

FLUTTER FLOW

APP AGENDAMENTO

Autora: Terezinha Márcia de Carvalho Lino

**JOÃO PESSOA
2024**

SUMÁRIO

1.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.....	8
3.1 Criar Novo Projeto.....	8
4. FIREBASE – BANCO DE DADOS	9
4.1 Criar conta e configurar o Firebase	9
4.2 Banco de Dados - fazer integração entre Firebase e FlutterFlow	10
4.2.1 Authentication (para fazer login no app)	10
4.2.2 Storage (para guardar as mídias utilizadas no app: imagens, vídeos etc)	11
4.2.3 Configurações.....	11
4.2.4 Volte ao FlutterFlow para terminar a integração	11
4.2.5 Firestore no FlutterFlow – Criar Coleção	12
4.2.6 Na engrenagem do Firestore	12
4.3 Criação de Coleção no Firebase.....	12
4.4 Corrigindo Erros Firebase	13
5. FLUTTERFLOW – APRESENTAÇÃO DA INTERFACE	14
5.1 Menu do lado esquerdo – Menu dos Elementos.....	14
5.2 Menu do lado esquerdo - Menu dos Componentes	15

5.2.1 Templates - terceiro ícone da barra de menu horizontal do lado esquerdo da tela.....	15
5.2.2 Widget Tree	15
5.2.3 Principais Widgets - Ferramentas	15
5.2.4 Storyboard	17
5.2.5 Firestore.....	17
5.2.6 DataTypes - App State.....	17
5.2.7 API Calls	18
5.2.8 Media Assets	18
5.2.9 Custom Code.....	18
5.2.10 Cloud Functions	18
5.2.11 Automated Tests.....	18
5.2.12 Theme Settings.....	18
5.2.13 Configurações.....	18
5.3 Menu Superior.....	19
5.3.1 FlutterFlow IA.....	19
5.4 Menu do lado direito.....	20
5.4.1 Primeiro Ícone: Propriedades	20
5.4.2 Segundo ícone: Actions	20
5.4.3 Terceiro Ícone: Backend Query	20
5.4.4 Quarto ícone - Animações	20
5.4.5 Quinto ícone: Documentação	20
6. CRIAÇÃO DO PROJETO AGENDAMENTO.....	21
6.1 Elementos de página Fundamentos Linhas e Colunas	21
6.2 Planejamento do App.....	21

6.2.1 Problema a ser resolvido	21
6.2.2 Solução proposta pelo uso do App	21
6.2.3 Design de Telas	21
6.3.1 Telas do Aplicativo	22
6.4 Telas Criadas	23
.....	24
6.5 Sistema de Agendamento	28
6.6 Teoria sobre Datas	29
7. MODELAGEM DO BANCO DE DADOS	29
7.1 Banco de Dados	29
7.2 Desenvolvimento	29
8. OPERAÇÕES NO BANCO DE DADOS	30
8.1 Interações com o Banco de Dados - CRUD	30
9. PÁGINA LISTA MÉDICOS	31
9.1 Criar App State	31
9.2 Categoria de Médicos – colocar ações	31
9.3 ListView Médicos Populares– Back End Query – Consultar Banco de Dados	32
.....	32
9.3.1 Colocar os Dados Dinâmicos da ListView Médicos Populares	32
9.4 ListView Especialidade Médico – Back End Query – Consultar Banco de Dados	32
.....	32
9.4.1 Colocar os Dados Dinâmicos da ListView Especialidade	33
9.5 ListView Todos os Médicos – Back End Query – Consultar Banco de Dados	33
.....	33
9.5.1 Colocar os Dados Dinâmicos da ListView Todos os Médicos	33
9.6 Visibilidade Condicional para as ListView	34

9.7 Botão Adicionar Médico	36
10. PÁGINA DETALHESMÉDICO.....	36
10.1 Depois desse procedimento ir para a Página DetalhesMedicos	36
10.2 Colocar os dados da Página DetalhesMédico de forma Dinâmica	37
10.3 GridView – Horários de Trabalho	37
10.4 Botão Editar Médico.....	38
10.5 Fazer Upload da Imagem na Página Editar Médico – Atualizar imagem ..	41
10.6 Botão Salvar Alterações.....	41
11.PÁGINA CRIAR NOVO MÉDICO.....	43
11.1 Fazer Upload da Imagem do Médico	43
11.2 Fazer Upload da Imagem local e no Firebase.....	43
11.2.1 Problemas em fazer o Upload	45
Imagem 1	45
Imagem 2	46
11.3 Horários Disponíveis	46
11.4 Botão Criar Médico	47
12. DRAWER – PAGINA SUSPENSA PARA APRESENTAR DADOS DO USUÁRIO, DENTRO DA PÁGINA LISTAMÉDICOS.....	49
Para aparecer a Drawer.....	49
12.1 Colocar elementos da Drawer Dinâmicos	49
12.2 Página Editar Perfil	50
12.2.1 Botão Atualizar Perfil	51
13. Editar o Administrador na Drawer	51
13.1 Página Todos Atendimentos	52
13.1.1 Container Calendar.....	52

14. PÁGINA DE LOGIN.....	59
14.1 Botão Login	59
14.2 Botão Esqueci minha senha	60
14.3 Botão Criar Conta	61
15. SISTEMA DE RESERVA DE HORÁRIOS.....	61
15.1 Para fazer o sistema de horário de um médico, siga o procedimento:.....	61
15.2 Criar Horários de Atendimento.....	63
15.2.1 Para criar horários, você deve criar uma Custom Functions Lista de Horário	63
15.2.2 Para criar os horários, você precisa criar uma App State	64
15.2.3 Criar horário de forma personalizada.....	67
15.2.4 Horários Indisponíveis	69
15.2.5 Colorir o Container das Horas se acordo com a disponibilidade	72
15.3 Botão Confirmar Agendamento	73
16. DRAWER – COLOCAR AÇÃO EM MEUS ATENDIMENTOS.....	75
17. PÁGINA DETALHES ATENDIMENTO (AGENDAMENTO).....	78
17.1 BackEnd Query – Dados do Médico	78
17.2 Colocar os dados de forma dinâmica	78
17.3 BackEnd Query – Dados do Paciente.....	80
17.4 Colocar os dados do usuário de forma dinâmica	80
17.5 Botão Editar Atendimento	81
18. PÁGINA EDITAR ATENDIMENTO.....	81
18.1 BackEnd Query Primeiro Container	81
18.2 Horário de Atendimento – Forma dinâmica.....	86
18.3 Descrição: Descreva o que está sentido.....	89

18.4 Botão Confirmar Agendamento.....	90
19. PUBLICAR O APP NA WEB	91
REFERÊNCIA	92

1.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

O objetivo deste projeto é criar o APP Agendamento, seguindo as orientações do professor Daniel Neto, da NoCode Start Up.

O App foi idealizado para registrar o agendamento de horário de empreendimentos que exigem uma boa organização de horários de atendimento, tais como: Barbearia, Salão de Beleza, Consultório Médico e Dentário, etc.

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 clicar 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: nome da coleção

- Clique em Firestore (menu lateral);
- Create Collection;
- Name:
- 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;
 - descricao: string;
 - imgUrl: imagePath;
 - investimento: 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: 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
- Adicionar novos campos, até o necessário;
- Campo: status; Tipo: boolean; Valor: False
- Campo: orcamento; Tipo: number; Valor: coloque um valor de referência;
- Salvar.
- OBS: O Firebase permite colocar uma coleção dentro de uma coleção (subcoleção).

Para repetir o processo acima e colocar novos campos na coleção criada anteriormente, faça o seguinte:

- Clicar em Mais no Google Cloud (ícone à direita, na linha do nome da coleção);
- Key Visualizer;
- Vincular;
- No Google Cloud é onde será gerenciado toda a parte do Backend, já aparece os dados da coleção que você cadastrou no Firebase, para continuar cadastrando, na coluna Adicionar Documento, clique nos três pontinhos e clique em Adicione Documento semelhante, então abre uma nova página com vários campos, só substituir com os dados novos.

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;

- Firestone 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 FuttlerFlow;
- 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 DO PROJETO AGENDAMENTO

6.1 Elementos de página Fundamentos Linhas e Colunas

Colunas

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

Linhas

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

6.2 Planejamento do App

6.2.1 Problema a ser resolvido

Para o planejamento do App é essencial identificar o **Problema** e qual a solução que será proposta: entender o problema existente no mercado e a persona envolvida. Exemplos:

- Tempo gasto pelo profissional para organizar a agenda;
- Esquecimento de marcar horários agendados pelos clientes;
- Esquecimento de desmarcar horários cancelados pelos clientes;
- Falta de informação sobre horários disponíveis.

6.2.2 Solução proposta pelo uso do App

A Solução proposta pelo uso do App: conceitualizar a tecnologia que irá solucionar o problema.

- A solução é um Aplicativo de Agendamento.

