

Documentação de Desenvolvimentos de aplicativos móveis

Com flutterflow

Projeto: atividade final

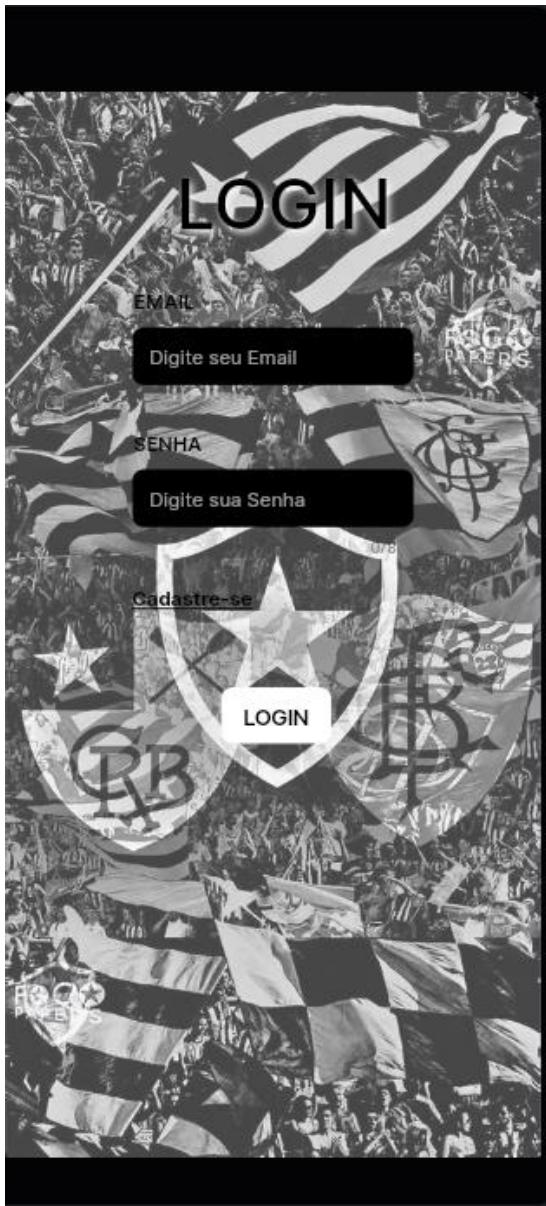
João Bernardo Costa de Oliveira

Dezembro/2025

Índice

1. tela de login
2. tela de cadastro
3. tela de home
4. tela de loja
5. tela de dados do usuário
6. tela de endereço

