firestore

Banco de Dados do Firebase, onde se faz o controle do banco criado.

5.2.6 DataTypes - App State

- A Appstate é como se fosse o cache de um site (evita que o app fique buscando dados no Banco de Dados, dessa forma economiza espaço);
- Uma variável AppState pode ser acessada e modificada em todo o aplicativo, em todas as páginas e componentes;

- Esse tipo de variável pode ser útil para armazenar dados que precisam ser compartilhados entre diferentes partes do aplicativo, como preferências do usuário e tokens de autenticação;
- Os dados salvos no appstate ficam armazenados de forma temporária. Quando o usuário sai do aplicativo aquele dado não é salvo. Então não deve ser usado para armazenar dados que precisam ficar pra sempre no banco de dados. Um exemplo: página de Consulta de CEP, dados de localidade podem ser armazenados em App State. Outro exemplo: Lista de agendamento de horário de um médico. Aqueles horários ficarão armazenados em App State. Pois, serão temporários.

5.2.7 API Calls

Onde se faz as chamadas às APIs, por exemplo: API do CEP.

5.2.8 Media Assets

Onde se faz a gestão das mídias para tela. Pode-se fazer o upload de uma foto que sempre será utilizada no APP, assim ela ficará disponível. **OBS: a foto de perfil não será guardada aqui, e sim no Banco de Dados.**

5.2.9 Custom Code

Pode criar Custom Functions, Custom Widget, Custom Actions, Custom Files.

5.2.10 Cloud Functions

Desenvolve funções para rodar na nuvem do Firebase.

5.2.11 Automated Tests

Para fazer testes automatizados.

5.2.12 Theme Settings

Onde define as fontes, tamanho da fonte, cores e design que serão usados no projeto.

5.2.13 Configurações

Diversas configurações do APP são definidas aqui. Por exemplo, quando o APP tiver entrada de usuário tem que habilitar no App Setting o **Authentication**.

5.3 Menu Superior

- Ícones de celular, tablet e desktop: clique nos ícones, para verificar se o APP está responsivo;
- Ícone do Besouro: se estiver laranja indica que tem algum erro, clique para saber qual o erro;
- Ícone de comentário;
- Ícone </>, clique para visualizar o código;
- Ícone da seta: quando tiver alguma dúvida sobre o desenvolvimento do projeto, você pode compartilhar o mesmo, clicando em Make Public e depois, basta copiar o link e compartilhar;
- Ícone do olho: para visualizar a parte de design do APP. Aqui não consegue ver a inteligência, conexão com o Banco de Dados;
- Ícone do Raio: Abre o modo teste do APP para ver o funcionamento do mesmo.

5.3.1 FlotterFlow IA

- Na tela central do Flutter, no menu superior clique no ícone de AI Generated Page, para criar uma nova página pela IA;
- Para criar um novo componente pela IA, clique no ícone +, New Component, Create with AI.
- No menu lateral à esquerda, vá até Theme Setting Colors e clique em Generate with AI, para gerar uma paleta de cores.
- No menu lateral, clique no ícone de Banco de Dados Firestore, clique para criar uma nova coleção, dê o nome da coleção e clica, aparece a opção de Generate with AI:
 - Entre no Firestore, Clique em criar nova coleção;
 - Clique em Generate AI;

- Escreva o nome da nova coleção. Deve-se colocar o nome em inglês, pois em Português está dando erro;
- Dar enter para criar a nova coleção. Se não estiver de acordo, pode-se repetir o processo que é criada uma nova coleção.

5.4 Menu do lado direito

Mostra as ferramentas para mudar as propriedades do elemento selecionado. A seguir será feita uma breve apresentação dessas ferramentas.

5.4.1 Primeiro Ícone: Propriedades

- Visibilidade condicional: você pode impor uma condição para que o elemento seja visualizado;
- Responsividade: escolhe se o elemento será visível, apenas no celular, tablet, desktop;
- Padding: controlar os alinhamentos;
- Propriedades: define a largura, altura, font, cor, bordas, sombreamento da borda, etc.

5.4.2 Segundo ícone: Actions

Aqui são definidas as ações para o elemento selecionado. Por exemplo: selecione um botão e coloque a ação de levar para outra página. Você pode colocar várias ações em sequência.

5.4.3 Terceiro Ícone: Backend Query

Aqui você vai fazer uma consulta ao Banco de Dados, faz a conexão do FlutterFlow com o Banco de Dados – Firebase.

5.4.4 Quarto ícone - Animações

Você pode colocar alguma animação no elemento selecionado.

5.4.5 Quinto ícone: Documentação

Você pode fazer toda a documentação do seu APP, para futuras consultas. Do lado do nome do elemento, estão 3 ícones, o primeiro é para salvar a

configuração do componente, o segundo para salvar o widget, para poder ser utilizado em outros projetos.