6.2.3 Design de Telas

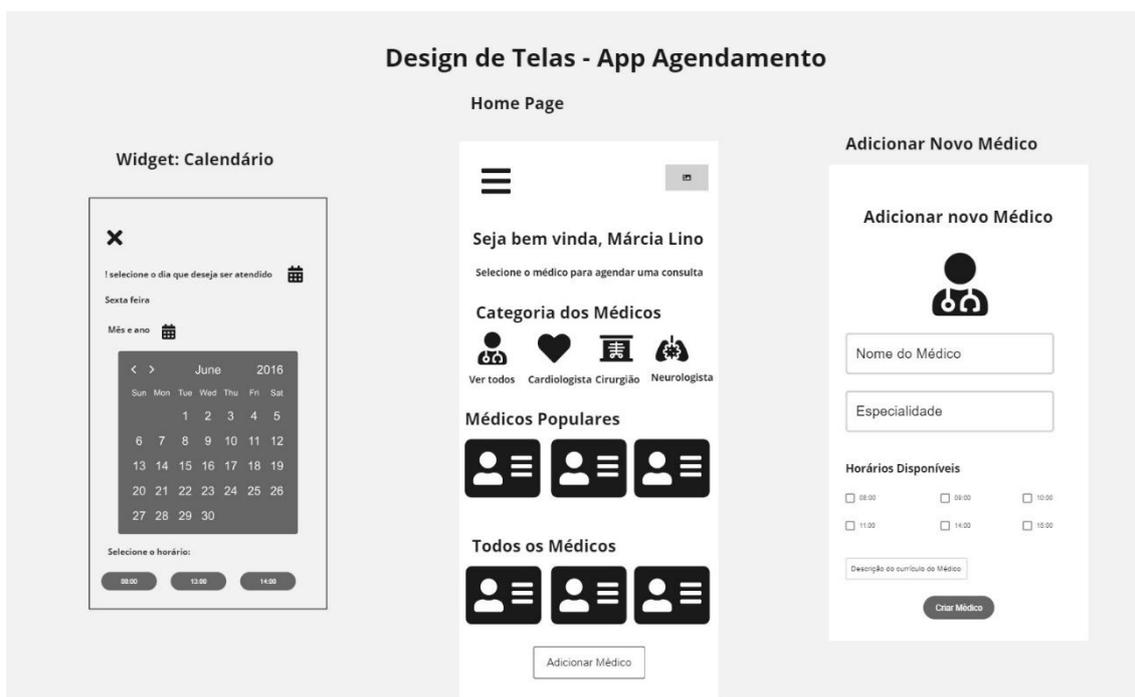
Design de Telas: elaborar telas por meio de um wireframe e fluxo de usuário.

- Fazer o desenho de todas as páginas;
- Desenvolver o fluxo do usuário, desde quando ele entra no aplicativo e todas as opções de navegação.

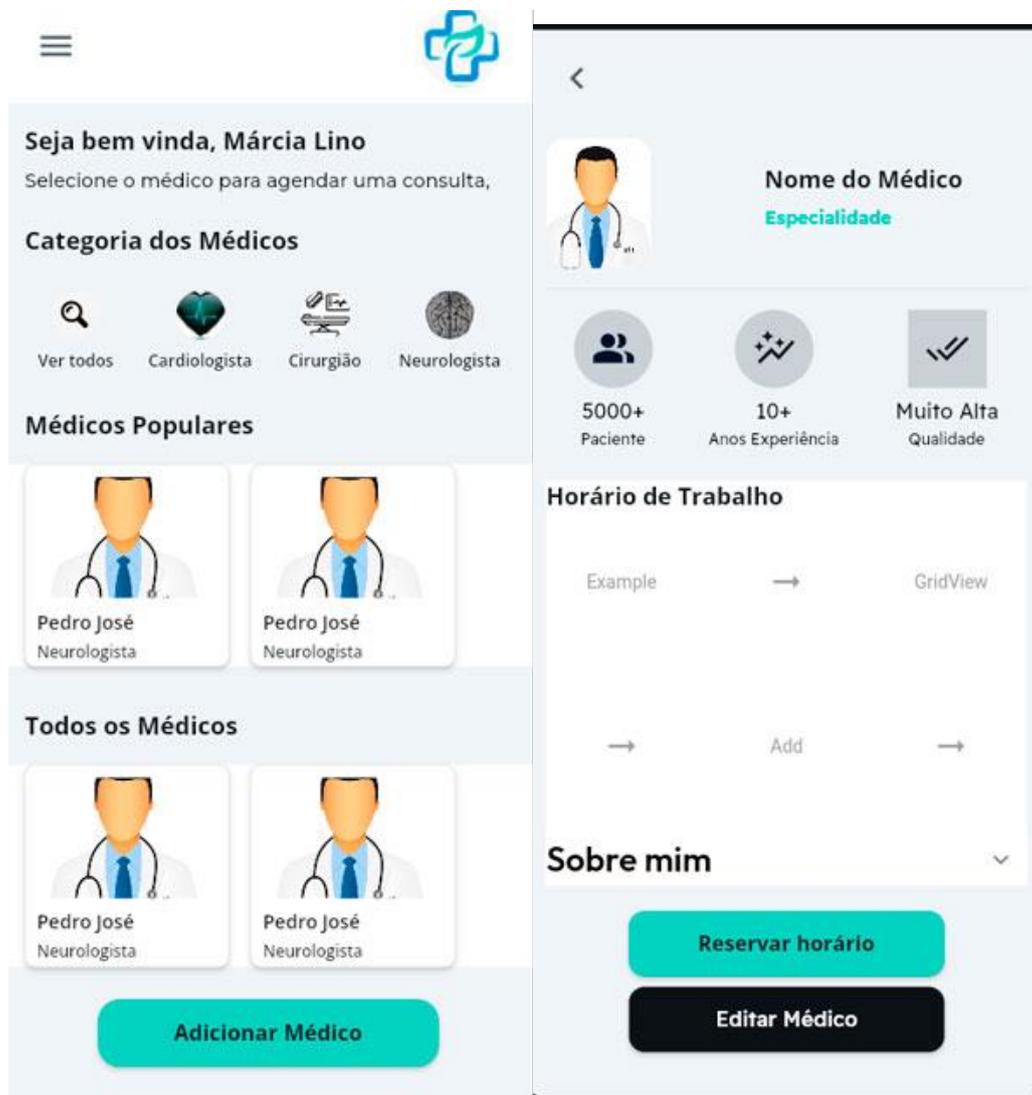
- **Miro** - Criar no Miro o design das páginas: <https://miro.com/app/>

6.3.1 Telas do Aplicativo

- Lista dos profissionais: poderão ser selecionados de acordo com o ramo de atividade;
- Detalhes e horários do profissional: nome, perfil, horários de atendimento;
- Agendamento com horário personalizado: o agendamento será feito de forma personalizada;
- Meus agendamentos: página que irá mostrar todos os agendamentos do usuário;



6.4 Telas Criadas



X

Editar Médico



Foto do Médico

Nome do Médico

Especialidade

Descrição e Currículo do Médico

<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00
<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00
<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00	<input type="checkbox"/> 08:00

Salvar Alteração

X

Adicionar Novo Médico



Foto do Médico

Nome do Médico

Função

← Detalhes do Agendamento

Nome Médico
Especialidade

Data:

Descrição dos sintomas do paciente:

Paciente:



nome
email
Telefone

Editar Agendamento

×

1ª - Selecione o dia que deseja ser atendido

Selecione o dia que deseja ser atendido

February 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	1	2

2ª Selecione o horário

Selecione o Horário

08:00

Dia de atendimento 

Dia de Atendimento

February 2024  < >

Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	1	2	3

Horário do atendimento:

Selecione o Horário

Example → GridView →

Add → Things →

Here!

Descreva o que você está sentindo...