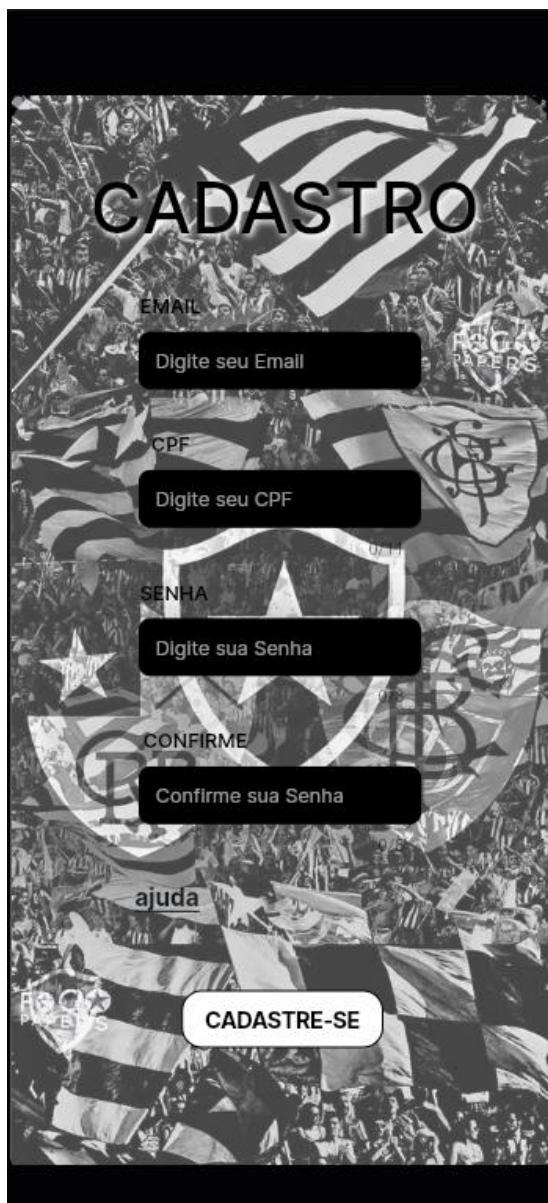
1. Tela de login



A tela de login foi criada seguindo o mesmo estilo das outras, com a imagem de fundo aplicada de maneira integral. O texto “Login” foi colocado no topo da página. Foram adicionados dois campos principais: email e senha, com suas respectivas instruções de preenchimento. Abaixo dos campos foi inserido um pequeno texto “Cadastre-se” que poderá futuramente servir como navegação para a tela de registro. Por fim, um

botão centralizado com a palavra “Login” foi adicionado para realizar a ação de entrada no sistema.

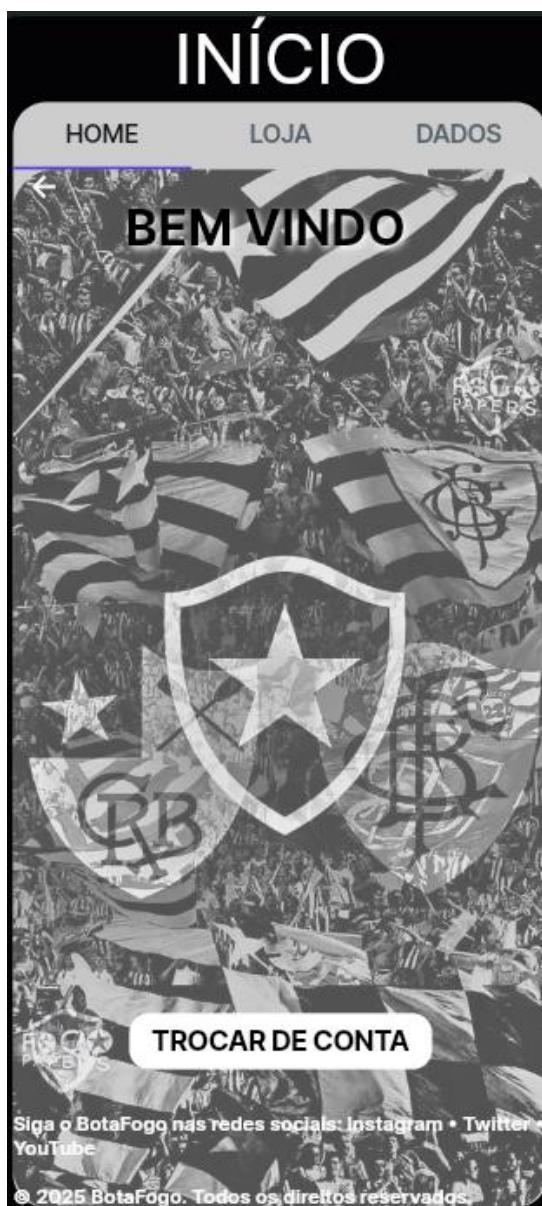
2. Tela de Cadastro



A tela de cadastro foi criada no FlutterFlow a partir de uma nova página, onde primeiro foi configurada uma imagem de fundo ocupando toda a tela. No topo, adicionou-se um texto centralizado com a palavra “Cadastro”. Logo abaixo foram inseridos os campos do formulário, cada um com um rótulo e um placeholder orientando o que o usuário deve digitar. Os campos utilizados foram: email, CPF, senha e confirmação de senha. Após os campos, foi colocado um pequeno texto de apoio (“ajuda”), seguido de um

botão principal “Cadastre-se”, posicionado ao centro, que servirá posteriormente para enviar os dados do formulário.

