



FACULDADE DE TECNOLOGIA DE TAUBATÉ

HENRIQUE MARQUES M. DA SILVA

NILSON ANEAS FERRAZ

***APLICATIVO DE BUSCA E PUBLICAÇÃO DE RECEITAS
COM BASE EM INGREDIENTES E UTENSÍLIOS PRÉ-
DEFINIDOS***

TAUBATÉ

2023



FACULDADE DE TECNOLOGIA DE TAUBATÉ

HENRIQUE MARQUES M. DA SILVA

NILSON ANEAS FERRAZ

***APLICATIVO DE BUSCA E PUBLICAÇÃO DE RECEITAS
COM BASE EM INGREDIENTES E UTENSÍLIOS PRÉ-
DEFINIDOS***

Trabalho de Graduação apresentado à Coordenação do Curso Superior de Tecnologia em Análise e Desenvolvimento de sistemas do Centro Estadual de Educação Tecnológica Paula Souza para a obtenção do diploma de Tecnólogo em junho de 2023

Orientador: Prof. Esp. Luis Felipe Feres Santos

Co-Orientador: Prof. Me. Luiz Eduardo Souza Evangelista

TAUBATÉ

2023

**HENRIQUE MARQUES M. DA SILVA
NILSON ANEAS FERRAZ**

***APLICATIVO DE BUSCA E PUBLICAÇÃO DE RECEITAS
COM BASE EM INGREDIENTES E UTENSÍLIOS PRÉ-
DEFINIDOS***

Trabalho de Graduação apresentado a Faculdade de
Tecnologia de Taubaté, como parte das exigências
para a obtenção do diploma de Tecnólogo em junho
de 2023

Orientador: Prof. Esp. Luís Felipe Feres Santos

**Co-orientador: Prof. Me. Luiz Eduardo Souza
Evangelista**

Taubaté, 29 de maio de 2023.

BANCA EXAMINADORA

Prof. Luís Felipe Feres Santos
Fatec de Taubaté

Prof. Me, Luiz Eduardo Souza Evangelista
Fatec de Taubaté

Prof. Me. Jose Jean Peixoto Negrão
Fatec de Taubaté

Dedicamos este trabalho aos nossos familiares e colegas de faculdade pelo apoio, incentivo e exemplo diário de luta e fé.

AGRADECIMENTOS

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Aos nossos pais, que proporcionaram toda condição necessária para a conclusão desta faculdade.

Ao professor Luis Felipe Feres e Luis Evangelista, pelo inestimável apoio na orientação deste trabalho.

Aos colegas de classe, com quem nesses anos de estudo tive a felicidade de conviver.

Sempre que te perguntarem se podes fazer um trabalho, responde que sim e te ponhas em seguida a aprender como se faz.

(F. Roosevelt)

RESUMO

Muitos cozinheiros profissionais e amadores procuram aplicativos com algumas facilidades, como buscas de receitas por nome ou tipo (doce ou salgado). Com o evento da pandemia do *covid-19*, que se iniciou em 2020, muitos cozinheiros se viram com dificuldades em cozinhar, devido à escassez de alimentos. Porém, no mercado não se encontram muitos aplicativos que dão a opção do usuário em realizar buscas de receitas por ingredientes e utensílios que o usuário possui em sua cozinha. Como é uma busca refinada, esses usuários gastam seu tempo em procura por aplicativo que o satisfaça. Durante a pandemia, a procura por aplicativos de receitas culinárias aumentou. Com isso, a demanda por novos aplicativos aumentou a tal ponto que mais pessoas estão buscando a área de programação para aprender, desenvolver ou trabalhar em projetos de aplicativos. Como muitos dos novos profissionais não dominam muito a lógica de programação devido aos códigos e linguagem, para eles, complexa, novas plataformas vêm desenvolvendo o método *no-code* para programação. O *no-code* ou a programação sem códigos” vem facilitando a vida de desenvolvedores amadores e diminuindo o tempo de programação. Visando a dificuldade dos usuários que buscam por aplicativos mobile que lhe possa propor uma busca por receita de maneira mais refinada e a produção de um aplicativo fácil de se desenvolver, os autores deste trabalho criaram um aplicativo mobile em que o usuário possa realizar buscas de receitas culinárias por nome, ingredientes específicos e por utensílios, desenvolvido pela tecnologia no-code da plataforma *Flutter Flow*.

Palavras-Chave: programação; *no-code*; *Flutter Flow*; Aplicação Mobile

ABSTRACT

Many professional and amateur cooks look for applications with some facilities, such as searching for recipes by name or type (sweet or savory). With the event of the covid-19 pandemic, which began in 2020, many cooks found it difficult to cook due to food shortages. However, there are not many applications on the market that give the user the option of searching for recipes by ingredients and utensils that the user has in their kitchen. As it is a refined search, these users spend their time looking for an application that satisfies them. During the pandemic, demand for culinary recipe apps has increased. As a result, the demand for new applications has increased to such an extent that more people are looking to the programming area to learn how to develop or work on application projects. As many of the new professionals do not master the programming logic due to the codes and language, for them, complex, new platforms have been developing the no-code method for programming. No-code or “programming without codes” has been making life easier for amateur developers and reducing programming time. Aiming at the difficulty of users who search for mobile applications that can propose a more refined search for fear and the production of an application that is easy to develop, the authors of this work created a mobile application in which the user can search for culinary recipes by name, specific ingredients and utensils, developed using the Flutter Flow platform's no-code technology. The work is still in development.

Keywords: programming; no-code; FlutterFlow; Mobile Application

LISTA DE FIGURAS

Figura 1 Interface do aplicativo “Receitas Fáceis” _____	23
Figura 2 Interface do aplicativo “Tasty Receitas” _____	24
Figura 3 Interface do aplicativo “Tudo Gostoso Receitas” _____	25
Figura 4 Interface do aplicativo “Cozinha Fit & Fat” _____	26
Figura 5 Interface do aplicativo “Livro de Receitas” _____	27
Figura 6 Interface do aplicativo “Receitas Veganas em Português” ____	28
Figura 7 Interface do aplicativo “Petitichef” _____	29
Figura 8 Diagrama de caso de uso UML do aplicativo Prato Feito ____	43
Figura 9 DER do aplicativo Prato Feito _____	44
Figura 10 Diagrama de sequência do Administrador _____	46
Figura 11 Diagrama de sequência do Usuário _____	47
Figura 12 MER do aplicativo Prato Feito _____	49
Figura 13 Prototipagem da tela “Home Page” _____	50
Figura 14 Prototipagem da tela “Menu principal” e da tela de autenticação para cadastrar ou editar receitas _____	51
Figura 15 Prototipagem da tela “Tela de CadastroRecLogin” _____	52
Figura 16 Prototipagem da tela “Cadastro de receita” _____	53
Figura 17 Prototipagem da tela “Edição de receita” _____	54
Figura 18 Prototipagem da tela “Escolha de Pesquisas” _____	55
Figura 19 Prototipagem da tela “Pesquisar por nome” _____	56
Figura 20 Prototipagem da tela “Pesquisa por utensílios” _____	57
Figura 21 Prototipagem da tela “Pesquisar por ingredientes” _____	58
Figura 22 Prototipagem da tela “Resultado de pesquisa por nome de receita” _____	59
Figura 23 Prototipagem da tela “Resultado de pesquisa por utensílios”	60
Figura 24 Prototipagem da tela “Resultado de pesquisa por ingredientes” _____	61
Figura 25 Exemplo do uso da coluna de propriedades de objeto ____	62
Figura 26 Exemplo de programação para atribuição do FlutterFlow ao Firebase _____	63

Figura 27 Exemplo de programação para atribuição do FlutterFlow ao Firebase _____	63
Figura 28 Exemplo de programação para atribuição do FlutterFlow ao Firebase _____	64
Figura 29 Criação da tabela usuário e variáveis para registro no banco de dados _____	65
Figura 30 Nomeação da receita e lista de escolhas de itens _____	66
Figura 31 Botão com a ação de salvar a receita _____	67
Figura 32 Atribuição dos "dropdows e textfields" (nome, utensílios, ingredientes e modo de preparo) à tabela receita _____	68
Figura 33 Criação da caixa de diálogo após-cadastro _____	68
Figura 34 Tela do Menu Principal _____	69
Figura 35 Tela pesquisa por utensílios. Ações atribuídas para busca no banco de dados _____	70
Figura 36 Tela pesquisa por utensílios. Programação do Algolia _____	71
Figura 37 Tela Menu Principal _____	72
Figura 38 Tela de "Edição de Receita" _____	73
Figura 39 Tela de "Edição de Receita", componente parametrizado _____	74
Figura 40 Tela de "Edição de Receita", componente parametrizado com o botão "Salvar" _____	74
Figura 41 Acesso ao teste do Prato Feito _____	75
Figura 42 Carregamento do teste do aplicativo _____	76
Figura 43 A figura representa o caminho da tela "Homepage" "a" até ao login "c" para cadastro de receita _____	76
Figura 44 A figura representa as etapas para o cadastro da receita _____	77
Figura 45 Galeria de imagens do computador _____	78
Figura 46 A figura apresenta a escolha da imagem ("a") e a mensagem da confirmação do registro da receita ("b") _____	78
Figura 47 A figura apresenta o caminho para pesquisa de uma receita pelo nome _____	79
Figura 48 A figura apresenta o caminho para pesquisa de uma receita por ingredientes _____	80
Figura 49 A figura apresenta os passos para a edição de uma receita _____	81

Figura 50 A figura apresenta a confirmação da alteração da receita no aplicativo_____ 82

Figura 51 A figura apresenta a confirmação da alteração da receita no Firebase_____ 82

LISTA DE ILUSTRAÇÕES

TABELAS

Tabela 1 Requisitos funcionais e não-funcionais_____	42
---	----

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicação
DER	Diagrama Entidade-Relacionamento
GPS	Sistema de Posicionamento Global
IDE	Ambiente de Desenvolvimento Integrado
MER	Modelo Entidade-Relacionamento
RH	Recursos Humanos
TI	Tecnologia da Informação
UML	Linguagem de Modelagem Unificada
WI-FI	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO	16
2 DESENVOLVIMENTO	18
2.1 BUSCA POR RECEITAS CULINÁRIAS	18
2.2 TECNOLOGIA <i>MOBILE</i>	18
2.2.1 APLICATIVOS <i>MOBILE</i>	20
2.3 TECNOLOGIA <i>NO-CODE</i>	29
2.3.1 <i>FLUTTER FLOW</i>	31
2.4 BANCO DE DADOS	31
2.4.1 TIPOS DE BANCOS DE DADOS	32
2.4.2 <i>FIREBASE</i>	33
2.4.3 <i>ALGOLIA</i>	34
3 MÉTODOS	35
3.1 LEVANTAMENTO DE REQUISITOS	35
3.2 DIAGRAMA <i>UML</i>	36
3.3 DIAGRAMA DE SEQUÊNCIA	36
3.4 DER	37
3.5 MER	39
3.6 PROTOTIPAGEM	40
4 RESULTADO E DISCUSSÕES	42
4.1 LEVANTAMENTO DE REQUISITOS	42
4.2 DIAGRAMA DE CASO DE USO <i>UML</i>	42
4.3 DER	43
4.4 DIAGRAMA DE SEQUÊNCIA	45
4.5 MER	48
4.6 PROTOTIPAGEM	50
4.7 RESULTADOS OBTIDOS	61

4.7.1 PROGRAMAÇÃO	61
4.7.2 BANCO DE DADOS	62
4.7.3 TESTE DO PROGRAMA	75
5 CONCLUSÕES	83
REFERÊNCIAS	85

1 INTRODUÇÃO

De acordo com Fluvele (2019), cozinheiros profissionais e amadores enfrentam certas dificuldades ao tentar preparar o prato perfeito, incluindo a falta de ingredientes e utensílios necessários. Em uma cozinha industrial ou comercial, o chef precisa manter a despensa sempre abastecida, mas ainda assim pode encontrar desafios quando os clientes pedem algo específico e o ingrediente necessário não está disponível. Para os cozinheiros domésticos, essas dificuldades podem ser exacerbadas pela diferença no poder aquisitivo de cada um.

Quando um cozinheiro inexperiente tenta criar suas próprias receitas, geralmente começa procurando por ideias em *sites* de busca ou em livros de culinária. No entanto, muitas vezes eles não têm todos os ingredientes e utensílios necessários para a receita desejada. Ao contrário, os cozinheiros profissionais geralmente verificam os ingredientes disponíveis antes de escolher uma receita (VALENTE, 2002). Durante a pandemia da COVID-19, em que as pessoas passaram mais tempo em casa, o sentimento de frustração em relação à falta de ingredientes se tornou mais comum. (NERI, 2021)

De acordo com Neri (2022), a pobreza tem aumentado em São Paulo, com um crescimento de 5,2 pontos percentuais no município em comparação com 4,54 pontos percentuais no estado entre 2019 e 2021. O índice de pobreza no estado é de 17,9%, e esses fatores afetaram a cozinha brasileira. No entanto, muitos cozinheiros amadores têm procurado maneiras de empreender.

Com base no *Google Trends* (2022), a busca por receitas culinárias em sites de busca de sites de receitas aumentou durante a pandemia, de entre zero e mil buscas diárias na média para dez mil e vinte mil buscas diárias, mas a busca por aplicativos de receitas não acompanhou esse crescimento, como relatado por Soprana (2021). Muitos brasileiros enfrentam dificuldades para encontrar um aplicativo móvel que atenda às suas necessidades e desejos, seja para entretenimento ou para funcionalidades diárias, como receitas culinárias. A complexidade do aplicativo pode tornar a escolha ainda mais difícil.

De acordo com Ferreira *et al.* (2017), o mercado de aplicativos de receitas culinárias tem crescido nos últimos anos, mas é importante ressaltar que o conhecimento de culinária é essencial para o sucesso de um aplicativo. Apenas seguir boas práticas de desenvolvimento de *software* pode não ser suficiente, e um

desenvolvedor que não entenda o processo de preparar um ovo, por exemplo, pode levar à frustração do usuário. (SOUZA, 2022)

Os desenvolvedores podem gastar horas ou dias criando um aplicativo móvel e enfrentam desafios para mantê-lo no ar, especialmente se a adesão for baixa. A vontade do usuário pode mudar constantemente, levando-os a procurar novas inovações, o que significa que os desenvolvedores estão sempre procurando maneiras de encurtar o tempo de programação. Uma dessas ferramentas é a programação *no-code*. (MARGARET, 2022)

A programação *no-code* permite que desenvolvedores e até mesmo pessoas sem conhecimento de programação criem um aplicativo móvel usando interfaces gráficas e configurações, em vez da programação tradicional. As plataformas de desenvolvimento sem código estão se tornando cada vez mais populares à medida que as empresas lidam com uma força de trabalho móvel e recursos limitados de desenvolvedores de *software* competentes. Essas plataformas visam acelerar o processo de desenvolvimento de aplicativos, contornando as restrições tradicionais de tempo, dinheiro e recursos humanos de desenvolvimento de *software* escassos. As ferramentas de programação sem código são frequentemente projetadas para usuários de linha de negócios e são usadas para atender às necessidades de empresas que desejam digitalizar processos por meio de aplicativos móveis baseados em nuvem. (MARGARET, 2022)

Nesta introdução, uma visão geral sobre o problema a ser resolvido é apresentada, bem como a importância desse trabalho de graduação dentro desse contexto. Além disso, são explanados os objetivos e a abordagem metodológica utilizada. O segundo capítulo, por sua vez, embasa mais profundamente o tema de pesquisa através de bibliografias, apresenta soluções semelhantes no mercado e trata, ainda, o lado técnico do trabalho em relação ao desenvolvimento de aplicativos móveis. Em seguida, no terceiro capítulo, aspectos como os requisitos, diagramas, a estrutura da aplicação, código de programação e a interface de usuário compõem tópicos essenciais do funcionamento do sistema apresentado neste TG (Trabalho de Graduação). No capítulo subsequente, a conclusão do que fora alcançada com este projeto é disposta e, por fim, as fontes de pesquisas utilizadas e apêndices encerram o trabalho.

2 DESENVOLVIMENTO

Neste capítulo, será explorado o processo de desenvolvimento do trabalho, abordando as etapas e estratégias utilizadas para a sua elaboração.

2.1 BUSCA POR RECEITAS CULINÁRIAS

Atualmente, as buscas por receitas culinárias são realizadas através de livros de receitas, artigos de revistas ou jornais e no momento em sites e aplicativos de receitas. Uma pesquisa mostrou que a busca na internet por "Como fazer uma comida diferente" ou "como fazer comida caseira" teve um crescimento muito expressivo durante o isolamento social. O preferido das buscas foi o *brownie*, uma sobremesa de chocolate popular nos Estados Unidos. (ANDRADE E BELARMINO, 2020)

Muitos cozinheiros com pouca ou nenhuma experiência foram atrás de diversas receitas nos períodos de quarentena. No Brasil, de acordo com o *Google Trends*, uma ferramenta que monitora as pesquisas de forma global, a busca por "como fazer" cresceu e a expressão "como fazer *brownie*" é a mais procurada, tendo aumentado 300% em 12 meses. Apenas no final de março de 2020, esse tipo de procura aumentou em 139%. (ANDRADE E BELARMINO, 2020)

2.2 TECNOLOGIA MOBILE

Neste tópico, foi descrita uma breve ideia sobre a tecnologia *mobile*, sua importância na sociedade e algumas das principais linguagens para a criação de aplicativos.

Tecnologia *mobile* nada mais é do que toda tecnologia que permite a mobilidade do seu usuário, sem perder o contato com a tecnologia. Logo, celulares, *tablets*, *notebooks* e demais aparelhos eletrônicos portáteis podem ser considerados parte da tecnologia *mobile*. (PORTO, 2019)

De acordo com Barra (2011), a tecnologia *mobile* vai bem além do *hardware*. Ela também está nos serviços *online*, programas, aplicativos e meios de contato digitais que podem ser utilizados durante a movimentação e o transporte. Neste sentido, podemos chamar de tecnologia *mobile* também:

- *GPS WI-FI* ;(BARRA, 2011)
- Aplicativos de rede social; (BARRA, 2011)
- Aplicativos de banco; (BARRA, 2011)
- Aplicativos de reprodução de música; (BARRA, 2011)
- Entre outros aplicativos de disponibilidade móvel. (BARRA, 2011)

Até 2020, 80% da população deve ter um telefone móvel. Em outubro de 2016, os acessos à internet por meio de dispositivos *mobile* somaram 51% do total mundial, ou seja, os *desktops* foram superados pela tecnologia *mobile*. (PORTO, 2019)

As tecnologias *mobile* são cada vez mais desenvolvidas para facilitar o dia a dia do usuário, o que gera uma identificação maior entre clientes e empresas. Dúvidas surgem sobre os motivos de utilizar meios simplórios para pagar contas, por exemplo, se o consumidor pode realizar pagamentos por meio do *smartphone*. Ou então, sobre o porquê de passar horas ao telefone com uma central de atendimento se é possível tirar dúvidas via *chat*, mensageiro instantâneo ou redes sociais. (PORTO, 2019)

Porém, para construir ou criar um aplicativo *mobile* é preciso utilizar linguagens de programação adequadas. Várias linguagens podem ser utilizadas, tais como:

- *Objective-C*; (PORTO, 2019)
- *Swift*; (PORTO, 2019)
- *Java*; (PORTO, 2019)
- *Kotlin*; (PORTO, 2019)
- *No-code*; (PORTO, 2019)

Muito popular, a *Objective-C* foi a primeira linguagem de programação criada pela *Apple* com a finalidade de programar aplicativos especificamente para o sistema *iOS*. Trata-se de uma linguagem guiada para "Objetos", que também acrescenta a transmissão de mensagens voltada para "C", o que acontece por meio do aspecto *SmallTalk*. Embora seja amplamente adotada pelos desenvolvedores e tida como madura e estável, a *Objective-C* é criticada por alguns profissionais devido ao fato de oferecer uma sintaxe complexa - condição que acaba por dificultar o *debugging* no aplicativo (processo de depuração de erros). (PORTO, 2019)

O *Swift* figura entre as linguagens de programação mais utilizadas para o sistema *iOS*. Foi lançado no ano de 2014 e se caracteriza por ser muito simples e prático de ser aplicado, o que agrega agilidade para a rotina de trabalho do programador. A alternativa tem uma sintaxe limpa, acesso que o uso do ponto e vírgula no final de uma declaração é opcional. Além disso, a sintaxe permite tratar funções como objetos que posteriormente são referenciados e servem como parâmetro, gerando maiores abstrações. Ela ainda é "tipada" com inferência de tipo, que traz uma segurança extra de que as linhas de programação não serão quebradas ao serem compiladas. (PORTO, 2019)

Em se tratando de aplicativos para *Android*, o *Java* é a linguagem de programação mais frequente entre os programadores. Uma das suas principais características é fazer uso do paradigma orientado a objetos. E, nos últimos anos, abriu espaço para a inserção de elementos de "programação funcional" dentro do seu escopo. Vale ressaltar que a programação *Java* para *Android* é diferente do *Java* padrão, com sistema GUI e *runtime* distintos. A localização do aplicativo também é implementada de modo diferente. Como tem um modelo de desenvolvimento nativo, essa opção não requer um *framework* específico. (BARRA, 2011)

Bastante semelhante à *Java*, a *Kotlin* é mais recente e tem conquistado a preferência dos desenvolvedores por ser vista como uma linguagem de programação muito acessível. Inclusive, ela é interoperável com a *Java*, permitindo o uso de todas as suas bibliotecas de subprogramas, o que desburocratiza a construção de aplicações. Caso o profissional utilize o *IntelliJ*, vai obter uma interoperabilidade do IDE excelente, podendo utilizar código navegado, refatorado, pesquisado ou preenchido automaticamente. Durante a programação, você também tem acesso a um suporte completo para todas as tarefas, como teste de unidade, depuração e criação de perfil. (BARRA, 2011)

2.2.1 Aplicativos *mobile*

Aplicativos móveis são *softwares* utilizados para funções específicas em dispositivos móveis, como smartphones e tablets. Eles estão disponíveis através de plataformas de distribuição de aplicações que são normalmente operadas pelo proprietário do sistema operacional móvel, como *App Store*, *Android Market*, *BlackBerry App World*, *Ovi Store*, entre outros. (DAVE, 2013)

Segundo Siddique (2013), alguns aplicativos são gratuitos, e outros têm um preço. Normalmente eles são baixados da plataforma para um dispositivo de destino, como um *iPhone*, *BlackBerry*, *Android* ou telefone, mas às vezes eles podem ser baixados para computadores menos móveis, tais como *laptops* ou *desktops*. Os aplicativos podem ser destinados a facilitar o desempenho de atividades práticas ou para entretenimento do usuário.