6. CRIAÇÃO DE UM PROJETO

6.1 Elementos de página Fundamentos Linhas e Colunas

Colunas

- Quando as informações são organizadas uma embaixo da outra;
- Sempre comece a organizar uma página usando uma coluna.

Linhas

- Quando as informações são organizadas uma ao lado da outra.

6.2 Criando Projeto do Zero

Dar nome ao projeto;

- Clicar em Create Blank;
- Project Setup Step – Clicar em Start Building. Depois disso, aparecerá a primeira página (chamada de canva) do seu projeto. Ela estará em branco, pronta para ser trabalhada;
- No menu à esquerda da Plataforma estão os elementos que podem ser colocados na página. No menu do lado direito, você utiliza para configurar o elemento selecionado.

7. HOME PAGE

7.1 Home Page Primeiro Elemento da Tela - Propriedade

- **Background Color:** escolha a cor da tela;
- **Safe area:** deixar habilitado, é o espaço no entorno da página;
- **Hide Keyboard da tela on tap:** deixar habilitado;
- **Disable Android Back Button:** deixar desabilitado;

OBS: Comece a criar a página colocando uma coluna. O espaço que fica no topo, é o espaço de segurança, conforme configurado.

- **Coluna Principal:** Deixar Scrollable: allow scrolling (seta), para dar a opção de rolar as listas para baixo.
- **Cabeçalho** (o cabeçalho pode ser colocado dentro da AppBar);
- **AppBar:** Deletar a AppBar que vem com a página. Clicar no nome da página. Procurar o elemento: AppBar e clicar. Na AppBar não tem como inserir elementos diretos, como fazer isso:
 - Clicar em Widget Pallet (na parte superior do menu lateral) e arraste um container para a AppBar. Vão aparecer três espaços, coloque o container no primeiro do lado direito. Insira nesse container uma imagem circular.
 - Faça o mesmo processo e insira outro container, deixe sem altura e largura, para ele se adaptar ao tamanho do elemento pai.
 - Insira dois textos e escreva: Que bom ter você por aqui; no outro: Márcia Lino.
- **Banner**
 - Inserir na coluna principal outro container com o nome de Banner. Largura 100%, altura: 100px.
 - Fazer o upload da imagem: ir até Media Assets; Upload Media e carregue a imagem do banner;
 - Ir para as propriedades e escolher background image e escolher a imagem do banner. Fazer os ajustes do tamanho, manualmente;

- Retirar a cor de fundo do container.
- **Listas de Lugares - Sonhos Futuros**
 - Inserir novo container na coluna principal. Dar espaçamento de 16 em todos os lados, Renomear o container de SonhosFuturos. Largura 100%, altura: 200px; retirar cor de fundo.
 - Inserir uma coluna dentro do container. Alinhar os elementos dentro da coluna à esquerda. Inserir na coluna: text e uma listview;
 - ❖ Clicar na listview para configurar: Axis: horizontal (para colocar os cards na horizontal), para isso funcionar, clique no último botão de expansion, a lista via expandir o máximo possível;
 - ❖ Inserir dentro da listview um container, largura: 120px, a altura não tem como mudar, pois está acompanhando a altura do container pai. Se precisar mudar, mude no pai. Dê o nome ao container de Sonho. Arredondar bordas: 12; Padding em todos os cantos: 12; Colocar sobre: habilitar o box shadow.
 - ❖ Dentro do Container, colocar uma image, com altura de 110px, com os cantos superiores arredondados: 12 . Coloque a imagem do seu sonho. Título do sonho: text: coloque o nome do seu sonho.Vai perguntar se quer organizar em linha ou coluna: clicar em coluna;
 - ❖ Duplicar o container Sonho: clicar no container Sonho, Control C e Control V na ListView. Faça isso duas vezes.
 - ❖ Clicar no texto para configurar:
 - ❖ **Listas de Lugares - Sonhos desse ano.** Ir até a o container de SonhosFuturos, copiar e colocar na coluna principal. Renomear como: Sonhos desse ano.
 - ❖ **Listas de Lugares - Sonhos Realizados** Ir até a o container de SonhosFuturos, copiar e colocar na coluna principal. Renomear como: Sonhos Realizados.
- **Botão Flutuante (FAB) na parte de baixo do App**
 - Ir para o nome da página;
 - Procurar Page Elements;
 - FAB e inserir

Para colocar ação no Botão FAB, primeiro tem que criar a página DetalhesSonho.

7.2 Visualizar dados em listas - Sonhos Atuais

- Entrar na Home Page;
- No container Sonhos Atuais, têm três container com o nome Sonho. Delete os dois últimos, pois, quando fizer a conexão da ListView com o Banco de Dados, tornando-a dinâmica, eles irão se multiplicar. Então, a lista irá se adaptar ao número de sonhos registrados pelo usuário no Firebase;

7.2.1 Backend Query

Consulta ao Banco de Dados - ação que faz o elemento se conectar com o Banco de Dados.