3. Tela de Home



Na primeira aba (Home), foi colocado um texto de boas-vindas centralizado no topo indicando que o usuário está na tela inicial. Abaixo, manteve-se a mesma imagem de fundo utilizada nas outras telas para padronizar o estilo visual. Também foi adicionado um botão com o texto “Trocar de Conta” posicionado ao centro da tela na parte

inferior, que pode ser configurado futuramente para realizar alguma ação de logout ou mudança de perfil. Elementos extras, como pequenos textos de rodapé com redes sociais e direitos autorais, foram colocados na base da tela para complementar a interface.

4. Tela da loja



A segunda aba (Loja) contém outra estrutura: o título “Loja” centralizado no topo abaixo da Tab Bar e dois cartões de produtos exibindo imagens de camisas com seus

valores (por exemplo, R\$250 e R\$300). Esses cartões foram inseridos em uma coluna ou list view para manter a apresentação organizada. Abaixo da lista de produtos, foi adicionado um botão “confirmar Compra”, centralizado, para que o usuário possa confirmar a ação de adquirir um item selecionado. Cada item visual foi montado com imagem e preço para simular um catálogo de produtos.

5. Tela dos Dados



A terceira aba (Dados) apresenta um formulário de preenchimento de informações do usuário. O título “Dados” aparece logo abaixo da Tab Bar. Foram inseridos campos de texto com rótulos e placeholders para que o usuário possa digitar seu email, nome, CPF, cidade e selecionar o estado a partir de um menu suspenso. Todos os campos ficaram alinhados verticalmente com espaçamento entre eles para facilitar a leitura e o preenchimento. Ao final da lista de campos, foi colocado um botão “Salvar” centralizado para gravar ou enviar os dados preenchidos.

Em todas as abas, a experiência visual foi mantida com a mesma imagem de fundo para reforçar a identidade do aplicativo, e os textos e botões foram distribuídos de forma clara e intuitiva. A navegação entre as abas acontece ao tocar sobre os nomes Home, Loja ou Dados na Tab Bar, trocando o conteúdo exibido sem sair da tela principal

6. Tela de Endereço



Na tela de endereço, uma nova página foi criada e recebeu a mesma imagem de fundo para manter padrão visual entre as telas. O título “Endereço” foi posicionado no topo. Abaixo, foram organizados diversos campos relacionados ao endereço do usuário, também contendo rótulos e placeholders. Os campos utilizados foram: CEP, número, bairro, complemento, cidade, logradouro e referência. Para o campo de estado foi utilizada uma lista suspensa (dropdown) permitindo a seleção da UF. Todos os campos foram distribuídos em coluna, com espaçamento regular para facilitar a leitura e o preenchimento.

Para todas as telas, manteve-se a estética unificada com planos de fundo iguais, campos escuros para boa leitura, textos centralizados nos títulos e botões claros para maior contraste. A navegação entre as páginas pode ser configurada diretamente no FlutterFlow utilizando as ações dos botões conforme o fluxo desejado.