O desenvolvimento de aplicativos móveis é o processo pelo qual aplicativos são desenvolvidos para dispositivos móveis, como *smartphones*, assistentes pessoais digitais e assistentes digitais empresariais. Essas aplicações são pré-instaladas durante a fabricação ou os clientes das mais variadas plataformas de distribuição de *software* móvel realizam o *download*. (DAVE, 2013)

Existem três tipos de aplicativos móveis. Os aplicativos nativos, aplicativos *web* e aplicativos híbridos. As aplicações desenvolvidas exclusivamente para uma plataforma específica, como o *iOS* ou o *Android*, são chamadas de aplicativos nativos. Ou seja, uma aplicação criada para a plataforma *Android* não funciona no *iOS*. Aplicações para *Android* são programadas em *Java*, e aplicações para *iOS* utilizam a linguagem *Objective-C* ou *Swift*. Nos aplicativos nativos, é possível utilizar os recursos existentes no *smartphone*, como câmera, *GPS*, etc. O *WhatsApp*, o *Facebook* e o *Uber* são exemplos de aplicativos nativos. São aplicativos recomendados e geralmente possuem melhor desempenho. (DAVE, 2013)

Os aplicativos *web*, na verdade, não são aplicativos reais. Os *web Apps* são executados pelo navegador e, uma vez que o programa reconhece que o usuário está acessando o site através de um *smartphone*, adapta-se a ele. Não estão disponíveis para instalação nas *apps stores*. São desenvolvidos utilizando *HTML5*, *CSS* e *Javascript*. Esses aplicativos não são recomendáveis para empresas ou ideias que querem proporcionar uma grande qualidade do aplicativo para os usuários. Geralmente, as fábricas de aplicativos usam essas tecnologias, mas não são recomendadas pela comunidade de desenvolvimento de aplicativos internacional. (DAVE, 2013)

Os aplicativos híbridos são uma junção de um aplicativo nativo e um *Web App*, gerando assim uma aplicação multiplataforma. Estão disponíveis nas *apps stores*. *Netflix* e *Steam* são exemplos de aplicativos híbridos. São desenvolvidos na linguagem *HTML5*, *CSS* e *Javascript*. (SIDDIQUE, 2013)

2.2.1.1 Distribuições

A *Amazon Appstore* opera como um distribuidor alternativo do sistema operacional *Android*. Começou a funcionar em março de 2011 e já conta com mais de 1 milhão de aplicações disponíveis. Além disso, compromete-se a lançar novas aplicações diariamente, e está classificada como a loja de aplicações móveis mais confiável para a plataforma *Android* no momento do desenvolvimento deste projeto. (POGUE, 2014)

A *App Store* é a loja de aplicações para a plataforma *iOS*, foi a primeira distribuidora, definiu o padrão para os serviços de distribuição de aplicações e ainda hoje serve de modelo para as restantes distribuidoras. Inaugurou em 10 de julho de 2008, e em janeiro de 2011, apresentou resultados surpreendentes, com mais de 10 bilhões de *downloads*. (POGUE, 2014)

A *Google Play* (anteriormente conhecida como *Android Market*) é uma loja de *software online* internacional desenvolvida pela Google para dispositivos *Android*. Foi inaugurada em outubro de 2008. Em julho de 2013, já contava com mais de 1 milhão de aplicações disponíveis exclusivamente para *Android*, conforme estimativa apresentada pela *Google Play*. (KART, 2015)

A *Windows Phone Store*, inicialmente denominada *Windows Phone Marketplace*, foi lançada em 7 de outubro de 2010 pela *Microsoft*, simultaneamente com o lançamento do *Windows Phone 7*. (POGUE, 2014)

A *Windows Store* foi introduzida pela *Microsoft* para as plataformas *Windows 8* e *Windows RT*. Possibilita a transferência de diversos programas construídos para computadores de mesa (*desktop*) com compatibilidade certificada para o *Windows 8*, contudo é usado maioritariamente para distribuir aplicações para operarem em *tablets* e outros dispositivos tácteis, mas ainda podem ser usados com teclados, *mouse*, e em *desktop* e em computadores portáteis (*laptops*). (KART, 2015)

A *Samsung Apps Store* é a loja de aplicações para celulares e smartphones *Samsung*, fundada em setembro de 2009. Em outubro de 2011, a *Samsung Apps* alcançou 10 milhões de *downloads*. Atualmente, a distribuidora está disponível em mais de 120 países e oferece aplicações para *Windows Mobile*, *Android* e plataformas *Bada*. Em dezembro de 2013, a *Samsung* e a *Fingerprint* uniram-se na criação de aplicações para dispositivos *Samsung*, exclusivamente para crianças. (POGUE, 2014)

2.2.1.2 Aplicativos mobile de buscas de receitas culinárias

Neste tópico, serão abordadas as ferramentas de buscas, aplicativos *mobile*, de receitas culinárias. Foram analisados sete aplicativos com suas diferenças e peculiaridades. Porém, esses aplicativos possuem algumas lógicas próximas à do trabalho em questão.

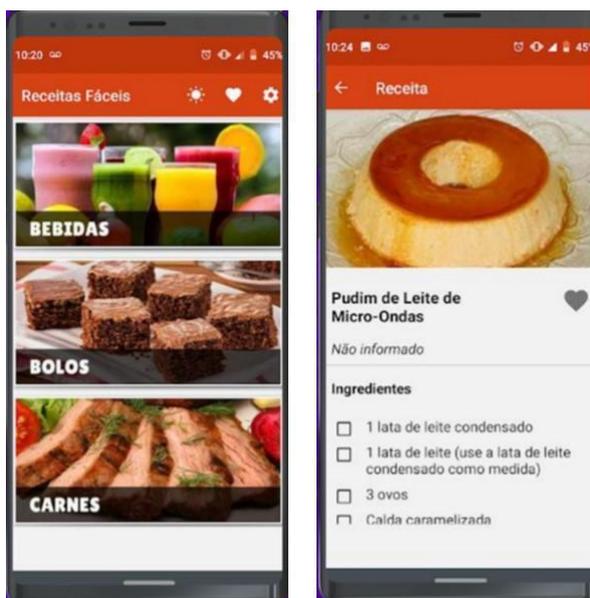
De acordo com Syozi (2021) vários aplicativos de buscas por receitas surgiram nos últimos anos. Syozi cita sete aplicativos, sendo eles o “Receitas Fáceis”, “Tasty Receitas”, “Tudo Gostoso Receitas”, “Cozinha *Fit & Fat*”, “Livro de Receitas”, “Receitas Veganas em Português” e “*Petitichef*”.

Segundo Syozi (2021) o “Receitas Fáceis” se descreve como:

Um dos aplicativos de receitas mais populares oferece uma enorme biblioteca para quem quer inventar na cozinha. Sua interface é bastante simples, mas dá conta do recado. Tudo é separado por categorias como “Bebidas”, “Bolos” e “Carnes”. Além disso, o usuário pode deixar as receitas que mais usar como suas favoritas para sempre ter acesso. Outra opção interessante é a possibilidade de marcar os ingredientes que a pessoa tem em mãos na hora de fazer a receita. Com isso, fica mais fácil de controlar e não ter que sair no meio da preparação. Há anúncios no aplicativo. O aplicativo é gratuito, mas com anúncios. Sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 1 ilustra um exemplo da interface do aplicativo “Receitas Fáceis”.

Figura 1 – Interface do aplicativo “Receitas Fáceis”



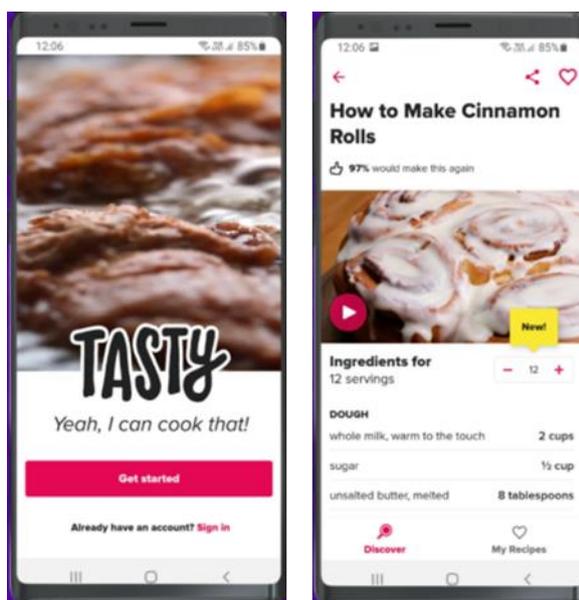
Fonte: Syozi (2021)

Segundo Syozi (2021) o “Tasty Receitas” se descreve como:

Um aplicativo de receitas que ganha popularidade, ainda mais por ser do famoso veículo *BuzzFeed*. Ele já se apresenta garantindo mais de 3000 opções de pratos variados. O aplicativo oferece preparações passo a passo para facilitar a vida do usuário, além de vídeos frequentes sobre novidades da cozinha. Há como personalizar o app para transformá-lo em um livro de receitas pessoal, mas também tem como permitir que ele envie sugestões periódicas do que preparar para a próxima refeição. Infelizmente, não há opções para o português, ou seja, é necessário conhecer um pouco de inglês para usufruir de todo o conteúdo. É totalmente gratuito e sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 2 ilustra um exemplo da interface do aplicativo “Tasty Receitas”.

Figura 2 – Interface do aplicativo “Tasty Receitas”



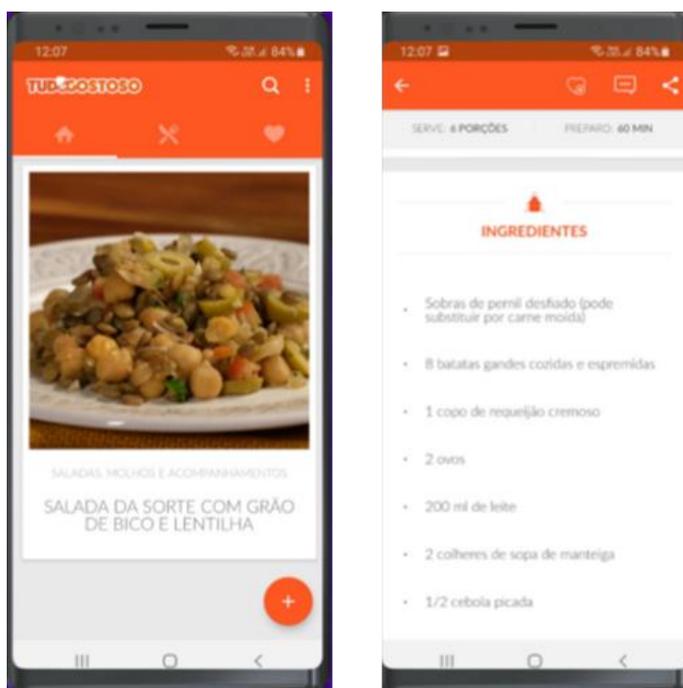
Fonte: Syozi (2021)

Segundo Syozi (2021) o “Tudo Gostoso Receitas” se descreve como:

Aplicativo mais de 170 mil receitas para os usuários, este aplicativo permite que tudo seja encontrado através de um botão de buscas de fácil acesso. As categorias também auxiliam na hora de achar aquele prato para aquele momento importante do dia. Uma característica que vale mencionar é que o TudoGostoso Receitas motiva os usuários a enviarem suas próprias sugestões com imagens e dicas. Assim, quase que uma comunidade de chefs acaba sendo criada entre as pessoas que mais gostam de aplicativos de receitas. Há muitas propagandas que aparecem enquanto o usuário faz uso do aplicativo. A busca da receita é realizada digitando um ingrediente, ou seja, receitas com o usuário tem em casa. É totalmente gratuito, mas possui anúncios e sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 3 ilustra um exemplo da interface do aplicativo “Tudo Gostoso Receitas”.

Figura 3 – Interface do aplicativo “Tudo Gostoso Receitas”

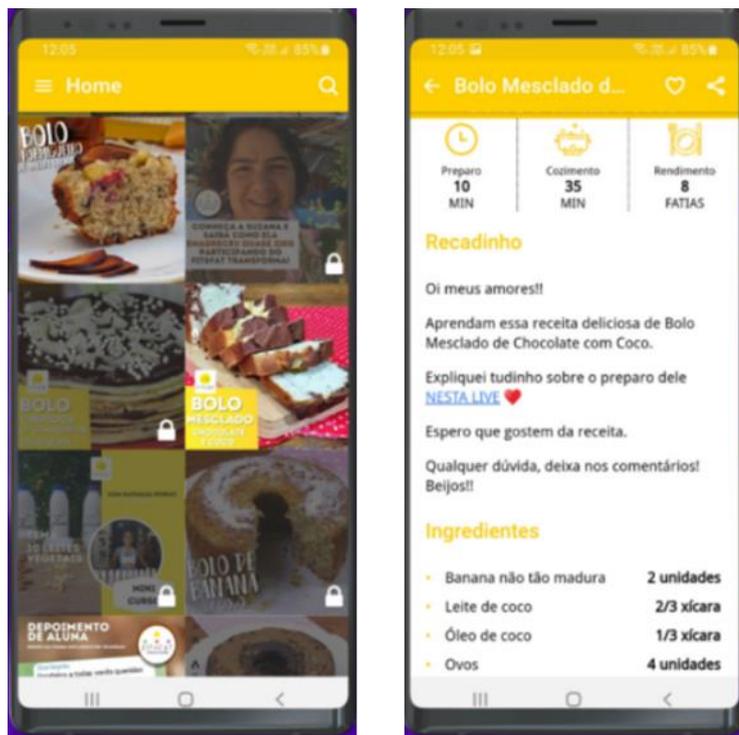


Fonte: Syozi (2021)

Com base em Syozi (2021) o “Cozinha *Fit & Fat*” se descreve como:

Aplicativo que oferece opções para quem quer criar refeições mais saudáveis, o Cozinha *Fit & Fat* permite buscas que vão desde o ingrediente usado na receita ao próprio nome do prato. Cada opção é apresentada com uma imagem da receita pronta e um passo a passo intuitivo e cheio de detalhes. Além disso, matérias informativas e miniaulas são parte da biblioteca do aplicativo. Há até mesmo entrevistas com personalidades para quem busca aquele conteúdo extra que vai além da preparação e dos ingredientes. Acessos na maioria dos aplicativos de receitas por aí. A busca da receita é realizada digitando um ingrediente, ou seja, receitas com o usuário tem em casa. É gratuito, mas com anúncios e compras no app e sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 4 ilustra um exemplo da interface do aplicativo “Cozinha *Fit & Fat*”.

Figura 4 – Interface do aplicativo “Cozinha *Fit & Fat*”

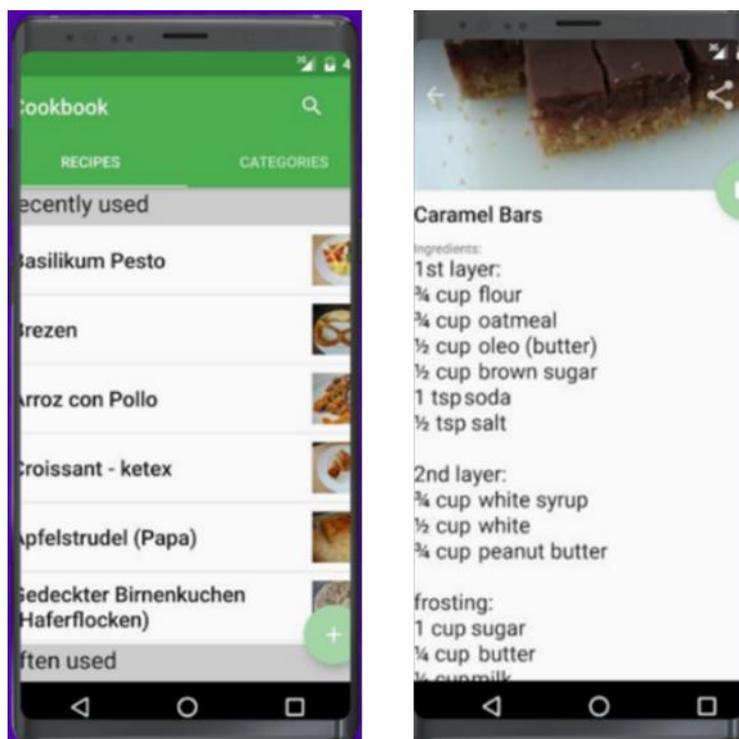
Fonte: Syozi (2021)

Segundo Syozi (2021) o “Livro de Receitas” se descreve como:

Ao contrário de todas as opções desta lista de aplicativos, o Livro de Receitas não oferece uma lista para que o usuário possa encontrar algum prato e prepará-lo. Na verdade, este aplicativo funciona como um livro no qual a própria pessoa pode adicionar texto e imagens para deixar tudo organizado. Assim, a ideia aqui é aposentar o caderninho físico e trocá-lo por um app para manter suas receitas guardadas de maneira segura. Outro fator interessante é que dessa maneira, o usuário pode compartilhar com outras pessoas os seus pratos preferidos. A busca da receita é realizada digitando um ingrediente, ou seja, receitas com o usuário tem em casa. É gratuito, mas com anúncios e compras no app e sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 5 ilustra um exemplo da interface do aplicativo “Livro de Receitas”.

Figura 5 – Interface do aplicativo “Livro de Receitas”



Fonte: Syozi (2021)

De acordo com Syozi (2021), o “Receitas Veganas em Português” se descreve como:

Direcionado tanto para aquelas pessoas interessadas em conhecer o veganismo quanto para os veganos de longa data, esse aplicativo oferece um *layout* simples, mas prático. A gama de pratos não é gigantesca como outros exemplos dessa lista, mas fornece o suficiente dentro do mundo apresentado. Há sugestões de café da manhã, almoço, jantar e até de sobremesa. Tudo sem usar um único ingrediente de origem animal. A busca da receita é realizada digitando um ingrediente, ou seja, receitas com o usuário tem em casa. Vale o destaque para os lanches (meus favoritos) que conseguem agradar até mesmo os carnívoros de plantão. É gratuito, mas com anúncios e compras no app e sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 6 ilustra um exemplo da interface do aplicativo “Receitas Veganas em Português”.

Figura 6 – Interface do aplicativo “Receitas Veganas em Português”



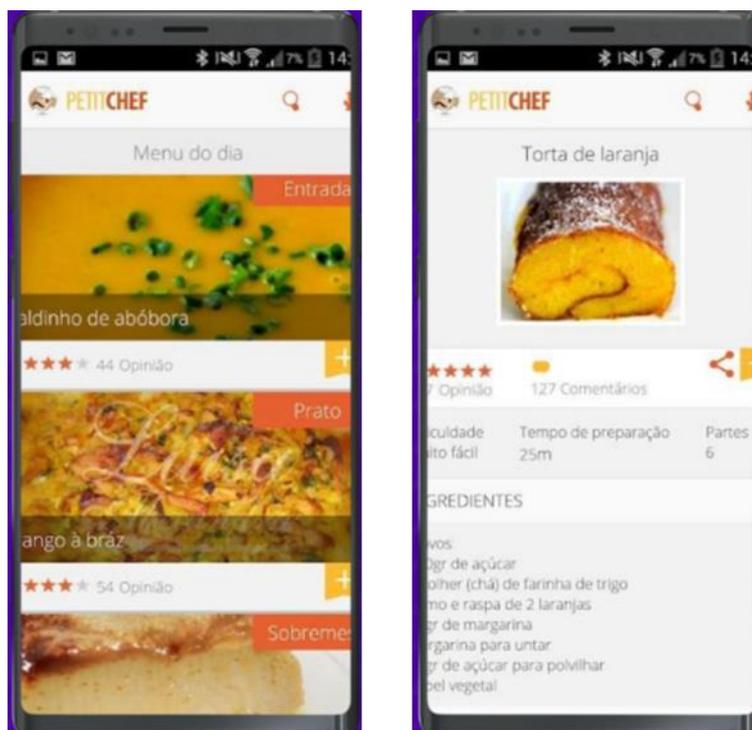
Fonte: Syozi (2021)

De acordo com Syozi (2021), o “*Petitichef*” se descreve como:

Este é mais um dos muitos aplicativos de receitas que prometem milhares de pratos para todos os gostos. O *Petitichef* divide suas categorias entre “Entrada”, “Prato”, “Sobremesa” e “Outros”. Sendo assim, fica mais fácil encontrar aquele petisco para a sua festa, por exemplo. A busca da receita é realizada digitando um ingrediente, ou seja, receitas com o usuário tem em casa. Um ponto de destaque deste aplicativo é que muitas receitas também trazem informações nutritivas, auxiliando quem busca por mais detalhes. É possível favorizar as que quiser, além de adicionar imagens para compartilhar com outras pessoas. Por último, vale ressaltar que o app envia sugestões de pratos diariamente para os usuários. É gratuito, mas com anúncios e compras no app e sua distribuição está na *Play Store*. (SYOZI, 2021)

A figura 7 ilustra um exemplo da interface do aplicativo “*Petitichef*”.

Figura 7 – Interface do aplicativo “Petitichef”



Fonte: Syozi (2021)

2.3 TECNOLOGIA NO-CODE

Neste tópico, foi abordada a tecnologia *no-code*, assim como a plataforma escolhida para a realização deste trabalho, o *FlutterFlow*.

No-code (ou "sem código", em português) é um movimento tecnológico e social que permite desenvolver soluções de *internet* e dispositivos móveis sem programação tradicional. O desenvolvimento de aplicações sem o uso de código é possível graças a ferramentas *no-code*. (MATOS, 2022)

Assim, a barreira de conhecimento que existia entre desenvolvedores e usuários de negócios deixou de ser um empecilho, pois o *no-code* permite que amadores desenvolvam suas próprias soluções. (MATOS, 2022)

Pode-se utilizar como exemplo uma pessoa do RH de uma empresa que está sempre cheia de solicitações de reembolso. O desejo dela é poder fazer esse processo de forma automática, desde a solicitação do colaborador, passando pelo RH e terminando no financeiro. Porém, os analistas de sistemas não conseguem dar atenção ao problema dela pois existem diversas outras situações críticas com que eles devem se preocupar. Sendo assim, uma ferramenta *no-code* permitiria que ela

mesma automatizasse esse processo, simplesmente conectando elementos gráficos de forma intuitiva e acessível. Problema resolvido, ela atingiu seu objetivo e não demandou a TI. (SATELL, 2022; MATOS, 2022)

Existem muitas vantagens na tecnologia *no-code*, como:

- Aumentam a produtividade das áreas de TI, pois elas conseguem entregar resultados muito mais rápidos; (MATOS, 2022)
- Aumentam a produtividade de outras áreas, pois é possível criar diferentes soluções para diferentes processos do dia a dia, possibilitando maior controle, produtividade e padronização para atividades recorrentes; (MATOS, 2022)
- Democratizam a tecnologia e dão autonomia a diferentes criadores, pois capacita o desenvolvedor cidadão a criar aplicações, mesmo sabendo pouco ou nada de programação; (MATOS, 2022)
- Facilita e acelera o *onboarding* das soluções nas empresas, devido a facilidade que oferecem, se torna mais fácil inserir soluções *low-code* e *no-code* nas organizações; (MATOS, 2022)
- Acelera a percepção nas Provas de Conceito (POC), como é muito mais rápido construir soluções em *low-code* e *no-code*, também é mais rápido testar e ajustar detalhes. Assim, rapidamente é possível entender como as ferramentas devem evoluir;
- Redução de custos, pois as plataformas com baixo ou zero código podem representar economia não só de tempo, mas também de dinheiro, pois, geralmente, se tratando de empresas, evita-se projetos gigantes com muitos desenvolvedores. (MATOS, 2022)

No entanto, podem ser citadas algumas desvantagens da tecnologia *no-code*. Muitas vezes, a escolha de uma ferramenta sem código para desenvolver um aplicativo pode ser um equívoco. Há momentos em que a maior parte do trabalho já foi feita e, de repente, fica claro que é simplesmente difícil implementar a função ou recurso necessário na plataforma. (SILOTTO, 2022)

Por exemplo, a ferramenta sem código *FlutterFlow* pode fazer *upload* de código *Flutter*, o desenvolvedor pode criar ações personalizadas nele. No entanto, no período de desenvolvimento deste projeto, o *FlutterFlow* não possuía várias funções simples, como pesquisa por palavra-chave, filtragem, visibilidade de objetos sob condições e assim por diante. Com isso, os desenvolvedores sem código geralmente não se preocupam com a segurança dos dados e da *API*, e os dados são constantemente vazados. Para muitos clientes, isso é crucial. A política de segurança da empresa muitas vezes não permite o armazenamento de dados em servidores de terceiros. Os serviços sem código muitas vezes armazenam dados em servidores estrangeiros, e as leis de muitos países obrigam os clientes a armazenar dados pessoais em servidores nesses países específicos e assim por diante. (SILOTTO, 2022)

2.3.1 Flutter Flow

Com base em Silotto (2022), o *Flutter* é um *framework* criado pelo *Google* para ajudar na criação e desenvolvimento de aplicativos para qualquer tela. É usado para desenvolver aplicativos para *Android* e *iOS*, além de ser o principal método de criação de aplicativos para o *Google Fuchsia*. A proposta é que o desenvolvedor encontre todas as ferramentas necessárias para criar rapidamente o aplicativo e não se preocupe com tarefas repetitivas e monótonas. Porém, existe ainda o *FlutterFlow*, uma plataforma *low-code online* que facilita ainda mais a vida de quem deseja criar uma aplicação *mobile* e *web*. A ideia é que o programador, profissional ou não, monte visualmente todas as *views* e interfaces de usuário e foque apenas no que realmente interessa, ou seja, os recursos principais do seu aplicativo.