Confirmar agendamento

 Carvalho Clínica

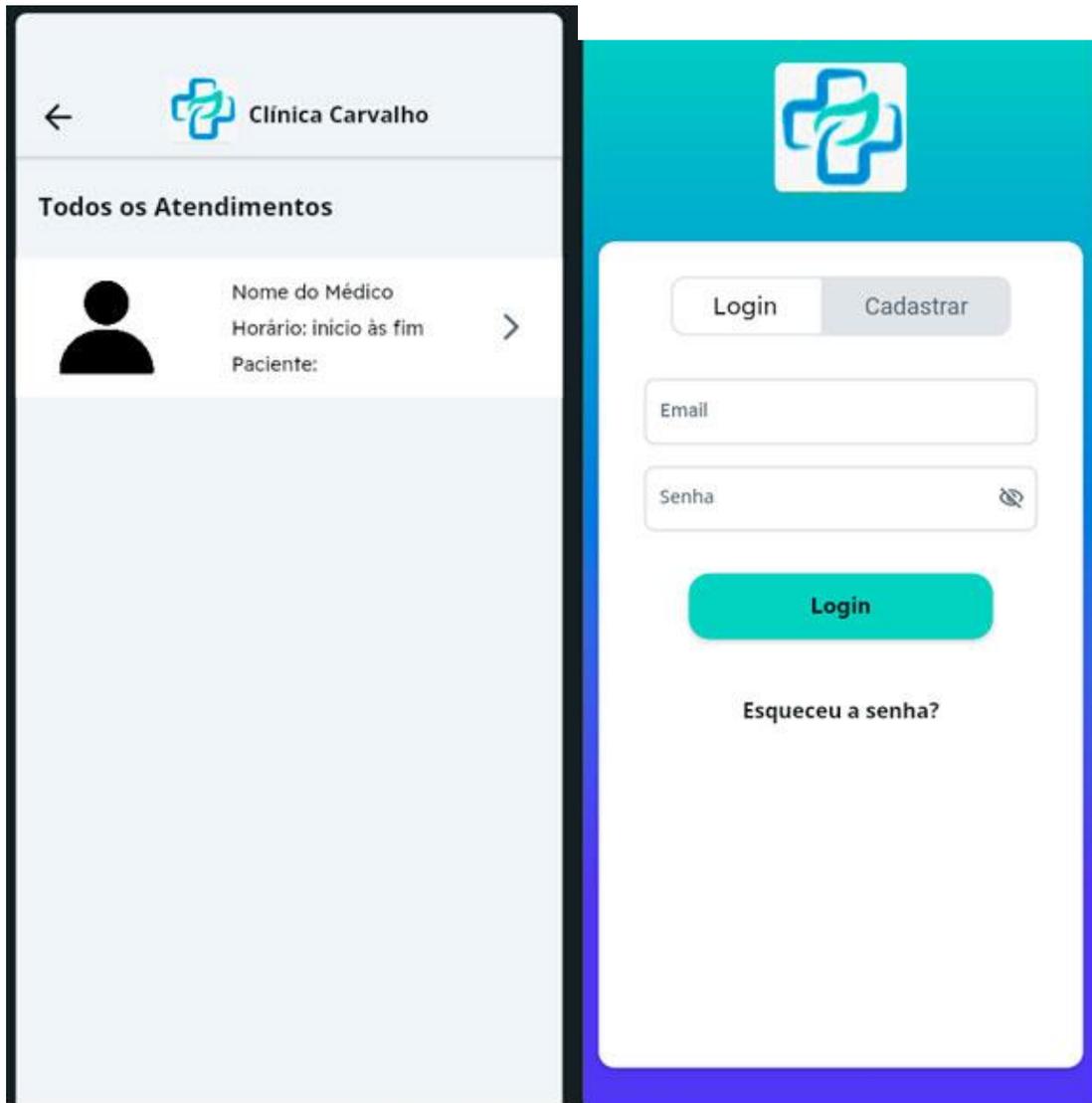
Meus agendamentos

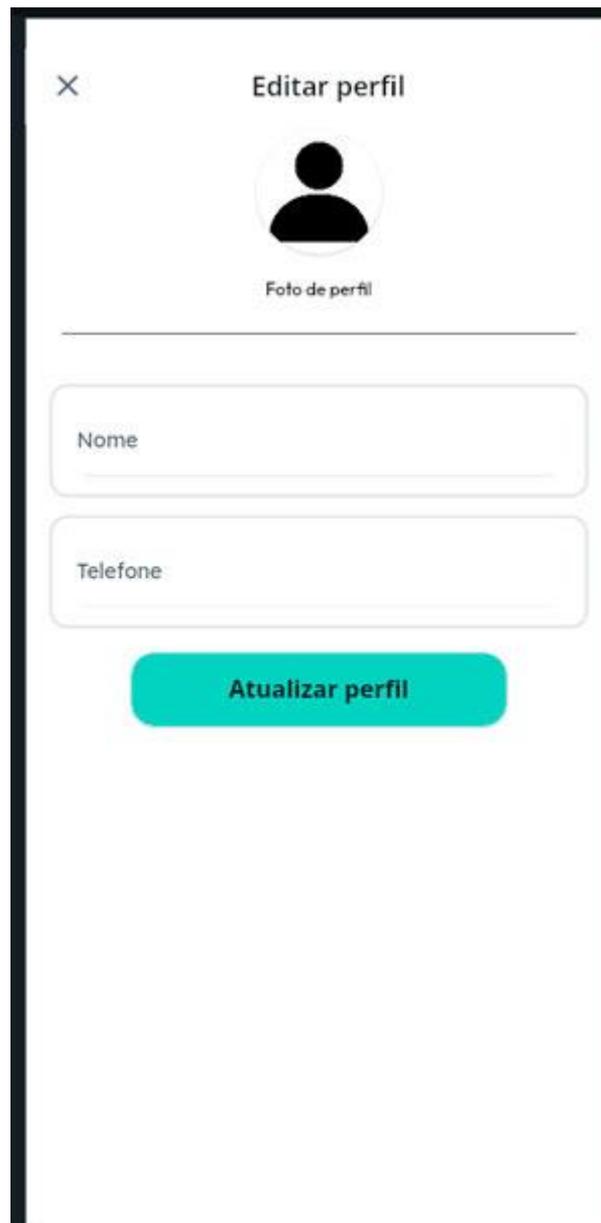


Nome do Médico

Horário: início às final

Horário >





×

Editar perfil

Foto de perfil

Nome

Telefone

Atualizar perfil

6.5 Sistema de Agendamento

- Criar profissional com horário: De forma totalmente personalizadas, com possibilidade de edição;
- Sistema de horário por profissional;
- Trabalhando com DateTime: sistema de verificação de datas com horário;
- Consumo de Leituras no Firebase: Melhor performance e menor gasto com banco de dados para maior escalabilidade;

6.6 Teoria sobre Datas

Quando utiliza-se o DateTime no FlutterFlow é possível personalizar o formato que irá aparecer a data selecionada. São várias as opções de formato, escolha a que melhor se encaixa no seu projeto.

Exemplo de data: 5 de outubro de 2023 às 23:16:34 UTC-3 (UTC-3: distingue o fuso horário).

Como colocar DateTime:

- Widget Calendar;
- Ação: Date/Time Picker

7. MODELAGEM DO BANCO DE DADOS

7.1 Banco de Dados

Banco de Dados: modelar o banco de dados do aplicativo e a relação entre as tabelas.

- Definir as coleções e quais são os dados que ficarão salvos;
- Quais são as informações que usuário e o profissional precisarão ter.

São 3 as coleções:

- Users;
- Médicos;
- Agendamentos;

7.2 Desenvolvimento

Construir o App conectando o Frontend (FlutterFlow) com o Backend (Firebase)

Serão criadas três coleções:

Coleção User – padrão FlutterFlow

- Uid..... id
- Display_name.....string
- Email..... string
- Photo_url..... image
- Create_time..... dateTime
- Phone_number..... string
- isAdmin..... boolean

Coleção Médicos

- nome.....string
- função.....string
- foto.....image
- curriculoMedico.....string
- horarioTrabalho..... listString

Coleção Agendamento

- SintomasPaciente..... string
- Medico.....refMedicos
- Usuário..... refUsers
- Data..... dateTime

8. OPERAÇÕES NO BANCO DE DADOS

8.1 Interações com o Banco de Dados - CRUD

- CREATE: criar novos dados;
- READ: Visualizar dados;
- UPDATE: Atualizar dados;
- DELETE: Deletar dados.

9.PÁGINA LISTA MÉDICOS

9.1 Criar App State

As AppStates são criadas para guardar dados temporários, assim, não sobrecarrega o Banco de Dados.

- No FlutterFlow, clicar na AppValues – App State (menu lateral);
- Add App State Value;
- Field Name: FiltroBuscaMedico;
- Data Type: string;
- Is List: não habilitar;
- Persisted: não habilitar (se o dado deve continuar no Banco de Dados após o usuário sair do App);
- Create.

9.2 Categoria de Médicos – colocar ações

Clicar na Coluna Ver Todos

- Action;
- Add Action;
- Update App State;
- AddField: FiltroBuscaMedico;
- Select Update Type: Clear value;

Clicar na Coluna Cardiologista

- Action;
- Add Action;
- Update App State;
- AddField: FiltroBuscaMedico;
- Select Update Type: Set value;
- Value to Set: Cardiologista

OBS: Repetir a operação para: Cirurgião e Neurologista.

9.3 ListView Médicos Populares– Back End Query – Consultar Banco de Dados

- Clicar na ListView;
- Back End Query;
- Query Collection;
- Unset: Coleção: Médicos;
- List Document;
- Limit: 4 (mostrar apenas 4 médicos, com maiores números de consultas);
- Ordering: Order by 1;
- Collection Field Name: Unset: NúmerodeAtendimento;
- Order: Decressing;
- Confirm.

9.3.1 Colocar os Dados Dinâmicos da ListView Médicos Populares

- Clicar na imagem do médico;
- Medicos Document;
- Document: foto;
- Repetir para nome do médico e função.

9.4 ListView Especialidade Médico – Back End Query – Consultar Banco de Dados

- Clicar na ListView;
- Back End Query;
- Query Collection;
- Unset: Coleção: Médicos;
- List Document;
- Filters;
- Collection: função;
- Unset: Equal;
- Unset: App State; FiltroBuscaMedico;

- Order;
- Colletction Field Name: nome;
- Increassing;
- Confirm.

9.4.1 Colocar os Dados Dinâmicos da ListView Especialidade

- Clicar na imagem do médico;
- Medicos Document;
- Document: foto;
- Repetir para nome do médico e função.

9.5 ListView Todos os Médicos – Back End Query – Consultar Banco de Dados

- Clicar na ListView;
- Back End Query;
- Query Collection;
- Unset: Coleção: Médicos;
- List Document;
- Order;
- Colletction Field Name: nome;
- Increassing;
- Confirm.

9.5.1 Colocar os Dados Dinâmicos da ListView Todos os Médicoa

- Clicar na imagem do médico;
- Medicos Document;
- Document: foto;
- Repetir para nome do médico e função.

9.6 Visibilidade Condicional para as ListView

Quando o usuário selecionar todos os médicos, no menu superior, as listas de Médicos Populares e Médicos por Especialidade não devem aparecer. O mesmo vai acontecer quando ele selecionar outra lista. Para que isso aconteça, faça os seguintes procedimentos:

Clicar na linha Médicos Populares;

- Propriedade;
- Condicional: habilitar;
- Condition;
- Single Condition;
- First Value: Value: App State: FiltroBuscaMedico; Is Not Set or is Empty;
- Confirm
- Repetir o procedimento para o Container MedicosPopulares

Clicar na linha de Especialidade Médico

- Propriedade;
- Condicional: habilitar;
- Condition;
- Single Condition;
- First Value: Value: App State: FiltroBuscaMedico; Is Set and not Empty;
- Confirm
- Repetir o procedimento para o Container Especialidade Médicos

Colocando o dado dinâmico na linha Especialidade Médico:

- Clicar na linha Especialidade Médico;
- Propriedade;
- Text: Especialidade Médico;
- App State;
- Filtrobuscamedico.

Navegar da Lista Médico para Detalhes Médico

- Clicar no container abaixo da ListView Médicos Populares;

- Action;
- Add Action;
- Navigation to;
- DetalhesMedico;

Toda vez que passa de uma página para outra, que tem um detalhe de uma própria lista, deve-se passar um parâmetro de página. Para isso, faça os seguintes procedimentos.