API CALLS

1. CADASTRO USUÁRIO

Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test

API Call Name: Cadastro

Method Type: POST API URL: https://agora-vai-lrkl.onrender.com/Usuario/salvar

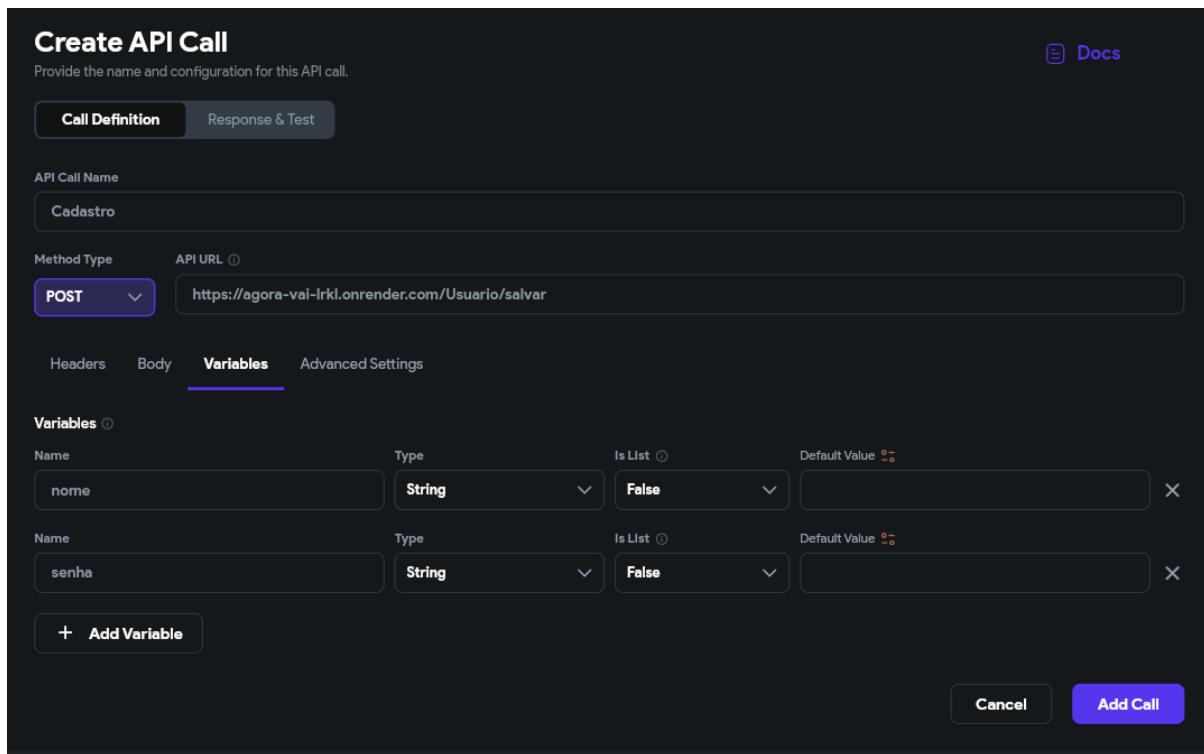
Headers Body Variables Advanced Settings

Variables

Name	Type	Is List	Default Value
nome	String	False	
senha	String	False	

+ Add Variable

Cancel Add Call



Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test Test API Call

Preview and Test ⓘ

Variables			Preview		
Name	Value	Include	API URL	Headers	Body
nome	joao	<input checked="" type="checkbox"/>			Test Response
senha	123	<input checked="" type="checkbox"/>			Body JSON

```
1 {  
2   "id": 14,  
3   "nome": null,  
4   "senha": null,  
5   "matricula": null  
6 }
```

Status: 200 (Success)

Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test Test API Call

Preview and Test ⓘ

Variables			Preview		
Name	Value	Include	API URL	Headers	Body
nome	joao	<input checked="" type="checkbox"/>			Test Response
senha	123	<input checked="" type="checkbox"/>			Body JSON

Response Type ⓘ

Data Type

Parse as Data Type ⓘ

JSON Paths Tutorial ⓘ

Selected

JSON Path ⓘ	Name	Response Preview ⓘ	Type	Preview
+ Add JSON Path				

Recommended

JSON Path	Response Preview	Preview
\$id	14	
		+ Add JSON Path

2. CADASTRO ENDEREÇO

Create API Call
Provide the name and configuration for this API call.