2.4 BANCO DE DADOS

Neste tópico, serão abordados a funcionalidade dos bancos de dados, seus tipos, sua importância e uma breve explanação sobre a plataforma do *Google*, o *Firebase*.

Bancos de dados são conjuntos de arquivos relacionados entre si, com registros sobre pessoas, lugares ou informações em geral. São coleções organizadas de dados que se relacionam de forma a criar algum sentido (informação) e dar mais eficiência durante uma pesquisa ou estudo científico. São de vital importância para

empresas e há mais de duas décadas se tornaram a principal peça dos sistemas de informação e segurança. Normalmente, existem por vários anos sem alterações em sua estrutura sistemática. (LAUDON, 2011)

Os bancos de dados são operados pelos Sistemas Gerenciadores de Bancos de Dados (SGBD), que surgiram na década de 1970. Antes deles, as aplicações usavam sistemas de arquivos do sistema operacional para armazenar suas informações. Na década de 1980, a tecnologia de SGBD relacional passou a dominar o mercado. Atualmente, é utilizada em praticamente todos os bancos de dados. Outro tipo notável é o SGBD Orientado a Objetos, implementado em bancos de dados com estruturas complexas ou aplicações que mudam constantemente. (MORAIS, 2018)

A principal aplicação de banco de dados é o controle de operações empresariais. Outra aplicação também importante é o gerenciamento de informações de estudos, como fazem os Bancos de Dados Geográficos, que unem informações convencionais e espaciais. (BATISTA, 2017)

2.4.1 Tipos de bancos de dados

Existem diversos tipos, sendo eles:

- Relacionais: padrão mais conhecido e utilizado em ERP, CRM e outras aplicações, por possuir armazenamento em tabela (com colunas e linhas), fornecendo uma maneira mais eficiente e flexível no acesso de informações estruturadas; (BATISTA, 2017)
- Orientados a objetos: armazenam seus registros no formato de objetos, como definido no paradigma orientado a objetos; (BATISTA, 2017)
- Distribuídos (BDD): são bancos de dados que estão armazenados em diferentes servidores; (BATISTA, 2017)
- *Data warehouses*: projetados especialmente para análises avançadas utilizando diversas fontes de dados centralizadas; (BATISTA, 2017)
- NoSQL: são bancos de dados que armazenam os dados em um formato onde não há uma relação direta entre os elementos; (BATISTA, 2017)

- Gráficos: responsáveis por armazenar as entidades de dados na borda de sua estrutura e os relacionamentos entre as entidades em seus nós; (BATISTA, 2017)
- OLTP: banco de dados projetado para muitas transações que são realizadas por vários usuários; por este motivo, os bancos de dados OLTP devem ser rápidos e analíticos. (BATISTA, 2017)

2.4.2 *Firestore*

O *Firestore* do *Google* é uma plataforma digital utilizada para facilitar o desenvolvimento de aplicativos *web* ou móveis de forma efetiva, rápida e simples. Graças às suas diversas funções, é utilizado como técnica de *Marketing Digital* com a finalidade de aumentar a base de usuários e gerar maiores benefícios econômicos. Seu principal objetivo é melhorar o desempenho dos aplicativos mediante a implementação de diversas funcionalidades que tornarão o aplicativo muito mais maleável, seguro e de fácil acesso para os usuários. (WAGNER e TOWNSEND, 2018)

Em outras palavras, o *Firestore* é um conjunto de serviços de hospedagem para qualquer tipo de aplicativo (*Android*, *iOS*, *Javascript*, *Node.js*, *Java*, *Unity*, *PHP* e *C++*). Ele oferece hospedagem em tempo real de bancos de dados *NoSQL*, conteúdo, autenticação social (*Google*, *Facebook*, *Twitter* e *Github*) e notificações, além de serviços como um servidor de comunicação em tempo real. (WAGNER e TOWNSEND, 2018)

A *Firestore* evoluiu da *Envolve*, uma *startup* anterior fundada por James Tamplin e Andrew Lee em 2011. A *Envolve* forneceu aos desenvolvedores uma API que permite a integração da funcionalidade de bate-papo on-line em seus sites. Depois de liberar o serviço de bate-papo, Tamplin e Lee descobriram que ele estava sendo usado para passar dados de aplicativos que não eram mensagens de chat. Os desenvolvedores estavam usando o *Envolve* para sincronizar dados de aplicativos, como o estado do jogo em tempo real em seus usuários. Tamplin e Lee decidiram separar o sistema de bate-papo e a arquitetura em tempo real que o alimentava. Eles fundaram *Firestore* como uma empresa separada em 2011 e foi lançada ao público em abril de 2012. (MELENDEZ, 2014)

2.4.3 Algolia

Algolia é uma plataforma de pesquisa e descoberta com tecnologia de IA para experiências dinâmicas que ajuda as empresas a maximizar a velocidade de pesquisa e descoberta, ao mesmo tempo em que resolve o problema do ajuste de relevância por meio da IA. Acessar o conteúdo ou produtos certos em sites e aplicativos nunca foi tão rápido ou intuitivo. (ALGOLIA, 2023)

No *Firebase* o *Algolia* é utilizado como uma extensão para converter o *Firebase* de um banco de dados não-relacional para um banco de dados **relacional**. Com isso, o *Algolia* pode ajudar a refinar as buscas realizadas em um aplicativo que utiliza o banco de *Firebase*. (ALGOLIA, 2023)

3 MÉTODOS

Neste capítulo, serão descritos os métodos e procedimentos adotados para realizar este trabalho.

3.1 LEVANTAMENTO DE REQUISITOS

Para a realização do levantamento de requisitos, foi necessário compreender, identificar e definir as necessidades que os desenvolvedores e os usuários terão. Desta forma, os requisitos foram divididos em funcionais, que são aqueles que abordam o que o sistema deve fazer, e os não funcionais, que tratam mais de características do que de funcionalidades em si.

Além disso, para concluir com eficácia a etapa de levantamento de requisitos, são necessárias técnicas para que estes sejam validados como necessários ao desenvolvimento do sistema e à utilização do usuário. Em primeiro lugar, foi feito um levantamento orientado ao ponto de vista dos desenvolvedores, por meio de análises de sistemas já existentes e de opiniões do orientador e de possíveis usuários deste sistema.

Para a realização deste, primeiro foi pensado nos requisitos funcionais, a fim de apontar quais são as funções que o aplicativo deve ter. A seguir, são apontados em tópicos cada um dos requisitos funcionais:

- Cadastro de novas receitas (pelos administradores);
- Consulta de receitas;
- Alerta de receita cadastrada com sucesso;
- Alerta de receita repetida cadastrada com sucesso;
- Alerta de consulta realizada com sucesso;
- Alerta de consulta sem resultados;
- Compartilhar receitas;

Prontamente, foram pensados nos requisitos não funcionais, a fim de apontar quais são as características do sistema. A seguir, são apontados em tópicos cada um dos requisitos não funcionais:

- Capacidade de atualização dos dados em tempo real;
- Garantia de manutenção e disponibilização de novas versões;
- Tratamento de textos e imagens;
- Compatibilidade com *software* de banco de dados.

3.2 DIAGRAMA UML

Após a conclusão da etapa de elicitação de requisitos, deu-se início à estruturação dos diagramas e modelos relacionais. Por meio destes, torna-se possível haver um melhor entendimento de como será a "circulação" do sistema como uma unidade. Inicialmente, foi desenvolvido um diagrama *UML*.

A *UML* trata-se de uma padronização para a elaboração da estrutura de projetos de *software*. Para a estruturação e a realização do diagrama *UML*, os desenvolvedores utilizaram a ferramenta *Dia*, que possui suporte necessário para a estrutura do diagrama de acordo com as normas. (BATISTA, 2017)

Assim, nesta etapa, foram definidas todas as entidades, bem como suas funções, seus objetivos e de que forma elas se relacionam entre todo o corpo do sistema.

Para a criação do diagrama, foram utilizados dois atores, sendo eles o administrador do aplicativo e o usuário. O usuário poderá baixar o aplicativo, consultar receitas e compartilhá-las em suas redes sociais. O administrador herdará todas as funções dos usuários.

3.3 DIAGRAMA DE SEQUÊNCIA

Para entender o que é um diagrama de sequência, é importante conhecer a função da Linguagem de modelagem unificada, conhecida como UML. UML é um kit de ferramentas de modelagem que orienta a criação e notação de muitos tipos de diagramas, incluindo diagramas de comportamento, interação e estrutura. (BATISTA, 2017)

Um diagrama de sequência é uma espécie de diagrama de interação, pois descreve como, e em qual ordem, um grupo de objetos trabalha em conjunto. Estes diagramas são usados por desenvolvedores de software e profissionais de negócios

para entender as necessidades de um novo sistema ou para documentar um processo existente. Diagramas de sequência são conhecidos como diagramas de eventos ou cenários de eventos. (BATISTA, 2017)

Para a criação do diagrama, foram utilizados dois atores, sendo eles o administrador do aplicativo, o usuário. O administrador cadastrará uma receita e ela será salva no banco de dados. Quando o administrador realizar a pesquisa será checado no banco de dados se a receita existe. Se ela não existir aparecerá na tela uma mensagem de erro. Porém, se a receita existir, o aplicativo retornará a receita.

No caso do usuário, ao realizar a pesquisa será checado no banco de dados se a receita existe. Se ela não existir aparecerá na tela uma mensagem de erro. Porém, se a receita existir, o aplicativo retornará a receita.

3.4 DER

A seguir, o próximo diagrama estruturado pelos desenvolvedores foi o Diagrama Entidade-Relacionamento, conhecido também pela sigla DER.

Este diagrama é a representação gráfica do Modelo Entidade-Relacionamento (MER), que será descrito no próximo tópico. O DER, assim como o *UML* deste projeto, tem como objetivo explicitar os relacionamentos das entidades do sistema. Além disso, deixa bem claro também quais são os atributos específicos de cada entidade, quais as relações de cardinalidade, quais entidades são fortes ou fracas, entre outros. (LAUDON, 2011)

Portanto, para a estruturação e realização do DER, os desenvolvedores também utilizaram a ferramenta *Dia.exe*, que também possui o suporte necessário para a estrutura deste tipo de diagrama de acordo com as normas.

Seguindo os passos dos requisitos de sistemas e o diagrama *UML*, o DER foi montado seguindo o levantamento de seis entidades de relacionamento, sendo elas em ordem alfabética: eletrodoméstico, etapas, ingredientes, receitas, usuários e utensílios. Para cada entidade são utilizados os dados referentes a ela.

Para a receita, são levantados os seguintes dados:

- Nome: nome da receita;
- Tipo: se a receita é para doces ou salgados;

- ID_receita: número de identificação da receita;
- Imagem: uma foto que o usuário poderá tirar da receita ou pegar da *internet*;
- Tempo: tempo de preparo da receita;
- Rendimento: quantidade em porções da receita.

Para os utensílios, são levantados os seguintes dados:

- Nome_utensílio: nome do utensílio, por exemplo, uma colher de sopa;
- ID_utensílio: número de identificação do utensílio.

Para os eletrodomésticos, são levantados os seguintes dados:

- Nome_eletrodoméstico: nome do utensílio, por exemplo, um liquidificador;
- ID_eletrodoméstico: número de identificação do eletrodoméstico.

Para os ingredientes, são levantados os seguintes dados:

- Nome_ingrediente: nome do utensílio, por exemplo, um ovo;
- ID_ingrediente: número de identificação do ingrediente;
- Quantidade_ingrediente: quantidade do ingrediente;

Para as etapas, são levantados os seguintes dados:

- Texto: texto da receita;
- ID_etapa: número de identificação da etapa.

Entre as entidades, serão colocadas as letras "N" ou o número "1", descrevendo o índice de cardinalidade de relação delas, como, por exemplo, quantas receitas um usuário poderá cadastrar no aplicativo. Além disso, serão utilizados losangos descrevendo qual será a ação entre as entidades, como a ação de um usuário cadastrar duas receitas.

3.5 MER

Como já antecipado na etapa anterior, o MER é a sigla para Modelo Entidade-Relacionamento, realizado de forma bastante similar ao DER (Diagrama Entidade-Relacionamento). Em geral, este modelo representa como será realizada a estruturação do banco de dados do sistema de forma mais abrangente do que os outros modelos explicitados.

Assim, para a estruturação e realização do MER, os desenvolvedores optaram por utilizar a ferramenta *BRModelo*, que é específica para tal estrutura, tendo o suporte necessário para este tipo de diagrama de acordo com suas normas. Foi possível também, através desta ferramenta, definir características específicas das entidades e dos atributos, como chaves primárias e estrangeiras (caso sejam necessárias), os tipos de cada atributo (*char*, *varchar* etc.), além dos já definidos relacionamentos e cardinalidades.

Assim, após a conclusão deste modelo, conclui-se a etapa de elicitação de requisitos e de estruturação de diagramas, tendo toda a base necessária para o seguimento do desenvolvimento geral do sistema.

Para a realização deste trabalho, é utilizada a trigramação para cada tabela, tais como:

- TBL_ADM: tabela do usuário administrador
- TBL_RECEITA: tabela da receita;
- TBL_UTENSILIO: tabela do utensílio;
- TBL_INGREDIENTE: tabela do ingrediente;
- TBL_MODALIDADE_DE_REPARO: tabela da etapa.

A trigramação utilizadas para a normalização das entidades são:

- ADM: Administrador;
- REC: receita;
- UTE: utensílio;
- ING: ingrediente;
- MP: modo de preparo.

Para identificar a relação entre as entidades, será utilizada a letra "n". Sendo assim, essa letra permitirá aos desenvolvedores saber, por exemplo, quantas receitas um usuário poderá cadastrar no aplicativo. Além disso, serão utilizados losangos descrevendo qual será a ação entre as entidades, como a ação de um usuário cadastrar duas receitas.

3.6 PROTOTIPAGEM

Com base em Carvalho (2020), o planejamento se torna uma etapa essencial para evitar o gasto de recursos financeiros e tempo da equipe em propostas que agregam pouco às empresas. Uma das fases fundamentais durante o desenvolvimento de um projeto de tecnologia, seja de um aplicativo, de um sistema ou até de processos de automação, é a prototipagem, que auxilia num melhor planejamento antes de evoluir para o desenvolvimento em si (codificação).

A prototipagem é primordial para o sucesso do projeto de TI e, neste texto, trataremos as principais vantagens de criar um protótipo antes de começar a produção e o porquê de ainda ser vista por alguns como uma etapa extra. (Carvalho, 2020)

A prototipagem de um sistema ou aplicativo é um processo iterativo de criação de um modelo do que virá a ser o *software*. É uma atividade de criação de uma versão inicial do *software* baseada no levantamento dos requisitos. Um processo que propõe a criação de um protótipo de *software* objetiva apoiar a fase de elaboração do escopo, a fim de prevenir possíveis falhas no projeto do *software*. O protótipo faz a simulação do aplicativo, ajudando clientes e profissionais de desenvolvimento a verem os requisitos do projeto e avaliar, alterar e aprovar as características mais marcantes na interface e suas funções. (KRIGER, 2022)

Para a montagem da prototipagem, os integrantes utilizaram a plataforma *FlutterFlow*, porém antes da própria programação. Cada tela, caixa de texto, cores, tipos e tamanhos de botões foram pensados durante a programação do aplicativo na plataforma. Foram criadas doze telas:

- *Home Page*: nesta tela, há um botão que leva o usuário para o menu principal;
- Menu principal: tela em que o usuário verá botões que o levarão à tela

de cadastro/edição de receita, pesquisa de receita ou de volta para a tela inicial;

- Tela de cadastro: nesta tela, há um botão que leva para o cadastro de receita e outro que leva para a edição de receitas;
- Cadastro de receitas: tela em que o usuário pode cadastrar uma receita escolhendo utensílios, ingredientes, modo de preparo e uma imagem da receita;
- Edição de receita: tela em que o usuário pode editar as receitas já cadastradas;
- Pesquisa receita: tela em que o usuário poderá escolher entre pesquisar por nome de receita, por utensílio, ou por ingrediente, e também poderá voltar ao menu principal;
- Pesquisa por nome de receita: nesta tela, o usuário poderá pesquisar a receita inserindo o nome dela;
- Pesquisa por utensílios: nesta tela, o usuário poderá pesquisar a receita baseando nos utensílios selecionados;
- Pesquisa por ingredientes: nesta tela, o usuário poderá pesquisar a receita baseando nos ingredientes selecionados;
- Resultado de pesquisa por nome de receita: nesta tela, o usuário verá o resultado de sua pesquisa com base no nome da receita inserida;
- Resultado de pesquisa por utensílios: nesta tela, o usuário verá o resultado com base em suas seleções na pesquisa por utensílios selecionados;
- Resultado de pesquisa por ingredientes: nesta tela, o usuário verá o resultado com base em suas seleções na pesquisa por ingredientes selecionados;

4 RESULTADO E DISCUSSÕES

Neste capítulo, são apresentados os principais resultados obtidos no desenvolvimento deste trabalho.

4.1 LEVANTAMENTO DE REQUISITOS

Com a utilização do programa *Excel*, foi criada a tabela com o levantamento dos requisitos funcionais e não-funcionais (Tabela 1).

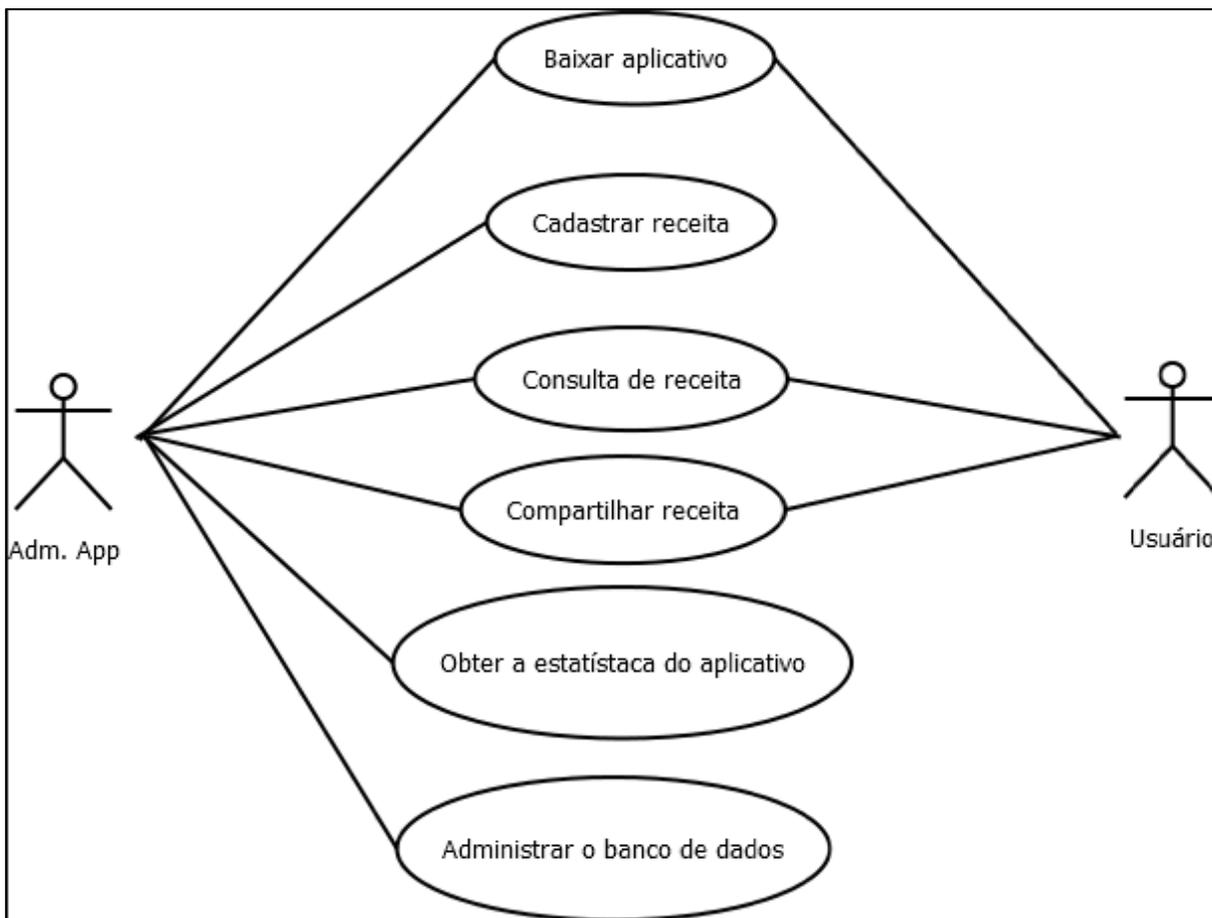
Tabela 1 – Requisitos funcionais e não-funcionais
Desenvolvimento próprio (2023)

Cod.	Requisitos funcionais
FN-1	Cadastro de novas receitas
FN-2	Consulta de receitas
FN-3	Alerta de consulta de receita realizada com sucesso
FN-4	Alerta de consulta sem resultado
FN-5	Usuário publica a receita em suas redes sociais
Cod.	Requisitos não-funcionais
NF-1	Capacidade de atualização dos dados em tempo real
NF-2	Garantia de manutenção e disponibilização de novas versões
NF-3	Tratamento de textos e imagens
NF-5	Compatibilidade com o <i>software</i> de banco de dados

4.2 DIAGRAMA DE CASO DE USO *UML*

A figura a seguir ilustra o diagrama caso de uso *UML* do sistema do aplicativo Prato Feito.

