

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“DR. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ARTHUR ALVES SILVA FARIA

**SISTEMA DE CONTROLE DE ESTOQUE E VENDAS PARA BARES E
MERCEARIAS**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

FRANCA/SP

2023

SISTEMA DE CONTROLE DE ESTOQUE E VENDAS PARA BARES E MERCEARIAS

Arthur Alves Silva Faria¹

Resumo

Os pequenos bares e mercearias que atendem na periferia das cidades, muitas vezes fundadas e administradas por uma mesma família, por décadas, e que se tornam conhecidas e famosas por oferecerem uma grande variedade de produtos e excelente qualidade aos clientes. É indispensável que esses pequenos comércios busquem uma gestão eficiente das suas atividades cotidianas, tais como o controle de estoque, vendas e relatórios. Como forma de resolver esse problema, emerge a necessidade do desenvolvimento de um Sistema de Gestão Integrada, e este é um dos objetivos deste Trabalho, qual seja, o desenvolvimento de uma solução sistêmica para este segmento comercial. Os outros objetivos são o de proporcionar uma maior eficiência, organização e qualidade nos serviços oferecidos através do aprimoramento dos processos administrativos. Os resultados conquistados com esta solução incluem a maior eficiência nos processos diários e um controle de estoque mais aprimorado. Esta gestão integrada ainda contribui na melhoria contínua do atendimento ao cliente, agregando valor ao negócio e gerando satisfação constante.

Palavras-chave: Comércio. Controle. Informação. Bares. Mercadoria. Sistema. Vendas.

Abstract

The small bars and grocery stores that serve the outskirts of cities, often founded and managed by the same family for decades, have become known and famous for offering a wide variety of products and excellent quality to customers. It is essential that these small businesses seek efficient management of their daily activities, such as inventory control, sales and reports. As a way to solve this problem, the need to develop an Integrated Management System emerges, and this is one of the objectives of this Work, that is, the development of a systemic solution for this commercial segment. The other objectives are to provide greater efficiency, organization and quality in the services offered through the improvement of administrative processes. The results achieved with this solution include greater efficiency in daily processes and improved inventory control. This integrated management also contributes to the continuous improvement of customer service, adding value to the business and generating constant satisfaction.

Keywords: Business. Control. Information. Bars. Grocery store. System. Sales.

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: [arthuralvessilvafaria07022003@gmail.com].

1 Introdução

Nas últimas décadas, os sistemas de informação têm ganhado cada vez mais espaço no mercado, pois a tecnologia vem evoluindo a cada dia que passa. Prova disto, foi o que ocorreu no período da pandemia da Covid-19, com muitas empresas aderindo o uso de soluções sistêmicas, principalmente os serviços de e-commerce. Mesmo com o fim da pandemia, muitas empresas continuaram aderindo aos sistemas de informação por causa dos benefícios que um sistema proporciona.

Entretanto, por mais que o uso de sistemas estejam em constante crescimento e dominando cada vez mais o mercado, algumas empresas ainda resistem à adoção do uso de soluções sistêmicas, principalmente àquelas geridas por uma família tradicional ou por um casal.

Podemos citar como motivo para essa resistência, a falta de conhecimento e prática no uso de ferramentas tecnológicas. Além disso, o preço considerado alto para manutenção e contratação de um software, também afasta os pequenos comerciantes.

O objetivo deste projeto é ajudar este pequeno comerciante na organização das vendas e, futuramente no gerenciamento de estoque. Neste momento, o foco essencial e a premissa, é ajudar o comerciante no monitoramento das vendas realizadas, para que essa informação seja usada na tomada de decisão. Afinal, é de grande importância que o comerciante busque uma gestão eficiente do seu negócio e apresente um diferencial, por ser um ramo de trabalho que apresenta uma vasta concorrência e também pelo fato de ser impactado diretamente pela inflação de diversos produtos.

As empresas costumam oferecer aos seus colaboradores os mais diversos tipos de sistemas como apoio e auxílio na execução das suas tarefas diárias. Realidade diferente em pequenos negócios, que geralmente tem apenas um ou nenhum sistema.

1.1 Termo de Abertura do Projeto (TAP)

O Termo de Abertura de Projeto (TAP) é um documento formal que tem como objetivo formalizar o início do projeto. Nele contém todas as informações detalhadas do projeto.

Geralmente, este documento é criado pelo gerente de projeto, e em seguida, ele deve ser aprovado pelos patrocinadores e clientes.

Objetivos deste documento: Autorizar o início do projeto, atribuir principais responsáveis e descrever de forma clara os requisitos iniciais, principais entregas, premissas e restrições do projeto. Neste documento será citado todos os processos para a realização do projeto e seus colaboradores para fins de esclarecimento.

Justificativa do projeto: O cliente deste projeto é o Bar & Merceria Elimar que foi fundado há mais de 22 anos. Neste período, não houve uma mudança na forma de trabalho nem definição concreta de função entre os colaboradores da empresa. Por esse motivo, atualmente a empresa sofre com a falta de informação a respeito das vendas. Não é possível saber se a empresa está tendo lucro ou prejuízo.

Proposta de solução: Como proposta de solução, este projeto visa a criação de um sistema que ajude a registrar as informações sobre os produtos e vendas do comércio e retornar, de maneira simples e de fácil entendimento, um relatório que contenha algumas informações sobre as vendas da empresa, com o objetivo de ajudar na tomada de decisão.

2 Viabilidade do Projeto

O método escolhido para avaliar a viabilidade do projeto foi o Business Model Canvas. Por meio dele, é possível identificar os principais elementos que constituem uma organização e como eles se relacionam entre si. A ferramenta ajuda a visualizar o modelo de negócio de uma empresa e em seu planejamento estratégico, mesmo em pequenos estabelecimentos ou empresas.

2.1 Canvas de Negócio (*Business Model Canvas* - BMC)

O Business Model Canvas foi criado pelo suíço Alex Osterwalder, por meio de suas pesquisas feitas na tese de doutorado.

Segundo Reis (2019), o Business Model Canvas (BMC) é um esquema de plano de negócios que pode ser apresentado em apenas um quadro, permitindo visualizar a correlação estratégica entre várias áreas de forma prática e resumida.

Tabela 01: Business Model Canvas (BMC)

Parcerias-Chave	Atividades-Chave	Oferta de Valor	Relacionamento	Segmentos de Clientes
- Fornecedores de Bebidas	- Produção de salgados	- Estabelecimento limpo e organizado	- Degustação	- Pessoas que moram no mesmo bairro

- Supermercados	- Reposição de estoque	- Ambiente agradável - Preços acessíveis	- Atendimento de qualidade	- Viajantes
	Recursos-Chave		Canais	
	- Copos, pratos e talheres - Funcionários		- Estabelecimento físico	
Estrutura de Custos		Fontes de Receita		
- Matéria prima - Contas a pagar - Pessoal		- Venda de comidas e bebidas - Venda de produtos de limpeza e utensílios		

Fonte: o autor

2.2 Matriz SWOT

A matriz SWOT é uma ferramenta de análise que engloba a análise de cenários para a tomada de decisão, a partir dos quatro fatores, sendo eles as Forças (Strengths), Fraquezas (Weaknesses), Oportunidades (Opportunities) e Ameaças (Threats).

Tabela 02: Matriz SWOT

Forças: Agilidade no atendimento; preços baixos; carisma dos funcionários.	Fraquezas: Desorganização; não padronização; falta de divulgação.
Oportunidades: Têm vários clientes; potencial para expandir.	Ameaças: Concorrência; inflação no preço dos produtos alimentícios.

Fonte: o autor

2.3 Plano de Ação: 5W2H

Segundo Pires (2019), a matriz 5W2H é um plano de ação para definir e executar atividades ou montar planos de ação. Essa ferramenta pode abranger desde projetos mais simples até os mais complexos. Geralmente, é muito utilizada também para registrar fluxos de trabalho em grandes empresas.

A matriz 5W2H é dividida em 7 etapas. Cada etapa da abordagem se refere a uma pergunta e resposta de cada uma dá espaço a próxima etapa.

Tabela 03: Matriz 5W2H

Ação	What? (O que?)	How (Como?)	Who? (Quem?)	When? (Quando?)	Where? (Onde?)	Why? (Por quê?)	How Much? (Quanto custa?)
Ação 1	Ajudar na tomada de decisões com relação a gerência de estoque	Criando um sistema que retorna informações referente as vendas dos produtos	Desenvolvedores e Gerentes	Aproximadamente 16 dias	Cidade: Franca Bairro: Recanto Elimar	* Falta de informações referentes a vendas dos produtos em estoque. * Produtos parados no estoque, possivelmente expirando sua data de validade e gerando prejuízo para o comercio.	Aproximadamente R\$ 2000
Ação 2	Ajudar a organizar as vendas feitas no estabelecimento	Implementando no sistema a funcionalidade de cadastro de clientes e histórico de compras	Desenvolvedores e Gerentes	Aproximadamente 16 dias	Cidade: Franca Bairro: Recanto Elimar	* O registro das vendas facilitará o gerenciamento do estoque. * Para facilitar na organização das contas a receber.	Aproximadamente R\$ 2000
Ação 3	Resgistrar os produtos da empresa	Implementando no sistema a funcionalidade de cadastro de produtos	Desenvolvedores e Gerentes	Aproximadamente 16 dias	Cidade: Franca Bairro: Recanto Elimar	* Funcionalidade necessária para controle de estoque e vendas	Aproximadamente R\$ 2000
Ação 4	Ajudar a organizar o estoque da empresa	Implementado no sistema a funcionalidade de controle de estoque	Desenvolvedores e Gerentes	Aproximadamente 16 dias	Cidade: Franca Bairro: Recanto Elimar	* Funcionalidade necessária para gerar relatórios de estoque dos produtos	Aproximadamente R\$ 2000

Fonte: o autor

3 Levantamento de Requisitos

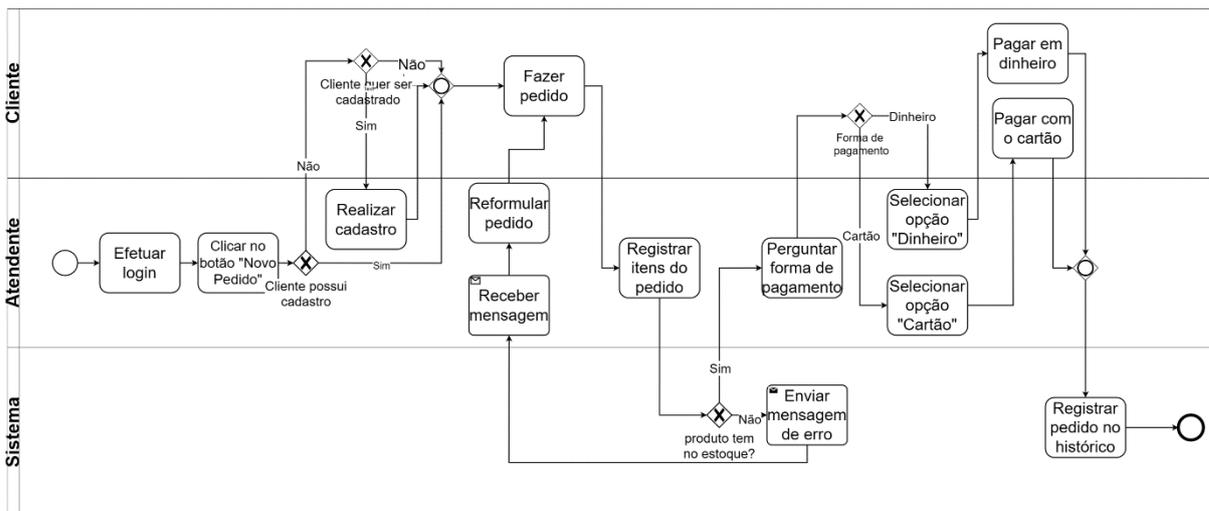
3.1 Elicitação e especificação dos Requisitos

O método utilizado para a elicitação de requisitos foi uma entrevista com o dono do estabelecimento e observar o ambiente de trabalho por um dia inteiro. O objetivo da entrevista e da observação do trabalho foi obter mais conhecimento sobre o funcionamento e a forma de trabalho, a fim de identificar possíveis problemas.