Call Definition Response & Test

API Call Name: endereco

Method Type: POST API URL: <https://agora-vai-lrkl.onrender.com/Lab/salvar>

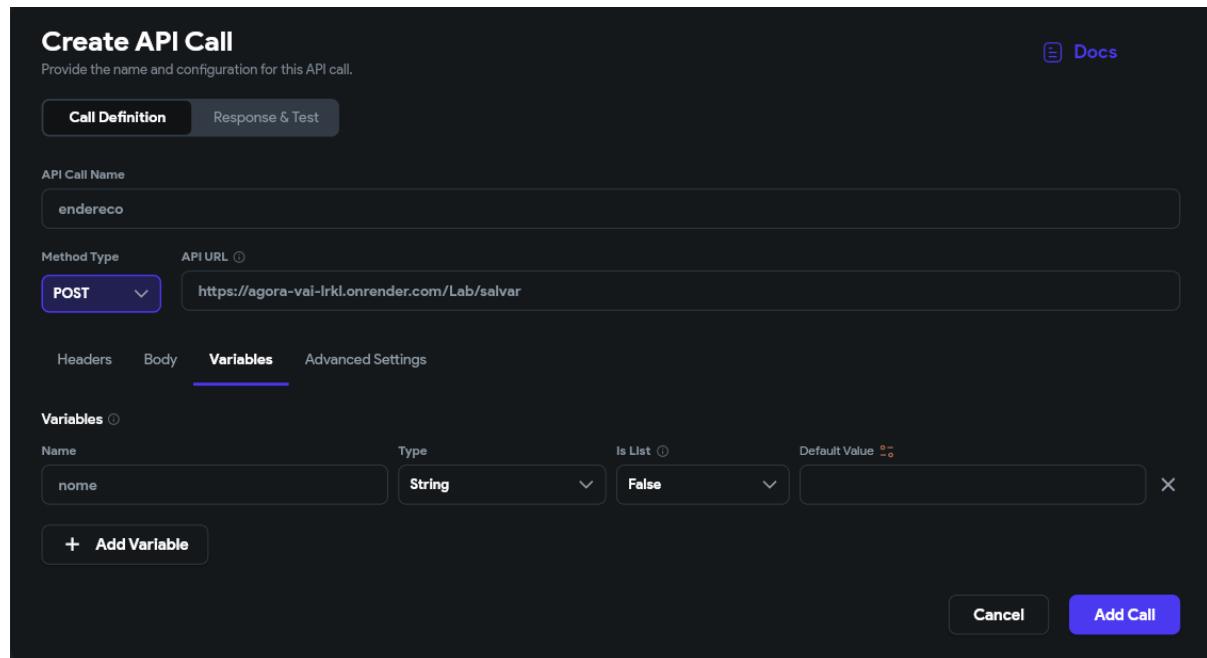
Headers Body Variables Advanced Settings

Variables:

Name: nome	Type: String	Is List: False	Default Value:
------------	--------------	----------------	----------------

+ Add Variable

Cancel Add Call



Define API Call
Provide the name and configuration for this API call.

Call Definition Response & Test Test API Call

Preview and Test:

Variables:	Name: nome, Value: caxias, Include: <input checked="" type="checkbox"/>	Preview:	API URL: https://agora-vai-lrkl.onrender.com/Lab/salvar
------------	---	----------	---

Response Type:

Data Type: Parse as Data Type:

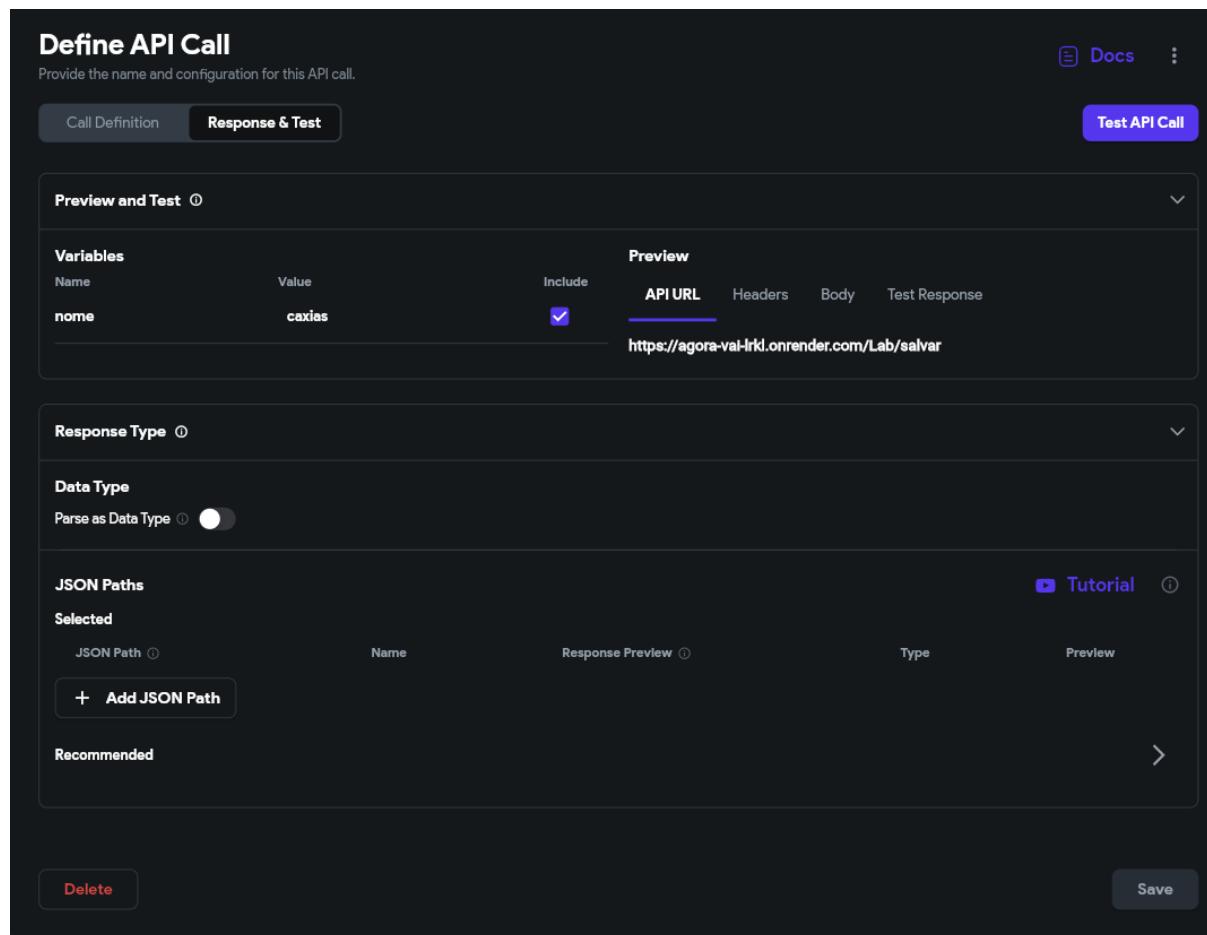
JSON Paths:

Selected:

JSON Path	Name	Response Preview	Type	Preview
+ Add JSON Path				

Recommended:

Delete Save



Define API Call

Provide the name and configuration for this API call.

Call Definition Response & Test **Test API Call**

Preview and Test ⓘ

Variables

Name	Value	Include
nome	caxias	<input checked="" type="checkbox"/>

Preview

API URL Headers Body **Test Response**

Body JSON Raw Body Headers

```
1 {  
2   "id": 7,  
3   "nome": null  
4 }  
5  
6
```

Status: 200 (Success)

3. DADOS DO USUÁRIO

Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test

API Call Name: Dados do usuário

Method Type: GET API URL: https://agora-vai-lrkl.onrender.com/Usuario/search/13

Variables tab selected. Variable: id (Type: Integer, Value: 13, Is List: False, Default Value: null)

+ Add Variable

Cancel Add Call

Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test

Test API Call

Preview and Test tab selected. Variables: id (Value: 13, Include checked). Preview tab selected. Test Response tab selected. Body JSON: { "id": 13, "nome": "joao", "senha": "123", "matricula": null }

Status: 200 (Success)

Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test Test API Call

Preview and Test

Variables

Name	Value	Include
id	13	<input checked="" type="checkbox"/>

Preview

API URL Headers Test Response

<https://agora-val-irki.onrender.com/Usuario/search/13>

Response Type

Data Type

Parse as Data Type

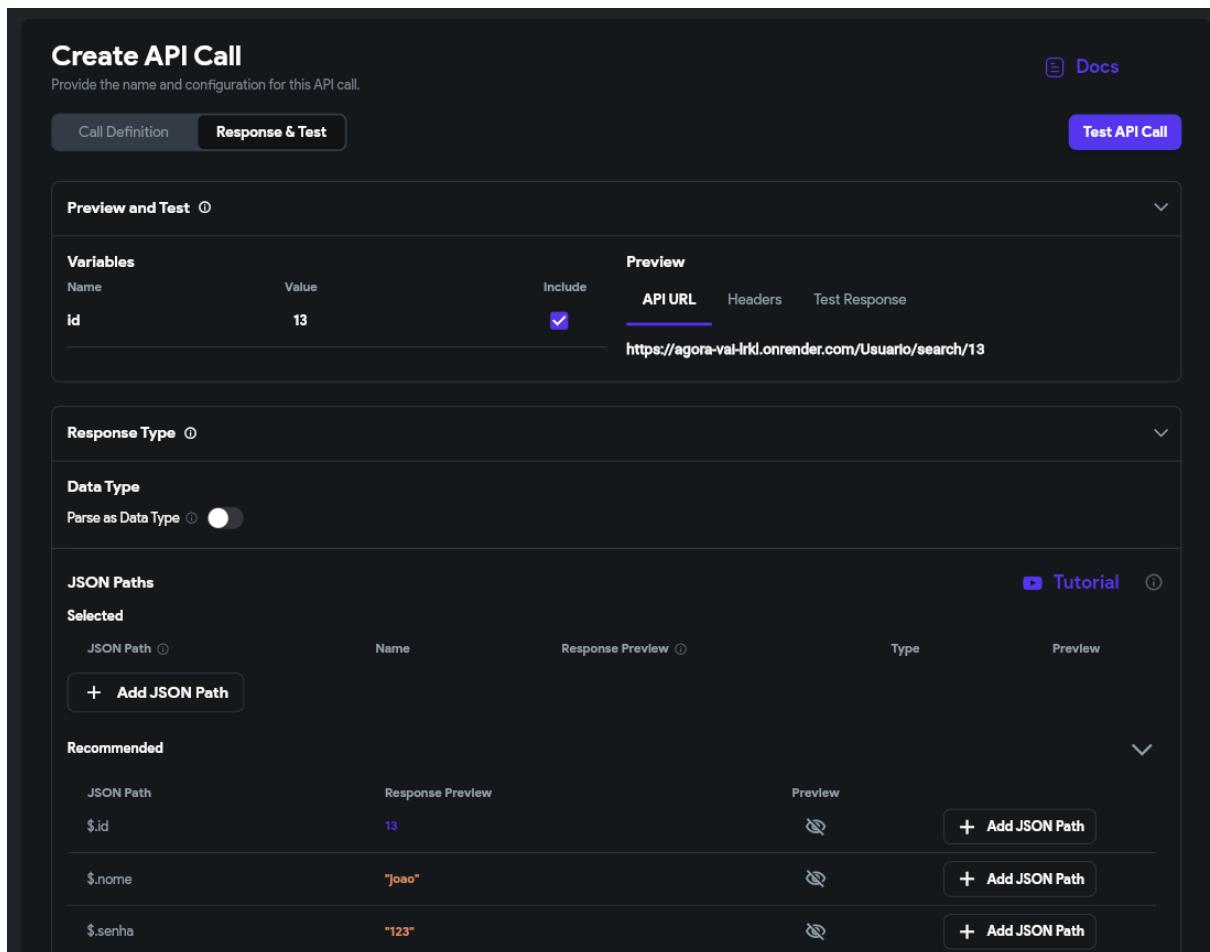
JSON Paths

Selected

JSON Path	Name	Response Preview	Type	Preview
+ Add JSON Path				

Recommended

JSON Path	Response Preview	Preview	+ Add JSON Path
\$.id	13		+ Add JSON Path
\$.nome	"Joao"		+ Add JSON Path
\$.senha	"123"		+ Add JSON Path



4. Login

Create API Call

Provide the name and configuration for this API call.