- Clicar em + Define;

Você é encaminhado para a Página DetalhesViagem, então clicar em

Add Parameter;

- Parameter Name: refMedico (referência de um determinado médico);
- Type: Document Reference;
- Collection Type: medico;
- Confirm;
- Volta para a Action da HomePage que estava adicionando e clica em Pass
- Parameters: refMedico;
- Type: DocumentReference;
- Parameters: refMedico;
- Unset: medicosDocument;
- Reference;
- Confirm
- OBS: Fazer o mesmo procedimento para a Lista Vertical

Depois desse procedimento ir para a Página DetalhesMedicos

- Clicar no nome da página: DetalhesMedicos, para colocar uma Backend para puxar os dados do Banco de Dados (Escolhe o elemento de maior nível hierárquico da página) puxar os dados do Banco de Dados)
- Backend Query;
- Add BackEnd Query;
- Query Type (tipo de consulta): Document from Reference;

- Collection: medicos;
- Unset: refMedicos;
- Confirm;
- Confirm

9.7 Botão Adicionar Médico

- Clicar no botão;
- Action;
- Add Action;
- Bottom Sheet;
- Show;
- Select Component: Criar Medico.

10. PÁGINA DETALHESMÉDICO

10.1 Depois desse procedimento ir para a Página DetalhesMedicos

OBS: o procedimento abaixo já foi realizado, quando estava sendo desenvolvida a inteligência da Página Lista Médicos.

- Clicar no nome da página: DetalhesMedicos, para colocar uma Backend para puxar os dados do Banco de Dados (Escolhe o elemento de maior nível hierárquico da página) puxar os dados do Banco de Dados)
- Backend Query;
- Add BackEnd Query;
- Query Type (tipo de consulta): Document from Reference;
- Collection: medicos;
- Unset: refMedicos;
- Confirm;
- Confirm

10.2 Colocar os dados da Página DetalhesMédico de forma Dinâmica

- Clicar na imagem do médico;
 - Path;
 - medicosDocument;
 - Foto;
 - Confirm
- Repetir para o Nome e Especialidade.

10.3 GridView – Horários de Trabalho

- Clicar na GridView;
- Clicar em Generating Children form Variable (ícone do lado da BackEnd Query);
- Variable name: ListaHorarios;
- Unset;
- medicosDocument;
- horarioTrabalho (List+string);
- Confirm;
- Save

Texto dentro da GridView

- Clicar na GridView e colocar um texto: Horas;
- Colocar o Horas de forma dinâmica;
- Clicar no texto;
- Unset;
- ListaHorario;
- Confirm.

Sobre Mim – colocar de forma dinâmica

- Clicar no texto, abaixo do título Sobre Mim;

- Max Caracteres Allowed: 80
- Text;
- Unset;
- medicosDocument;
- curriculoMedico

Depois disso, colocar de forma dinâmica a visão expandida

- Clicar em expanded View;
- Text (que está dentro do container);
- Text;
- Unset;
- medicosDocument;
- Curriculo

10.4 Botão Editar Médico

Este procedimento só poderá ser feito depois que criar e configurar a Página Criar Novo Médico

- Clicar no botão Editar Médico;
- Action;
- Add Action;
- Bottom Sheet;
- Show;
- Select Component: Editar Médico;
- Pass;
- refMedico;
- Value: Unset;
- refMedico;
- Confirm

Depois disso, entrar na Página Editar Médico

- Clicar no segundo Container (abaixo do container EditarMedico);
- Query Collection;
- Add Query;
- Document Reference;
- Medicos;
- Unset: refMedico;
- Confirm;
- Confirm;

Colocar os dados de forma dinâmica

- Imagem: Path;
- Unset: medicosDocument;
- Foto;
- Nome Médico:
- Initial Value;
- medicosDocument;
- nome;
- Função:
- Incitial Value;
- medicosDocument;
- função;
- Descrição;
- Initial Value;
- Médicos Document;
- Currículo Medico;

Editar os horários

Para não precisar criar ação para todos os horários: delete a coluna onde estão os horário. Vá até a Página Criar Médico e copie a coluna onde estão os horários já definidos e cole na Página Editar Médicos. Aí, todas as ações já estarão definidas. Depois disso:

- Os horários deverão ser carregados de acordo com o que foi definido na criação do médico. Para isso, faça o seguinte:
 - Entrar na Página DetalhesMedico;
 - Clicar no Botão Editar Médico;
 - Add Action, esta seria a primeira ação, antes do Bottom Sheet. Add Action (ela ficará em segundo lugar, então clique na setinha na parte superior da ação, assim, ela se torna a primeira ação);
 - Update App State;
 - Add Field;
 - listaHorarioMedicoCriacao;
 - Unset;
 - Set Value;
 - Unset;
 - medicosDocument;
 - horarioTrabalho;
 - Confirm

Dessa forma você atualiza os horários do médico.

Voltar para a Página Editar Médico:

- Clicar no primeiro check box;
- Em propriedade, clique em checkbok Initial Value;
- Condition;
- App State;
- listaHorarioMedico;
- Unset;
- ListContains Item;
- Unset;
- Combine Text;
- Text 1: 08:00;
- Confirm;
- Confirm

- Faça isso para todos os horários. Você pode copiar a ação e colocar, mudando depois os horários.

10.5 Fazer Upload da Imagem na Página Editar Médico – Atualizar imagem

- Na Página Editar Médico, clicar na imagem do Médico;
- Action;
- Add Action;
- Upload Media;
- Firebase.

10.6 Botão Salvar Alterações

- Clicar no botão;
- Action;
- **Add Action – 1 Ação**
- Open;
- Alert Dialog;
- Confirm Dialog;
- Você tem certeza?
- Isso irá alterar todas as informações do Médico.
- Cancelar;
- Confirmar.
- **Add Action Condicional - 2ª Ação**
- True;
- Add Action;
- Firestore;
- Update Document;
- Unset;
- refMedico;

- Confirm;
- Add Field;
- Nome;
- From variable;
- Widget State;
- Nome;
- Repetir isso para: função e currículo;
- **Para horário:**
 - List Update Type: Set Value;
 - App State;
 - listaHorarioMedicoCriação;
- **Para a foto:**
 - From Variable;
 - Condition;
 - If/Then/Else;
 - Unset:
 - Widget State;
 - Upload File URL;
 - Is Set and Not Empty;
 - Value: Widget State; Upload File URL;
 - Else: Unset; medicoDocument; foto;
 - Confirm;
 - **Add Action – 3ª Ação**
 - Bottom Sheet;
 - Dismiss;

Na condicional False:

- Bottom Sheet;
- Dismiss;

11. PÁGINA CRIAR NOVO MÉDICO

11.1 Fazer Upload da Imagem do Médico

- Clicar na coluna, onde está circle image;
- Action;
- Add Action;
- Upload Midia;
- Banco de Dados: Firebase;

Mudar para propriedade para deixar a imagem dinâmica

- Clicar na circle image;
- Unset;
- Widget State;
- Upload File Url

11.2 Fazer Upload da Imagem local e no Firebase

O FlutterFlow fez uma atualização para evitar fazer upload direto no Banco de Dados. Desta forma evita sobrecarregá-lo. O procedimento do item 12.1, faz o upload direto no Firebase, o que não é recomendável. Abaixo da circle image, coloque um botão para salvar a imagem.

<https://www.youtube.com/watch?v=iviDnzo50Ww>

- Entrar na Página Criar Médico;
- Clicar na Circle Image (aqui a imagem vai aparecer para que o usuário defina se é a imagem que ele quer fazer o upload no Firebase)
- Action;
- Open
- Add Action;
- Upload/Save Media;
- Upload Type: Local Upload (Widget State);
- Vá para propriedade;
- Clique em Network e troque para Upload File;

- Unset;
- Widget State;
- UpLoad Local File;
- Confirm.

Botão Fazer Upload

- Clicar no Botão;
- Action;
- Open;
- Add Action;
- Upload/ Save Media;
- Unset: Firebase;
- UpLoad Media;
- Unset;
- Widget State;
- Upload Local File 1;
- Confirm.

Add Action Conditional – 2ª Ação

- Condition;
- Single Condition;
- Upload Load First Value
- Widget State;
- Upload Local File
- Is Set and Not Empty
- **True**
- **Add Action – 3ª Ação**
- Backend Call – Create Document;
- Collection: Medicos;
- Foto;
- From Variable;

- Uploaded File URL 2

11.2.1 Problemas em fazer o Upload

Caso tenha problema em enviar a imagem para o Firebase, siga o procedimento abaixo.

- Entre no Firebase;
- Storage;
- Regras – o sistema traz as regras como consta na imagem abaixo. Substitua o False e o request.auth.uid por True, conforme a segunda imagem.