3.2 BPMN

Um diagrama de notação de modelagem para processos de negócios, também conhecido como BPMN, é uma ferramenta útil para exibir um processo de negócios interno. A meta do BPMN é apresentar uma representação visual de um procedimento de negócios específico para que seja compreensível a todas as partes interessadas.

Figura 01: BPMN



Fonte: o autor

3.3 Requisitos Funcionais

Os requisitos funcionais podem ser definidos como funções necessárias para as rotinas dos usuários em um sistema.

Segundo Canguçu (2021), dentro da engenharia de uma empresa de software pode-se destacar o requisito funcional, onde há a materialização de uma necessidade ou solicitação realizada por um software.

Tabela 04 – Requisitos Funcionais do sistema

RF001 - Cadastrar cliente	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve cadastrar novos clientes		
RF002 - Alterar cadastro de cliente	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve alterar cadastros existentes de clientes.		
RF003 - Cadastrar produto	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve cadastrar novos produtos.		
RF004 - Alterar cadastro de produto	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve ter a opção de alterar cadastros de produtos existentes.		
RF005 - Cadastrar de tipo de produto	Categoria: <input type="checkbox"/> Oculto	Prioridade: <input checked="" type="checkbox"/> Altíssima

	(X) Evidente	() Alta () Média () Baixa
Descrição: O sistema deve cadastrar novos tipos de produtos.		
RF006 - Alterar cadastro de tipo de produto	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve ter a opção de alterar novos tipos de produtos.		
RF007 - Cadastrar pedido	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve cadastrar novos pedidos.		
RF008 - Registrar histórico de pedido	Categoria: (X) Oculto () Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve registrar os pedidos corretamente no histórico.		
RF009 - Filtrar histórico de pedidos	Categoria: (X) Oculto () Evidente	Prioridade: () Altíssima (X) Alta () Média () Baixa
Descrição: O sistema deve filtrar os pedidos do mais recente ao mais antigo em questão de data e horário.		
RF010 - Gerar dashboard	Categoria: () Oculto (X) Evidente	Prioridade: () Altíssima (X) Alta () Média () Baixa
Descrição: O sistema deve gerar um dashboard com base nos registros de vendas.		
RF011 - Ativar tela de login	Categoria: (X) Oculto () Evidente	Prioridade: () Altíssima (X) Alta () Média () Baixa
Descrição: O sistema deve entrar na tela de login toda a vez que for iniciado.		
RF012 - Debitar estoque	Categoria: (X) Oculto () Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá descontar a quantidade de um determinado produto no estoque		

Fonte: o autor

3.4 Requisitos Não Funcionais

A descrição de Noledo (2022) a respeito dos requisitos não funcionais, aborda como algo que não interfere diretamente no desenvolvimento do sistema, mas que é um requisito muito importante para as funcionalidades externas, sendo necessário

para determinar o que será feito no software.

Tabela 05: Requisitos Não Funcionais do sistema

RNF001 - Bloquear acesso por meio da tela de login	O acesso ao sistema deverá ser bloqueado por meio de uma tela de login.	Tipo: Segurança	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF002 - Complexidade e de Senha Alta	Para maior segurança do sistema, a senha do usuário deverá ter alto nível de complexidade.	Tipo: Segurança	(X) Desejável () Obrigatório	(X) Permanente () Transitório
RNF003 - Tempo de resposta do sistema	O sistema deverá ter um de resposta de 0,1 segundos para cada ação.	Tipo: Desempenho	(X) Desejável () Obrigatório	(X) Permanente () Transitório
RNF004 - Compatibilidade	O sistema deverá ser compatível com todos os navegadores por ser um sistema web.	Tipo: Compatibilidade	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF005 - Desenvolvimento	O sistema ser desenvolvido na plataforma FlutterFlow.	Tipo: Compatibilidade	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF006 - Banco de dados usado	O sistema deverá ser desenvolvido no banco de dados não relacional Cloud Firestore devido a integração com a FlutterFlow.	Tipo: Compatibilidade	() Desejável (X) Obrigatório	(X) Permanente () Transitório

Fonte: o autor

3.5 Regras do Negócio

Segundo a jornalista Schemes (2020), as regras do negócio podem ser definidas como padrões que condicionam o funcionamento do software. Entretanto, elas não estão restringidas somente a esta definição. As regras do negócio também podem ser consideradas as diretrizes do negócio, ou seja, são partes fundamentais para a estratégia de um negócio.

Tabela 06: Regras do Negócio do sistema.

RN001 – Campos obrigatórios no cadastro de cliente
Descrição: Só serão aceitos os cadastros que estiverem com os campos “código” e “nome” preenchidos.
RN002 – Tabelas com todos os campos obrigatórios
Descrição: Todos os campos das tabelas produtos, tipos de produto, pedido e carrinho deverão ter preenchimento obrigatório.
RN003 – Impedir estoque negativo
Descrição: O sistema não deverá permitir que a quantidade de qualquer produto no estoque seja menor que 0.
RN004 – Acesso restrito

Descrição: O sistema não deve permitir o acesso ao sistema a contas usuários não cadastrados.
RN005 – Duplicação de cadastros
Descrição: O sistema não deverá permitir cadastro repetidos sob com base nas chaves primarias das tabelas.
RN006 – Exclusão de pedidos finalizados
Descrição: O sistema só poderá permitir a exclusão de pedidos finalizados por meio do usuário administrador.

Fonte: o autor

3.6 Casos de Uso

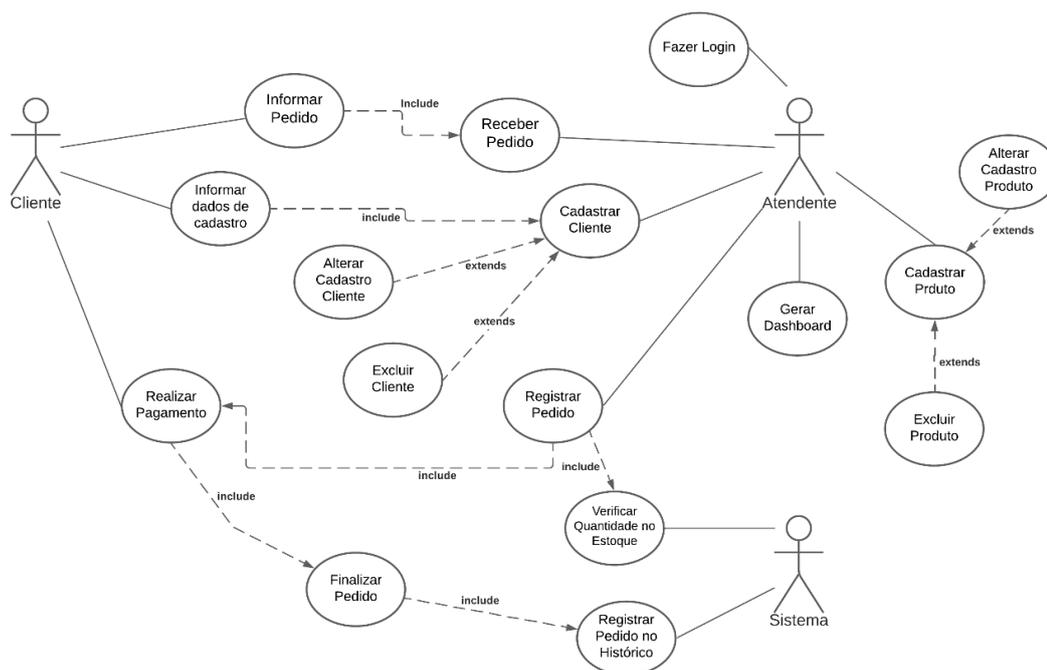
Na Linguagem de Modelagem Unificada (UML), um diagrama de caso de uso é uma representação que condensa as informações relativas aos usuários de um sistema, também chamados de atores, e suas interações com o sistema em questão.

Através de símbolos e conectores especializados, esse diagrama permite a visualização e discussão de cenários nos quais o sistema interage com pessoas, organizações ou sistemas externos.

Além disso, ele auxilia na compreensão das metas que o sistema ajuda os atores a alcançarem e na definição clara do escopo do sistema, fornecendo um meio eficaz de documentar e comunicar as funcionalidades e interações envolvidas no projeto.

Os diagramas de casos de uso são usados para descrever as interações entre um sistema e seus usuários. O diagrama modela o objetivo necessário para alcançar o sucesso no negócio.

Figura 2: Diagrama de Caso de Uso



Fonte: o autor

Sua documentação também é essencial para demonstrar o cenário no qual vai ser usado, juntamente com as regras em que o sistema deverá seguir com outras opções por meio da escolha do usuário.

Serão descritos os eventos necessários para efetuar as ações no cenário principal e alternativo, considerando pré-condições ou aquelas que forem posteriores a sua execução.

Novamente são destacados os atores, parâmetros a que devem ser fornecidos, restrições e validações.

Tabela 07: Use Case Cadastrar Usuários

Caso de Uso – Informar pedido	
ID	UC 001
Descrição	Este caso de uso tem como objetivo informar o que o cliente vai pedir
Ator Primário	Cliente
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicial quando o cliente chama o atendente 2. O cliente informa verbalmente ao atendente o seu pedido
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. O pedido pode ser realizado a qualquer momento
Caso de Uso – Receber pedido	
ID	UC 002
Descrição	Este caso de uso tem como objetivo receber o pedido informado pelo cliente
Ator Primário	Atendente

Pré-condição	UC 001 – Informar pedido
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso de inicia quando o cliente informa o pedido 2. O pedido é recebido verbalmente pelo atendente
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Um atendente deve estar disponível durante todo o período de funcionamento do estabelecimento
Caso de Uso – Fazer login	
ID	UC 003
Descrição	Este caso de uso permite que o usuário acesse o sistema por meio de uma tela de login
Ator Primário	Atendente
Pré-condição	Ter um usuário cadastrado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o usuário (Atendente) abre o sistema 2. Uma tela de login aparece 3. O usuário precisa preencher os campos “Nome de Usuário” e “Senha” corretamente para acessar o sistema
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Caso o usuário esqueça seus dados de login, ele deve informar ao responsável por fazer a manutenção do sistema
Caso de Uso – Informar dados do cadastro	
ID	UC 004
Descrição	Este caso de uso tem como objetivo informar os dados do cliente para realizar o seu cadastro
Ator Primário	Cliente
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando o cliente precisar de um cadastro no estabelecimento
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. O cliente já pode estar cadastrado 2. O cliente pode não ter o seu número de CPF em mãos
Caso de Uso – Cadastrar cliente	
ID	UC 005
Descrição	Este caso de uso tem como objetivo cadastrar um cliente novo
Ator Primário	Atendente
Pré-condição	UC 004 – Informar dados do cadastro
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o usuário clica em “Novo Cliente” 2. O usuário preenche os campos com as informações passadas pelo cliente
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Os campos “Endereço” e “Telefone” não precisam ser obrigatoriamente preenchidos
Caso de Uso – Alterar cadastro cliente	
ID	UC 006
Descrição	Este caso de uso tem como objetivo fazer alterações em cadastros de clientes existentes
Ator Primário	Atendente
Pré-condição	UC 005 – Cadastrar cliente

Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o usuário clica no ícone de “Editar Cadastro” que fica na frente do nome do cliente que se deseja alterar 2. O usuário realiza as alterações necessárias 3. O usuário clica no botão “CONFIRMAR” e em seguida clica em “SIM” para salvar as alterações
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Caso o usuário não queira salvar as alterações, ele deve clicar em “CANCELAR” para não salvar as alterações feitas
Caso de Uso – Excluir cliente	
ID	UC 007
Descrição	Este caso de uso tem como objetivo excluir um cadastro de um cliente existente
Ator Primário	Atendente
Pré-condição	UC 005 – Cadastrar cliente
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o usuário clica no ícone “EXCLUIR CADASTRO” 2. Um caixa de confirmação aparecerá na tela do usuário 3. Caso ele deseje realmente excluir o cadastro ele deverá clicar em “SIM”
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Case o usuário queira cancelar a ação de deletar, ele deverá clicar em “NÃO” na caixa de diálogo
Caso de Uso – Registrar pedido	
ID	UC 008
Descrição	Este caso de uso tem como objetivo registrar o pedido do cliente
Ator Primário	Atendente
Pré-condição	UC 005 – Cadastrar cliente; UC 009 – Cadastrar produto
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando o atendente recebe as informações sobre o pedido do cliente 2. O atendente preenche o formulário de registro de pedidos 3. O pedido é gravado no sistema 4. O pedido é encerrado quando o cliente efetua o pagamento
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. O cliente pode não estar cadastrado no sistema 2. O produto pedido pode ter acabado
Caso de Uso – Cadastrar produto	
ID	UC 009
Descrição	Este caso de uso tem como objetivo cadastrar um novo produto no sistema
Ator Primário	Atendente
Pré-condição	Nenhum
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o usuário clica em “Novo Produto” 2. O usuário preenche os campos com as informações passadas pelo cliente 3. O usuário clica em “CONFIRMAR” para salvar o cadastro
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Caso algum dos campos não esteja preenchido, o usuário não conseguirá salvar o cadastro

Caso de Uso – Alterar cadastro produto	
ID	UC 010
Descrição	Este caso de uso tem como objetivo fazer alterações em cadastros de clientes existentes
Ator Primário	Atendente
Pré-condição	UC 009 – Cadastrar produto
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o usuário clica no ícone de “Editar Cadastro” que fica na frente do nome do cliente que se deseja alterar 2. O usuário realiza as alterações necessárias 3. O usuário clica no botão “CONFIRMAR” e em seguida clica em “SIM” para salvar as alterações
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Caso o usuário não queira salvar as alterações, ele deve clicar em “CANCELAR” para não salvar as alterações feitas
Caso de Uso – Excluir produto	
ID	UC 011
Descrição	Este caso de uso tem como objetivo excluir o cadastro de um produto existente
Ator Primário	Atendente
Pré-condição	UC 005 – Cadastrar cliente
Cenário Principal	<ol style="list-style-type: none"> 2. O caso de uso se inicia quando o usuário clica no ícone “EXCLUIR CADASTRO” 3. Um caixa de confirmação aparecerá na tela do usuário 4. Caso ele deseje realmente excluir o cadastro ele deverá clicar em “SIM”
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 3. Case o usuário queira cancelar a ação de deletar, ele deverá clicar em “NÃO” na caixa de diálogo
Caso de Uso – Gerar dashboard	
ID	UC 012
Descrição	Este caso de uso tem como objetivo gerar um dashboard de vendas com base nas vendas registradas
Ator Primário	Atendente
Pré-condição	Nenhum
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando o usuário clica em “Gerar Relatórios” 2. O usuário escolhe “Dashboard de vendas”
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. O usuário pode não ter nenhuma venda registrada no sistema 2. Se a quantidade de vendas for baixa, as informações geradas pelo dashboard podem não ajudar muito
Caso de Uso – Verificar quantidade no estoque	
ID	UC 013
Descrição	Este caso de uso tem como objetivo verificar se o produto que foi pedido tem no estoque
Ator Primário	Atendente
Pré-condição	UC 008 – Registrar pedido
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando um pedido é registrado

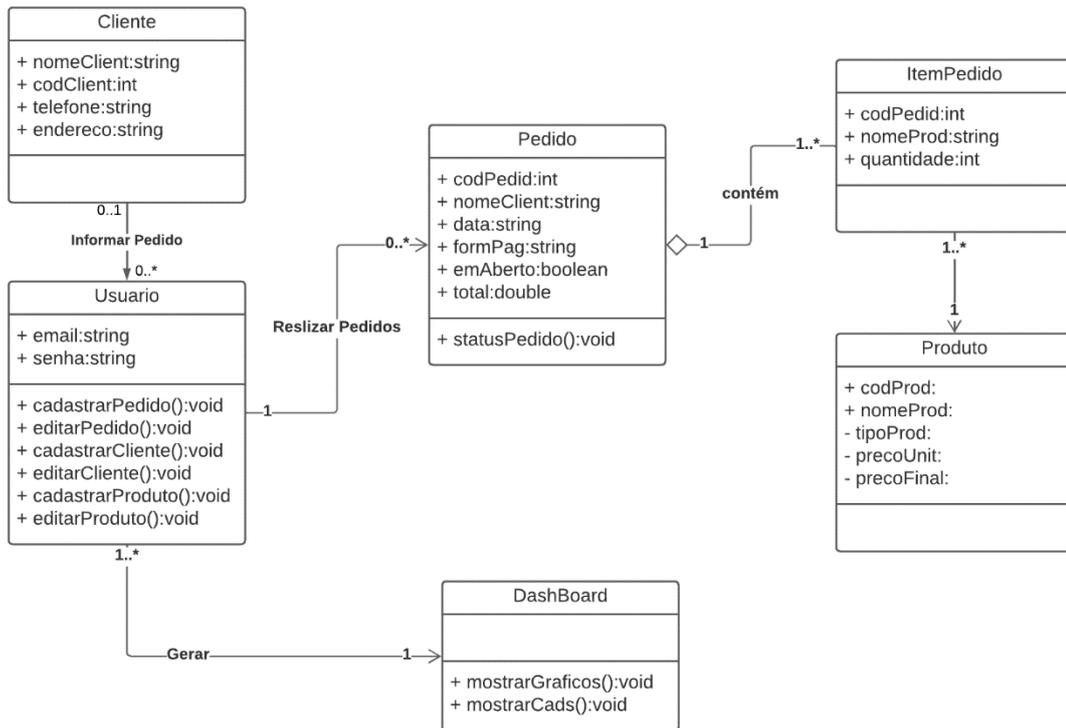
	<ol style="list-style-type: none"> 2. O sistema verifica a quantidade de cada item pedido e subtrai do sistema 3. Caso algum produto esteja negativo no estoque, o sistema retorna uma mensagem para o usuário "O produto x não tem quantidade suficiente no estoque"
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. A quantidade do produto pedido pode não ser suficiente.
Caso de Uso – Realizar pagamento	
ID	UC 014
Descrição	Este caso de uso tem como objetivo finalizar o pedido do cliente
Ator Primário	Atendente
Pré-condição	UC 008 – Registrar pedido
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando o cliente informa a forma de pagamento 2. O usuário seleciona a forma de pagamento
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. O pagamento pelo cartão pode ser recusado caso não haja a quantidade necessária para o pagamento
Caso de Uso – Finalizar pedido	
ID	UC 015
Descrição	Este caso de uso tem como objetivo finalizar o pedido do cliente
Ator Primário	Atendente
Pré-condição	UC 005 – Cadastrar cliente
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando o cliente realiza o pagamento 2. Quando o pagamento é confirmado o pedido é finalizado
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. O pagamento pelo cartão pode ser recusado caso não haja a quantidade necessária para o pagamento
Caso de Uso – Registrar pedido no histórico	
ID	UC 016
Descrição	Este caso de uso tem como objetivo registrar os pedidos feitos no histórico de vendas
Ator Primário	Atendente
Pré-condição	UC 015 – Finalizar pedido
Cenário Principal	<ol style="list-style-type: none"> 1. Este caso de uso se inicia quando o usuário finaliza algum pedido 2. O pedido então é armazenado no histórico de vendas na ordem do mais recente para o mais antigo
Pós-condição	Nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1. Pode ter vários pedidos que ainda não foram finalizados

Fonte: o autor

3.7 Diagrama de Classes

O diagrama de classes é usado para representar a estrutura estática de um sistema, representando as classes e objetos que os compõe, assim como a relação entre eles. Cada classe contém atributos e métodos com a finalidade de descrever o comportamento e as ações do sistema.

Figura 3: Diagrama de Classes

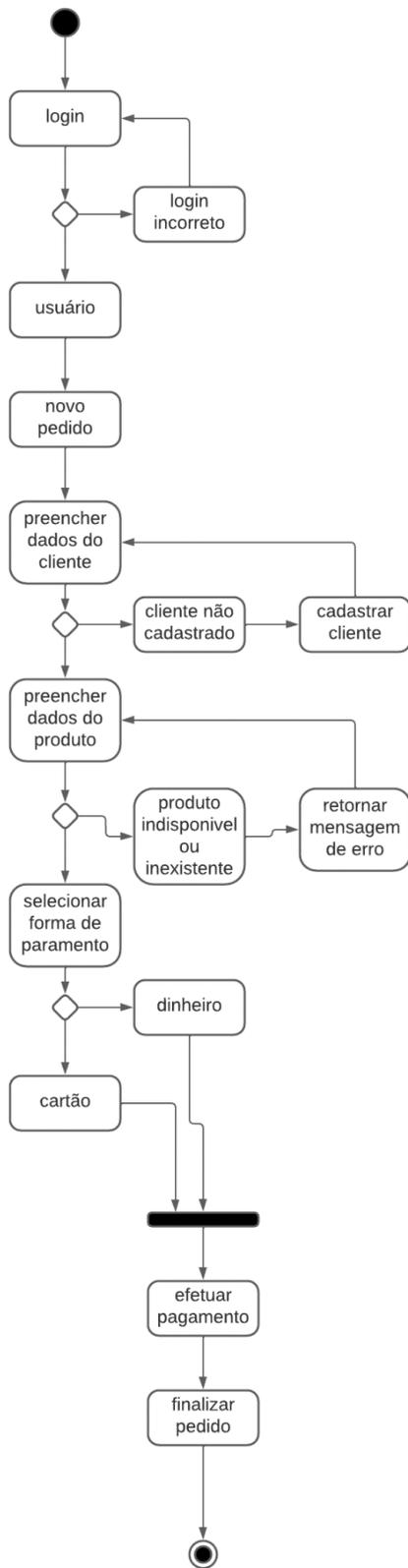


Fonte: o autor

3.8 Diagrama de Atividades

O diagrama de atividades representa as atividades de um sistema. Nele são capturadas as ações, seus resultados e como elas se relacionam dentro de uma instância de um objeto.