Figura 8 – Diagrama de caso de uso UML do aplicativo Prato Feito

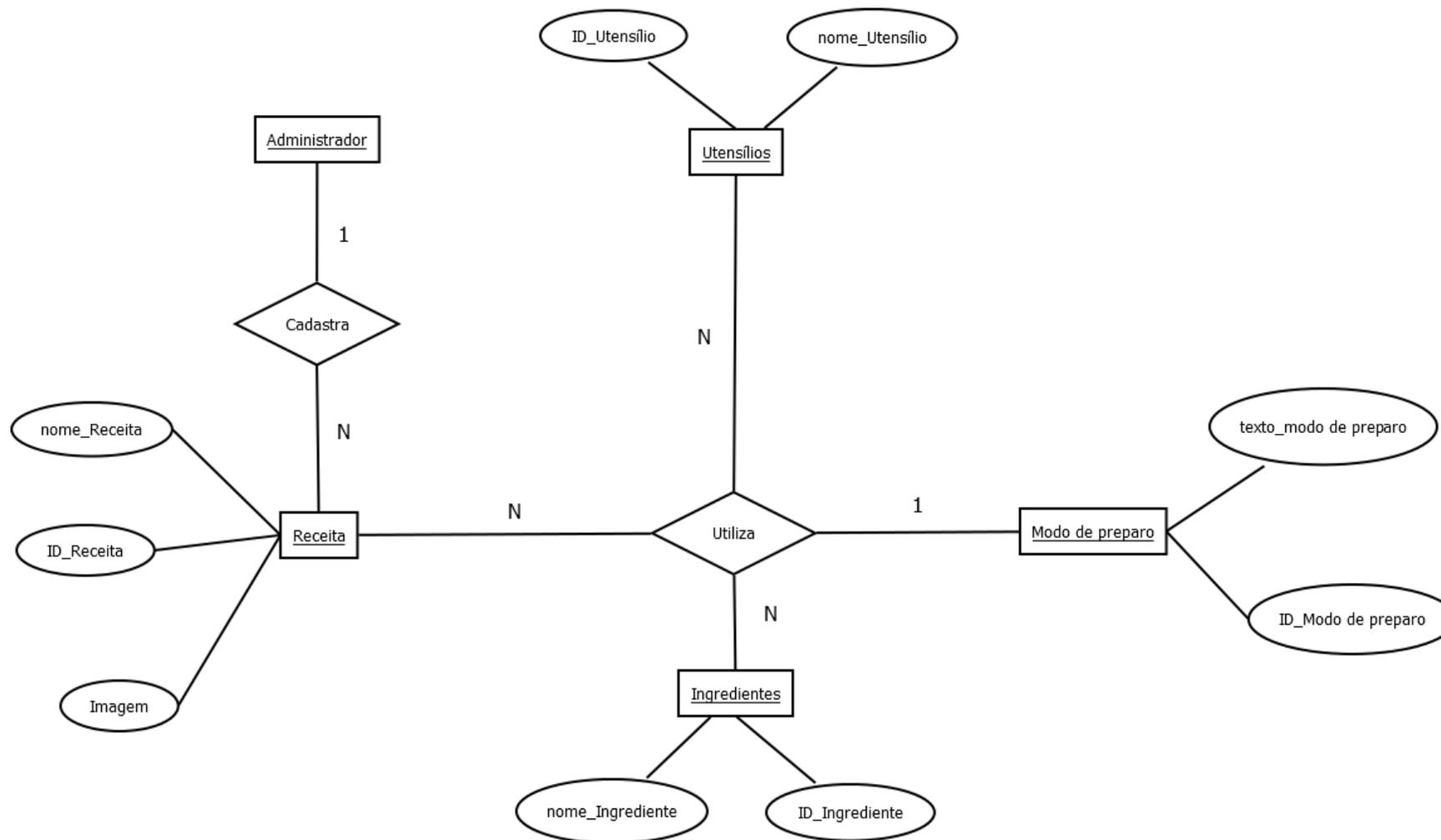


Fonte: autoria própria (2022)

4.3 DER

A figura a seguir ilustra o diagrama DER do sistema do aplicativo Prato Feito. Nota-se que os relacionamentos com cada entidade são descritos com a letra “N” e com o número “1”. Por exemplo, entre as entidades “usuário” e “receita” existe a ação “cadastra”, a letra “N” e com o número “1”, ou seja, um usuário pode cadastrar muitas receitas. Outro exemplo é entre as entidades “receita” e “ingredientes”. Muitas receitas podem precisar de muitos ingredientes.

Figura 9 – DER do aplicativo Prato Feito



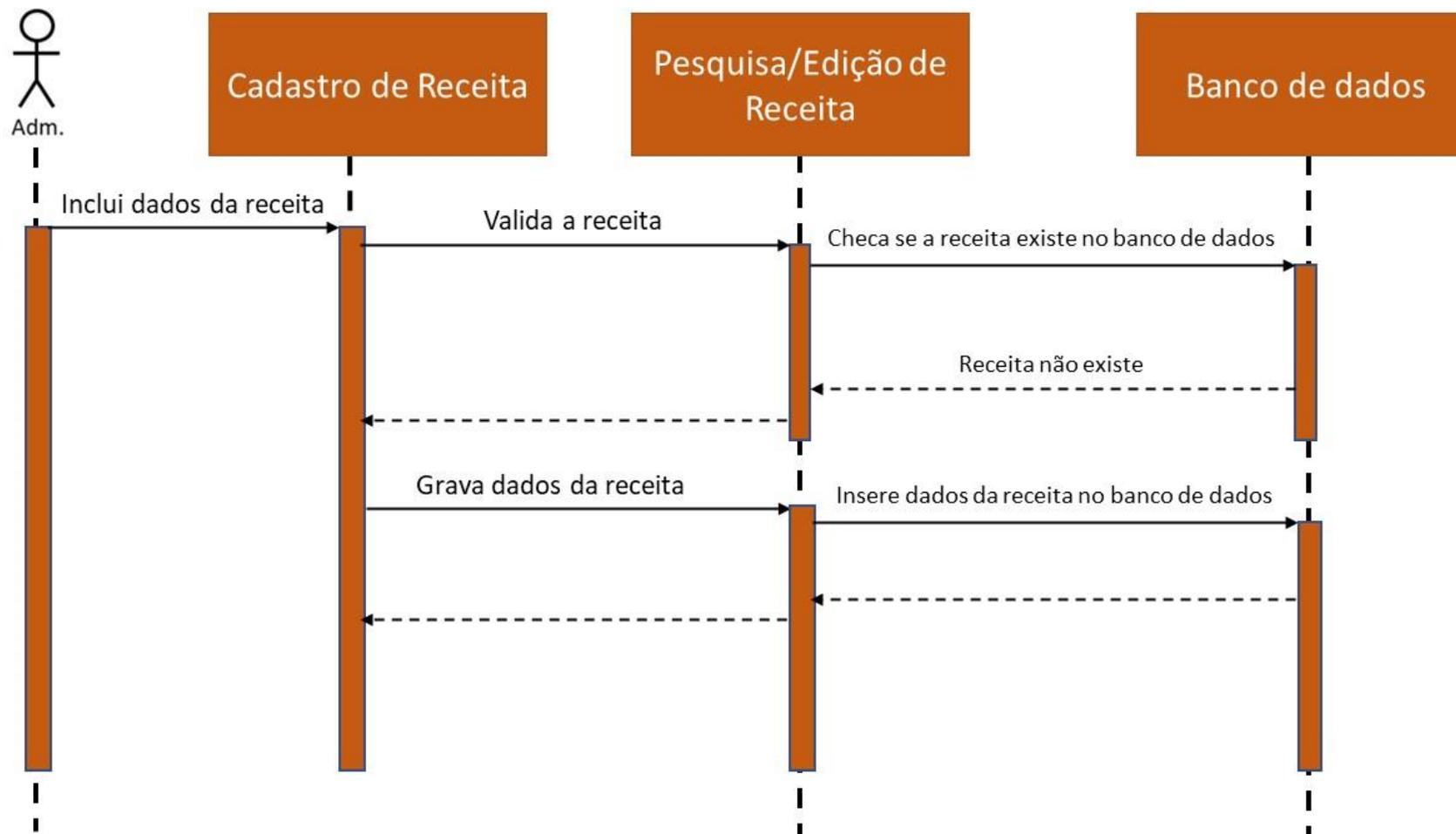
Fonte: autoria própria (2022)

4.4 DIAGRAMA DE SEQUÊNCIA

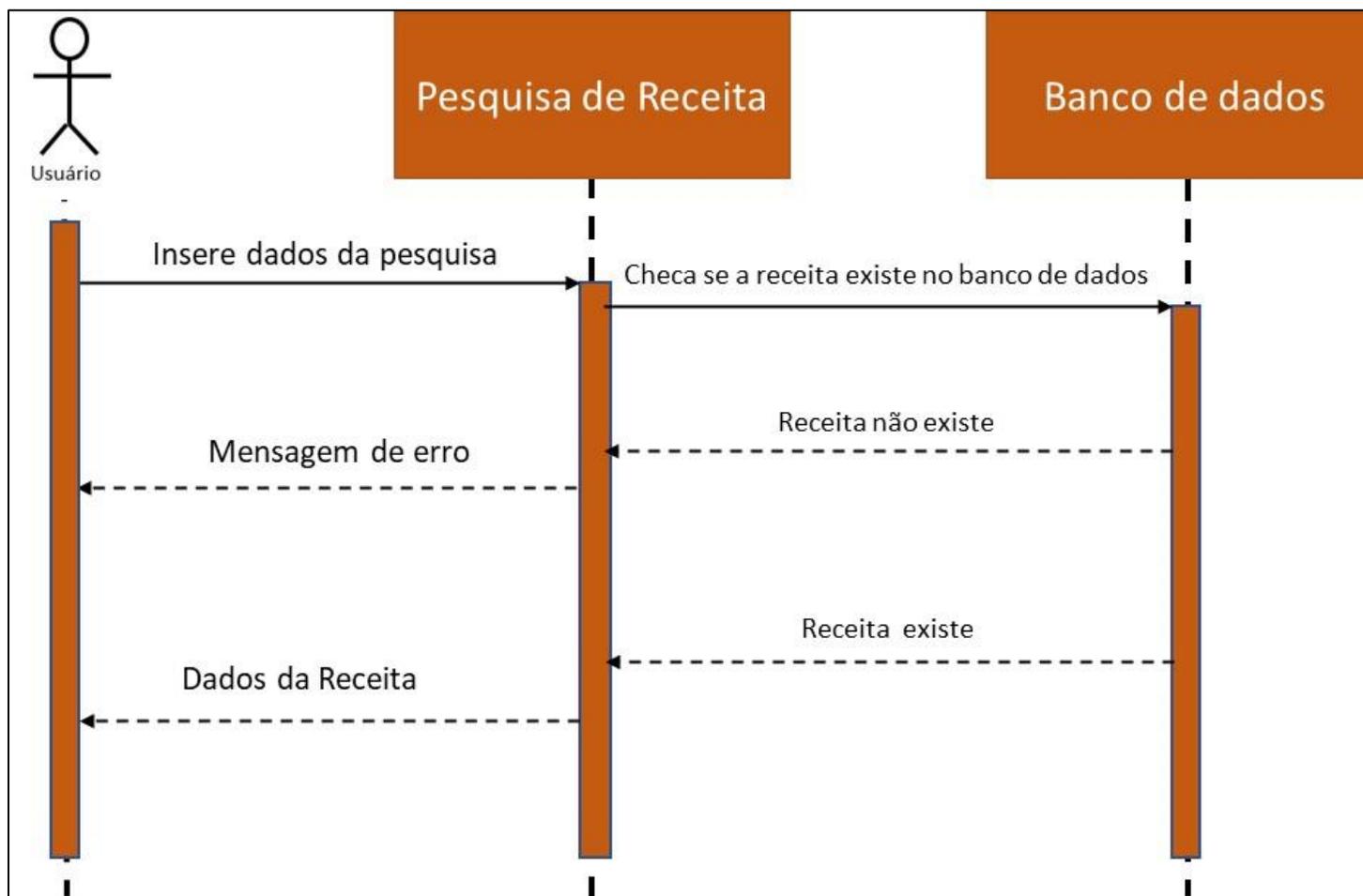
As figuras a seguir ilustram o diagrama de sequência do sistema do aplicativo Prato Feito. Na figura a seguir, o administrador cadastrará uma receita e ela será salva no banco de dados. Quando o administrador realizar a pesquisa será checado no banco de dados se a receita existe. Se ela não existir aparecerá na tela uma mensagem de erro. Porém, se a receita existir, o aplicativo retornará a receita.

Na próxima figura, no caso do usuário, ao realizar a pesquisa será checado no banco de dados se a receita existe. Se ela não existir aparecerá na tela uma mensagem de erro. Porém, se a receita existir, o aplicativo retornará a receita.

Figura 10 - Diagrama de sequência do Administrador



Fonte: autoria própria (2023)

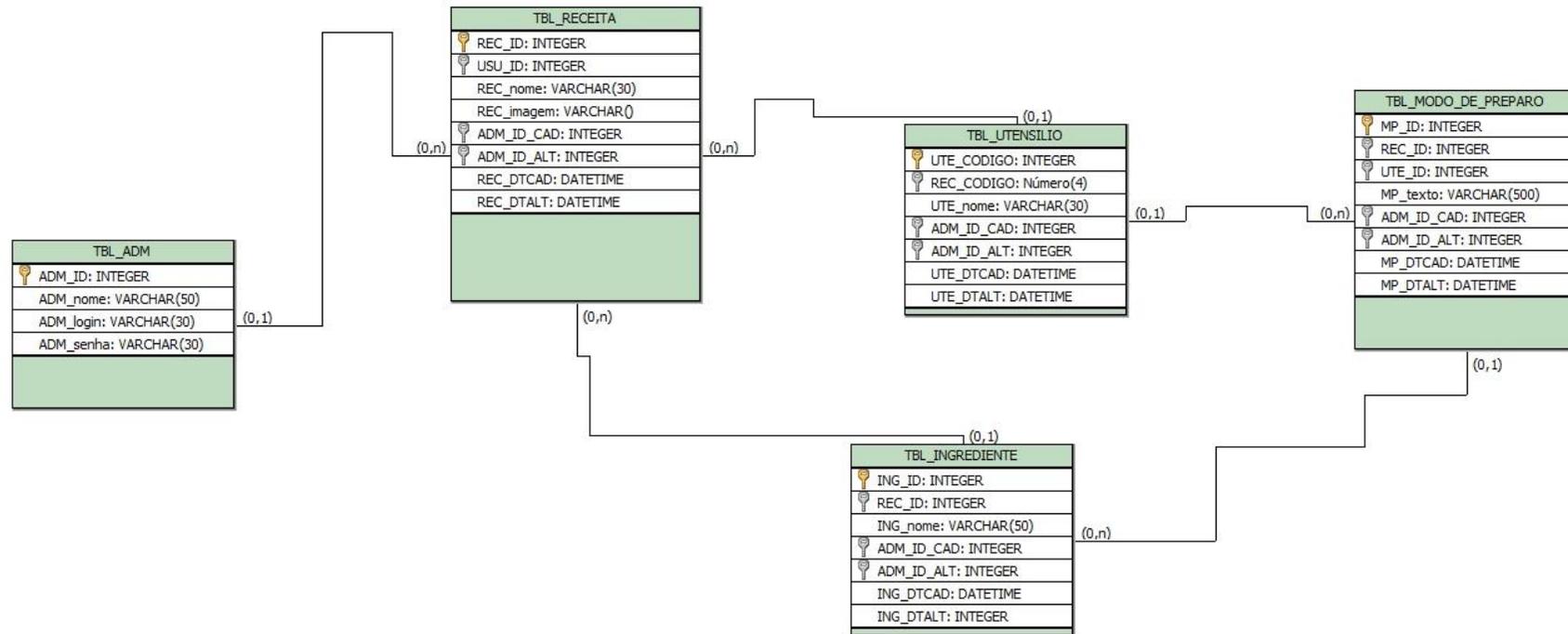


Fonte: autoria própria (2023)

4.5 MER

A figura a seguir ilustra o diagrama MER do sistema do aplicativo Prato Feito. Nota-se que o primeiro item de cada tabela ficou sendo a chave principal (*Primary key*). Por exemplo, a tabela do usuário (TBL_USUARIO) ficou sendo o número de identificação do usuário (ID_USU). Observa-se a presença de alguns itens necessários na criação do banco de dados, como as datas. Um exemplo disso pode ser visto na tabela do ingrediente, sendo o DTCAD e o DTALT como data de cadastro do dado e a data de alteração do dado, respectivamente. Observa-se a presença de chaves estrangeiras (*Foreign key*) na maioria das tabelas, por exemplo na tabela utensílio (TBL_UTENSILIO), sendo o número de identificação da receita (ID_REC) uma chave estrangeira na tabela em questão.

Figura 12 – MER do aplicativo Prato Feito



Fonte: autoria própria (2023)

4.6 PROTOTIPAGEM

Como explicado na metodologia, a prototipagem foi realizada antes da programação. Antes de tudo, foi realizado o cadastro de um dos autores deste trabalho. Em seguida, foi criado um projeto que foi nomeado como “Prato Feito”. Durante a criação do projeto, a plataforma solicitou o vínculo do banco de dados *Firebase* ao projeto e, em seguida, a escolha da versão de dispositivo móvel para que o aplicativo funcionasse. Foi selecionado o dispositivo de sistema operacional *Android* na versão 13.

A primeira etapa da prototipagem foi a criação da primeira tela, nomeada de “*Home Page*”. A figura a seguir ilustra essa tela, que possui a logo do aplicativo Prato Feito e um botão que leva ao menu principal, a tela seguinte.

Figura 13 – Prototipagem da tela “*Home Page*”



Fonte: autoria própria (2023)

A segunda etapa da prototipagem foi a criação da tela nomeada de "Menu principal". A figura a seguir ilustra essa tela. Diferente dos aplicativos pesquisados no mercado, apenas os aplicativos "Livro de Receitas" e "Cozinha Fit & Fat" apresentam uma tela

principal, que chamam de Homepage, semelhante ao Prato Feito, e apenas os aplicativos "Livro de Receitas" e "Tudo Gostoso Receitas" dão ao usuário a possibilidade de cadastrar uma receita. No entanto, a função que permite o usuário a cadastrar ou editar uma receita será apresentada em projetos futuros. Então, apenas os administradores podem realizar o cadastro ou edição da receita. Para isso, foi criada uma tela "*Login Cadastro/Edição*", em que o administrador pode colocar uma senha para ter o acesso da tela de cadastro ou edição de receitas. Além da opção de cadastrar ou editar receitas, essa tela também possui a opção de levar para a tela de pesquisa de receitas, assim como a opção de voltar à tela inicial.

Figura 14 – Prototipagem da tela "Menu principal" e da tela de autenticação para cadastrar ou editar receitas



Fonte: autoria própria (2023)

A terceira etapa da prototipagem foi a criação da terceira tela, nomeada de "*CadastoRecLogin* ", que o usuário acessa ao clicar na opção "Cadastrar/Editar" na tela acima. Nesta tela de cadastro, é importante notar que, no momento de sua criação, ela apenas disponibiliza o cadastro para os administradores do sistema. Além disso, nela existem as opções de cadastro e edição de receitas. A figura a seguir ilustra essa tela.

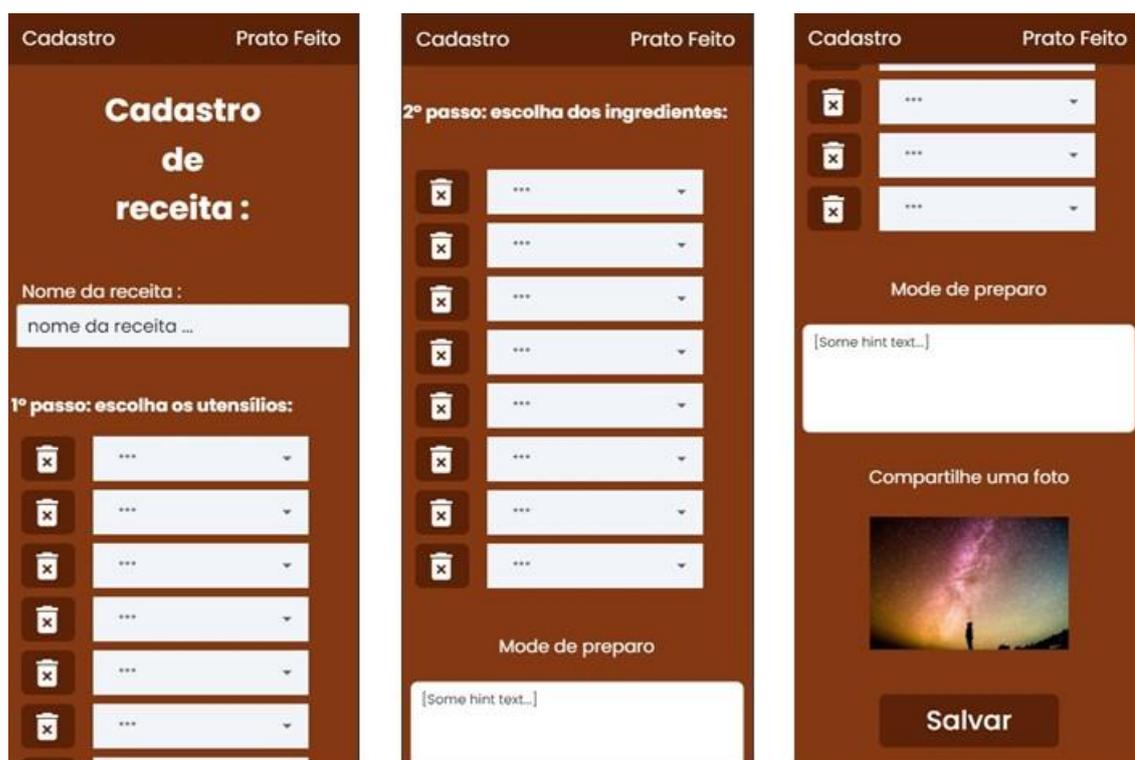
Figura 15 – Prototipagem da tela “Tela de *CadastoRecLogin*”

O prototipo da tela "Tela de CadastroRecLogin" apresenta um design com fundo marrom escuro. No topo, há uma barra de navegação com os botões "Cadastro" e "Prato Feito". O conteúdo principal começa com o alerta "Atenção!!!" em branco, seguido pelo texto "No momento apenas os administradores podem cadastrar e editar receitas." também em branco. Abaixo, o título "Login" em branco precede um campo de entrada de senha com o placeholder "senha". Na base da tela, há dois botões de ação em branco: "Cadastrar receita" e "Editar receita".

Fonte: autoria própria (2023)

A quarta etapa da prototipagem foi a criação da quarta tela, nomeada de "Cadastro de receita". Nela, o usuário pode cadastrar uma receita escolhendo utensílios, ingredientes, o modo de preparo e uma imagem da receita. Diferente dos aplicativos pesquisados no mercado, apenas os aplicativos "Livro de Receitas" e "Tudo Gostoso Receitas" dão ao usuário a possibilidade de cadastrar uma receita. A figura a seguir ilustra essa tela, onde são exibidas, respectivamente, as opções de inserir o nome da receita, de escolher os utensílios, de escolher os ingredientes, de descrever o modo de preparo e de compartilhar uma foto da receita.

Figura 16 – Prototipagem da tela “Cadastro de receita”.



Fonte: autoria própria (2023)

A quinta etapa da prototipagem foi a criação da quinta tela, nomeada de "Edição de receita". Nesta tela, o usuário possui a capacidade de editar e excluir as receitas já cadastradas na tela anterior. Enquanto o ícone simbolizado por uma lixeira faz com que a receita seja apagada, o ícone simbolizado por caneta e papel faz com que a receita seja editada. A figura a seguir ilustra essa tela.

Figura 17 – Prototipagem da tela “Edição de receita”



Fonte: autoria própria (2023)

A sexta etapa da prototipagem foi a criação da sexta tela, nomeada de “*Escolha de Pesquisas*”. A figura a seguir ilustra essa tela. Nesta tela, é possível realizar pesquisas de receitas por três maneiras diferentes: nome de receita, utensílios ou ingredientes, conforme a figura. É também possível voltar ao menu principal clicando no botão “Voltar”.