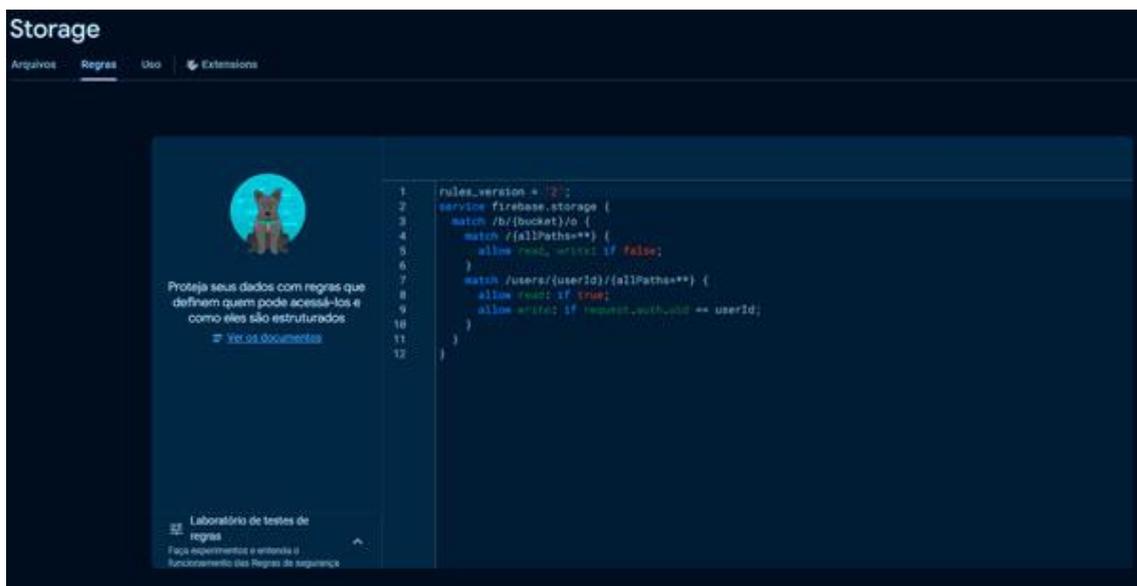


Imagem 1

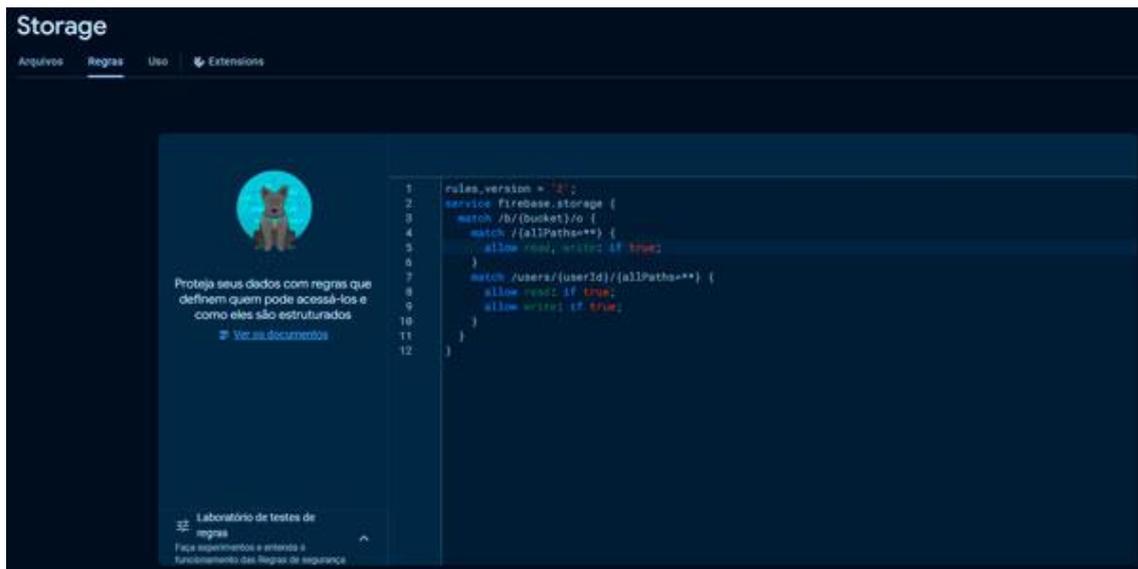


Imagem 2

11.3 Horários Disponíveis

Cada médico terá seu horário de forma personalizada. Para que isso, deve-se criar uma App State listaHorarioMedicoCriacao;

- Entrar em App Values;
- Add App State;
- Field Name: listaHorarioMedicoCriacao;
- String;
- Is list;
- Create;

Voltar para HorariosDisponiveis

- Clicar em cada check box;
- Action;
- Add Action
- Open;
- Clicar em On Toggled One, para colocar a ação
- Update App State;

- Add Field;
- listaHorarioMedicoCriacao;
- Unset;
- Add is List;
- Value to Add;
- 08:00

Voltar na ação anterior e clicar no On Toggled Off

- Update App State;
- Add Field;
- listaHorarioMedicoCriacao;
- Unset;
- Removed is List;
- Value to Add;
- 08:00

OBS: Repetir as duas ações anteriores para todos os horários.

11.4 Botão Criar Médico

- Clicar no botão Criar Médico;
- Action;
- **Add Action – 1ª Ação**
- Firestore;
- Create a Document;
- Unset;
- Coleção: médicos;
- Add Field;
- Nome;
- Unset;
- Widget State;
- Nome;

- Add Field;
- Função;
- Unset;
- Widget State;
- Função;
- Add Field
- Foto;
- Value Source: Uploaded Photo;
- Add Field;
- curriculoMedico;
- Unset;
- Widget State;
- Currículo;
- Add Field;
- horarioTrabalho;
- List Update Type;
- Unset;
- Set Value;
- Unset;
- App State;
- listaHorariosMedicoCriacao;
- Confirm

Add Action – 2ª Ação

- Alert Dialog;
- Informational Dialog;
- Value: Sucesso;
- Value: O médico foi criado;
- Value: Ok

Add Action – 3ª Ação

- Botton Sheet;

- Dismiss

12. DRAWER – PAGINA SUSPENSA PARA APRESENTAR DADOS DO USUÁRIO, DENTRO DA PÁGINA LISTAMÉDICOS

- Ir para a página Lista Médicos;
- Clicar no elemento de maior nível hierárquico (Lista Médico), e inserir o elemento de página: Drawer;
- Colocar os elementos: image, circle image, nome, email, Editar perfil, Meus agendamentos, Administrador, Botão

Para aparecer a Drawer

- Ir para a Página Lista de Médicos;
- Clicar no ícone de hambúrguer, menu superior à direita;
- Action
- Drawer;
- Open Drawer.

12.1 Colocar elementos da Drawer Dinâmicos

- Entrar na ListaMédicos;
- Clicar na Drawer;
- Nome;
- Authenticator;
- Display name.
- Repetir o procedimento para e-mail.
- Clicar Container da Editar Perfil;
- Action;
- Add Action;
- Bottom Sheet;
- Show
- Select Component;

- Página Editar Perfil;

Depois disso, entrar no componente EditarPerfil

12.2 Página Editar Perfil

- **Clicar na imagem;**

- Action;
- Add Action;
- Upload Media;
- Unset;
- Firebase;

Voltar para propriedades;

- Network;
- Path;
- Condition Conditional
- Condition If Then
- **IF** Unset Widget State Upload File URL Is Set and Not Empty Confirm
- Value Unset Widget State File URL;
- **Else** Value Authenticated User photo_URL Confirm.

- **Clicar no nome**

- Ir para propriedades;
- Initial Value;
- Authenticator User;
- Display name;

- **Clicar no telefone**

- Ir para propriedades;
- Initial Value;
- Authenticator User;
- Phone Number.

12.2.1 Botão Atualizar Perfil

- Clicar no botão;
- Action;
- Add Action;
- Backend
- Firestore;
- Update Document;
- Unset;
- Authenticated User;
- User Reference;
- Confirm;
- **Add Field;**
- Display_name;
- User Document;
- Display_name;
- **Add Field;**
- Photo_url;
- From Variable;
- Conditional If/The/Else;
- IF Unset Widget State, Up LoadFile URL Is Set and not Empty Confirm Than Value Widget State UpLoad File URL;
- Else Authenticator User photo_URL. Confirm
- **Add Field**
- Phone_number;
- From Variable;
- Unset;

13. Editar o Administrador na Drawer

- Abrir a Página ListaMédicos;

- Clicar na Drawer;
- Edit Drawer;
- Clicar no Container Administrador;
- Action
- Add Action;
- Navigation to;
- Todos os Atendimentos;

13.1 Página Todos Atendimentos

Depois disso, entrar na Página de Todos os Atendimentos:

13.1.1 Container Calendar

No container calendar criar dois calendários, o primeiro com a data inicial e o segundo com a data final. Este procedimento permitirá filtrar os atendimentos dentro do período de tempo estabelecido.

Criando Duas Variáveis: dataIn e dataFim

- Clique no Canvas;
- Clicar no ícone de variável (terceiro ícone na linha de ações);
- Add Field;
- Field Name: dataIn;
- Type: Date Time;
- Confirm;
- Add Field;
- Field Name: dataFim;
- Type: Date Time;
- Confirm

Clicar no Texto dataIn – do lado do primeiro calendar

- Text;
- Page State;
- dataIn;

- Date Time Format: d/M/y

Clicar no Texto dataFin – do lado do segundo calendar

- Text;
- Page State;
- dataFim;
- Date Time Format: d/M/y

Depois disso, coloque uma Ação na Página, para que as datas se atualizem, quando a página for aberta.

- Clicar no elemento TodosAtendimentos (maior nível hierárquico);
- Action;
- Add Action;
- Update Page State;
- Add Field;
- dataIn;
- Unset;
- Set value;
- Unset;
- Global Properties;
- Current Time;
- Confirm;
- Add Field;
- dataFim;
- Set Value;
- Unset;
- Global Properties;
- Current Time;
- Confirm.

Colocar ação no Botão Calendar dataIn (Data Inicial)

- Clicar no botão Calendar dataIn;

- Action;
- **Add Action- Primeira Ação**
- Date/Time Picker;
- Default Date/Time;
- Value;
- Page State;
- dataIn;
- Confirm;
- Habilitar a linha: Allow Past Date/Times: assim irá permitir selecionar datas anteriores a data atual;
- **Add Action – Segunda Ação**
- Update Page State;
- Add Field;
- dataIn;
- Set Value;
- Unset;
- Widget State;
- Date Picked;
- Confirm.

Colocar ação no Botão Calendar dataFim

- Clicar no botão Calendar dataFim;
- Action;
- **Add Action- Primeira Ação**
- Date/Time Picker;
- Default Date/Time;
- Value;
- Page State;
- dataFim;
- Confirm;

- Habilitar a linha: Allow Past Date/Times: assim irá permitir selecionar datas anteriores a data atual;
- **Add Action – Segunda Ação**
- Update Page State;
- Add Field;
- dataFim;
- Set Value;
- Unset;
- Widget State;
- Date Picked 2;
- Confirm.