Figura 4: Diagrama de Atividades

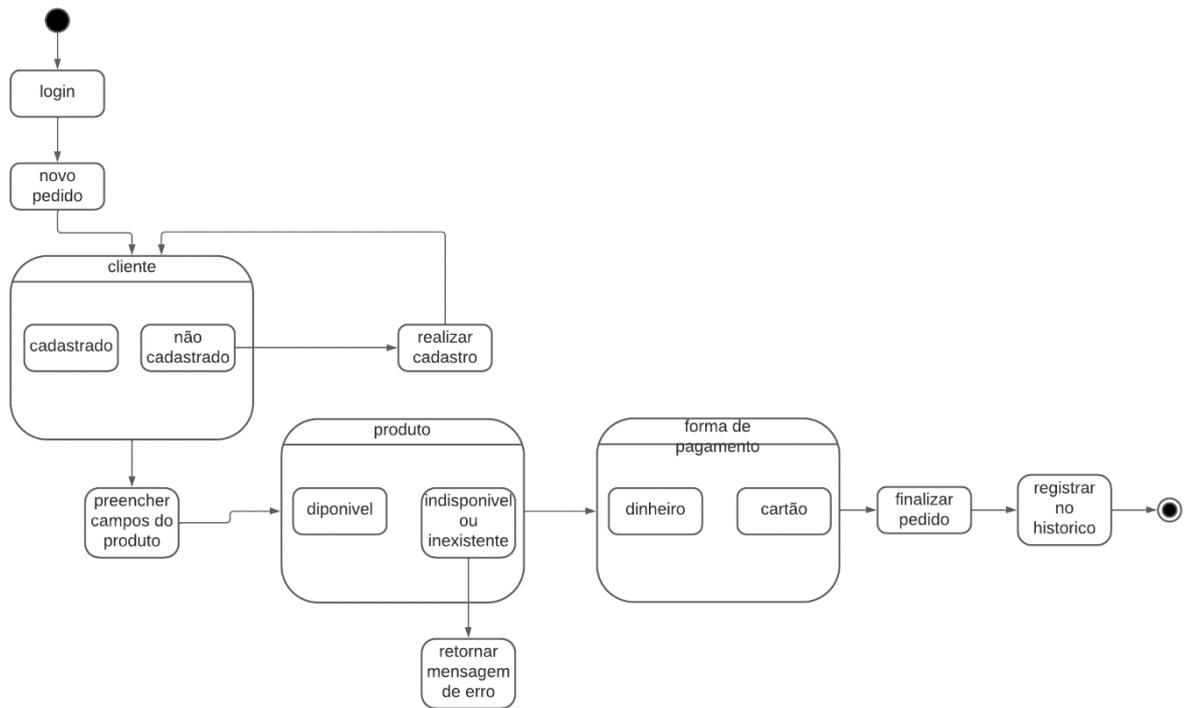


Fonte: o autor

3.9 Diagrama de Estados

O diagrama de estados é um complemento para a descrição das classes. Este diagrama mostra os estados de um objeto dentro levando em conta os eventos que podem causar cada mudança. Ele não é utilizado em todas as classes, apenas naquelas que apresentam mudança no comportamento devido aos diferentes estados.

Figura 5: Diagrama de Estados

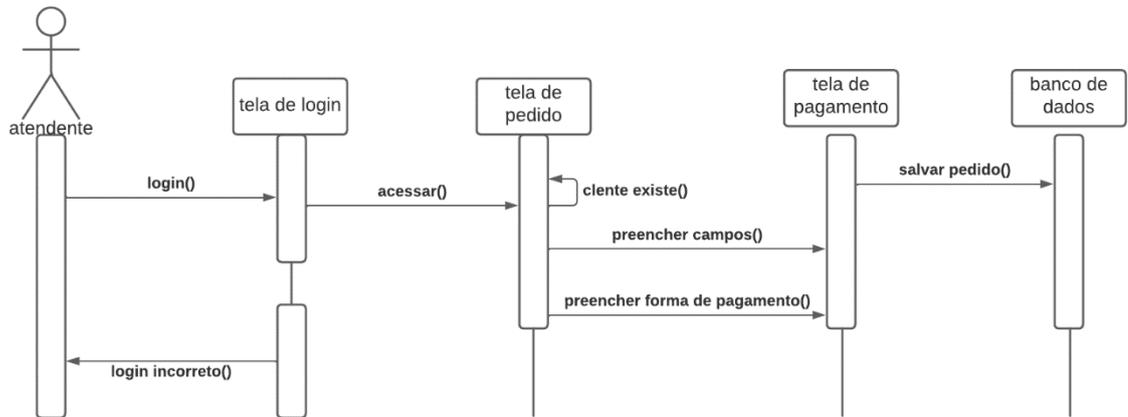


Fonte: o autor

3.10 Diagrama de Sequência

O diagrama de sequência mostra a interação dinâmica entre os objetos do sistema em uma sequência temporal. Ele também pode ser usado para descrever o fluxo de trabalho em um processo de negócios.

Figura 6: Diagrama de Sequência



Fonte: o autor

3.11 Matriz de Rastreabilidade

A matriz rastreabilidade é um documento que ajuda os requisitos para o teste e implementação da sua aplicação. Ela avalia e relaciona os diferentes componentes do sistema. Além disso, ela traz vários benefícios para as empresas como a facilidade na análise de impacto e auxílio nas alterações do produto.

Tabela 8: Matriz de rastreabilidade: Regras do negócio x Requisitos funcionais

	RF001	RF002	RF003	RF004	RF005	RF006	RF007	RF008	RF011
RN001	x	x							
RN002			x	x	x	x	x		
RN003								x	
RN004									x
RN005	x	x	x	x	x	x	x		
RN006							x		

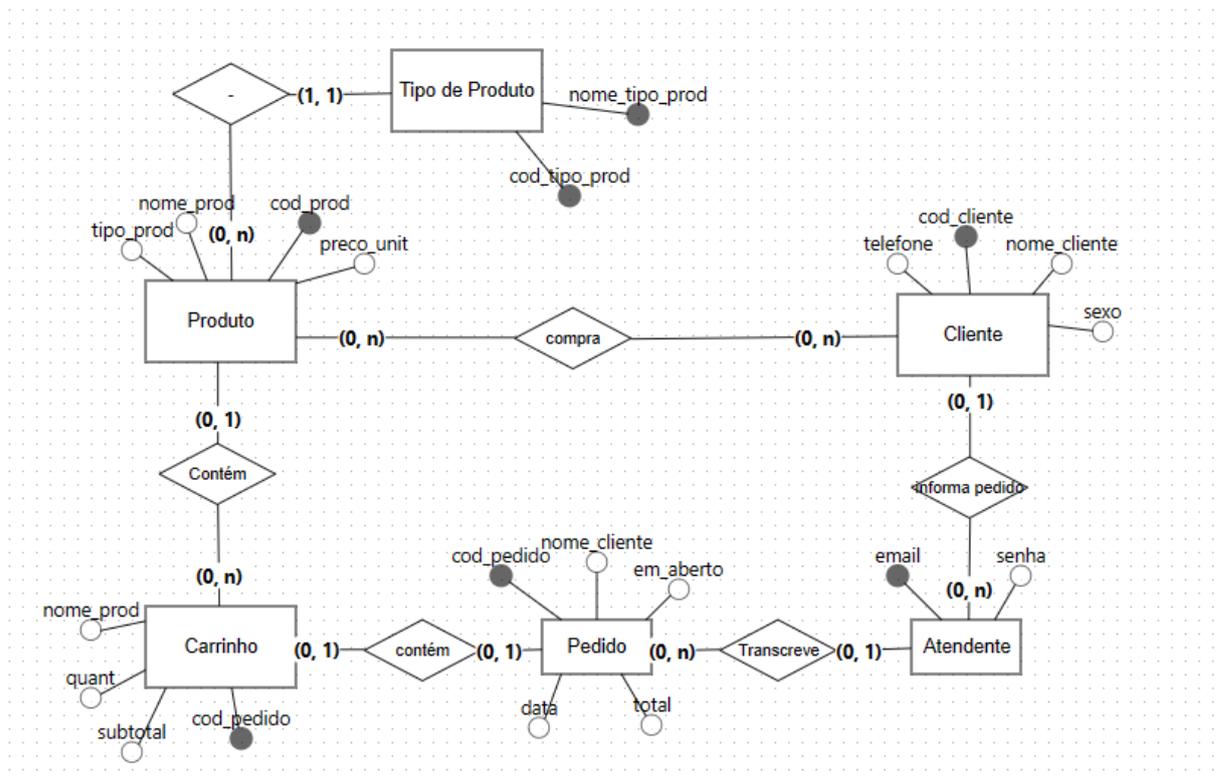
Fonte: o autor

3.12 Diagrama Entidade-Relacionamento

O diagrama de entidade relacionamento (DER) é um tipo de fluxograma que representa de forma gráfica o modelo entidade relacionamento. Seu objetivo é trazer para a realidade informações pertinentes de forma mais visual e intuitiva.

O DER é constituído de entidades, atributos e relacionamentos. As entidades são objetos de característica únicas. Os atributos são as características da entidade e os relacionamentos são as associações entre as entidades.

Figura 7: Diagrama Entidade-Relacionamento



Fonte: o autor

4 Ferramentas e Métodos

As ferramentas utilizadas no projeto foram escolhidas por serem intuitivas, de fácil acesso e pela qualidade em seu resultado. Todas são de acesso gratuito, e estão disponíveis de forma segura na internet.

4.1 - Frontend: FlutterFlow

Segundo o site oficial da ferramenta (FLUTTERFLOW, 2023), é uma plataforma low-code que utiliza o framework Flutter para a criação de aplicativos móveis nativos. A criação de interfaces funciona a partir de widgets de arrastar e soltar. Ela tem a opção de trabalhar com funções, APIs e Media Assesst. Além disso, também tem integração com os serviços da FireBase o que facilita na integração com o beckend.

A ferramenta foi escolhida pois possibilita o desenvolvimento de aplicações mobile, além de apresentar uma facilidade no desenvolvimento de aplicações e por ser muito intuitiva. A FlutterFlow tem vários recursos gratuitos e uma documentação detalhada com vários tutoriais e recursos em seu site oficial e comunidade. O fato de ser

uma plataforma específica para desenvolvimento mobile, permite futuramente, a expansão do projeto para as plataformas móveis.

4.1.1 Framework Flutter

O Flutter é um framework de desenvolvimento de aplicativos móveis criado pela Google e que é utilizado pela ferramenta FlutterFlow. Foi lançado em maio de 2017. Desde então, o Flutter tem sido amplamente adotado por desenvolvedores em todo o mundo, graças à sua velocidade de desenvolvimento e interface simples de arrastar e soltar (FLUTTERFLOW, 2023)

O framework Flutter é construído com a linguagem de programação Dart e é conhecido por sua compilação rápida, que permite aos desenvolvedores criar aplicativos rapidamente e com eficiência. O Flutter também fornece uma interface simples de arrastar e soltar que torna fácil a criação de aplicativos sem ter que escrever muito código (FLUTTERFLOW, 2023).