Colocar uma Backend Query nas ListView:

- Clicar na ListView para criar uma BackEnd Query;
- Clicar na ListView;
- No menu lateral direito, clique no ícone da BackEnd Query (do lado do ícone da Action);
- Add Backend Query;
- Query Type;
- Unset;
- Query Collection;
- Unset;
- sonhos;
- List of Document;
- Limit: deixar em branco, quando quiser pegar todos os elementos da lista;
- Filter: escolher Status; Unset: Equal To; Unset: Quando a tela abrir clique no lixo, para tirar a opção de escolha. Volte ao espaço e escreva: Sonhos Atuais; Confirm

- Ordering: não tem;
- Não habilitar os dados seguintes;
- Confirm;
- Confirm.

7.2.2 Colocando os elementos da ListView de forma dinâmica - Sonhos Atuais

- **Nome - Texto abaixo da imagem**
 - Clicar no texto;
 - Clicar no ícone do lado do text;
 - sonhos Document;
 - nome
- **Imagem**
 - Clique na imagem;
 - Clicar no ícone do lado do Path;
 - sonhos Document;
 - imgUrl;
 - Default Variable Value: colocar aqui uma imagem padrão, ela será mudada automaticamente, quando o usuário registrar a imagem do seu sonho.

OBS: Repetir o mesmo procedimento para a ListView de Sonhos Futuros e de Sonhos Realizados, observando que no Filter colocar: Sonhos Futuros e Sonhos Realizados.

7.3 Action Navigation Back e Navigation To

- Clicar no ícone Voltar;
 - Action;
 - Add Action;
 - On Tap;
 - Navigation Back.

7.4 Ver detalhes dos sonhos - Ir da HomePage para DetalhesSonhos

- Antes desta ação, você precisa criar a página DetalhesSonho;
- Entrar na Home Page
- Clicar no container Sonho, logo abaixo da ListView;
- Action;
 - Add Action;
 - Botton Sheet;
 - Show,
 - Selection Component;
 - Detalhes Sonho
 - Botton Sheet (para abrir o popup);
 - Show;
 - Background Color: texto primário, com um pouco de transparência;

OBS: Fazer isso para os Sonhos Futuros, Sonhos Atuais e Sonhos Realizados.

7.5 User - tornando o app multi usuário

7.5.1 Campo userRef na Coleção Sonhos

- No FlutterFlow, clicar em Firestore;
- Coleção Sonhos;
- Adicionar um novo campo:
 - Field Name: userRef;
 - Data Type: Document Reference;
 - Reference Type: Users;
 - Check.
- No Flutter Flow, entrar no Firebase;
- Criar uma nova coleção: users (aceitar a sugestão do firebase)

7.5.2 Atualizar a ListView co userRef - Entrar na Home Page

- ListView de SonhosFuturos;
- Edit Backend Query (coloque depois da query existente)
- + Filter;
- Unset;
- userRef (para mostrar apenas os sonhos do usuário que está logado);
- Equal to;
- Unset: Authenticated User;
- User Ref;
- Confirm;
- Confirm.
- Repita a última edição para os SonhosAtuais e SonhosRealizados;
- Para testar saia do app e crie um novo usuário, cadastrando pra eles alguns sonhos.

7.5.3 Dados Dinâmicos – Menu superior HomePage

- Entrar na HomePage;
- Clicar no nome, abaixo do texto: Bem-vindo...
- Propriedades;
- Text;
- Unset;
- Authenticated User;
- Display name.

OBS: Os sonhos criados antes de tornar o App multiusuário, não terão o campo userRef, no Firebase. Mas, você pode fazer isso manualmente:

- Entre no Firebase;
- Users e copie o id do usuário cadastrado;
- Clicar nos sonhos cadastrados e acrescente um novo campo: userRes;
Tipo: reference; Caminho: coloque o id copiado, tipo este:
/users/iN5ja3qPJfYih7maSjq4ltD2pEV2;
- Adicionar (desta forma este sonho fica associado ao usuário e questão).

7.5.4 Associar um Novo Sonho ao Usuário Logado

- Entrar na Página NovoSonho;
- Clicar no botão: Criar um novo Sonho;
- Clicar em Action (já tem 3 actions criadas)
- Open;
- Na primeira Action:
 - Add Field;
 - userRef;
 - Unset: Authenticated User;
 - User reference (userRef);
 - Confirm

7.6 Passar dados da Home Page para DetalhesSonhos

Para fazer este procedimento a página DetalhesSonhos deve estar criada.

Quando o usuário clicar no Sonho que está na Home Page, ele será levado para a página de DetalhesSonhos, onde serão visualizados os detalhes que ele descreveu sobre os seus sonhos.