Clicar na ListView – BackEnd Query, para buscar dados no Banco de Dados

- Banck End Query;
- Add;
- Query Collection;
- Collection: Agendamentos;
- List of Documents;
- Filters: AND
- Filter 1;
- Collection Field Name: data;
- Relation: Greater Than or Equal To;
- Value:Page State: dataIn;
- Confirm;
- Filter 2;
- Collections Field Name: data;
- Relation: Less Than Equal To;
- Page State: dataFim;
- Confirm.

OBS: Quando você aplica filtros compostos, você precisa criá-lo no Firebase. Às vezes o FlutterFlow acusa o erro e pede pra você criar, direto no FlutterFlow. Mas, nem sempre isso acontece. Assim, você deverá criar direto do Firebase

- Abrir o Firebase;
- Firestore Database;
- Clicar na aba de Índices;
- Adicionar índice;
- Veja os modelos e crie o seu: exemplo:

ID da coleção	Campos indexados	help_outline	Escopo da
Agendamentos	medico Crescente data Crescente	__name__ Crescente	Coleta

Corrigindo Erro na DataFim

A data que irá aparecer na dataFim precisa ser menor que a dataFim, foi assim configurada. A formatação da dataFim é de começar me 0:00 hora, assim qualquer data maior do 0:00 não será computada. Para pegar todas as horas do dia 24, por exemplo, o filtro deverá ser até 23h e 59min. Para corrigir isso, siga o procedimento abaixo:

Volte na ListView e altere o segundo filtro:

- Filter 2;
- Collections Field Name: data;
- Relation: Less Than Equal To;
- Date Time from UNIX (para converter a data para segundo);
- Unset;
- Code Expression;
- Add Argument;
- Var1; Name: dataFim;
- Type: Interger;
- Unset;

- Page State;
- dataFim;
- Unset;
- Seconds;
- Confirm;
- Expression (escreva a expressão, espaço abaixo de Return Type: dataFim + 23h54min (você tem que converter em segundos: 86399 segundos em um dia): dataFim + 86399. Clicar em checar erros; Confirm;
- Time Unit: Seconds; Available options: No Further Changes; Confirm; Confirm.

Clicar no ctnMenuItems – primeiro container abaixo da ListView

- BackEnd Query;
- Add Query;
- Document Reference;
- Medicos;
- Unset;
- Agendamentos Documents;
- Medico (medicos ref); Confirm

Clicar na Imagem – colocar de forma dinâmica

- **Clicar na imagem:**
 - Path;
 - Medicos Document;
 - Foto.
- **Clicar no nome do médico:**
 - Text;
 - Medicos Document;
 - Nome.
- **Clicar na Data:**
 - Text;

- Combine Text;
 - Text 1: Data:
 - Text 2::
 - Agendamentos Documents
 - Data;
 - Date Time;
 - ddMMHm;
 - Confirm;
 - Confirm.
- **Clicar no Container Paciente**
 - BackEnd Query;
 - Add Query;
 - Document Reference;
 - Unset;
 - Users;
 - Unset;
 - Agendamentos Documents
 - Usuário (users ref);
 - Confirm;
 - **Ir para as Propriedades;**
 - Text;
 - Combine Text;
 - Text 1: Paciente:
 - Text 2;
 - Unset;
 - Users Document;
 - Display_name;
 - Confirm.

Volta para o ctn Menultems

- Action;
- Add Action;

- Navigation to;
 - DetalhesAtendimento;
 - Pass;
 - refAgendamento;
 - Unset;
 - Agendamento Document;
 - Reference (Agendamentos Ref);
 - Confirm;
 - Confirm.
-
- **Clicar em Meus Agendamentos;**
 - Navigation to;
 - Meus Atendimentos.
 - Clicar no Botão Sair;
 - Action;
 - Add Action;
 - Authentication;
 - Log Out.

14. PÁGINA DE LOGIN

14.1 Botão Login

- Clicar no botão;
- Auth Log In;
- Auth Provider: e-mail;
- Email Field: EmailAdress;
- PassWord Field: Password;

14.2 Botão Esqueci minha senha

- Antes de fazer a ação do Botão Esqueci minha senha, crie uma Página EsqueciSenha;
- No primeiro container cabeçalho, coloque: ícone de Voltar e um texto: Esqueci minha senha;
- No container corpo: coloque:
 - Text: Email para recuperação de senha;
 - TextField: Coloque seu e-mail; propriedade do textfield:
 - Input Border Type: outline;
 - Filled: habilitar.Border Color[: Alternative, Fill Color: Secudary BackGround;
 - Focus Border; Primary;
 - Error Border Color: red;
 - Border Width: 2;
 - Border Radium: 12;
 - Content Padding:: 24;
 - Habilitar: Auto Fill Hint;
 - Auto Fill Hint Options: email;
 - Cursor color: primary.
- **Botão Recuperar Senha**
 - Action;
 - **Add Action – Primeira Ação**
 - Auth Reset Password;
 - Email;
 - **Add Action – Segunda Ação**
 - Dialog Informational;
 - Atenção; Recuperação de senha enviado para o seu s-mail; Ok.

Depois disso, volte à Página de Login

- Clique no botão EsqueceuSenha;
- Action;
- Add Action;

- Navigation to página EsqueceuSenha

14.3 Botão Criar Conta

- Clicar no botão criar conta;
- Action;
- **Add Action – Primeira Ação**
- Create Account;
- Auth Provider: Email;
- Email Field: Unset: emailAddress_create;
- Password Field: password_create;
- **Add Action – Segunda Ação;**
- Informational Dialog;
- Sucesso, conta criada; Seja bem-vindo ao App de Finanças

15. SISTEMA DE RESERVA DE HORÁRIOS

15.1 Para fazer o sistema de horário de um médico, siga o procedimento:

- Entrar na Página Detalhes Médico;
- Clicar no Botão Reservar Horário;
- Action;
- Add Action;
- Bottom Sheet;
- Show;
- Select Document;
- Horário Agendamento;
- Pass;(passar o Parâmetro para identificar o médico que está reservando o horário);
- refMedico;
- Unset;

- refMedico;
- Confirm

Depois disso, entrar na Página Horário Agendamento;

- Selecionar o 2º container (depois do ctn HorárioAgendamento);
- BackEnd Query;
- Add Query;
- Query Collection;
- Document Reference;
- Unset;
- Collection: médicos;
- Unset;
- refMedico;
- Confirm;
- Confirm

Depois disso, entrar em a Página Horário Agendamento;

- Clicar no Calendar;
- Propriedades;
- Calendar;
- Initial Date;
- Unset;
- Global Properties;
- Current time; (dessa forma, o calendário sempre abrirá no dia e horário atuais);
- Confirm;

OBS: Nestas propriedades, você pode configurar cores, fonte, se quer que o calendário mostre apenas os dias de uma semana, dentre outras.

15.2 Criar Horários de Atendimento

15.2.1 Para criar horários, você deve criar uma Custom Functions Lista de Horário

Para criar os horários é necessário criar uma função, para isso:

- Clicar em Custom Code (menu lateral do FlutterFlow);
- + Add Custom Function;
- Name: criarListaDeHoras;
- Apagar o Fuction Code lá existente e colar o código abaixo:

```

• import 'dart:convert';
• import 'dart:math' as math;
•
• import 'package:flutter/material.dart';
• import 'package:google_fonts/google_fonts.dart';
• import 'package:intl/intl.dart';
• import 'package:timeago/timeago.dart' as timeago;
• import '/flutter_flow/lat_lng.dart';
• import '/flutter_flow/place.dart';
• import '/flutter_flow/uploaded_file.dart';
• import '/flutter_flow/custom_functions.dart';
• import '/backend/backend.dart';
• import 'package:cloud_firestore/cloud_firestore.dart';
• import '/auth/firebase_auth/auth_util.dart';
•
• List<DateTime>? criarListaDeHoras(
•   DateTime? datetime,
•   List<String>? times,
• ) {
•   /// MODIFY CODE ONLY BELOW THIS LINE
•
•   // get a datetime and a list of times in format 08:00 and create
•   a list of times for that day
•   if (datetime == null || times == null) {
•     return null;
•   }
•
•   List<DateTime> listaHoras = [];
•
•   for (String time in times) {
•     List<String> parts = time.split(':');
•     int hour = int.parse(parts[0]);
•     int minute = int.parse(parts[1]);
•     DateTime horaAtual =

```

```

•     DateTime(datetime.year, datetime.month, datetime.day, hour,
minute);
•     listaHoras.add(horaAtual);
•     }
•
•     return listaHoras;
•
•     /// MODIFY CODE ONLY ABOVE THIS LINE
• }

```

- Na janela lateral adicionar As Function Setting:
 - Return Value;
 - Type: Date Time; habilitar: isList, Nullable;
 - Define Arguments;
 - Argument 1: datetime, habilitar Nullable;
 - Arguments 2: times, string, habilitar: isList, Nullable;
 - Save.
- Para testar a função:
 - Clicar em Test Function;
 - Clicar no calendário: definir o dia, ok; definir o horário, ok;
 - Clicar em listString e colocar um horário: 08:00;
 - Add itens: coloque outro horário e assim por diante e depois clique em run. Se estiver tudo certo os horários definidos irão aparecer para aquele dia.

15.2.2 Para criar os horários, você precisa criar uma App State

Primeira App State

- Clicar em App Values;
- Add App State Variable;
- First Name: DiaAgendamento;
- String;
- Não é uma lista;
- Não vai persistir;
- Create

Segunda App State

- First name: HorárioAgendamento;
- Data Type: Date Time;
- Create;

Voltar para a Página Horário Agendamento

Agendar o Dia

- Clicar no text: Selecione o dia que deseja ser atendida;
- Text;
- App State;
- DiaAgendamento;
- Confirm.