4.1.2 Dart

Dart é uma linguagem de programação de código aberto criada pela Google. Ela foi lançada em outubro de 2011. O Dart foi projetado para ser uma linguagem de programação moderna, com recursos como compilação rápida e análise estática. A linguagem é orientada a objetos, com sintaxe semelhante à do Java e do C++. (DART PROGRAMMING, 2023)

O Dart é usado principalmente para o desenvolvimento de aplicativos móveis, especialmente com o framework Flutter. Ele também pode ser usado para criar aplicativos da web, servidores e outros tipos de software. O Dart é conhecido por sua produtividade, graças à sua compilação rápida e ferramentas de análise estática. (DART PROGRAMMING, 2023)

4.2 Backend: Firebase

Firebase é uma plataforma de desenvolvimento de aplicativos móveis e web criada pelo Google. Ela ajuda os desenvolvedores a criarem aplicativos e jogos que os usuários adoram, fornecendo uma ampla gama de ferramentas e recursos. O Firebase é usado por milhões de empresas em todo o mundo e é conhecido por sua confiabilidade e escalabilidade (FIREBASE, 2023)

As soluções oferecidas pelo Firebase têm como objetivo acompanhar todo o

processo de desenvolvimento. Na criação de um sistema o usuário pode utilizar o banco de dados Cloud Firestore, o Authentication para autenticação de usuário e dentre outras extensões presentes na plataforma. O Firebase também permite monitorar as falhas do seu sistema pelo Crashlytics, a performance com o Performance Monitoring e gerar relatórios com o Google Analytics.

Além disso, é possível aumentar o engajamento do seu software através de um controle detalhado sobre o comportamento de e aparência do aplicativo pelo Remote Config, o envio de notificações com o Firebase Cloud Messaging e, por fim, realizar testes com o FireBase A/B Testing que está na fase beta de desenvolvimento. (FIREBASE, 2023)

4.2.1 Autenticação: FireBase Authentication

Firestore Authentication é um serviço de autenticação de usuários para aplicativos móveis e web. Ele fornece serviços de back-end, SDKs fáceis de usar e bibliotecas de interface prontas para autenticar usuários no seu aplicativo.

O Firestore Authentication oferece suporte à autenticação usando senhas, números de telefone, provedores de identidade federados conhecidos, como Google, Facebook e Twitter, entre outros.

Para conectar um usuário ao seu aplicativo, primeiro você precisa ter as credenciais de autenticação do usuário. Essas credenciais podem ser o endereço de e-mail e a senha do usuário ou um token do OAuth. O Firestore Authentication tem integração com outros serviços do Firebase e aproveita os padrões do setor, como OAuth 2.0 e OpenID Connect, para que possa ser facilmente integrado ao seu back-end personalizado.

Ao fazer upgrade para o Firestore Authentication com o Identity Platform, você consegue mais recursos, como autenticação multifator, funções de bloqueio, registros de atividades dos usuários e de auditoria, suporte a SAML e a OpenID Connect genérico, multilocação e suporte de nível empresarial. (FIREBASE, 2023)

A ferramenta FireBase Authentication foi utilizada, pois, além de possuir vários recursos gratuitos, tem integração direta com a FlutterFlow. Com isso, criar a autenticação de usuário se tornou mais eficiente e precisa. Além de ter vários recursos e opções de autenticação.

4.2.2 Banco de dados: Cloud Firestore

Cloud Firestore é um banco de dados NoSQL hospedado na nuvem que os aplicativos móveis e da web podem acessar diretamente por meio de SDKs nativos. Ele é um dos produtos do Firebase, uma plataforma de desenvolvimento de aplicativos móveis e da web que ajuda a criar e expandir aplicativos e jogos.

O Cloud Firestore é um banco de dados flexível e escalável para desenvolvimento móvel, da web e do servidor, construído pela Firebase e Google Cloud. Ele armazena e sincroniza dados para o desenvolvimento do lado do cliente e do servidor. O Cloud Firestore também oferece suporte à integração com outros produtos Firebase, incluindo funções em nuvem.

O modelo de dados NoSQL do Cloud Firestore suporta estruturas de dados hierárquicas flexíveis. Os dados são armazenados em documentos organizados em coleções. Os documentos podem conter objetos aninhados complexos, além de subcoleções.

As consultas no Cloud Firestore podem ser usadas para recuperar documentos individuais específicos ou para recuperar todos os documentos em uma coleção que correspondam aos seus parâmetros de consulta. As consultas podem incluir vários filtros encadeados e combinar filtragem e classificação. Eles também são indexados por padrão, portanto, o desempenho da consulta é proporcional ao tamanho do conjunto de resultados, não ao conjunto de dados.

O Cloud Firestore é projetado para lidar com as cargas de trabalho mais difíceis dos maiores aplicativos do mundo. Ele oferece replicação automática de dados multirregião, garantias de consistência forte, operações em lote atômicas e suporte a transações reais (FIRESTORE, 2023)

Este banco de dados foi escolhido por também ter integração direta com a FlutterFlow e facilitar o desenvolvimento. Porém, mesmo sendo um banco de dados NoSQL, tem uma documentação detalhada e traduzida, o que facilita na construção. O fato de ser um banco de dados em nuvem, também ajuda na economia de recursos para infraestrutura para os pequenos bares e comércios.

4.3 Diagramas de UML: Lucidchart

Lucidchart é uma ferramenta online que permite criar diagramas profissionais de forma fácil e colaborativa. Na Lucidchart, é possível criar fluxogramas, mapas mentais, organogramas, wireframes, mockups, diagramas UML e entre outros.

Ela oferece uma interface intuitiva, recursos avançados e integrações com plataformas de produtividade muito usadas, como Google Workspace, Microsoft Office, Slack e Atlassian. A Lucidchart também ajuda na comunicação e visualização de ideias de forma clara e eficiente (LUCIDCHART, 2023)

Essa ferramenta foi utilizada também pela sua imensa biblioteca, o que facilitou no desenvolvimento dos diagramas de UML. Também é uma ferramenta com interface intuitiva e com a possibilidade de construir até três projetos por conta, de forma gratuita, e com acesso a todas as bibliotecas de elementos.

4.4 BPMN – draw.io

Draw.io é uma ferramenta de diagramação gratuita e de código aberto que pode ser usada online em app.diagrams.net ou offline com o draw.io desktop. Como uma ferramenta de diagramação, consistente e segura para equipes, ela fornece a funcionalidade de diagramação e você escolhe onde manter os dados do seu diagrama.

Com o draw.io, é possível criar diagramas de fluxo, diagramas de rede, diagramas UML, diagramas ER, diagramas BPMN e muito mais. Essa ferramenta foi escolhida por ser uma solução documentar, analisar e melhorar os processos de negócio de forma ágil e eficiente (DRAW.IO, 2023)

4.5 Artefatos relevantes: Matriz SWOT, 5W2H, Canvas, Matriz de rastreabilidade

A ferramenta Excel é um programa de planilha eletrônica desenvolvido pela Microsoft. Ela permite criar, editar e manipular dados numéricos e alfanuméricos em tabelas, gráficos, fórmulas e funções. O Excel é amplamente utilizado em diversas áreas, como finanças, contabilidade, estatística, engenharia, educação e administração. Com o Excel, é possível realizar cálculos complexos, análises de dados, simulações, projeções, relatórios e dashboards de forma rápida e eficiente. (MEYER, 2018)

A ferramenta Excel foi utilizada pela estrutura de células facilitar a construção dos artefatos com formato de tabela como a SWOT, 5W2H, Canvas e Matriz rastreabilidade.

4.6 Diagrama de Entidade Relacionamento – Br Modelo Web

A ferramenta BR Modelo Web é online e gratuita para modelagem de banco de dados. Com ela, é possível criar e armazenar modelos de banco de dados na nuvem, além de acessá-los em qualquer dispositivo.

A ferramenta oferece recursos como criação de modelos conceituais, lógicos e físicos, templates de base de dados, diagrama de entidade e relacionamento, entre outros. Também foi escolhida por ser uma ferramenta gratuita e intuitiva (BR MODELO, 2023)

5 Desenvolvimento

O desenvolvimento na FlutterFlow, por ser uma ferramenta low-code, não foi feito a partir da construção de códigos. O Low-code é uma abordagem de desenvolvimento de software que usa codificação mínima para criar sites, aplicativos e processos. Em vez de escrever linhas de código complexo por meio de programação de computação tradicional, o desenvolvimento low-code usa modeladores visuais do tipo “arrastar e soltar” e criação de interface do tipo “apontar e clicar” para criar aplicativos completos de forma rápida e eficiente. A FlutterFlow disponibiliza vários componentes e ações já prontas. Nela é necessário apenas que o desenvolvedor escolha um componente e arraste-o para tela de design e lá ele pode editar esse componente e atribuir uma ação.

Todos os componentes podem ser editados e programados dependendo da sua função por meio de seus menus de configuração. No menu “Properties” podemos alterar a parte visual e as propriedades daquele componente.

No menu “Actions” podemos atribuir ações ao componente e qual será o gatilho para iniciar essas ações. No menu “Backend Query” podemos fazer uma chamada de dados em uma tabela do banco de dados ou por uma chamada de API.

5.1 Componentes utilizados

Abaixo serão apresentados todos os componentes da FlutterFlow que foram utilizados para a criação da aplicação.

5.1.1 Column

O componente Column é usado para criar um layout vertical. Dentro do componente existem os eixos principal e o cruzado, ambos são duas propriedades importantes da coluna. Para uma coluna, o eixo principal é executado verticalmente e o eixo cruzado é executado horizontalmente. Por padrão, todos os componentes dentro da coluna serão organizados de forma vertical.

5.1.2 Row

O componente Row é usado para criar um layout horizontal. Assim como o componente Column, ele também tem um eixo principal e um eixo cruzado. Porém, todos serão os componentes dentro da linha serão organizados de forma horizontal.

5.1.3 Container

O componente Container é como uma caixa que permite adicionar uma cor de fundo, cantos arredondados ou definir algumas restrições de tamanho para um componente filho.

O Container pode ser utilizado para ajudar a criar sua estrutura de interface do usuário. Os contêineres podem ser úteis para dividir a tela em partes lógicas menores.

5.1.4 Image

O componente Image é usado para mostrar uma imagem dos ativos carregados ou o link da URL. O widget Imagem pode exibir os diferentes tipos de imagens, como JPEG, PNG, GIF, GIF animado, WebP, WebP animado, BMP e WBMP.

5.1.5 Text

O componente Text é usado para mostrar texto na tela.

5.1.6 Icon

O componente Icon é usado para mostrar uma pequena imagem ou um símbolo. Este componente pode ser usado para permitir que os usuários entendam rapidamente o significado do texto.

5.1.7 Textfield

O componente TextField permite que os usuários insiram texto, números e símbolos em seu aplicativo. O componente pode ser usado para criar formulários, enviar mensagens, diálogos, pesquisar e entre outras funcionalidades.

5.1.8 ListView

O componente ListView é extremamente útil para exibir uma lista de itens em um arranjo linear (vertical ou horizontal). ListView é uma versão avançada do componente Coluna que fornece rolagem automaticamente quando a lista de itens não cabe

na tela.

ListView fornece uma maneira de exibir eficientemente uma lista de dados de uma maneira visualmente atraente e prática para gerenciar grandes quantidades de dados. Ele pode suportar dados estáticos e dinâmicos.

5.1.9 VerticalDivider

O componente Divisor Vertical é uma linha vertical usada para separar visualmente os widgets. Por exemplo, pode ser usado ao exibir uma lista de itens que têm várias colunas de informações ou para separar uma barra de navegação do conteúdo de uma página e etc.

5.1.10 IconButton

O componente IconButton permite que os usuários executem ações com um único toque. Ao contrário do widget Button normal, este widget mostra uma pequena imagem ou um símbolo que representa a ação a ser executada quando o botão é clicado.

O componente pode ser usado para permitir que os usuários entendam rapidamente o significado de uma ação.

5.1.11 DropDown

O componente DropDown é usado para permitir que um usuário selecione em uma lista de itens. O desenvolvedor precisa fornecer a ele uma lista de itens a serem exibidos como opções e um valor inicial que represente o item selecionado no momento. Quando um usuário seleciona um item na lista suspensa, o valor é atualizado para refletir o item selecionado.