Criar Parâmetro

- Abrir a Home Page;
- SonhosFuturos;
- Sonho;
- Action: abrir a action que já foi criada;
- Clicar em Parameters Define (passar o parâmetro, indicando qual é o ID, qual é o sonho, para que a página DetalhesSonho possa carregar aquele sonho.
- Add Parameter;
- Parameter Name: sonhoRef (dê um nome);
- Unset: Document Reference (pois, está passando uma referência do documento);
- Is List: não habilitar;
- Required: habilitar;
- Unset: sonhos;

- Confirm;

Passar o Parâmetro

- Clicar em Pass;
- Unset: sonhosDocument;
- Reference(sonhosRef);
- Confirm.

OBS: Repetir esse procedimento para o SonhosAtuais e SonhosRealizados.

8. PÁGINA DETALHES SONHO

8.1 Criar um PopUp

Criar um popup no lugar da página. Quando for preencher algum dado, a página inicial leva ao pop up.

- Clique na WidGet Tree;
- Clicar em componentes (ícone do diamante);
- New Component;
- Create blank;
- ComponentName: DetalhesSonho
- Create Component.

8.2 Propriedade do Componente DetalhesSonho

- Colocar um container;
- Largura: 100%, altura: 650px;
- Max W: 450px (para ele se encaixar melhor, quando for no WebApp);
- Arredondamento das bordas: 16;
- Deixar centralizado;
- Padding: 16

8.3 Colocando os elementos no PopUp

- Colocar uma coluna com o nome de **colunaprincipal**
- **Dentro da coluna:** colocar um stack (lembrando, que no stack, pode-se colocar um elemento sobre o outro. **Stack:** largura: 100%, altura:350px)
- No Stack, colocar uma image (largura: 100%, altura:350px)
- No stack, colocar um container.
 - No container colocar. largura: 50px, altura: 50px; alinhamento: x = - 0.9; y= -0.9. Cor: clicar na cor e deixar com 80% de transparência.
 - Dentro do container, colocar um icon. Colocar uma seta de voltar. Borda radius: 16. Dê nome para os container, para facilitar a organização: ctnVoltar

- No stack, cole `ctnVoltar`, copiado do container anterior. Denomine esse container de `ctnEditar`. Clicar no `ctnEditar`, em propriedades, vá até o alinhamento, no x, coloque 0.9, Y=-0.9. Assim, ele irá para a outra extremidade. Troque o ícone, para um lápis
- No stack, colocar outro container: largura: 100%, altura:75px e arraste para a parte inferior da figura. Deixar cor com transparência de 70%;
 - ❖ No container, colocar um texto: título do sonho;
- Coluna Principal, colocar um container, que vai ficar abaixo do stack. Largura: 100%, altura: 100px. Dê o nome de Detalhes. Colocar uma coluna dentro de Detalhes;
 - Dentro da coluna, colocar um container, com o nome de Ano. Colocar um texto e outro container (vai perguntar se é para organizar em coluna: sim. Nome deste container: Visualizar. Colocar um texto dentro de Visualizar. Na coluna dentro de Detalhes alinhar todos os elementos para a esquerda;
 - Copiar o container Visualizar e colocar na coluna do Ano. Dê o nome neste novo container Editar;
 - No container Editar, delete o text e coloque o TextField, para que o usuário possa escrever um novo texto. Colocar o Editar com 100% de largura e o TextField com 200px de largura.
- Ir para o Stack, copiar o `ctnEditar` e colar no Stack. Os ícones ficarão um em cima do outro. Dê o nome nesse novo container de Check. Troque o ícone para check. Como os ícones estão um sobre o outro, ícone de check só vai aparecer se o usuário pedir para editar.

OBS: Já fica totalmente responsivo.

8.4 Criar uma AppState

- Entrar na ferramenta App State;
- Criar uma variável, que irá mudar quando pedir uma determinada ação;

- Add App State Variable;
- Name: modoeditavel;
- Data Type:boolean;
- Is list: não habilitar;
- Persisted: não habilitar;
- Na próxima tela deixe valor padrão desabilitado. Então, sempre começar com o valor false.

Após ter criado a AppState – Container Editar

Página DetalhesSonho - Container Editar (ícone de editar, menu superior)

- Inserir uma nova Action;
- Add Action;
- Update App State;
- Add Field, aparece todas as App States criadas;
- Escolha: modoeditavel;
- Set Value;
- True (assim, quando clicar no container Editar, ele muda para a variável True.

Fazer a visibilidade condicional no Container Editar

- Clicar no container Editar;
- Propriedades;
- Habilitar a visibilidade condicional;
- Clicar em Unset;
- Conditions;
- Single Condition;
- First Value;
- Unset;
- App State;
- modoEditavel;
- Equal to;
- Second Value;
- False;

- Confirm

Página DetalhesSonho - Container Check – Ícone Check – Menu superior

- Inserir uma nova Action;
- Add Action;
- Update App State;
- Add Field, aparece todas as App States criadas;
- Escolha: modoeditavel;
- Set Value;
- False (assim, quando clicar no container Check, ele muda para a variável False. Significando que o usuário já terminou a edição.