Agendar o Horário

- Clicar no text: Selecione o horário;
- Text;
- App State;
- HorárioAgendamento; (não precisa colocar o formato);
- Dê um espaço no Valor Default, para permitir confirmar;
- Confirm.

Colocar Ação no Calendar – na tabela de dias

- Clicar em calendar;
- Action;
- Add Action;
- State Manager;
- UpDate App State;
- Add Field;
- DiaAgendamento;
- Unset;
- Set Value;

- Unset;
- Widget State;
- CalendarSelectDay
- Available Options: Start;
- DateTimeFormat;
- DateTimeFormat Options: EEEE=>wednesday (dia atual);
- Confirm.

Colocar Ação no Calendar – Ícone Calendário na parte superior da folha

- Clicar no ícone;
- Action;
- **Add Action – 1ª Ação**
- Date Time Picker: quando clicar, aparece as possibilidades de colocar: data padrão, data mínima, data máxima, data passada, data futuras; cores. Para o nosso App, fazer as seguintes configuração:
 - Allow Future Date: habilitar;
 - Coloque as cores do seu theme;
- **Add Action – 2ª Ação**
- State Manager;
- UpDate App State;
- Add Field;
- DiaAgendamento;
- Unset;
- Set Value;
- Unset;
- Widget State;
- Date Picket;
- Available Options: Start;
- DateTimeFormat;
- DateTimeFormat Options: EEEE=>wednesday (dia atual);

- Confirm.
- Add Field;
- HorarioListaAgendamento;
- Set Value;
- Unset;
- Custom Function: CriarListaDeHoras;
- Date Time;
- Unset;
- Widget State;
- Date Picked;
- Times;
- Unset;
- medicosDocument;
- HorariodeTrabalho;
- Confirm.

OBS: Voltar na AppState HorárioAgendamento e definir um horário, senão dará erro.

OBS: Em um projeto você não irá usar os dois calendários. Aqui, foi usado, para fins pedagógicos.

Add Action – 3ª Ação

15.2.3 Criar horário de forma personalizada

Criar uma App State:

- Clicar em App Values;
- Add App State Variable;
- HorarioListaAgendamento; Data Type: Date Time; IsList: habilitar;
- Create.

Depois disso, entrar na Página Horário Agendamento

- Clicar no gradview de horário agendamento;
- Clicar em Generate Dymanic Children (ícone depois da BackEnde Query);
- Variable Name: horarioAgendamento;
- Unset;
- App State;
- ListaHorarioAgendamento;
- Confirm.

Depois disso, clique no text dentro do GradView Horário

- Text;
- HorarioAgendamento;
- Format: Date Time;
- Unset: Hm=>;
- Confirm;

Clicar no Calendar

- Action;
- Add Field;
- HoráriodeListadeAgendamento;
- Set Value;
- Custom Function;
- criarListaDeHoras;
- argument;
- Date Time;
- Unset;
- WidGet State;
- Select Day Calendar;
- times;
- Unset;
- MedicosDocument;
- listaHorariodeTrabalho;
- Confirm;

- Confirm;

Clicar no Container onde está o Text de Horário

- Action;
- Add Action;
- App Date Up State
- Set Field;
- HorárioAgendamento;
- Set Value;
- Custom Function.
- HorárioAgendamentoitem;
- Confirm.

15.2.4 Horários Indisponíveis

Criar uma App State

- Clique em App Values;
- Add App State Variable;
- Name: HorariosIndiponiveis;
- Type: Date Time;
- Is Lit: habilitar;
- Create.

Abrir a Página Horário Agendamento

- Clicar no container Calendar;
- Action;
- Add Action – 2ª ação;
- Firestore;
- Query Collection;
- Unset;
- Agendamentos;
- List of Document;

- Filter;
- + Filter; (1º filtro)
- Medico;
- Equal;
- Unset;
- refMedico;
- Confirm;
- Filter + (2º filtro);
- Data;
- Unset;
- Maior ou igual (greater than or Equal To);
- Unset;
- Widget State;
- Calendar Selected Day
- Start;
- Confirm;
- Action Output variable name: RespostadaQueryAgendamento;

Add Action – 3ª Ação

- UpDate App State;
- Add Field;
- HorariosIndisponiveis;
- Set Value;
- Unset;
- Action Out Put;
- RespostaQuereAgendamendo;
- Unset;
- Map List Items;
- Get Document Propriety;
- Unset;
- Data;

- Confirm;
- Confirm.

Abrir a Página Horário Agendamento

Colocar as mesmas ações no ícone do calendario

- Clicar no ícone do Calendário, abaixo da TabBar
- Action;
- Add Action – 2ª ação;
- Firestore;
- Query Collection;
- Unset;
- Agendamentos;
- List of Document;
- Filter;
- + Filter; (1º filtro)
- Medico;
- Equal;
- Unset;
- refMedico;
- Confirm;
- Filter + (2º filtro);
- Data;
- Unset;
- Maior ou igual (greater than or Equal To);
- Unset;
- Widget State;
- Date Time Picker;
- Start;
- Confirm;
- Action Output variable name: RespostadaQueryAgendamento2;

Add Action – 3ª Ação

- UpDate App State;
- Add Field;
- HorariosIndisponiveis;
- Set Value;
- Unset;
- Action Out Put;
- RespostaQueryAgendamendo;
- Unset;
- Map List Items;
- Get Document Propriety;
- Unset;
- Data;
- Confirm;
- Confirm.

15.2.5 Colorir o Container das Horas se acordo com a disponibilidade

- Clicar no container da hora;
- Fill Color;
- Clicar no Unset, para trazer de uma variável;
- Condicional Value;
- If, Then Else;
- If Unset: condition, combine contidion; Unset: OR; colocar 3 condições: 1ª condição: Unset: App State; HorariosIndisponiveis; Unset: List Contain Hour, Unset: HorariodeAgendamento Item; Confirm Confirm.
Or Condition; Singel Condition, First Value, Unset, HorárioAgendamento Item, Confirm, Less Then, Second Value, Unset, HorárioAtual, Global Propertie, Current Time;
Or Unset Condition Single Condition, First Value Unset App State diadeAgendamento Equal Value: domingo; Confirm

- If Then: colocar a cor cinza (indisponível);
- Else If Then: colocar a cor verde padrão (já está selecionado para aquele usuário). Colocar condições para verificar se está selecionado: Unset; Condition, Single Condition, First Value: Unset: App State, HorariodeAgendamento Equal To, Second Value Unset: HorarioAgendamentoItem, Confirm,
- Else: colocar a cor branca (horário disponível)
- Confirm

Depois disso, você precisa alterar as ações do Container de Horas;

- Copiar as três condições do **IF** anteriores; (só as do IF);
- Clicar no container das horas;
- Action
- Open;
- Add Action Conditional – 2ª Ação;
- True: cole as três ações copiadas do IF;
- False: copie a Ação de App State que estava pronta e cole no False. Depois delete a Ação de App State que estava como sendo a primeira.

15.3 Botão Confirmar Agendamento

- Entrar na Página Horário Agendamento;
- Clicar no Botão Confirmar Agendamento;
- Action;
- Open;
- **Add Action 1ª Ação**
- Firestore;
- Create a Document;
- Coleção: Agendamento;
- Set Field;

- Sintomas;
- From Variable;
- Unset;
- Widget State;
- Descreva o que está sentido;
- AddField;
- Médico;
- refMedico;
- Add Field;
- Usuário;
- Authenticador User;
- User reference (users ref);
- Data;
- From Variable;
- Unset;
- App State;
- HorárioAgendamento
- **Add Action – 2ª Ação**
- Firestore;
- Update;
- Unset;
- refMedico;
- Add|Fied;
- NumerodeAtendimento;
- Update (Increment, Delete)
- Increment;
- Value: 1;
- **Add Action – 3ª ação;**
- Alert Dialog;
- Informational Dialog;
- Parabéns;

- Agendamento feito com sucesso;
- Ok;
- Parabéns
- **Add Action – 4ª Ação;**
- Bottom Sheet;
- Dismiss

16. DRAWER – COLOCAR AÇÃO EM MEUS ATENDIMENTOS

- Entrar na página lista de médicos;
- Clicar na Drawer (ela fica escondida, até precisar ser usada);
- Clicar em Meus Agendamentos;
- Action;
- Navigator To;
- Meus Atendimentos.

Depois disso entrar na Página Meus Atendimentos

- Clicar no elemento de maior nível hierárquico;
- Back End Query;
- Add Query;
- Query Collection;
- Unset;
- Agendamentos;
- List of Documet;
- Filter;
- Collection: usuário;
- Unset;
- Equal;
- Unset;
- Authentication;
- User Reference;

- Confirm;
- Confirm.

Depois disso, deixe os dados da Página Meus Atendimentos de forma dinâmica:

- Clicar no text Meus Agendamentos;
- Propriedades;
- Combine Text;
- Text1: escreva: Meus Agendamentos:
- Add Text;
- Text2: Agendamentos Documents;
- Aviable Options: Number of Itens;
- Confirm;
- Confirm

Clicar na List View

- Clicar na Generations Children from Variable (ícone ao lado da BackEnd Query);
- Variable Name: MeusAgendamentos;
- Unset;
- Agendamentos Document;
- Save.

Depois disso, clicar no container abaixo da ListView, para colocar os dados dinâmicos

- BackEnd Query;
- Unset;
- Document from Reference;
- Unset;
- Collection: médicos;
- Unset;
- MeusAgendamentos Item;
- medicosRef;

- Confirm;
- Confirm

Colocar os dados dinâmicos

- Clicar no nome do Médico;
- Propriedade;
- Medicos Document;
- Nome;
- Clicar na Imagem;
- Propriedade;
- MedicosDocument;
- Foto;
- Clicar em Horário;
- Text;
- Combine Text;
- Text1: Horário;
- Text2;
- Unset;
- MeusAgendamentosItem;
- Data;
- Available Options;
- DateTime Format;
- Unset;
- ddMMHHmm;
- Confirm;
- Confirm.