Figura 18 – Prototipagem da tela “Escolha de Pesquisas”



Fonte: autoria própria (2023)

A sétima etapa da prototipagem foi a criação da sétima tela, nomeada de “Pesquisa por nome”. A figura a seguir ilustra essa tela. Nesta tela, o usuário poderá pesquisar a receita inserindo o nome dela na caixa de texto para encontrar uma receita com o mesmo nome inserido.

Figura 19 – Prototipagem da tela “Pesquisar por nome”



Fonte: autoria própria (2023)

A oitava etapa da prototipagem foi a criação da oitava tela, nomeada de “Pesquisa por utensílios”. A figura a seguir ilustra essa tela. O usuário poderá realizar a busca selecionando ou removendo os utensílios disponíveis nas caixas de seleção.

Figura 20 – Prototipagem da tela “Pesquisa por utensílios”



Fonte: autoria própria (2023)

A nona etapa da prototipagem foi a criação da nona tela, nomeada de “Pesquisa por ingredientes”. A figura a seguir ilustra essa tela. O usuário poderá pesquisar a receita selecionando ou removendo os ingredientes disponíveis nas caixas de seleção, semelhante à tela anterior.

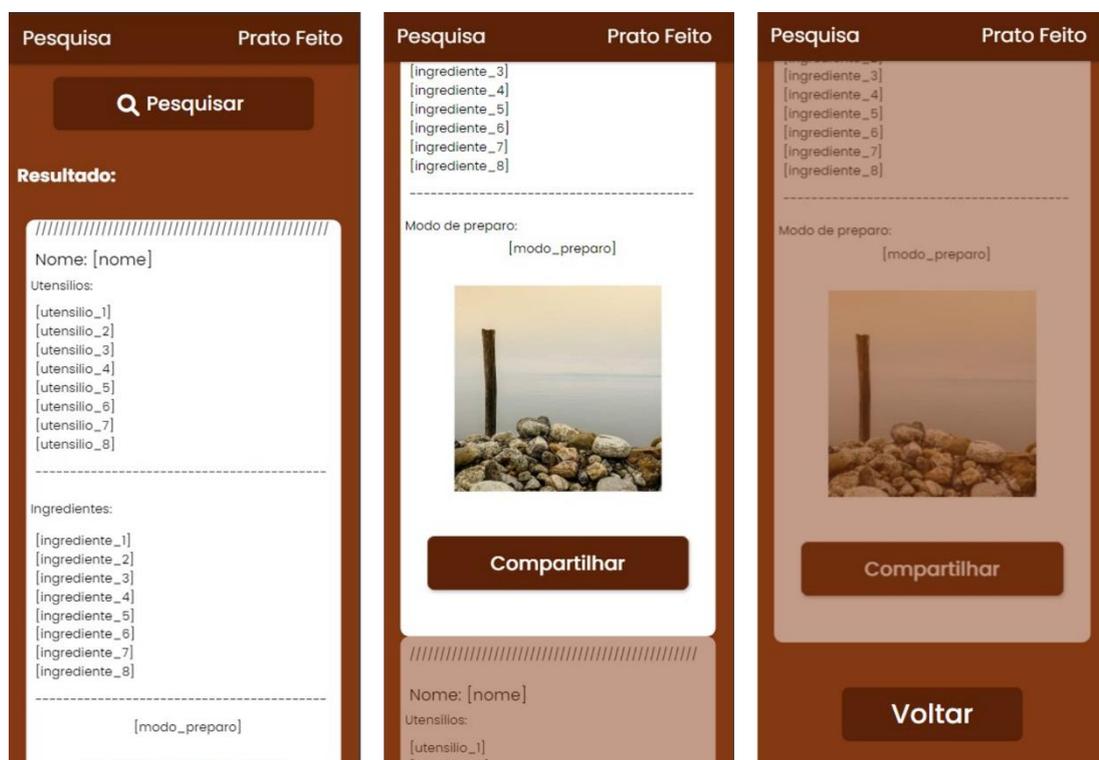
Figura 21 – Prototipagem da tela “Pesquisar por ingredientes”

O prototipo da tela "Pesquisar por ingredientes" apresenta um cabeçalho com duas opções de navegação: "Pesquisa" e "Prato Feito". O título principal da tela é "Pesquisa de receita por ingredientes:". Abaixo do título, há uma seção intitulada "Escolha dos ingredientes:" que contém sete campos de entrada. Cada campo é composto por um ícone de lixeira com um 'x' dentro, seguido por um campo de texto contendo "..." e um ícone de seta para baixo, indicando que se trata de uma lista suspensa.

Fonte: autoria própria (2023)

A décima etapa da prototipagem foi a criação da décima tela, nomeada de “Resultado de pesquisa por nome de receita”. A figura a seguir ilustra essa tela. Nesta tela, é possível observar os resultados da busca pelo nome da receita inserido na tela “Pesquisar por nome de receita” pelo usuário.

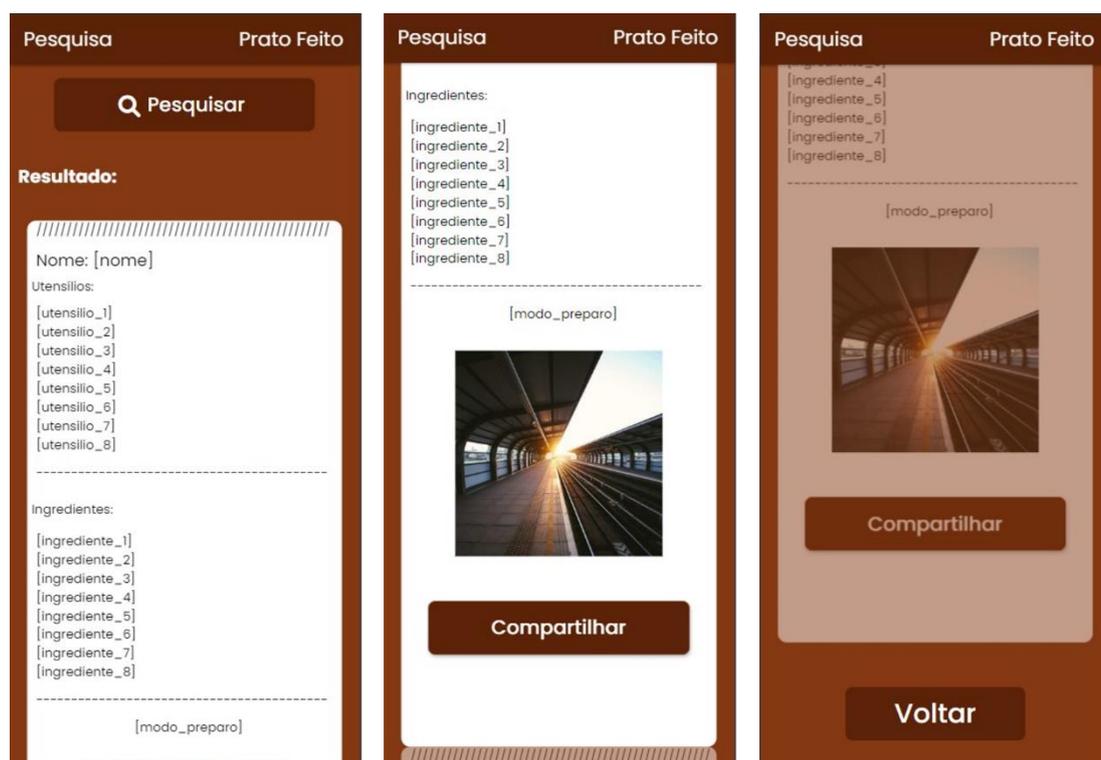
Figura 22 – Prototipagem da tela “Resultado de pesquisa por nome de receita”



Fonte: autoria própria (2023)

A penúltima etapa da prototipagem foi a criação da décima primeira tela, nomeada de “Resultado de pesquisa por utensílios”. A figura a seguir ilustra essa tela. Nesta tela, é possível observar os resultados da busca de receitas com base na seleção de utensílios feita pelo usuário na tela “Pesquisar por utensílios”.

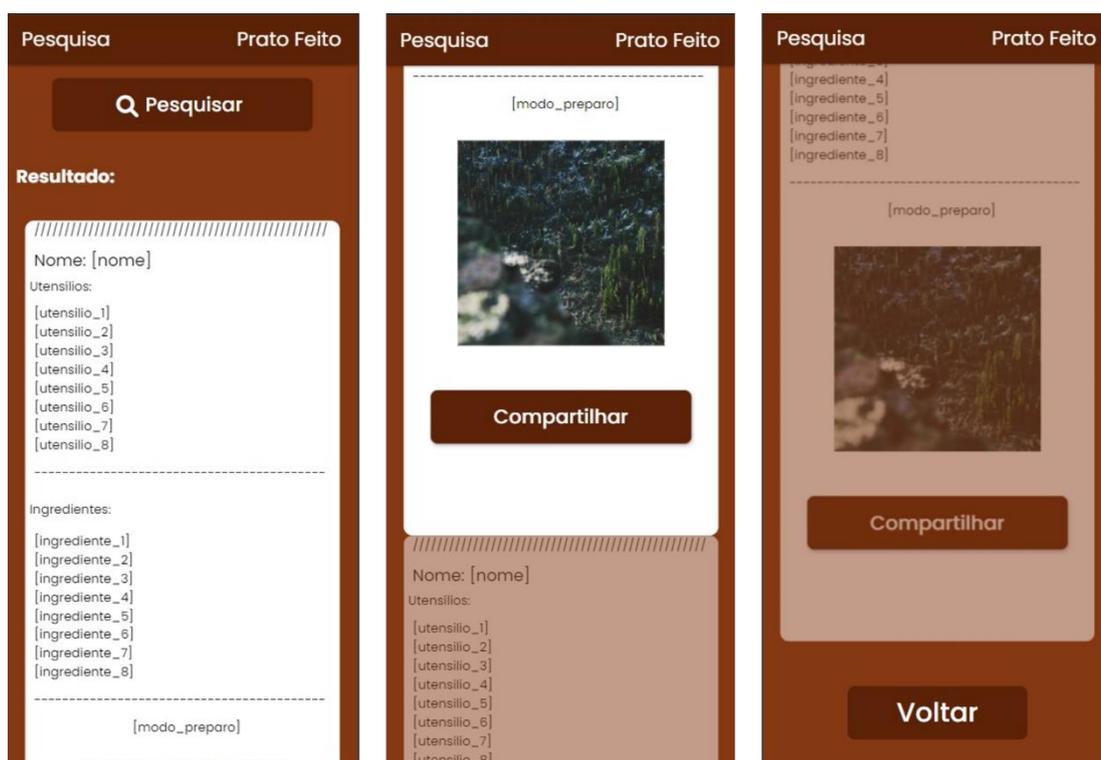
Figura 23 – Prototipagem da tela “Resultado de pesquisa por utensílios”



Fonte: autoria própria (2023)

A última etapa da prototipagem foi a criação da décima segunda e última tela, nomeada de “Resultado de pesquisa por ingredientes”. A figura a seguir ilustra essa tela. Nesta tela, é possível observar os resultados da busca de receitas com base na seleção de ingredientes feita pelo usuário na tela “Pesquisar por ingredientes”, em um processo bem semelhante ao de pesquisa por utensílios.

Figura 24 – Prototipagem da tela “Resultado de pesquisa por ingredientes”



Fonte: autoria própria (2023)

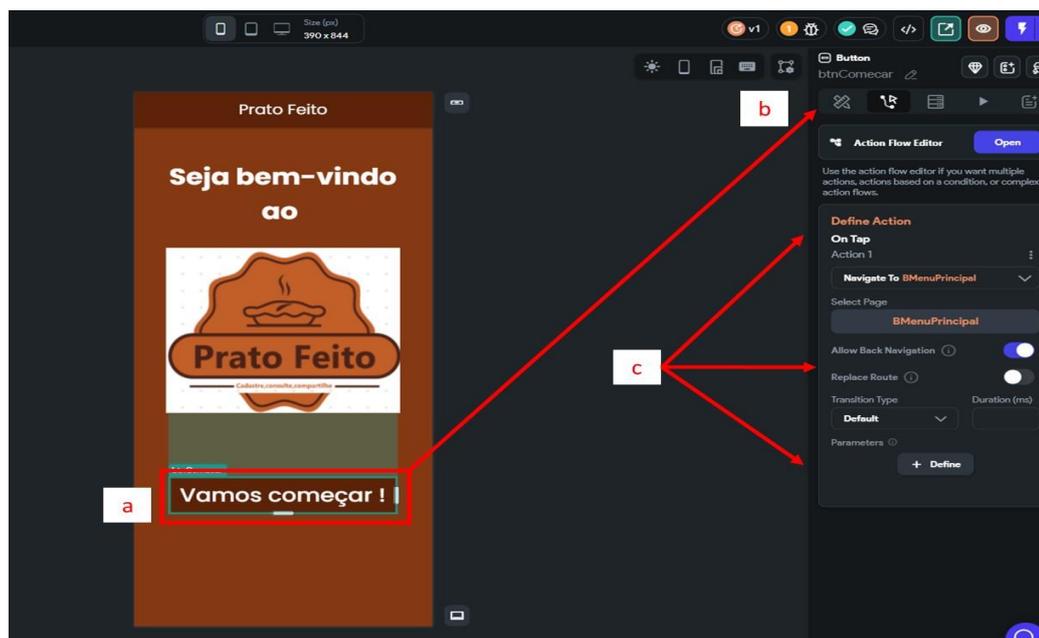
4.7 RESULTADOS OBTIDOS

Este tópico está dividido em três partes: a primeira sobre a programação, a segunda sobre o banco de dados e a terceira sobre a fase de testes do aplicativo Prato Feito. Nesses tópicos serão descritos exemplos do que foi realizado para a programação do aplicativo.

4.7.1 Programação

A programação foi realizada adicionando ações conforme descrito no *FlutterFlow*. A ação foi aplicada no botão que leva para a tela “Menu Principal”. Então, foi selecionado o botão nomeado como “btnComecar” (Fig. 25 “a”) e clicado na barra de ações (Fig. 25 “b”), que fica na coluna de propriedades do objeto (Fig. 25 “c”).

Figura 25 – Exemplo do uso da coluna de propriedades de objeto



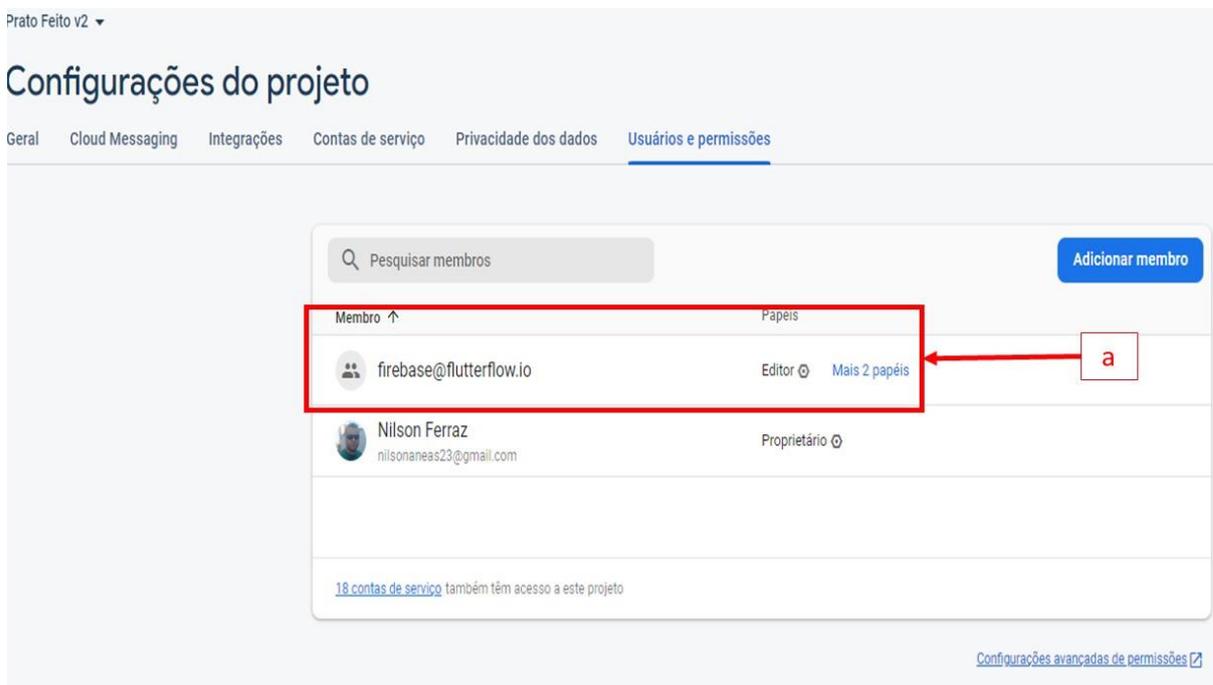
Fonte: autoria própria (2023)

4.7.2 Banco de dados

Como visto no primeiro parágrafo da prototipagem, o banco de dados teve que ser selecionado na hora de criar o projeto. Por isso, o banco de dados foi criado antes do projeto. Então, primeiro um dos desenvolvedores deste projeto vinculou o seu e-mail ao *Firestore*. Em seguida, foi criado um projeto cujo nome foi Prato Feito. Após a criação do projeto, o *Firestore* leva o desenvolvedor a ajustar a autenticação do banco de dados ao projeto do *FlutterFlow*. No *Google Cloud* (Fig. 26), o proprietário do banco de dados, um dos autores, aciona o acesso de superadministrador (Fig. 26 “a”) do banco e libera para os usuários cadastrarem suas receitas e imagens delas no banco de dados do proprietário.

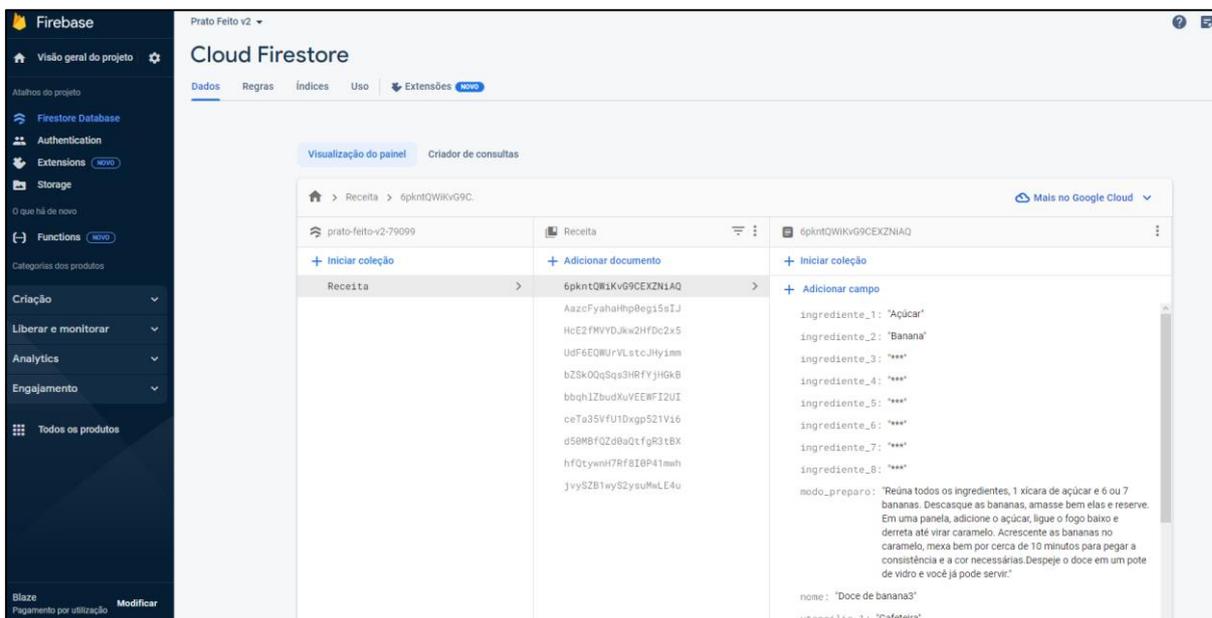
O *Google Cloud*, junto ao *Firestore*, libera uma chave de acesso, códigos com letras e números, para que o desenvolvedor a vincule ao *FlutterFlow*. Após a liberação dessa chave, o desenvolvedor, quando está desenvolvendo o programa, deve acessar a parte que gerencia o banco de dados no *FlutterFlow*, conforme ilustra a figura 27 e o aplicativo têm o vínculo com o banco de dados.

Figura 26 – Exemplo de programação para atribuição do *FlutterFlow* ao *Firebase*.