Fazer a visibilidade condicional no Container Check

- Clicar no container Check;
- Propriedades;
- Habilitar a visibilidade condicional;
- Clicar em Unset;
- Conditions;
- Single Condition;
- First Value;
- Unset;
- App State;
- modoEditavel;
- Equal to;
- Second Value;
- True;
- Confirm

Voltando às propriedades, desabilitar a opção: Show in Ur Builder

Página DetalhesSonho - Container Visualizar

Fazer a visibilidade condicional no Container Visualizar – faça esse procedimento para: ano, orçamento e comentário.

- Clicar no container Visualizar;
- Propriedades;
- Habilitar a visibilidade condicional;
- Clicar em Unset;
- Conditions;
- Single Condition;
- First Value;
- Unset;
- App State;
- modoEditavel;
- Equal to;
- Second Value;
- False;
- Confirm.

Página DetalhesSonho - Container Editar - abaixo do visualizar

Fazer a visibilidade condicional no Container Editar - faça esse procedimento para: ano, orçamento e comentário.

- Clicar no container Editar;
- Propriedades;
- Habilitar a visibilidade condicional;
- Clicar em Unset;
- Conditions;
- Single Condition;
- First Value;
- Unset;
- App State;
- modoEditavel;

- Equal to;
- Second Value;
- True
- Confirm

Voltando às propriedades, desabilitar a opção: Show in Ur Builder.

8.5 Orçamento

Para criar o campo de Orçamento, logo abaixo do Ano.

- Clicar em Ano copiar e colar na coluna, onde está o ano, ou então faça os seguintes procedimento:
- Na Coluna Detalhes, colocar um container. Largura: 100%, altura: deixar em branco. Assim, ele acompanha o pai. Dê o nome de Orçamento. Colocar uma coluna dentro de Orçamento;
 - Dentro da coluna, colocar um container, com o nome de Orçamento. Colocar um texto e outro container (vai perguntar se é para organizar em coluna: sim. Nome deste container: Visualizar. Na coluna dentro de Detalhes alinhar todos os elementos para a esquerda;
 - Copiar o container Visualizar e colocar na coluna. Dê o nome neste novo container Editar;
 - No container Editar, delete o text e coloque o TextField, para que o usuário possa escrever um novo texto. Colocar o Editar cm 100% de largura e o TextField com 200px de largura

8.6 Comentário

- Clicar em Ano e colar na coluna, onde está o ano, dê o nome neste container de Comentário, ou então faça os seguintes procedimento:
- Na Coluna Detalhes, colocar um container. Largura: 100%, altura: deixa em branco. Dê o nome de Comentário.
- Colocar uma coluna dentro de Comentário:
 - Dentro da coluna, colocar um container, com o nome de Comentário. Colocar um texto e outro container (vai perguntar se é para organizar em

coluna: sim. Nome deste container: Visualizar. Na coluna dentro de Detalhes alinhar todos os elementos para a esquerda;

- Copiar o container Visualizar e colocar na coluna. Dê o nome neste novo container Editar;
- No container Editar, delete o text e coloque o TextField, para que o usuário possa escrever um novo texto. Colocar o Editar cm 100% de largura e o TextField com 200px de largura

8.7 Sonho Realizado

8.7.1 Criar os Botões Sonho Realizado, Editar/Deletar

- Na coluna principal, colocar um container;
- Colocar um botão dentro do container; largura 100%, altura: 50px; espaçamento: 16 nas laterais. Dê o nome de btnSonhoRealizado;
- Colocar nome do botão: Sonho Realizado;
- Escolher a cor do botão e fonte; Colocar um ícone: carinha feliz;
- Copiar o container SonhoRealizado e cole na Coluna Principal;
- Denominar o novo container de B-EditarDeletar;
- Copiar o botão do container B-EditarDeletar e colar no B-EditarDeletar. e pedir para organizar em linha. Assim, ficarão dois botões nessa linha. Um será para Deletar e o outro para Concluir Edição;
- Fazer a configuração dos botões: para separar os botões, ir até a linha, em Itens Spacing e coloque 16;
- Clique no segundo botão, vá até o Expansion e habilite o último ícone.