Colocar Ação no Container (abaixo da ListView)

- Clicar no Container;
- Action;
- Add Action;
- Navigation To;

- DetalhedoAtendimento;
- Define;
- refAgendamento;
- Collection: Agendamentos;
- Pass;
- refAgendamento;
- Unset;
- Meus Agendamentos Item;
- Reference.

17. PÁGINA DETALHES ATENDIMENTO (AGENDAMENTO)

17.1 BackEnd Query – Dados do Médico

- Clicar no elemento de maior nível hierárquico;
- BackEnd Query;
- Document fro Reference;
- Unset;
- Agendamento;
- refAgendamento;
- Confirm;
- Confirm.

17.2 Colocar os dados de forma dinâmica

- Selecionar a linha onde estão a foto e o nome do médico;
- BackEnd Query;
- Add Query;
- Document from Reference;
- Collection: medicos;
- Unset;
- Agendamento Documents;

- Medico;
- Confirm;
- Confirm;

Depois disso ir para Propriedade

- **Clicar na Imagem;**
- Unset;
- Medicos Document;
- Foto;
- **Clicar no nome;**
- Unset;
- MedicosDocument;
- Nome;
- **Clicar em especialidade;**
- Unset;
- Medicos Document;
- Função;
- **Clicar em Data;**
- Combine Text;
- Text1: Data;
- Text2:
- Unset;
- Agendamento Document;
- Data;
- Available Options;
- Date Time Format;
- Unset;
- ddMhhmm;
- Confirm;
- Confirm;
- **Clicar em Descrição dos Sintomas do Paciente**

- Unset;
- Combine Text;
- Text1: Descrição dos Sintomas do Paciente;
- Text2;
- Unset;
- Agendamentos Document;
- Sintomas do paciente;
- Confirm

17.3 BackEnd Query – Dados do Paciente

Selecionar o Container do Paciente

- BackEnd Query;
- Query Type;
- Document reference;
- Unset;
- Collection: users;
- Unset;
- Agendamento Document;
- userRef;
- Confirm;
- Confirm.

17.4 Colocar os dados do usuário de forma dinâmica

- Clicar na imagem do usuário;
- Users Document;
- Foto url;
- Confirm;
- Clicar no nome;
- Users Document;
- Display name;
- Clicar no email;

- Users Document;
- Email;
- Clicar em telefone;
- Users Document;
- Phone_number.

17.5 Botão Editar Atendimento

- Clicar no botão Editar Atendimento;
- Action;
- Add Action;
- Navegation to;
- Editar Atendimento;
- Define;
- refAgedamento;
- Collection: Agendamentos;
- Confirm
- Pass;
- refAgendamento;
- Unset;
- refAgendamento;
- Confirm;

18. PÁGINA EDITAR ATENDIMENTO

18.1 BackEnd Query Primeiro Container

- Clicar no primeiro container abaixo do nome Editar Atendimento
- BackEnd Query;
- Add Query;
- Query Type: Document from Reference;
- Agendamentos;
- Unset;

- refAgendamento;
- Confirm;
- Confirm.

Depois disso, entrar na Página DetalhesAtendimento

- Clicar no botão Editar Agendamento;
- Action;
- Add Action – 2ª Ação (esta ação deverá ficar antes da ação que já está feita lá)
- Update App State;
- Add Field;
- DiaAgendamento;
- Unset;
- Set Value;
- Agendamento Documents;
- Data;
- Date Time Format;
- Format: EEE=>
- Confirm;
- Add Field;
- HorárioAgendamento;
- Unset;
- Agendamento Document;
- Data;
- Confirm.

Feito isso, voltar para a Página Editar Atendimento

- Clicar no texto **Dia Atendimento**;
- Propriedades
- Text;
- Unset;

- App State;
- DiaAgendamento.
- Clicar em **Selecione o horário;**
- Text;
- App State;
- HorarioAgendamento.

Clicar em Calendar

- Initial Date;
- Unset;
- Agendamento Document;
- Data.

Adicionar ação no Calendar

Antes adicionar ação no Calendar, será necessário inserir uma BackEnd Query no Container anterior, pra buscar as referências do médico.

- Clicar no container;
- BackEnd Query;
- Add Query;
- Document from Reference;
- Collection medicos;
- Unset;
- Agendamentos Document;
- Medicos ref;
- Confirm;
- Confirm

Ação do Calendar

- Clicar em Action;
- **Add Action – 1ª Ação**
- Update App State;
- DiaAgendamento;
- Unset.

- Set Value;
- Widget State;
- Calendário;
- Start;
- Date Time Format;
- EEEE=>;
- Confirm;
- Add Field;
- HorarioListaAgendamento;
- Unset;
- Set Value;
- Unset;
- Custom Function;
- criarListaDeHoras;
- DateTime;
- Type DateTime;
- Unset;
- CalendarSelectDay;
- start
- Confirm;
- Times;
- Unset;
- Medicos Document;
- horarioTrabalho;
- Confirm;
- Confirm;
- **Add Action – 2ª Ação**
- Firestore;
- Query Collection;
- Collection: Agendamentos;
- Lis of Document;

- Filter+
- Filter 1;
- Unset;
- Medico;
- Relation: Equal;
- medicosRef;
- Reference;
- Confirm;
- Filter +;
- Filter 2;
- Data;
- Relation: Greater or Equal To;
- Unset;
- Widget State;
- CalendarSelect Day;
- Unset;
- Start;
- Action OutputVarient name: Resposta QueryAgendamento;
- **Add Action – 3ª Action;**
- Update App State;
- Add Field;
- HorariosIndisponiveis;
- Unset;
- Set Value;
- Action Output;
- Unset;
- RespostaQueryAgendamento;
- Unset;
- Map List Items;
- Item in List;
- Get Document Property;

- Unset;
- Data;
- Confirm;
- Confirm.

Clicar no GridView

- Clicar no Generating Children from Variable (ícone após a BackEnd Query);
- Variable Name: ListaHoras;
- Unset;
- App State;
- HorarioListaAgendamento;
- Confirm;
- Save

18.2 Horário de Atendimento – Forma dinâmica

Clicar no Container Horário Atendimento – 1º ctn dentro do GridView

- Propriedades;
- Clicar no text do horário;
- Unset;
- listaHoras Item;
- DateTime Format;
- Unset;
- Hm=>;
- Confirm

Para colocar cor no Container

- Clicar no container do horário;
- Fill Color;
- Unset;
- Conditional Value;
- **IF;**

- Condition;
- Combinada;
- **OR;**
- First Condition;
- Unset;
- App State;
- HorariosIndisponiveis;
- Available Options;
- List Contains Items;
- Unset;
- ListaHoraltem;
- Confirm;
- Confirm
- **OR;**
- Unset;
- Condition;
- Single Condition;
- First Value;
- Unset;
- ListaHora Item
- Confirm;
- Less Than;
- Second Value;
- Global Properties;
- Current Time;
- Confirm;
- Confirm;
- Add Condition;
- Unset;
- Single Condition;
- App State;

- DiaAgendamento;
- Equal to;
- Second Value;
- Value: domingo;
- Confirm;
- Then: (para colocar a cor no botão), Unset, Cinza (escolha a cor na paleta);
- **Else IF**
- Unset;
- Condition;
- Single Condition;
- Unset;
- App State;
- HorárioAgendamento;
- Confirm;
- Equal To;
- Unset;
- ListHoras Item;
- Confirm;
- Confirm;
- **Then:** (para colocar a cor no botão) Unset: Verde
- Else (se o horário estiver livre coloque a cor branca), Unset, branca;
- Confirm.

Container Horário Atendimento – Colocar Ação

Ainda no Container selecionado, ir para Action;

- Add Action;
- Update App State;
- Add Field;
- HorariodoAgendamento;
- Unset;

- Set Value;
- Unset;
- ListaHora Item;
- Confirm;

Como não pode agendar em um dia anterior, deve-se colocar as condições que foram feitas no Container, na área de Propriedades;

- Ir para propriedade deste Container;
- Clicar em Fill Color (onde já foram feitas as condições);
- First Conditions: clique nas 3 condições e copie;
- Volte para as Ações deste container;
- Add Action Conditional, cole as ações copiadas;
- True:
- Add Action;
- Alert Informational: Cuidado! Este horário não pode ser selecionado; Ok.
- No False cole a primeira Ação, realizada (Update App State), depois delete esta primeira ação.

Clicar no ícone do Calendário (acima do DiaAgendamento);

- Action;
- Add Action;
- Update App State;
- HorárioAgendamento;
- Set Value;
- Data.

18.3 Descrição: Descreva o que está sentido

- Selecione o text Fiel Descricao;
- Initial value;
- Agendamento Document;

- Sintomas do paciente.

18.4 Botão Confirmar Agendamento

- Clicar no botão;
- Action;
- Open;
- **Add Action – 1ª Ação**
- Firestore;
- UpDate Document;
- Unset;
- refAgendamento;
- Confirm;
- Add Field;
- Sintomas do Paciente;
- From Variable;
- Unset;
- Widget State;
- Sintomas paciente;
- Não precisa atualizar o médico e nem o usuário;
- Data;
- From Variable;
- Unset;
- App State;
- HorárioAgendamento;
- Confirm
- **Add Action – 2ª Ação**
- Dialog Alert;
- Confirmation Alert;
- Você tem certeza?
- Isso irá atualizar as informações desse agendamento;
- Cancelar;

- Confirmar;
- **Add Action – 3ª Ação**
- Conditional Action;
- Copie a primeira ação e cole em True (depois apague a primeira ação);
- False;
- Add Action;
- Close Dialog.

19. PUBLICAR O APP NA WEB

- Entrar no FlutterFlow;
- 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, 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 no 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>.