Fonte: autoria própria (2023)

Figura 27 – Exemplo de programação para atribuição do *FlutterFlow* ao *Firebase*

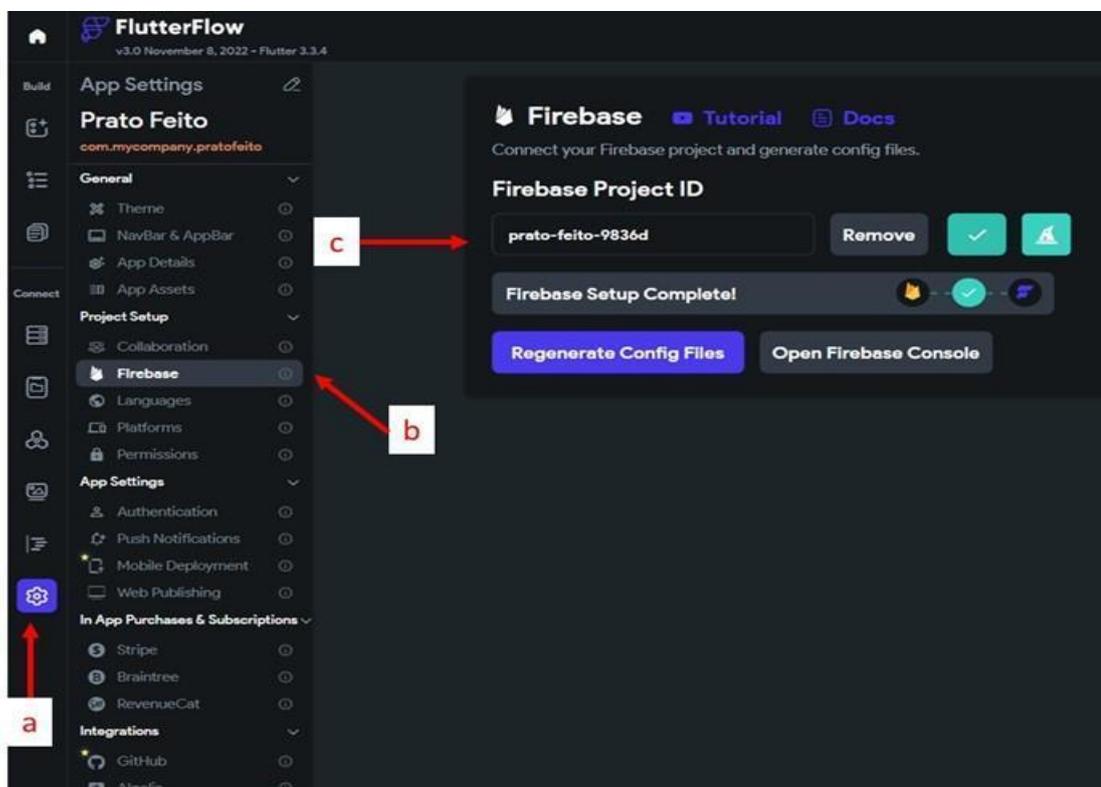


Fonte: autoria própria (2023)

O desenvolvedor acessou o ícone de propriedades (Fig. 28 "a"), em seguida acessou o item *Firebase* (Fig. 28 "b") e depois colocou a chave liberada pelo *Google Cloud* e *Firebase* (Fig. 28 "c"), que aciona o vínculo com o banco de dados. Com

isso, o projeto

Figura 28 – Exemplo de programação para atribuição do *FlutterFlow* ao *Firebase*

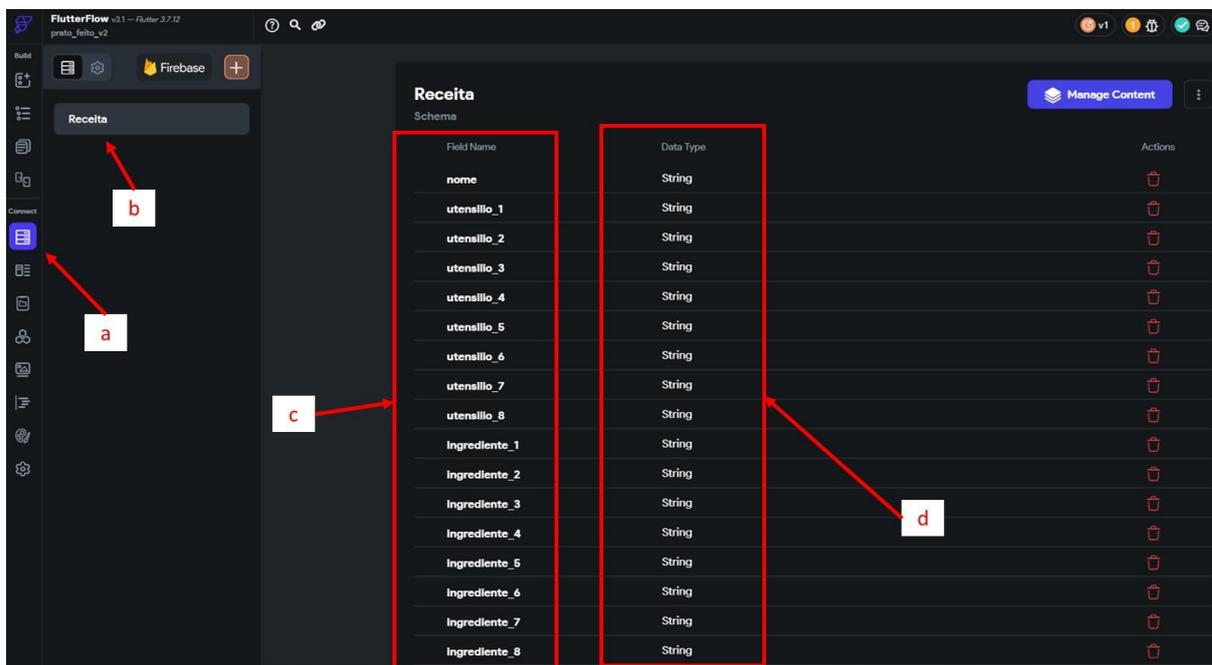


Fonte: autoria própria (2023)

4.7.2.1 Programação *FlutterFlow* para envio de dados ao banco de dados

Para continuar a programação do aplicativo, os desenvolvedores tiveram que criar uma tabela no item de nome *Firestore* (Fig. 29 "a") para o vínculo entre o programa e o banco de dados, nomeada de Receita (Fig. 29 "b"). Na tabela Receita foram criadas variáveis (Fig. 29 "c") com o seu tipo de dado (Fig. 29 "d"), ou seja, **nome** (nome da receita) sendo do tipo *string*.

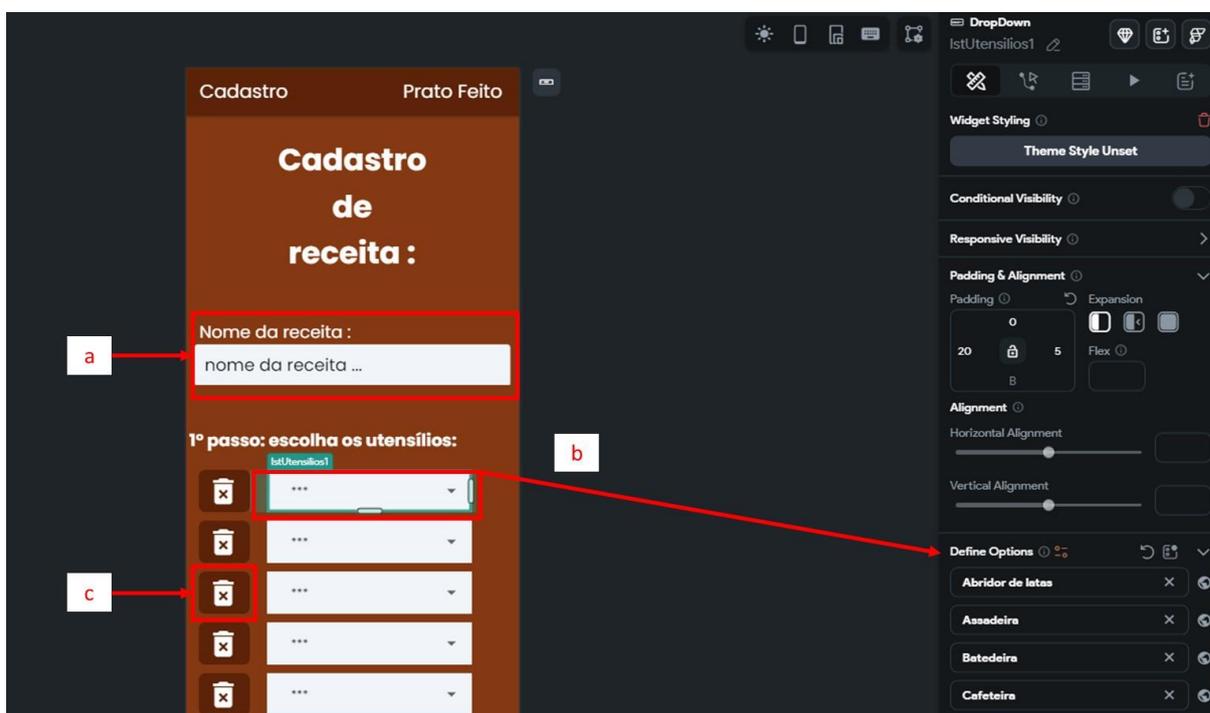
Figura 29 – Criação da tabela usuário e variáveis para registro no banco de dados



Fonte: autoria própria (2023)

Em seguida, os desenvolvedores acessaram a tela de desenvolvimento "Cadastrar receita", por exemplo, acrescentaram uma caixa de texto ("*text field*") (Fig. 30 "a") para ser digitado o nome da receita. Na Fig. 30 "b" foi acrescentado uma lista ("*dropdown*") de utensílios e botões de deletar (Fig. 30 "c") com a ação de excluir o item selecionado.

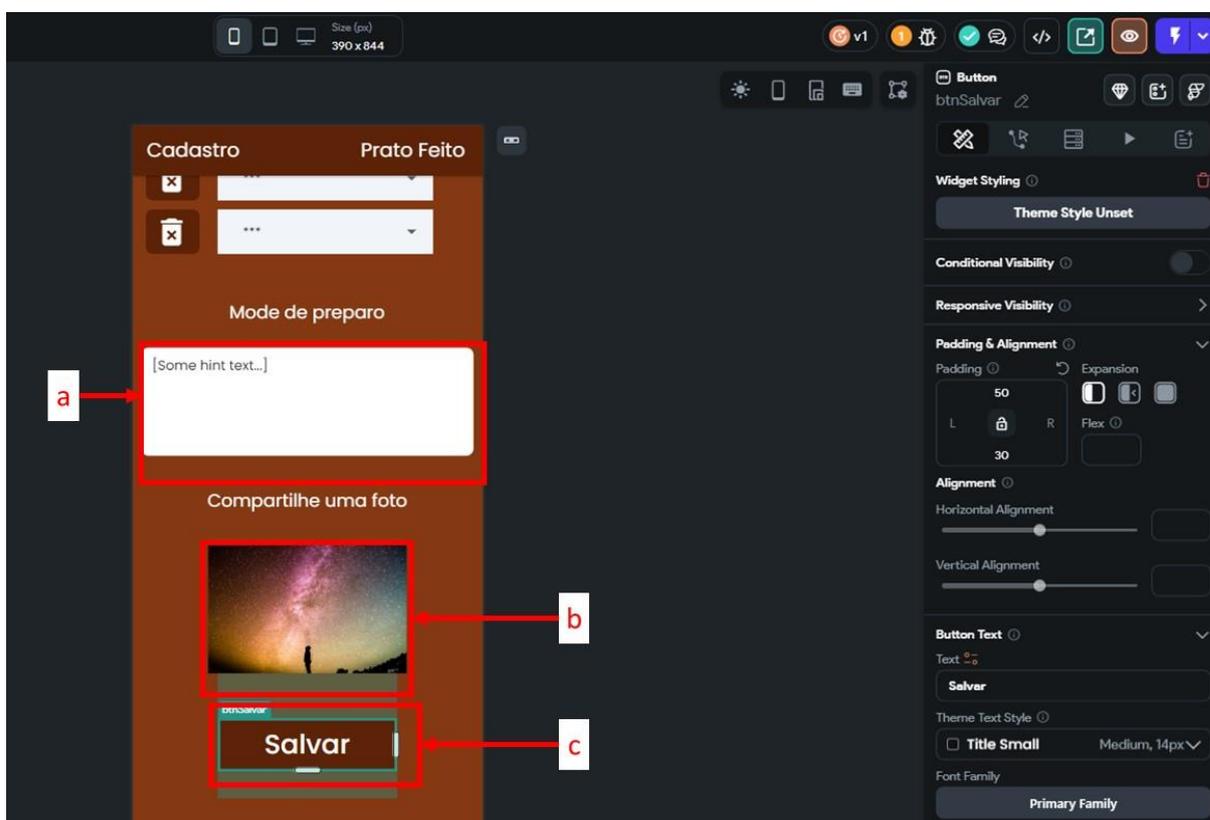
Figura 30 – Nomeação da receita e lista de escolhas de itens



Fonte: autoria própria (2023)

Em seguida, os desenvolvedores acrescentaram uma caixa de texto, nomeada com `txtModoDePreparo` (Fig. 31 “a”), para que o seja escrita o modo de preparo da receita. Em seguida, um botão em forma de imagem (Fig. 31 “b”), nomeado como `fotoReceita`, foi acrescentado para com a ação de buscar uma foto da galeria do aparelho ou o acionamento da câmera para tirar uma foto da receita pronta. Após, um botão nomeado como `btnSalvar` (Fig. 31 “c”) foi acrescentado com a ação salvar toda a receita.

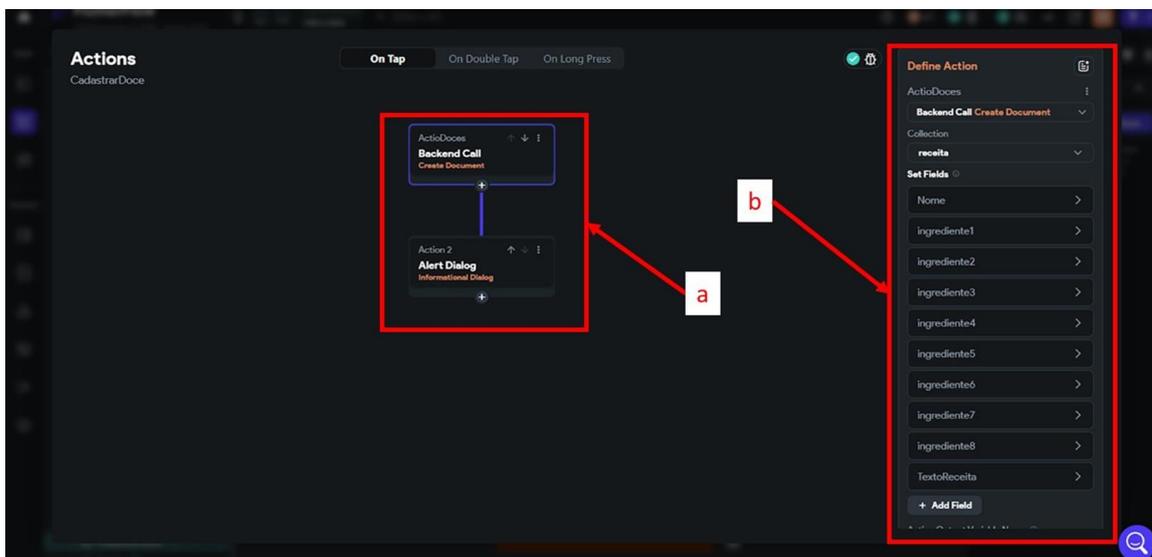
Figura 31 – Botão com a ação de salvar a receita



Fonte: autoria própria (2023)

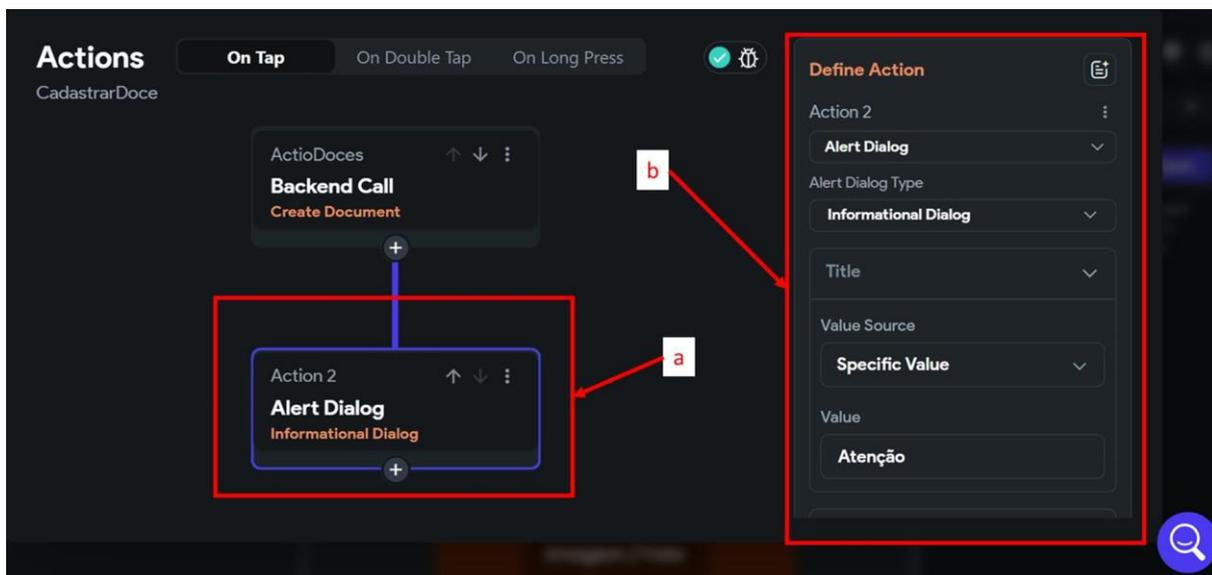
Após isso, foram criadas duas ações denominadas "*backend call*" (Fig. 32 "a") para criar a documentação no banco de dados e outra para a caixa de diálogo (*Alert Dialog*, Fig. 32 "b" e 33 "a"), a fim de informar se a receita foi cadastrada com sucesso ou não. Todas as listas (Fig. 33 "b") de utensílios, de ingredientes, caixas de texto, imagem da receita e o modo de preparo foram atribuídas à tabela receita criada com vínculo da plataforma em questão ao *Firebase*, conforme ilustrado na figura 29. Nota-se que a segunda ação (Fig. 33 "a"), denominada como "*Alert Dialog*", está abaixo da "*backend call*". Isso teve que seguir essa sequência lógica para que, após realizar o preenchimento das caixas de texto com o nome, ingredientes e o modo de preparo da receita, a caixa de diálogo surja na tela informando o status da receita.

Figura 32 – Atribuição dos "dropdowns e textfields" (nome, utensílios, ingredientes e modo de preparo, à tabela receita



Fonte: autoria própria (2023)

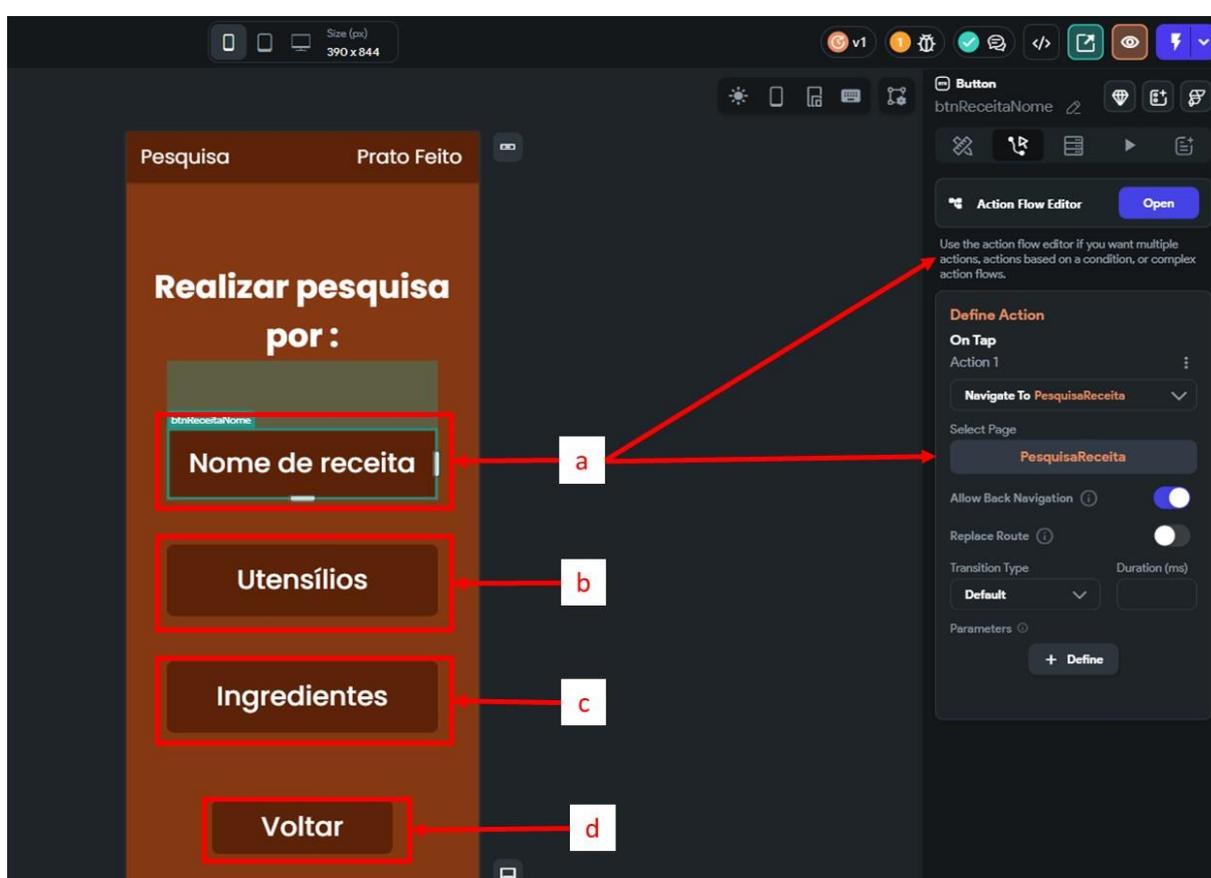
Figura 33 – Criação da caixa de diálogo após-cadastro



Fonte: autoria própria (2023)

Depois, na tela “*Pesquisa*” (Fig. 34), foram realizadas as programações para cada um deles, sendo o botão nomeado de “btnReceitaNome” (Fig. 34 “a”) para transferir o usuário para a tela de pesquisa de receita por nome, “btnUtensilio” (Fig. 34 “b”) para transferir o usuário para a tela de pesquisa de receita por utensílios, “btnIngredientes” (Fig. 34 “c”) para transferir o usuário para a tela de pesquisa de receita por ingredientes e “btnVoltar” (Fig. 34 “d”) para transferir o usuário para a tela “*Menu principal*”.

Figura 34 – Tela do *Menu Principal*



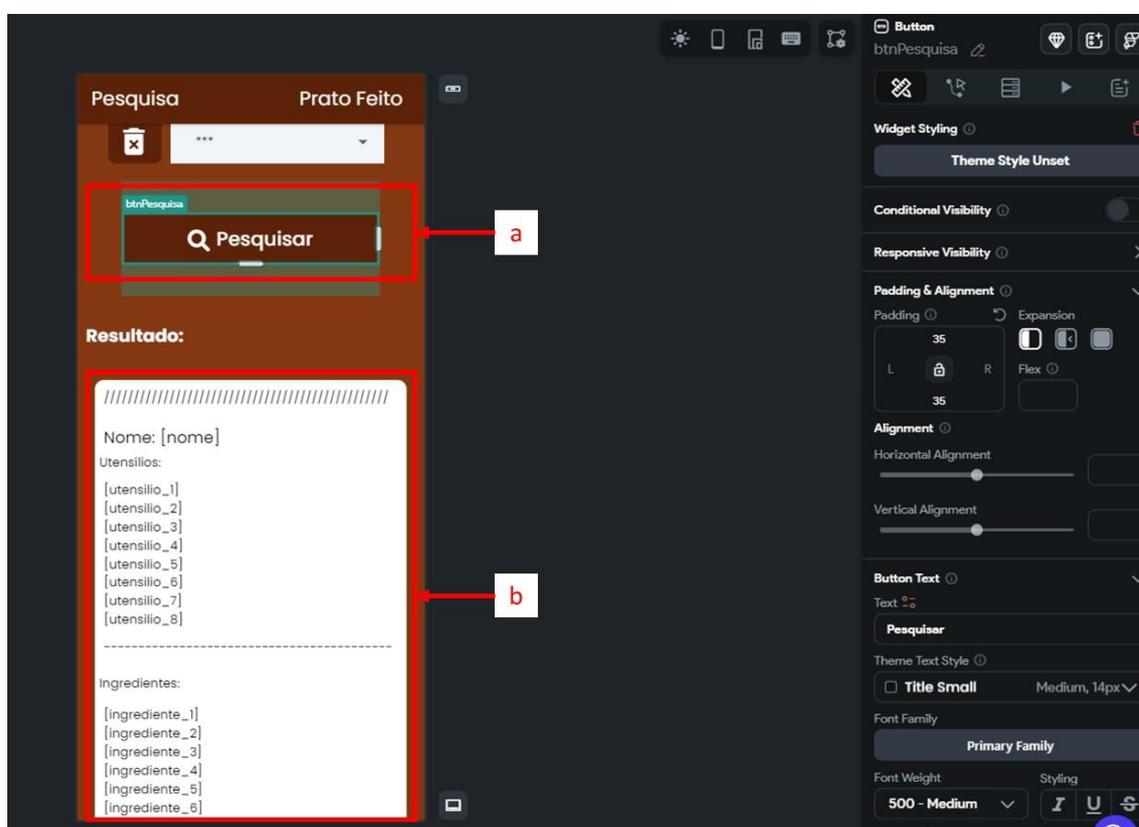
Fonte: autoria própria (2023)

As telas “*Pesquisa por nome*”, “*Pesquisa por utensílio*” e “*Pesquisa por ingredientes*” têm as mesmas funcionalidades. A seguir será apresentado um exemplo da programação da tela “*Pesquisa por utensílio*” (Fig. 35).