O componente pode ser utilizado em qualquer situação em que deseja que os usuários selecionem entre um conjunto de opções, como selecionar um país, escolher um idioma ou escolher uma cor.

5.1.12 RadioButton

O componente RadioButton é usado para permitir que um usuário selecione uma opção de várias seleções. Pode ser usado para implementar uma única seleção, como seleção de gênero, preferências de notificação, etc.

5.1.13 CountController

O componente CountController é usado para incrementar e diminuir a contagem ou o número. Pode ser usado para definir a quantidade de qualquer produto ao comprar.

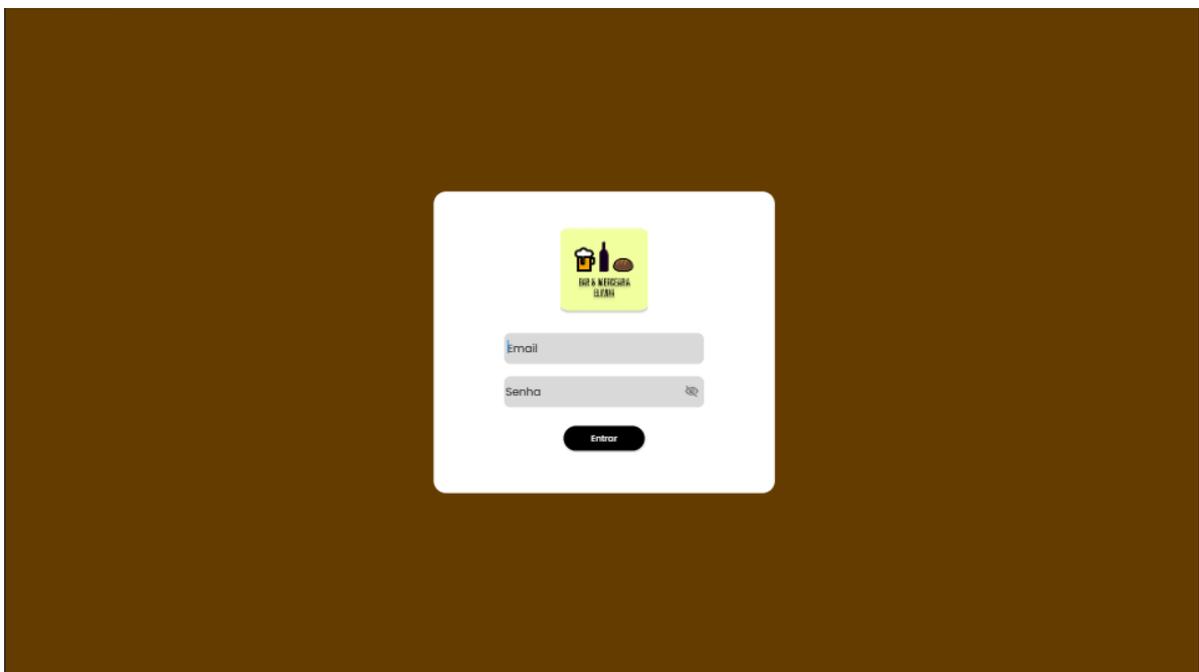
5.2 Telas do sistema

Abaixo estão documentadas todas as telas funcionais do sistema.

5.2.1 Tela de login

A tela de login tem como objetivo prover uma autenticação de usuário para fins de segurança. Essa tela será acessada toda vez que o usuário iniciar a aplicação e terá os campos de e-mail e senha para preencher.

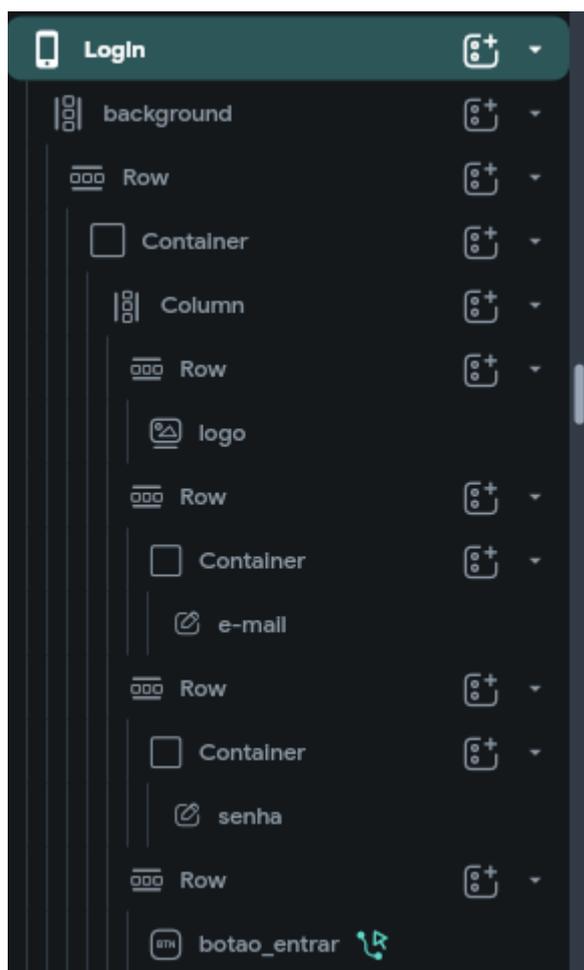
Figura 08: Tela de login



Fonte: o autor

Apresentamos abaixo a estrutura de componentes utilizados para a construção da tela de login.

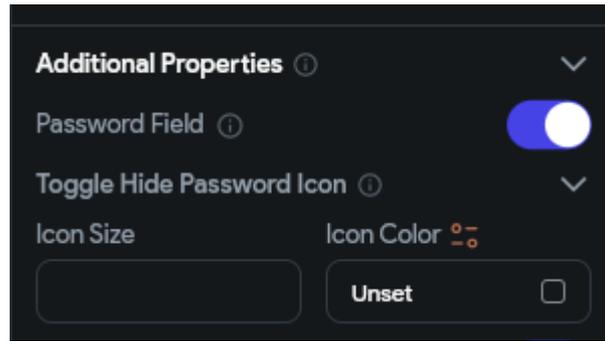
Figura 09: Estrutura de componentes da tela de login



Fonte: o autor

Ambos os campos do e-mail e senha são componentes “TextField”. O campo senha tem a opção “Password Field” ativada para apontar as propriedades de senha.

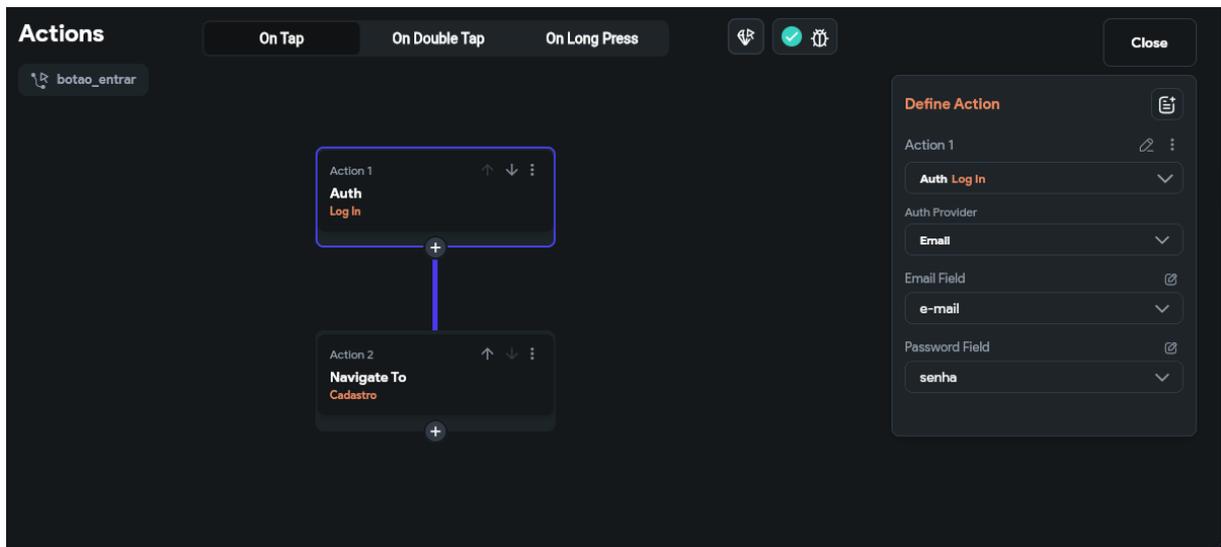
Figura 10: propriedades de senha



Fonte: o autor

O componente “Entrar” é do tipo Button e executa uma série de ações quando o usuário clica nele. A primeira ação executada é a autenticação de usuário, onde ele recebe o conteúdo dos componentes e-mail e senha e procura na lista de usuários cadastrados no sistema. Caso o e-mail ou senha serem inválidos será retornada uma mensagem de erro, caso contrário prosseguirá para a segunda ação. Na segunda ação o usuário será direcionado à página “Cadastro”.

Figura 11: Árvore de ações do botão “Entrar”.



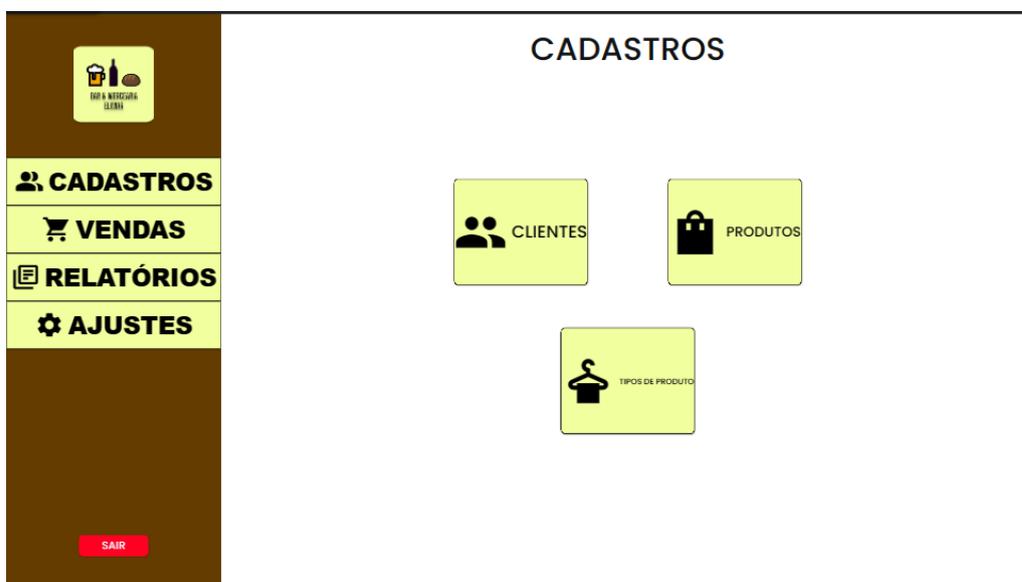
Fonte: o autor

5.2.2 Tela de cadastros

A tela de cadastros mostra todas as opções de cadastro do sistema, como o cadastro de clientes, cadastro de produtos e cadastro de tipos de produtos. Foram aplicados para todos os botões da tela uma ação de navegação. Essa é a tela inicial do sistema e aparece quando o usuário faz o login ou quando clica em “CADASTROS”

no menu lateral esquerdo.