8.7.2 Botões Editar Deletar e Sonho Realizado – Visibilidade Condicional

Container do B-EditarDeletar - Visibilidade Condicional

- Visibilidade condicional;
- Condition;
- Single Condition,
- First Value: App State; modoEditavel;
- Equal to
- Second Value: True;

- Confirm

Container do B-SonhoRealizado - Visibilidade Condicional

- Visibility Conditional;
- Unset: Conditional;
- Combine Conditions;
- Unset:conditional;
- Single Condition;
- First Value: unset: App State; modoEditavel;
- Equal;
- False;
- Confirm;
- AND;
- Unset:
- Conditions;
- Single Condition;
- First value: Unset: sonhosDocument; status
- Second Value: value: Sonho Atual;
- Confirm;

Esta condicional também foi criada no Botão SonhoEsteAno, por isso coloquei a observação abaixo. Você pode repetir o processo ou então colar. Seguindo os passos abaixo:

- Nos três pontinhos ao lado do lápis de editar: clicar em copy variable. Esta condição será colada no Botão SonhoRealizado, mudando apenas o segundo valor da segunda condição para SonhoAtual. Confirm

8.7.3 UpDate

- Clicar no botão SonhoRealizado;
- Action
- Open;

- AddAction;
- UpDate Document;
- Unset: sonhoRef;
- Confirm;
- Set Field: Status; Specific Value: Sonho Realizado;
- Add Action;
- Navigation to: Home Page.

8.7.4 Botões Concluir Edição

Fazer uma Action para atualizar os dados (Update dos dados).

Botão Concluir Edição – Actions

Primeira Action

- Clicar no botão Concluir Edição;
- Action;
- Open Action Flow Editor;
- Add Action;
- Update Document;
- Unset: sonhoRef (documento que será atualizado);
- AddField: comentario; From Variable; Unset: Widget State: EditarComentario
- AddField: ano; From Variable; Unset: Widget State: EditarAno
- AddField: orçamento; From Variable; Unset: Widget State: EditarOrçamento
- Esses três são mais passíveis de mudança. Mas, se quiser todos poderão ser atualizados.

Segunda Action - Alert Dialog (para avisar que a operação deu certo)

- Add Action;
- Alert Dialog;
- Informational Dialog;
- Deu Certo!

- Sonho Atualizado.

Terceira Action - Atualizar a App State

- Add Action;
- Update App State;
- Unset: selecione modoEditavel;
- Set Value;
- False.

8.7.5 Botão Deletar - UpDate

Fazer uma Action para atualizar os dados (Update dos dados).

Primeira Action

- Action;
- Open; Add Action;
- Dialog: confirm dialog;
- Value: "Você deseja deletar o sonho?"
- Message: Remover Sonho;
- Desmin Text: Value: Cancelar;
- Confirm Text: Sim, deletar.

Segunda Action

Add Ação Condicional - se o usuário escolher sim, tem uma ação; caso escolha não terá outra ação.

- False: clicar em Add Terminate;
- True:
- Add Action (deletar o sonho do Banco de Dados);
- Delete Document;
- Unset: sonhoRef;

Terceira Action

- Add Action;
- Informational Dialog:
- Value: Sonho Removido;

- Value: Ok.

Quarta Action

- Add Action
- Navigation to: Home Page

8.7.6 UpDate – Edição de Ano, Orçamento, Comentário

Se precisar mudar editar alguma informação e/ou dado da página DetalhesSonhos,

- Clicar no container Ano, no textField renomear para EditarAno;
- Clicar no container Orcamento, no testField renomear para EditarOrcamento;
- Clicar no container Comentario, no textField renomear para EditarComentario;

EditarAno

- Clicar no EditarAno;
- initial Value;
- Unset: sonhos Document;
- ano;
- Confirm;

EditarOrcamento

- Clicar em EditarOrcamento;
- Initial Value;
- Unset: sonhos Document;
- Orcamento;
- Available options: Number Format;
- Number Format Options: Decimal;
- Decimal type: Comma for Decimal;
- Display as Currency: habilitar;
- Currency Symbol: R\$
- Confirm.

EditarComentario

- Clicar em EditarCometario;
- Initial Value;
- Unset: sonhos Document;
- Orcamento;
- Confirm.

8.7.7 Botão Tornar um Sonho para este Ano - Update

Se o usuário quiser mudar o status de um sonho, que está definido como sonho futuro, para sonho atual.

- Entrar na página DetalhesSonho;
- Copiar o botão SonhoRealizado e colar na coluna principal;
- Renomear o Container e o Botão para: SonhoEsteAno;
- Clicar no Botão SonhoEsteAno;
- Open Action (como foi copiado do botão SonhoRealizado, as actions já estão criadas. Mude apenas o Status para Sonho Atual

Volte para as propriedades do Container Botão SonhoEsteAno;

- Visibility Conditional;
- Unset: Conditional;
- Combine Conditions;
- Unset:conditional;
- Single Condition;
- First Value: unset: App State; modoEditavel;
- Equal;
- False;
- Confirm;
- AND;
- Unset:
- Conditions;
- Single Condition;

- First value: Unset: sonhosDocument; status
- Second Value: value: Sonho Futuro;
- Confirm;
- Nos três pontinhos ao lado do lápis de editar: clicar em copy variable. Esta condição será colada no Botão SonhoRealizado, mudando apenas o segundo valor da segunda condição para SonhoAtual. Confirm.