Na tela “*Pesquisa por utensílio*”, foi copiado e colado todas as listas (dropdowns) da tela “*Cadastrar receita*”. Em seguida, foi realizada ação do botão nomeado como “btnPesquisa” (Fig. 35 “a”) o vínculo do tipo de busca, no caso pelo *Algolia Search* (Fig. 36), e sendo atribuídos todas as listas de ingredientes, imagem e

o campo com a resposta da busca (*container*), conforme a figura 33 “b”. O container (Fig. 35 “b”) foi acrescentado para que quando a busca for solicitada, cada item da busca se auto localize em forma de texto. Para que a resposta da busca faça toda “varredura” no banco de dados, foi realizada a conversão do container em uma “*query*”, que é baseada em um código pré-definido para obter as respostas que deseja. Com isso, no *container* foram acrescentados itens como nome, utensílio e ingrediente e depois foram atribuídos cada um desses itens ao banco de dados.

Figura 35 – Tela pesquisa por utensílios. Ações atribuídas para busca no banco de dados

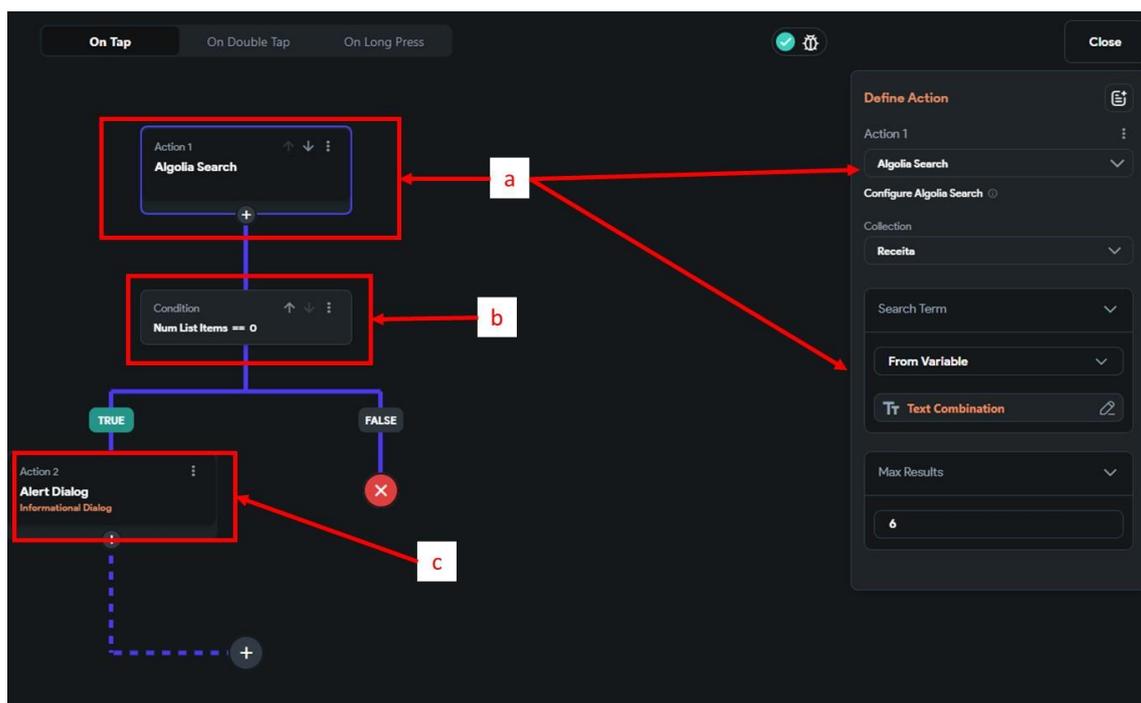


Fonte: autoria própria (2023)

Nota-se que na figura 36, a busca do *Algolia* (Fig. 36 “a”) está atribuída as variáveis (*From Variable*), que nada mais são que os itens selecionados nas listas de utensílios. Para que a busca seja realizada de forma “refinada”, foi utilizado o método de busca com texto **concatenado** (*Text Combination*) para que apenas seja devolvida a resposta de busca de uma receita que tenha os itens selecionado. Uma lógica foi selecionada (Fig. 36 “b”) para que se o usuário realizar uma busca de receita

apresentado uma lista de utensílio que não tem em nenhuma receita no banco de dados (**Num List Items == 0**, Fig. 36 “b”), uma mensagem de alerta (Fig. 36 “c”) será apresentada ao usuário constando que não foram encontradas receitas com os itens selecionados. Caso contrário, a resposta apresentará o *container* com as receitas encontradas no banco de dados.

Figura 36 – Tela pesquisa por utensílios. Programação do *Algolia*

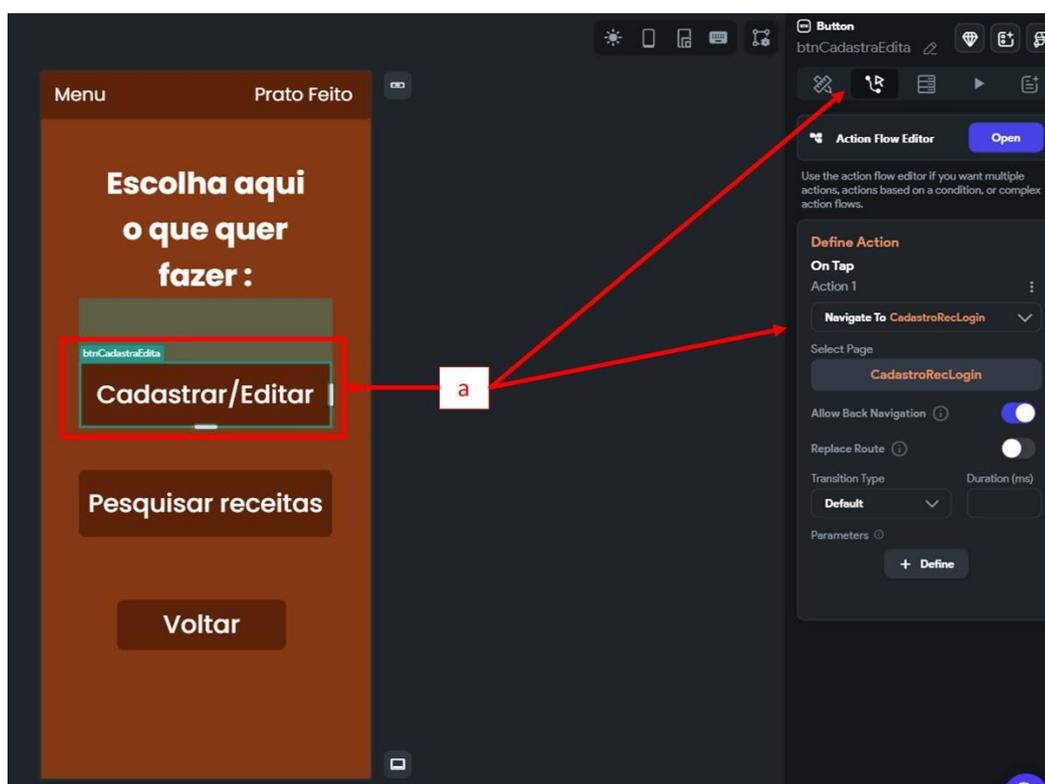


Fonte: autoria própria (2023)

4.7.2.2 Edição de receitas

Ao acessar a tela “*Menu Principal*” (Fig. 37) foi atribuída a ação para cada botão, sendo o botão nomeado “*btnCadastraEdita*” que transfere o administrador para a tela de “*CadastoRecLogin*”, “*btnEscolhaPesquisa*” que transfere o administrador para a tela de “*Escolha de Pesquisas*” e o botão nomeado de “*btnVoltar*” que transfere o usuário para a “*Home Page*”.

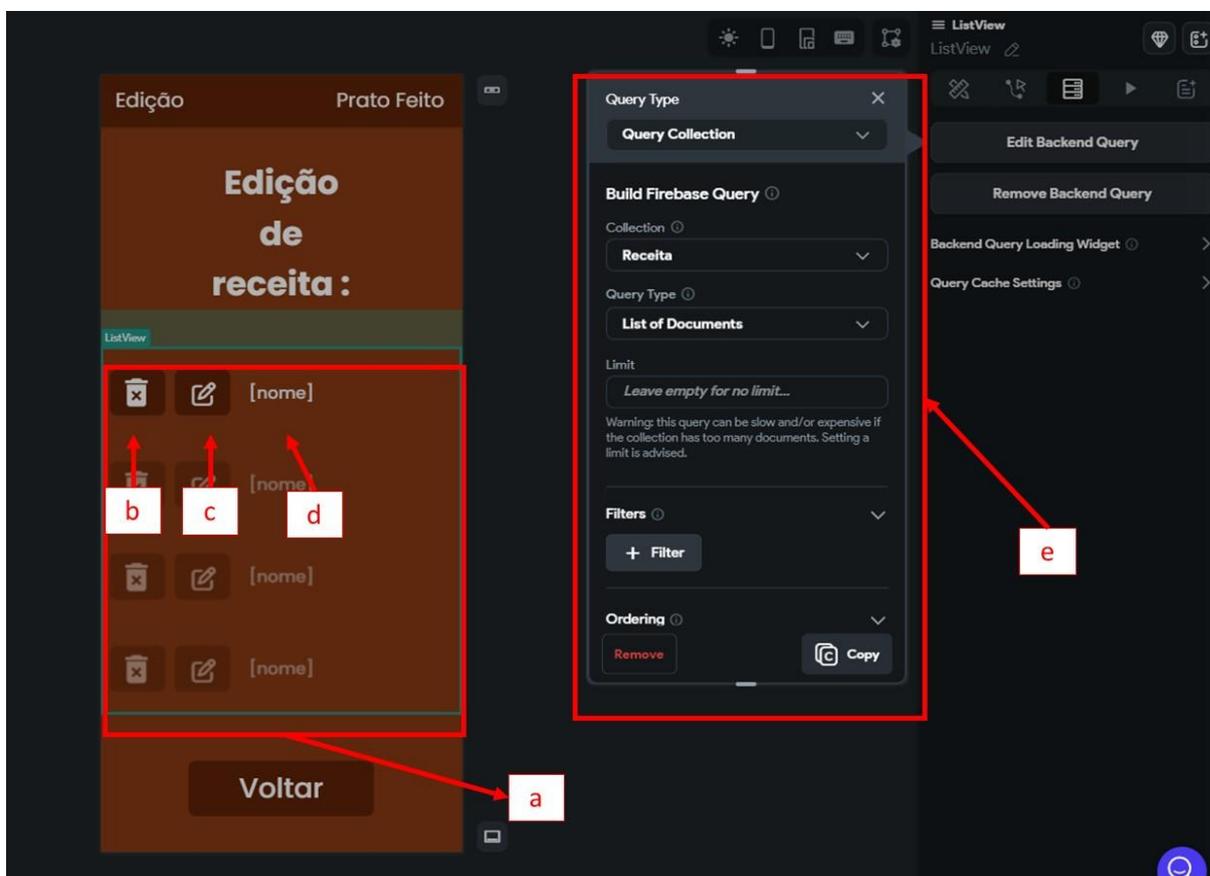
Figura 37 – Tela Menu Principal



Fonte: autoria própria (2023)

Ao acessar a tela “*Edição de Receita*” (Fig. 38) foi criada uma “*listView*” (Fig. 38 “a”) que apresenta a lista com os nomes das receitas (Fig. 38 “d”) e para cada receita foram colocados dois botões, sendo um para deletar (Fig. 38 “b”) e outro para editar a receita (Fig. 38 “c”). Porém, para que a lista possa ser carregada com as receitas cadastradas no banco de dados, o “*listView*” foi atribuído a uma “*query*” Fig. 38 “e”) que por sua vez foi atribuída a coleção “*Receita*” criada no *Firestore*.

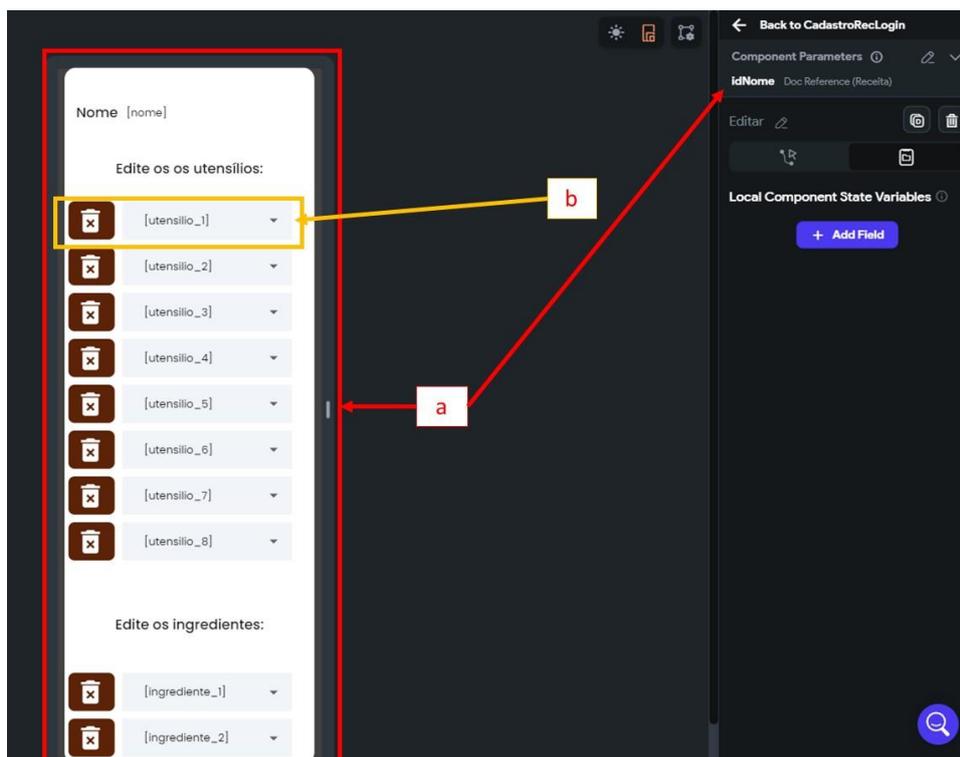
Figura 38 – Tela de “Edição de Receita”



Fonte: autoria própria (2023)

Em seguida, foi criada uma um componente parametrizado (Fig. 39 “a”) que apresenta a lista com os “*dropdowns*” de utensílios, ingredientes, a imagem e o modo de preparo da receita registrada no banco de dados. A parametrização foi realizada para que quando o administrador clicar no botão de edição, seja apresentado o componente em questão com todos os dados da receita para ser editado (Fig. 39 “b”). Nota-se a presença do botão de deletar, para que o administrador delete o item desejado. Para que o administrador possa salvar as alterações que deseja fazer, basta clicar no botão “btnSalvar”, atribuído ao banco de dados e coleção “Receita” no *Firestore* (Fig. 40).

Figura 39 – Tela de “Edição de Receita”, componente parametrizado



Fonte: autoria própria (2023)

Figura 40 – Tela de “Edição de Receita”, componente parametrizado com o botão “Salvar”

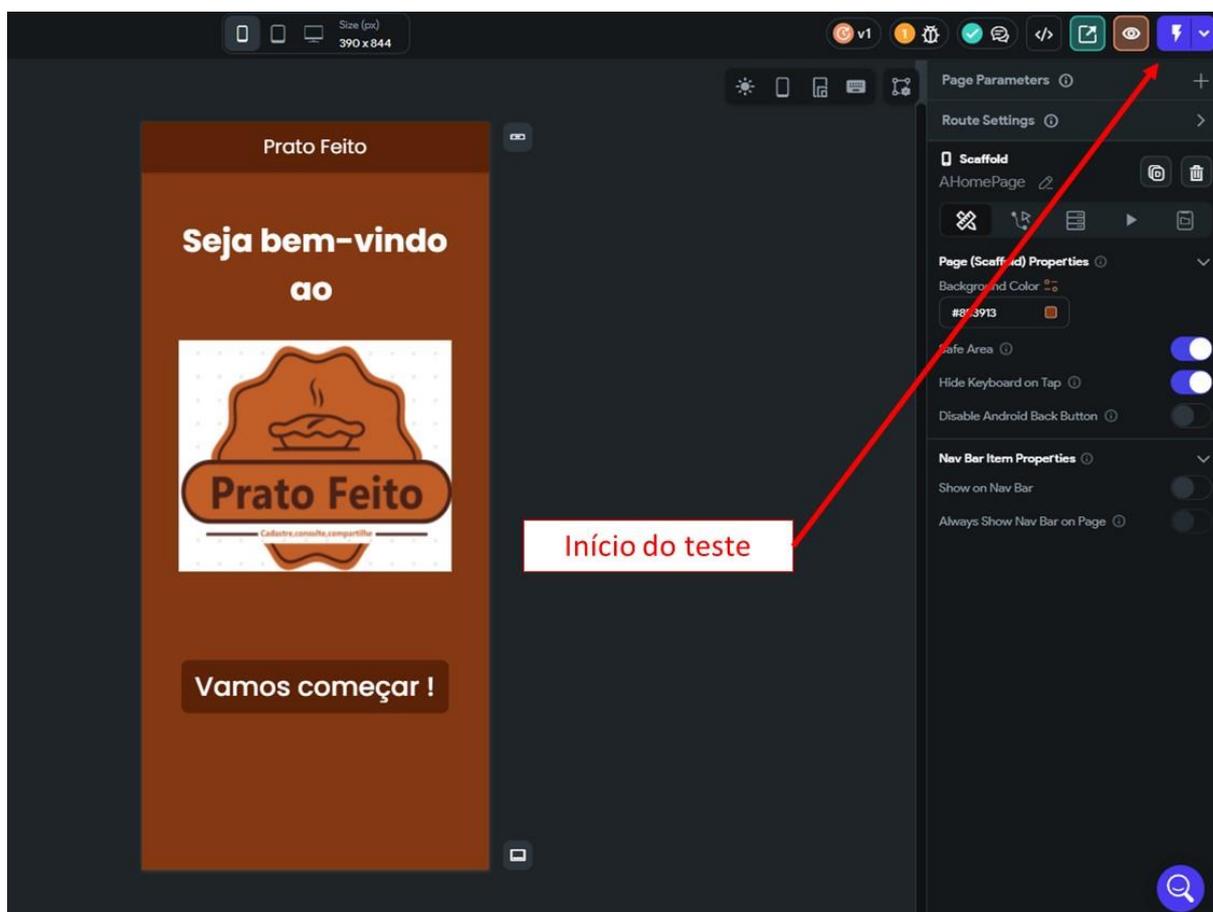


Fonte: autoria própria (2023)

4.7.3 Teste do programa

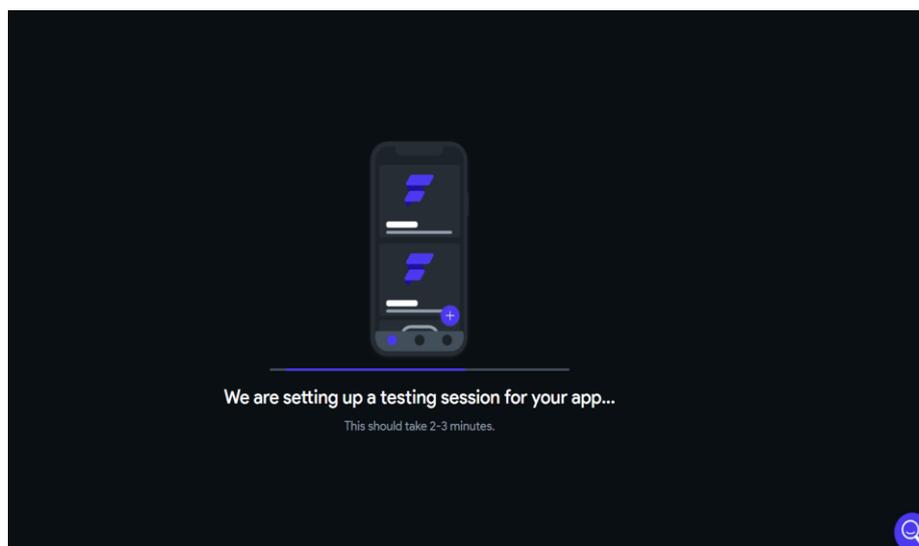
Para iniciar o teste, foi clicado no botão "run", com o ícone de um raio, conforme ilustra a figura 41. Em seguida, a plataforma carregou a tela de "carregamento do sistema" (Fig. 42). Assim que a tela "Home Page" (Fig. 43) carregou, o desenvolvedor clicou no botão "Vamos começar!", a fim de prosseguir para a tela "Menu Principal". Depois, o desenvolvedor preencheu os campos de texto com os dados de nome, e-mail e senha para cadastro e, em seguida, clicou em "cadastrar". Uma caixa de texto apareceu, confirmando que o cadastro foi realizado com sucesso, conforme ilustrado nas figuras 37 e 38, respectivamente.

Figura 41 – Acesso ao teste do Prato Feito



Fonte: autoria própria (2022)

Figura 42 – Carregamento do teste do aplicativo



Fonte: autoria própria (2023)

No próximo passo, o desenvolvedor clicou o botão “Vamos Começar!” (Fig. 43 “a”) que o levou para a tela “Menu principal”, depois clicou em no botão “Cadastrar/Editar” (Fg.43 “b”) e preencheu a caixa de texto com uma senha (Fg.43 “c”). Prontamente, o desenvolvedor acessou a tela de “Cadastro de Receita” (Fig. 44)

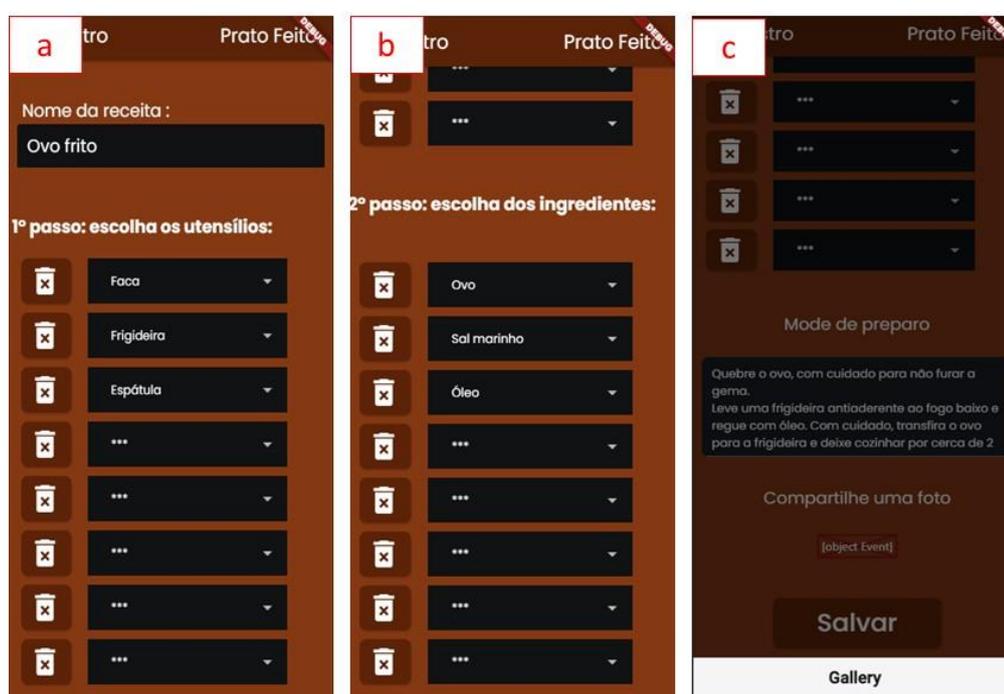
Figura 43 – A figura representa o caminho da tela “Homepage” “a” até ao *login* “c” para cadastro de receita.