Figura 12: Tela de cadastros



Fonte: o autor

5.2.3 Tela de cadastro de clientes

Essa tela mostra todos os clientes cadastrados no sistema e dá a opção de realizar um novo cadastro pelo botão “NOVO CLIENTE” ou de editar um cadastro existente pelo ícone de lápis ao lado de cada registro.

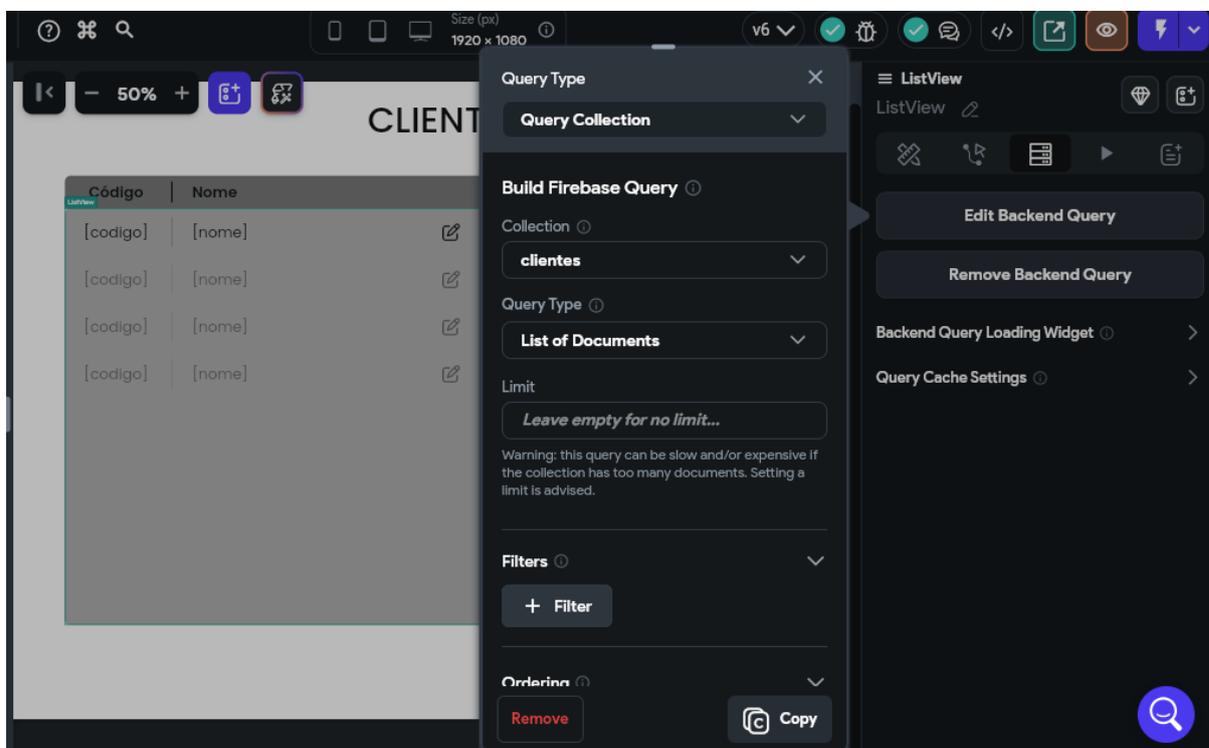
Figura 13: Tela cadastro de clientes.



Fonte: o autor

Para mostrar todos os cadastros disponíveis foi utilizado o componente ListView. Primeiro, ele foi vinculado a tabela de clientes através de uma listagem de documentos pelo menu Backend Query.

Figura 14: componente ListView

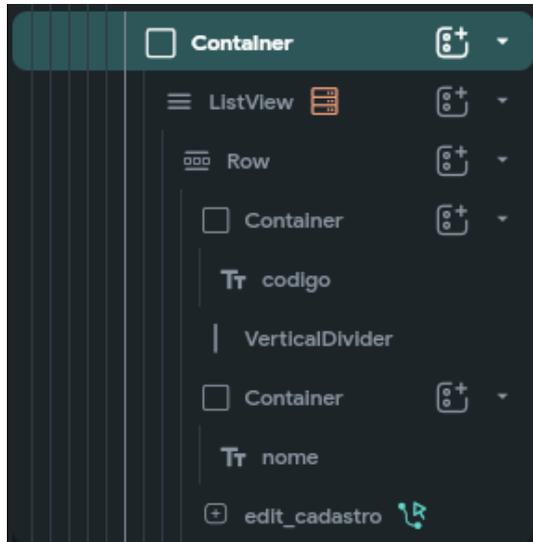


Fonte: o autor

Em seguida, foi inserido uma Row dentro da ListView e como a ListView já está

vinculado a tabela de clientes como uma lista de documentos, a FlutterFlow vai criar um Row para cada registro da tabela clientes.

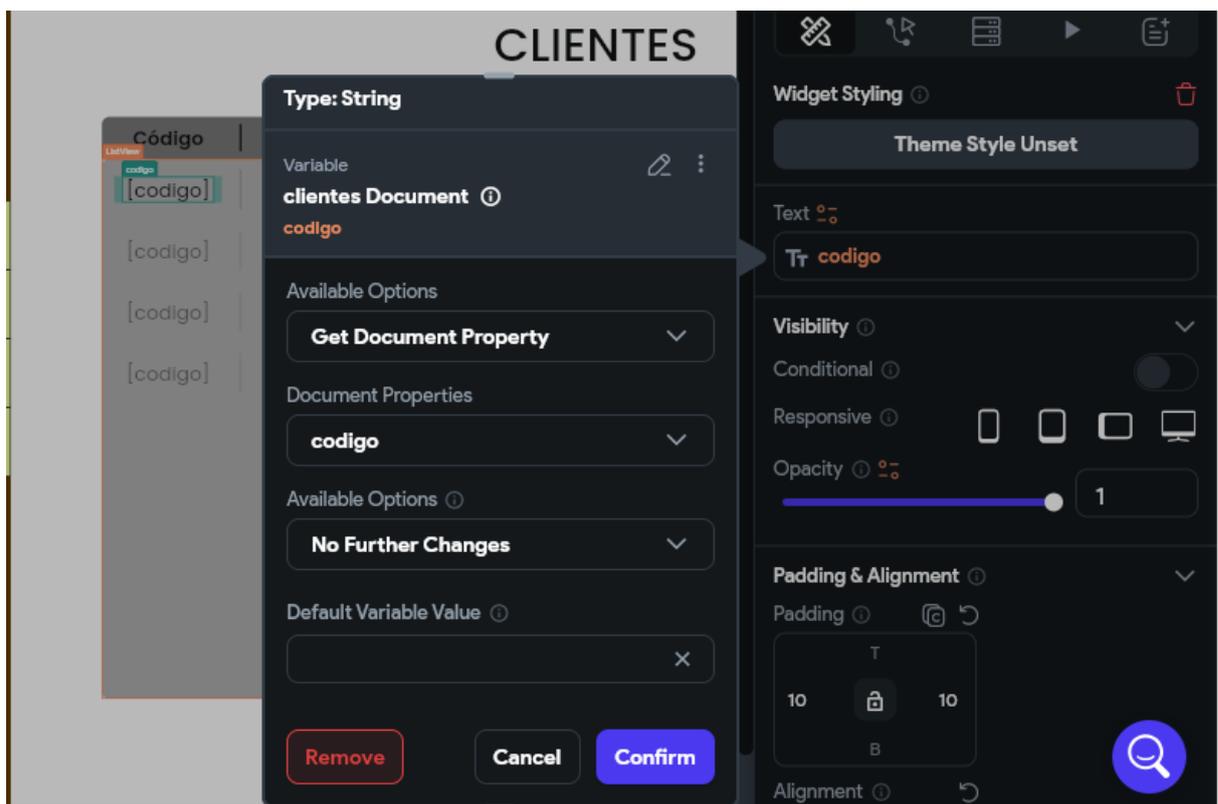
Figura 15: Componentes da lista de clientes



Fonte: o autor

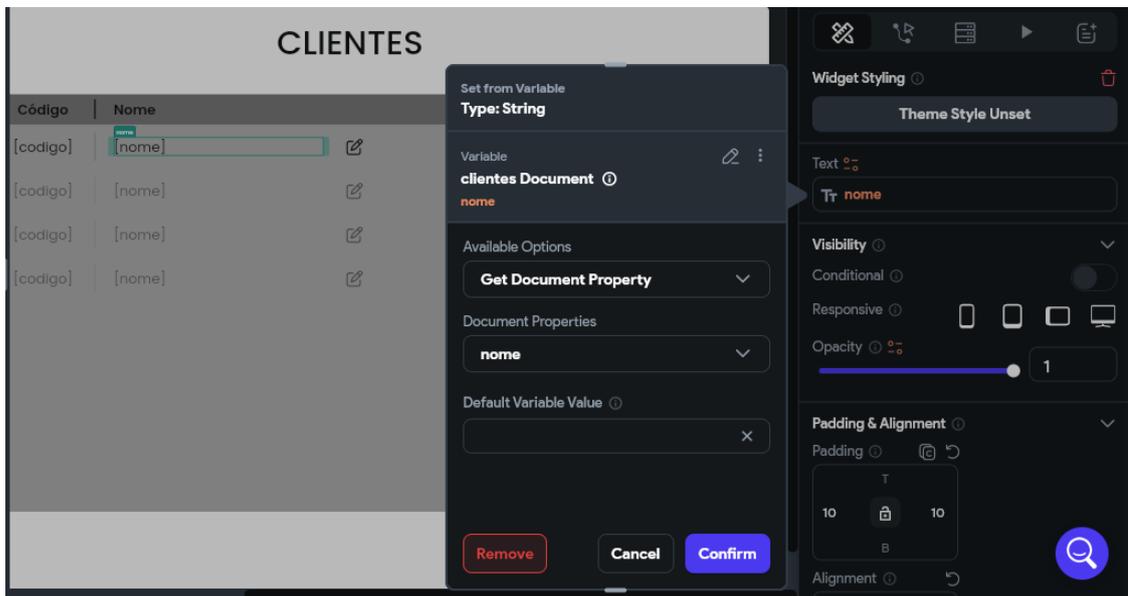
Para mostrar os campos código e nome da tabela clientes, foram usados dois componentes Text cujos textos foram as suas respectivas colunas através do menu Properties.

Figura 16: Atribuindo a coluna código ao componente.