Call Definition Response & Test Test API Call

Preview and Test

Variables

Name	Value	Include
nome	joao	<input checked="" type="checkbox"/>
senha	123	<input checked="" type="checkbox"/>

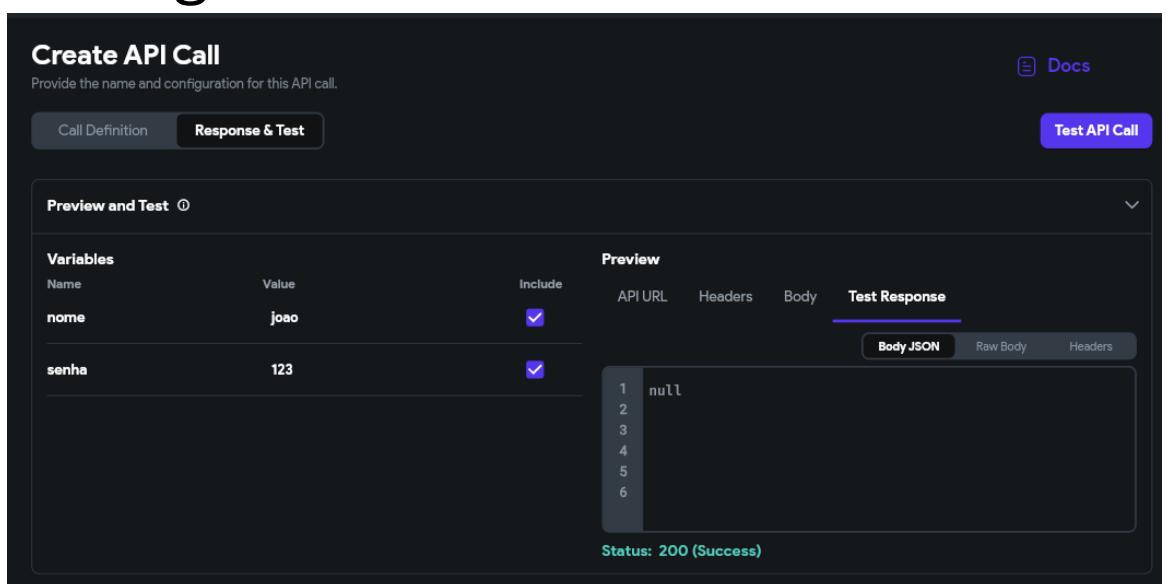
Preview

API URL Headers Body Test Response

Body JSON Raw Body Headers

1 null
2
3
4
5
6

Status: 200 (Success)



Create API Call

Provide the name and configuration for this API call.

[Docs](#)

[Call Definition](#) [Response & Test](#)

API Call Name: `login`

Method Type: `POST` API URL: `https://agora-va-lrkl.onrender.com/Usuario?nome=joao&senha=123`

Headers Body **Variables** Advanced Settings

Variables

Name	Type	Is List	Default Value
nome	String	False	
senha	String	False	

+ Add Variable

[Cancel](#) [Add Call](#)

Create API Call

Provide the name and configuration for this API call.

[Docs](#)

[Call Definition](#) [Response & Test](#) [Test API Call](#)

Preview and Test

Name	Value	Include
nome	joao	<input checked="" type="checkbox"/>
senha	123	<input checked="" type="checkbox"/>

Preview

API URL Headers Body Test Response

`https://agora-va-lrkl.onrender.com/Usuario?nome=joao&senha=123`

Response Type

Data Type

Parse as Data Type

JSON Paths [Tutorial](#)

Selected

JSON Path	Name	Response Preview	Type	Preview

+ Add JSON Path

Recommended

[Cancel](#) [Add Call](#)

Para criar uma API Call no FlutterFlow, primeiro vá em **Settings**, abra a seção **API Calls** e clique em **Add API Call** para adicionar uma nova chamada. Dê um nome para essa chamada e escolha o método HTTP que você vai usar (GET, POST, PUT ou DELETE). Em seguida, insira a URL do seu endpoint. Se a API precisar de cabeçalhos, adicione-os na seção **Headers**, como por exemplo "Content-Type: application/json" ou "Authorization". Caso a API envolva o envio de dados, como nas requisições POST ou PUT, insira o JSON do corpo (body) e defina variáveis para os campos que serão preenchidos pelo seu app. Após isso, clique no botão **Test API Call** para testar e verificar se a resposta da API está funcionando corretamente. Se a API retornar dados, configure o **Response JSON Path** para mapear os campos da resposta. Por fim, salve a API Call e ela estará pronta para ser usada em ações do seu app, como em botões, selecionando a opção **Backend Call** e vinculando os valores dos campos da página.