Na Página de Detalhes Sonhos - Fazer uma Consulta no Banco de dados

- Entrar na Página Detalhes Sonhos;
- Clicar no primeiro container, depois do nome DetalhesSonho;
- Backend Query;
- Add Backend Query;
- Document from Reference (não precisa pegar uma coleção, apenas um documento referência);
- Unset: sonhos;
- Unset: sonhosRef;
- Confirm;
- Confirm (para ter acesso ao documento escolhido, no caso sonhosRef. Assim, os valores da detalhesSonhos ficam dinâmicos).

Deixando os dados Dinâmicos

- Clicar no nome;
- Unset: sonhos Document;
- nome;
- Confirm;
- Faça isso para todos os dados.

9. PÁGINA NOVO SONHO

9.1.Criar a Página NovoSonho

- Clicar no componente DetalhesSonho e duplicar;
- Renomear para NovoSonho;
- Clicar no botão Voltar e retirar de dentro do Stack (que será deletado);
- Deletar o Stack;.
- Deletar no Ano, Orçamento e Comentário o Visualizar, deixar apenas o textField para receber o texto. Renomear os TextField com os nomes: InputNome, InputAno, InputComentario, InputOrçamento, InputImgUrl. Nos InputAno e InputOrçamento: colocar o Keyboard Type: Number e Number com decimal, respectivamente. O KeyBoard Type está no final das ferramentas;
- Renomear o botão para Criar novo Sonho;
- Inserir mais dois espaços. Para isso, copie o container Ano e cole duas vezes na Coluna Principal;
- Renomear os dois campos: Título do Sonho; e ImgUrl;
- Copiar o container da ImgUrl e colar na Coluna. Depois renomear como Status
 - Dentro do Status;
 - Editar;
 - Inserir um DropDown (para permitir que o usuário escolha o tipo de sonho: Atual, Futuro, Passado);
 - Define Options: 1ª opção: Atual; 2ª opção: Sonho Futuro;
 - Hint Text: Selecione quando irá realizar esse sonho

OBS: Como boa prática, não esquecer de nomear os container e os textField. Isso irá facilitar quando for fazer as integrações com o Banco de Dados.

9.2 Abrir Página NovoSonho, a partir da HomePage

- Abrir o NovoSonho
- Entrar na HomePage;
- Clicar no Botão +;
- Action;
- Add Action;
- Botton Sheet;
- Show;
- Select Componente:
- NovoSonho;
- Transparência: colocar cor preta copm 70% de transparência;

9.3 Criar Novos Dados na Página NovosSonhos - Create

- Entrar em NovoSonho;
- Clicar no Botão: Criar Novo Sonho;
- Action;
- Open;
- **Add Action - Primeira Ação**
- Create Document;
- Unset: sonhos;
- AddField:
- nome - From Variable - Unset - Widget State - InputNome
- Repita esse procedimento para todos: comentario, orçamento,
- **AddAction - Segunda Ação:**
- Alert Dialog;
- Informational Alert: O sonho foi criado - Sucesso - Ok
- Add Action - terceira ação;
- Navegation to;
- Home Page.

OBS: Quando voltar para a Home Page, o sonho ainda não irá aparecer, pois, temos que fazer novas ações.

10. POPUP MEU PERFIL

O perfil será necessário para fazer o login no app.

- Entrar nas páginas do app;
- Novo Componente;
- Create Blank
- Component Name: MeuPerfil;
- Create Component;
- inserir um container;
- Largura: 100%; altura: 350px
- Max Width: 250px;
- Alinhamento: centro;
- Arredondar bordas: 16px.

Dentro do container

- Inserir uma Coluna - espaçamento superior: 16px, 16 a esquerda e 16 pra direita;
- Dentro da Coluna:
 - Inserir uma circle image;
 - Um Text: com nome; tamanho 18, negrito; espaçamento inferior 32
 - Outro Text: com e-mail; cor secundária, fonte: 16
 - Inserir uma coluna - deixar esta coluna expandida; Expansion: todo pra baixo (terceiro ícone). Espaçamento Inferior: 16px;
 - Dentro desta coluna:
 - ❖ Inserir um botão; Logout, vermelho.
- Dentro da coluna principal, inserir um ícone de voltar;
- Colocar o ícone acima da imagem (coloque o ícone abaixo da imagem, depois coloque a imagem abaixo do ícone);
- Alinhar o ícone à esquerda.