Fonte: autoria própria (2023)

No próximo passo, o desenvolvedor digitou um nome de receita, sendo ela “Ovo frito” e selecionou três diferentes utensílios, sendo eles “Faca, Frigideira e Espátula” (Fig. 44 “a”). Após, o desenvolvedor selecionou três diferentes ingredientes, sendo eles “Ovo, sal marinho e óleo” (Fig. 44 “b”). Prontamente, foi digitado o modo de preparo da receita “Ovo frito” e clicado no botão “[object Event]” para acessar a galeria de imagens para a receita. Nota-se o surgimento na parte inferior da tela (Fig. 44 “c”) de um botão denominado “*Gallery*”, ou seja, galeria.

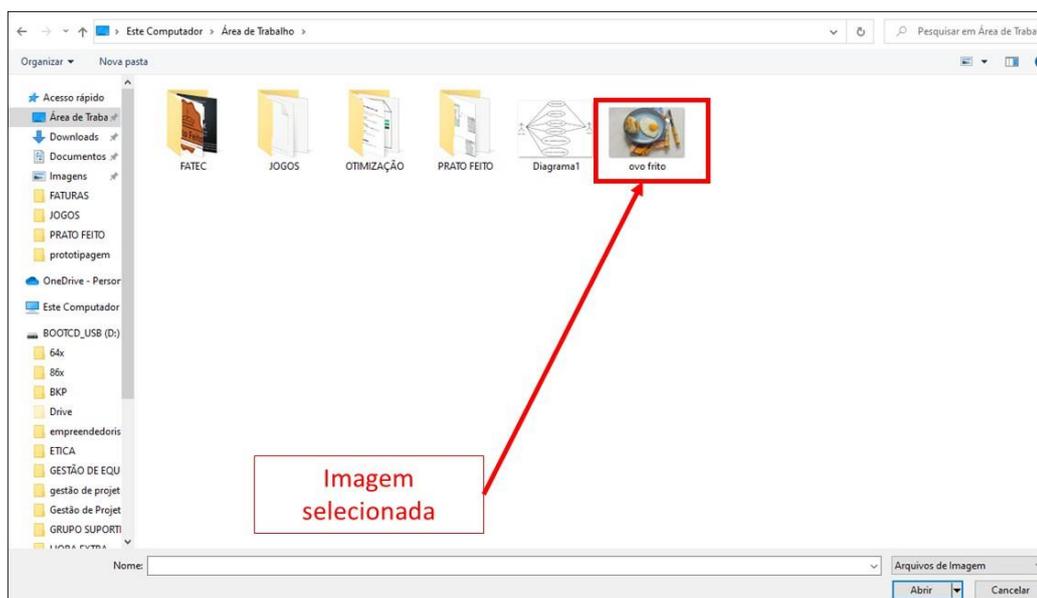
Figura 44 – A figura representa as etapas para o cadastro da receita.



Fonte: autoria própria (2023)

No passo seguinte, o desenvolvedor acessou a galeria de imagens do computador e selecionou uma imagem (Fig. 45). Logo depois a imagem foi carregada e o desenvolvedor clicou em salvar (Fig. 46 “a”). Depois, apareceu a mensagem “Salvamento, receita salva com sucesso!” (Fig. 46 “b”).

Figura 45 – Galeria de imagens do computador



Fonte: autoria própria (2023)

Figura 46 – A figura apresenta a escolha da imagem (“a”) e a mensagem da confirmação do registro da receita (“b”).



Fonte: autoria própria (2023)

No próximo passo, para demonstrar que a receita foi salva no aplicativo, o

desenvolvedor acessou a tela de “escolha de Pesquisa” (Fig. 47 “a”) e clicou no botão “Nome de receita” para acessar a tela de “Pesquisa por nome de receita” (Fig. 47 “b”). Ele digitou o nome da receita, “Ovo frito”, no campo correspondente e clicou no botão “Pesquisar” (Fig. 47 “b”). Logo se apresentou o resultado com o nome da receita, utensílios, modo de preparo e a imagem (Fig. 47 “c”, “d”, “e” e “f”).

Figura 47 – A figura apresenta o caminho para pesquisa de uma receita pelo nome

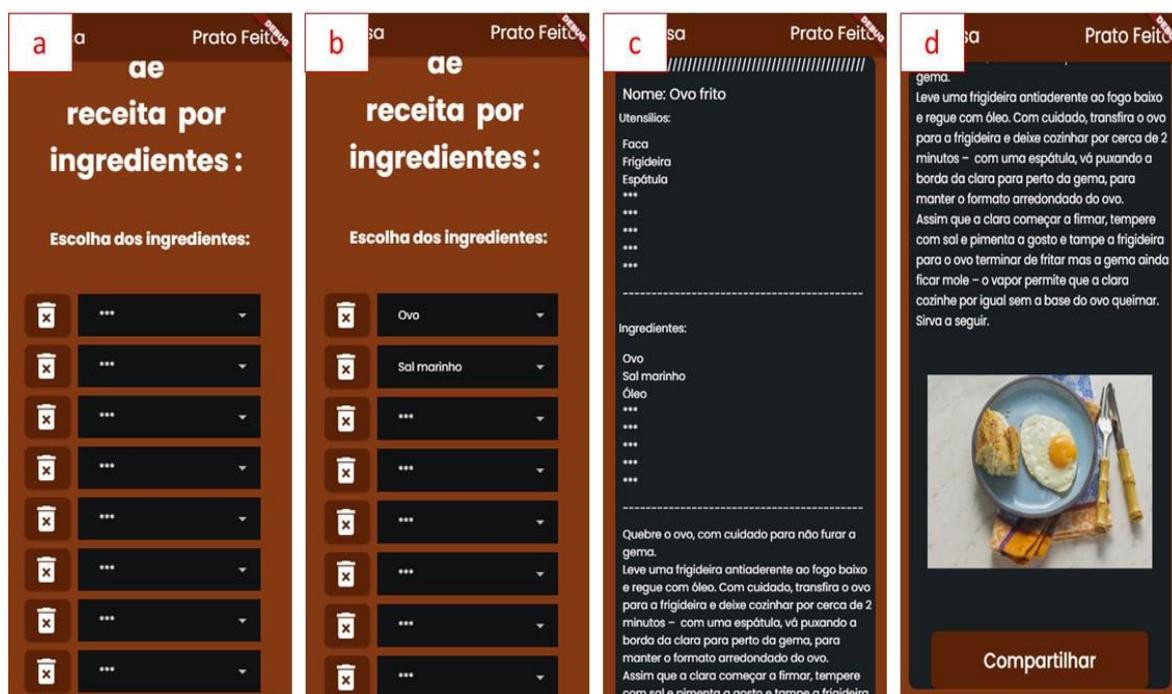


Fonte: autoria própria (2023)

No próximo passo, para demonstrar que a receita foi salva no aplicativo, o desenvolvedor acessou a tela de “escolha de Pesquisa” (Fig. 47 “a”) e clicou no botão

“Ingrediente” para acessar a tela de “Pesquisa por ingrediente” (Fig. 48 “a”). Ele selecionou os “*dropdowns*” com os ingredientes “Ovo” e “Sal marinho” (Fig. 48 “b”) e depois clicou no botão “Pesquisar”. Logo se apresentou o resultado com o nome da receita, utensílios, modo de preparo e a imagem (Fig. 48 “c” e “d”)

Figura 48 – A figura apresenta o caminho para pesquisa de uma receita por ingredientes

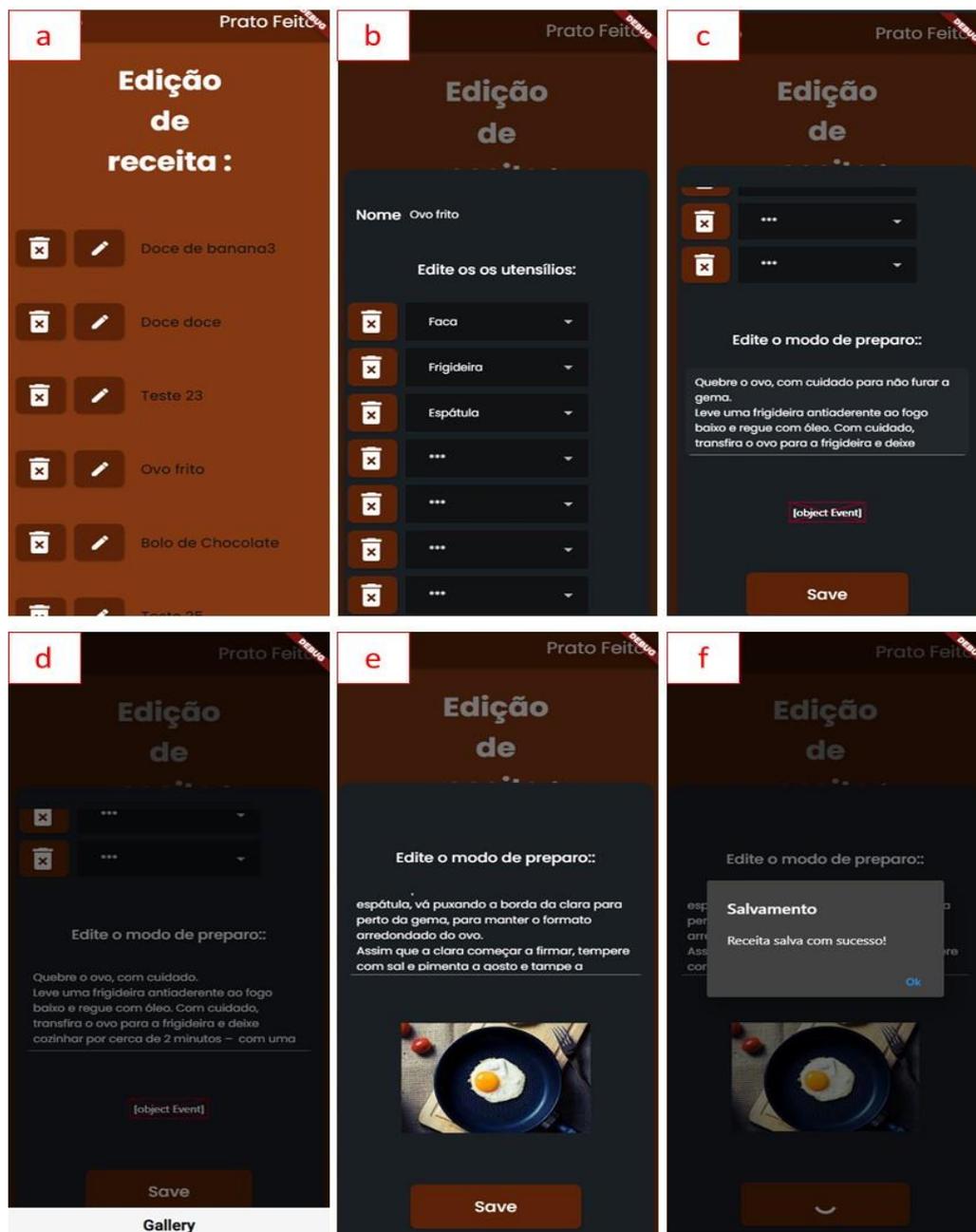


Fonte: autoria própria (2023)

No passo seguinte, para demonstrar que a receita pode ser editada, o desenvolvedor acessou a tela “*Menu principal*”, depois clicou em no botão “Cadastrar/Editar” (Fig. 43 “b”) e preencheu a caixa de texto com uma senha (Fig.43 “c”). Prontamente, o desenvolvedor acessou a tela de “Editar Receita” (Fig. 44), que o levou para tela de “Edição de Receita” (Fig. 49). Ele selecionou a receita “Ovo frito” clicou no botão de edição, simbolizado por um lápis (Fig. 49 “a”). Logo, surgiu o componente de parametrização com todos os “*dropdowns*” com os ingredientes, utensílios, modo de preparo (Fig. 49 “b” e “c”), porém sem a imagem (Fig. 49 “d”), pois a plataforma apenas permite quando a edição é realizada pelo celular. Então, foi editado o início da receita que constava a frase “Quebre o ovo, com cuidado para não furar a gema” por “Quebre o ovo, com cuidado” (Fig. 49 “d”). Em seguida, foi escolhida uma outra imagem para substituir a antiga (Fig. 49 “e”). Assim, foi clicado em “Save”

(salvar) (Fig. 49 “f”).

Figura 49 – A figura apresenta os passos para a edição de uma receita



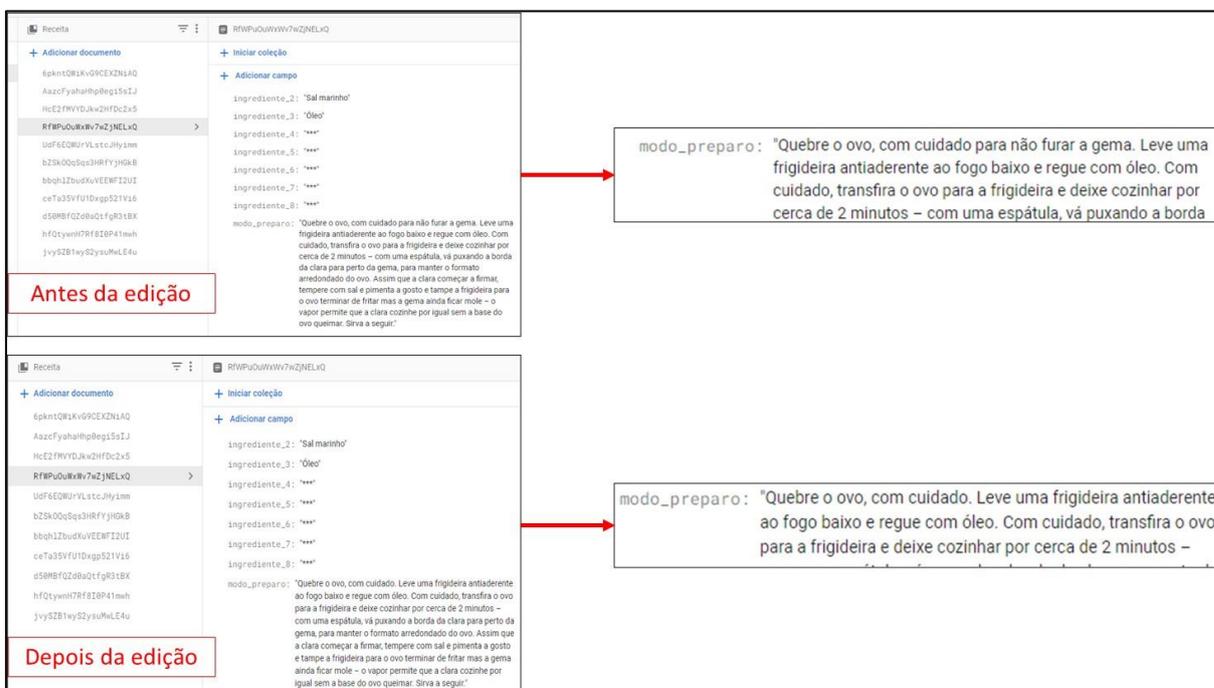
Fonte: autoria própria (2023)

Para demonstrar que a receita foi editada no aplicativo (Fig. 50) e no *Firebase* (Fig. 51), o desenvolvedor acessou a tela de "Pesquisa por nome" e digitou o nome da receita, "Ovo frito". Após clicar em pesquisar o aplicativo carregou a receita apresentando a alteração do texto do Modo de preparo e da imagem, conforme a figura a seguir. Nota-se que os dados foram alterados

Figura 50 – A figura apresenta a confirmação da alteração da receita no aplicativo



Fonte: autoria própria (2023)

Figura 51 – A figura apresenta a confirmação da alteração da receita no *Firebase*

Fonte: autoria própria (2023)

5 CONCLUSÕES

O presente estudo permitiu uma análise aprofundada da eficácia da programação *no-code* pelo *FlutterFlow*. Constatou-se que a maioria dos requisitos funcionais e não-funcionais do sistema foram atendidos com sucesso, demonstrando a eficiência desta tecnologia, sobretudo no registro de dados e na criação de novos itens nas interfaces do aplicativo.

Contudo, a análise identificou também pontos que demandam futuras intervenções e melhorias. Dentre os aspectos identificados, destacam-se:

- a) Habilitação de cadastro e edição de receitas por usuário: A implementação dessa funcionalidade possibilitará uma interação mais personalizada e dinâmica para os usuários, permitindo que contribuam ativamente com suas receitas e as modifiquem conforme suas necessidades.
- b) Associação das receitas cadastradas pelo usuário ao respectivo perfil, permitindo a edição tanto pelo próprio usuário quanto pelos administradores do sistema: Esta funcionalidade proporcionará uma experiência de usuário aprimorada, fortalecendo a relação entre os usuários e suas receitas, além de permitir um controle mais eficaz do conteúdo por parte dos administradores.
- c) Implementação de um sistema de classificação das receitas: A introdução dessa ferramenta será fundamental para proporcionar aos usuários uma visão rápida e clara da aceitação e popularidade das receitas na comunidade do aplicativo.

Essas diretrizes formam a base para as futuras etapas de desenvolvimento do projeto, cujo objetivo é enriquecer ainda mais a experiência do usuário e explorar mais profundamente as possibilidades da tecnologia *no-code*.

Importante destacar que, apesar das melhorias necessárias, a versão atual do aplicativo já apresenta significativo potencial. O sistema de cadastro e pesquisa de receitas baseado em ingredientes pré-selecionados demonstrou-se eficiente e prático, destacando-se pela capacidade de otimizar o tempo de preparo dos alimentos e promover a descoberta de novas combinações culinárias.

Portanto, conclui-se que o aplicativo desenvolvido possui forte potencial como ferramenta de suporte para entusiastas da culinária e indivíduos buscando soluções inovadoras para o preparo de alimentos no cotidiano. As melhorias planejadas para as próximas fases de desenvolvimento visam potencializar estas características, proporcionando uma experiência cada vez mais rica e intuitiva para os usuários.

REFERÊNCIAS

ALGOLIA. **O que é a ferramenta Algolia.** Disponível em: <https://resources.algolia.com/videos/video-whatiscalgolia>. Acesso em 20/04/2023

ANDRADE, Cássio; BELARMINO, João. **Pandemia aumenta procura por receitas diferentes na internet; aprenda como fazer brownie, o campeão em buscas.** Disponível em: <https://g1.globo.com/sp/mogi-das-cruzes-suzano/noticia/2020/06/24/pandemia-aumenta-procura-por-receitas-diferentes-na-internet-aprenda-como-fazer-brownie-o-campeao-em-buscas.ghtml>. Acesso em 20/09/2022

BARRA, Hugo. **Android: momentum, mobile and more at Google I/O.** Disponível em: <https://googleblog.blogspot.com/2011/05/android-momentum-mobile-and-more-at.html>. Acesso em 13/08/2022

BATISTA, Emerson O. **Sistemas de informação: o uso consciente da tecnologia para o gerenciamento.** Editora Saraiva, São Paulo, 2017

CARVALHO, Silvio. **Prototipagem e planejamento.** Disponível em: <https://kenzie.com.br/blog/prototipagem/>. Acesso em 15/09/2022

DAVE, Lee. **App Store claim on Apple anniversary.** Disponível em: <https://www.bbc.com/news/technology-23240971>. Acesso em 05/09/2022

FERREIRA, Giovanna Samires et. Al. **O Que Tem Pra Hoje – Sistema de Busca de Receitas com Base nos Ingredientes Disponíveis.** Trabalho de conclusão de curso, Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), Divinópolis, Brasil, 2017

FLUVELE. **Principais desafios de um chefe de cozinha.** Disponível em: <https://www.fruvele.com.br/desafios-chefe-de-cozinha/>. Acesso em 25/10/2022

GOOGLE TRENDS. Buscas por receitas culinárias. Disponível em: <https://trends.google.com.br/trends/explore?date=2020-01-03%202022-11-24&geo=BR&q=receitas%20culin%C3%A1rias>. Acesso em 01/08/2022

KART, Whitfield. **Mobiles platforms.** Disponível em: <http://www.portioresearch.com/en/blog/2013/fast-growth-of-apps-user-base-in-booming-asia-pacific-market.aspx>. Acesso em 29/09/2022

KRIGER, Daniel. **Prototipagem.** Disponível em: <https://kenzie.com.br/blog/prototipagem/>. Acesso em 15/09/2022

LAUDON, K.C.; LAUDON, J.P. **Sistemas de Informação Gerenciais.** São Paulo: Pearson Prentice Hall. p. 114, 2011

MARGARET, Rouse. **Low-code and no-code development platforms.** Disponível em: <https://www.techtarget.com/searchsoftwarequality/definition/low-code-no-code-development-platform>. Acesso em 20/09/2022

MATOS, Lucas Wolff. **O futuro da tecnologia é o no-code.** Disponível em: <https://www.inconnect.com.br/blog/Futuro-tecnologia-no-code-low-code/>. Acesso em 29/06/2022

MELENDEZ, Steven. Sometimes You're Just One Hop From Something Huge. Fast Company. Recode, San Diego, 2014.

MORAES, Vicente. **Sistemas de Banco de Dados.** Cognus, 6ª edição, São Paulo, Brasil, 2017

NERI, Marcelo. **Empobrecimento e queda na renda aumentam fila do Auxílio Brasil.** Disponível em: <https://economia.ig.com.br/2022-07-23/empobrecimento-e-queda-na-renda-aumentam-fila-do-auxilio-brasil.html>. Acesso em 28/09/2022.

POGUE, David. A Place to Put Your Apps. **New York Times.** Nova York, 2014

PORTO, Marcelo. **A evolução da tecnologia mobile e seu reflexo para a área de RH.** Disponível em: <https://blog.lg.com.br/evolucao-tecnologia-mobile-rh/>. Acesso em 19/08/2022

SATELL, Greg. **The Future of Software is No-Code.** Disponível em: <https://www.inc.com/greg-satell/how-no-code-platforms-are-disrupting-software.html>. Acesso em 01/11/2022

SIDDIQUE, F. Utilization of Smartphones to Access Health-related Information: A Descriptive Analysis. **Elsevier Technology**, Pasadena, 2013

SILOTTO, S. **FlutterFlow: Plataforma Low-Code para criação de Aplicativos Android e iOS**. Disponível em: <https://tekzoom.com.br/flutterflow-plataforma-low-code-para-criacao-de-aplicativos-android-e-ios/>. Acesso em 15/07/2022

SOPRANA, Paula. **45% dos brasileiros têm dificuldade para usar o serviço público digital**. Disponível em: <https://www1.folha.uol.com.br/mercado/2021/03/45-dos-brasileiros-tem-dificuldade-para-usar-servico-publico-digital.shtml>. Acesso em 30/09/2022.

SOUZA, Sandro Delage de. **Porque tantos profissionais de TI abandonam a carreira**. Disponível em: <https://pt.quora.com/Por-que-tantos-profissionais-de-TI-abandonam-a-carreira>. Acesso em 15/08/2022

SYOZI, Ricardo. **7 apps de receitas gratuitos**. Disponível em: <https://tecnoblog.net/responde/7-apps-de-receitas-gratuitos/>. Acesso em 29/06/2022

VALENTE, Marcos. **Principais desafios na cozinha**. Disponível em: <https://www.fruvele.com.br/desafios-chefe-de-cozinha/>. Acesso em 28/10/2022

WAGNER, Kurt; TOWNSEND, Tess. **Google has acquired most of Twitter's developer products, including Fabric and Crashlytics**. Recode, San Diego, 2018