10.1 UpLoad da Imagem do perfil

- Entrar na página MeuPerfil;
- Clicar na CircleImage;
- Action;
- Open;
- Action 1:
 - Upload Media;
 - Upload type: Firebase;
 - Media Source: Either Camera or Gallery.
- Add Action;
- Action 2:
 - Update Document;
 - Authenticated User;
 - User Reference;
 - AddField;
 - Photo_url;
 - Value Source: Uploaded Photo/Video
- Add Action;
- Action 3
 - Alert Dialog;
 - Informational Dialog;
 - Value: Foto Atualizada;
 - Value: Sucesso;
 - Value: Ok.

10.2 Colocar os dados da MeuPerfil dinâmicos

- **Clicar na CircleImage:**
 - Propriedade;
 - Image Type: Network;
 - Path: clicar
 - AuthenticatedUser;

- Photo Url;
- Coloque uma foto padrão, no Default Variable Value: que será substituída quando o usuário colocar a sua foto.
- **Clicar no nome:**
 - Propriedade;
 - Text;
 - Authenticated User;
 - Display Name.
- **Clicar no Email**
 - Propriedade;
 - Text;
 - Authenticated User;
 - Email

10.3 Botão de Logout – Ação

- Clicar no Botão Logout;
- Action;
- On Tap;
- Authenticated;
- Logout

11. PÁGINA DE CADASTRO E LOGIN

- Criar uma nova página;
- Usar um template: Authentication. Escolha um que tenha Sign In e Sign Up (login e cadastro);
- Nome da página: PaginaLogin;
- TabBar: te oferece duas opções de Login e Cadastro. Em duas abas, cada aba apresenta dados diferentes;
- Colocar a cor da página azul escuro;
- Tirar as social media, pois elas não serão utilizadas para fazer login;
- Clicar na TabBar: Mudar as cores das abas, para que ambas apareçam na tela: Selected Color: branco; Unselected Color: cinza; TagBar Style: Indicador;

Aba de Login:

- Nome: sem nada;
- Hint text: entrar com seu e-mail;
- Nome: sem nada;
- Hint Text: Senha;
- Fazer alteração no botão: Entrar;
- Fazer alteração no botão: Esqueceu a senha, deixar a cor 100% transparente.

Aba de Cadastro

- Hint Text: e-mail;
- Hint Text: senha.

Copiar a TextField de e-mail e colar na coluna, para colocar o nome.

Renomear o nome do elemento para InputNome;

Botão Cadastrar

- Nome: Cadastrar;
- Largura: 150px, altura: 50px.

12. AÇÃO DE CADASTRAR E DE FAZER LOGIN

12.1 Fazer o setup da coleção de usuário:

- Flutter Flow;
- Configurações;
- Firebase;
- Clicar no terceiro ícone na linha do id do projeto (um chapeuzinho) para configurar a tabela de usuário;
- Next Step;
- Habilitar Enable Authentication;
- Habilitar Create User Collection;
- Next Step;
- Fazer o Deploy;
- Start Builder;
- Clicar em Generate Config Files;
- Generate Files;

OBS: As regras para a coleção users não poderão ser mudadas, deixar como foram criadas pelo Firebase: users collections, primeiro e último ícone assinalados.

12.2 Ação de Cadastrar

- Entrar na páginaLogin;
- Clicar no botão Cadastrar;
- Action;
- Delete a ação que aparece (pois, como usamos um template, a ação acompanhou);
- Add Action;
- Firebase Authentication:
 - São várias as opções de fazer a autenticação. Escolher Create Account;

- Unset: escolher: e-mail;
- Unset: emailAddressCreate;
- Unset: passwordCreate;
- Unset: passwordConfirm
- Create Document;
- Collection: users;
- SetField:
- email: From Variable; Unset: widget State: emailAddress-Create
- display-name: From Variable; Unset: Widget State: Inputname

12.3 Ação do Login

- Entrar na aba de login;
- Clicar no botão Entrar;
- Action;
- On Tap;
- Auth Login;
- Auth Provider: email;
- Email Field: emailAddress;
- Password Field: password.

13. PUBLICAR O APP NA WEB

- Entrar no FlutterFlow;
- Entre no Flutter Flow
- Configuração
 - Platform;
 - Deixe habilitado: Android, IOS e Web
- Vá para Web Publishing
 - Site URL: coloque o nome personalizado para a URL, mas deixando sempre no final .flutterflow.app;
 - Habilitar o Enable PWA
 - As demais informações, pode deixar em branco;
 - Depois disso, clique em publish,
 - Depois de publicado, clique no olho do lado da url, para visualizar o APP;
 - Na barra de endereço do APP, clique num ícone de um computador com uma seta;
 - Ele vai perguntar se é para instalar o APP, sim. Ele instala no computador

Instalar no celular

- Copie a URL e abra no Celular no Chrome;
- Clique nos 3 pontinhos da barra de endereço;
- Procure a opção Instalar Aplicativo;
- Instalar.

REFERÊNCIA

Curso NocoCode Start Up: <https://flix.nocodestartup.io/courses>.