Fonte: o autor

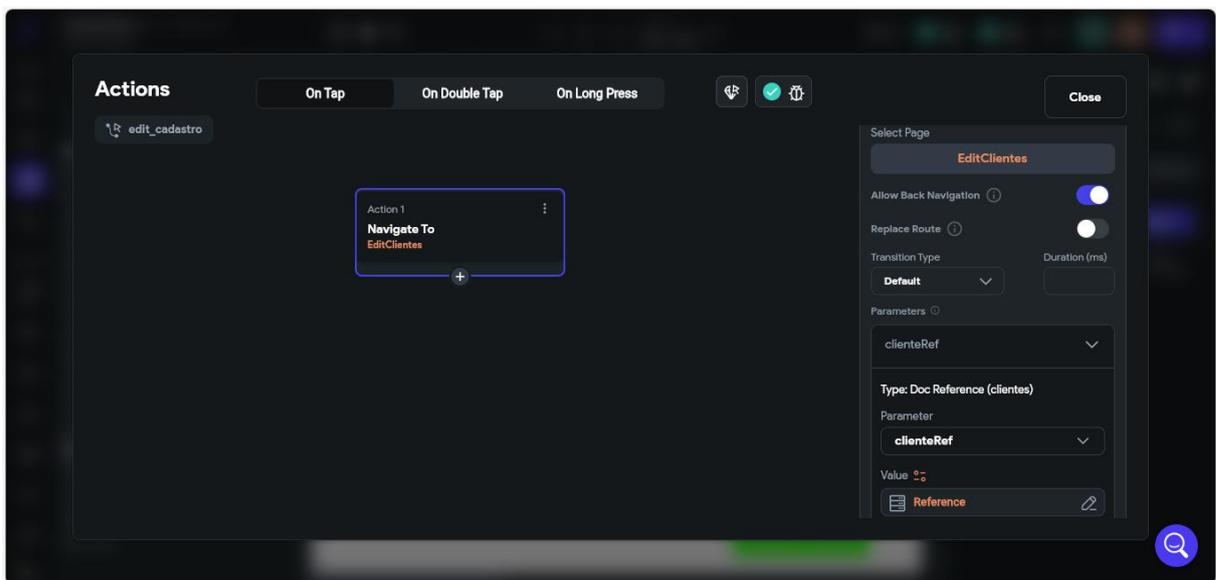
Figura 17: Atribuindo a coluna nome ao componente.



Fonte: o autor

No ícone de editar cadastro, além de ter uma ação que leva o usuário a tela de editar clientes, também é passado como parâmetro uma referência de documento que servirá para repassar os dados do registro do cliente para que possam ser editados.

Figura 18: Ações do botão de editar cadastro.



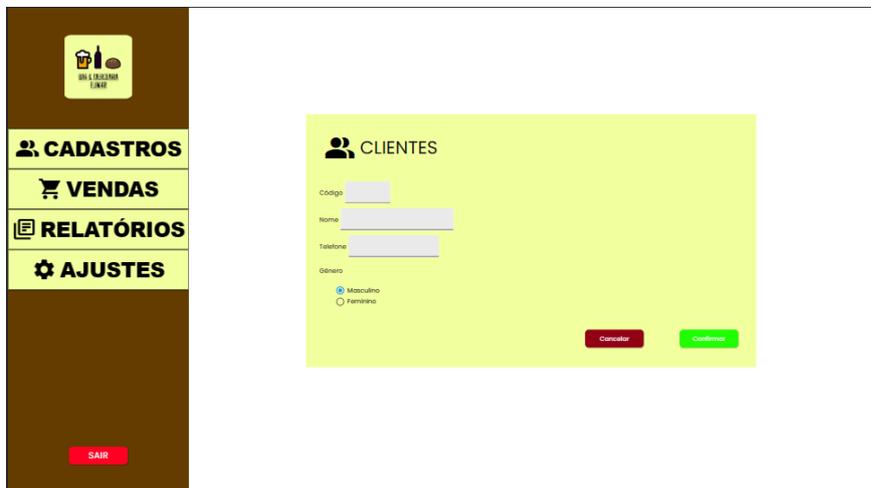
Fonte: o autor

As telas cadastro de produtos e cadastro de tipos de produto seguem o mesmo modelo da tela cadastro de clientes.

5.2.4 Formulário de cadastro de clientes

Essa tela mostra um formulário para cadastrar novos clientes. É possível acessá-la através do botão “NOVO CLIENTE” na tela cadastro de clientes.

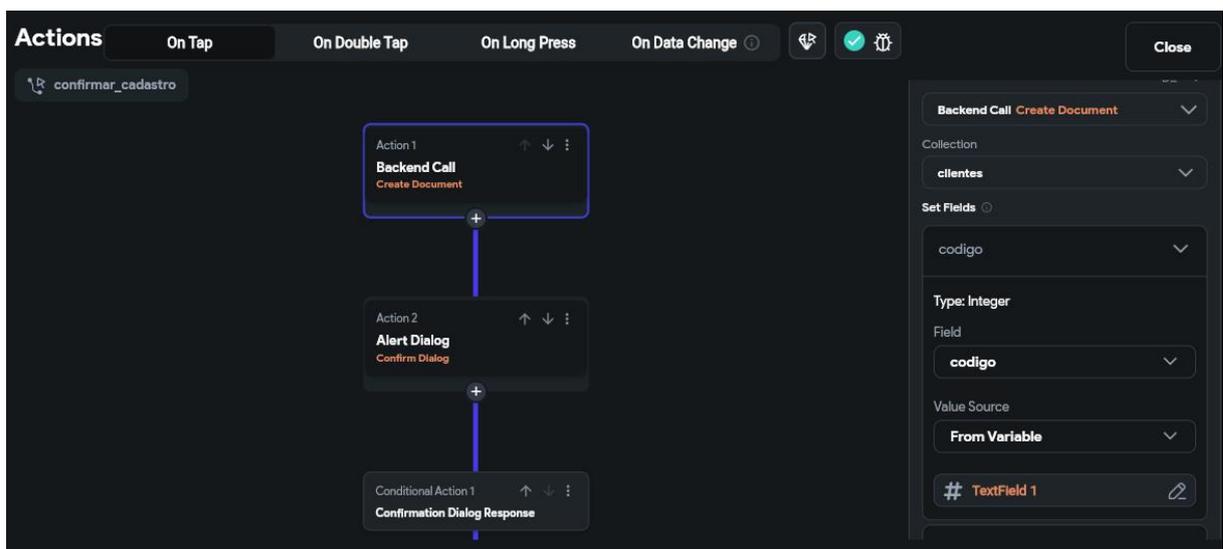
Figura 19: Formulário de cadastro



Fonte: o autor

O formulário contém os campos código, nome, telefone e gênero. No botão cancelar, o usuário poderá voltar a página anterior e no botão confirmar, foi atribuído uma série de ações mostradas abaixo.

Figura 20: Árvore de ações do botão confirmar_cadastro.



Fonte: o autor

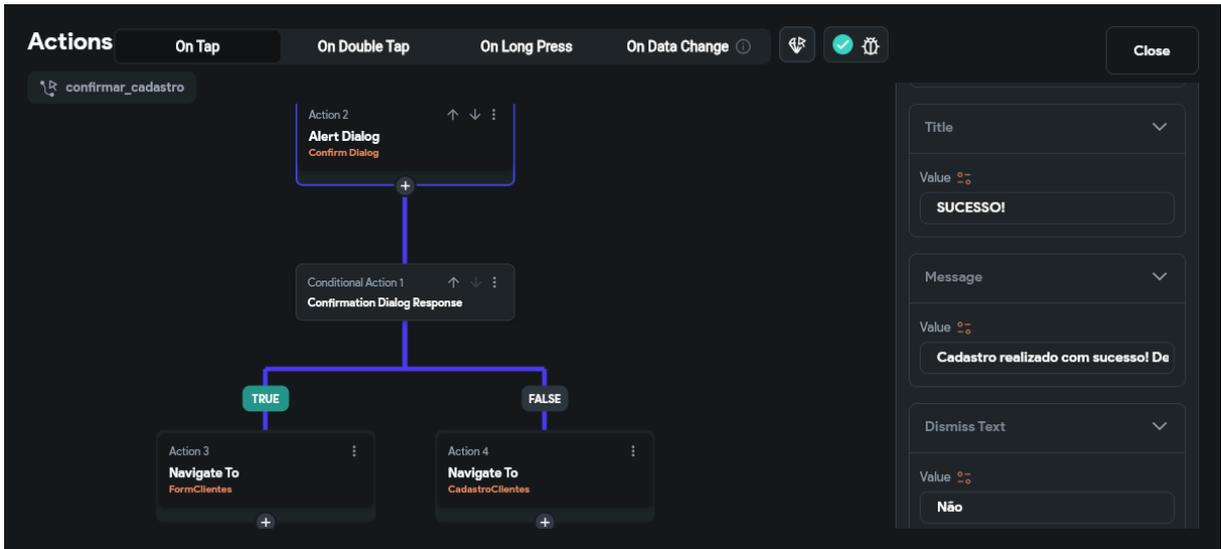
A primeira ação faz uma chamada de backend para criar um documento da tabela clientes.

Os valores, por sua vez, são extraídos dos campos do formulário de cadastro. O banco de dados não permitirá cadastro com códigos iguais e deverá responder uma mensagem de erro encerrando a árvore de ações.

A segunda ação gera uma caixa de diálogo e uma ação condicional, que confirma o registro do cadastro e pergunta ao usuário se deseja realizar um novo cadastro.

Caso o usuário clique em “sim”, ele será redirecionado a de formulário de cadastro, senão voltará a tela de cadastro de clientes.

Figura 21: Continuação da árvore de ações



Fonte: o autor

5.2.5 Editar cadastro de cliente

Nessa tela são apresentadas as informações de um registro para que possam ser alteradas ou excluídas. Primeiro, será apresentado o parâmetro de referência dado pelo botão de editar cadastro. Essa ação é feita por uma Backend Query, porém agora é solicitado um documento de uma referência.

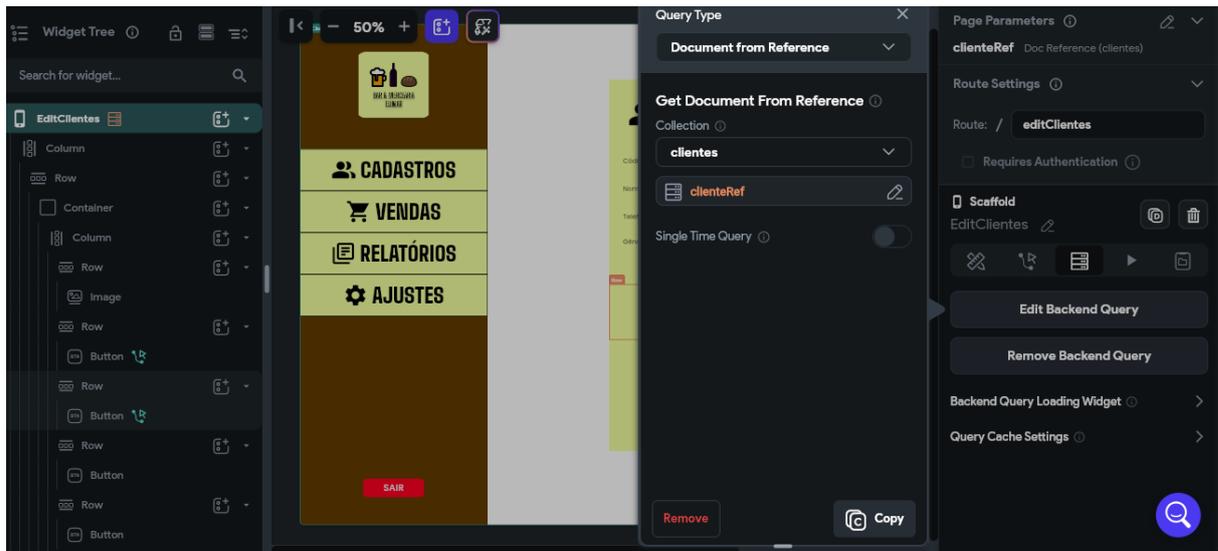
Figura 22: Tela editar cadastro de clientes



The 'CLIENTES' edit form has a light green background. It contains the following fields: 'Código' with the value '1'; 'Nome' with the value 'Arthur Alves'; and 'Telefone' with the value '16 999123'. Below these is a 'Gênero' section with radio buttons for 'Masculino' (selected) and 'Feminino'. At the bottom of the form are three buttons: 'Cancelar' (red), 'Excluir Cadastro' (red), and 'Salvar Alterações' (green).

Fonte: o autor

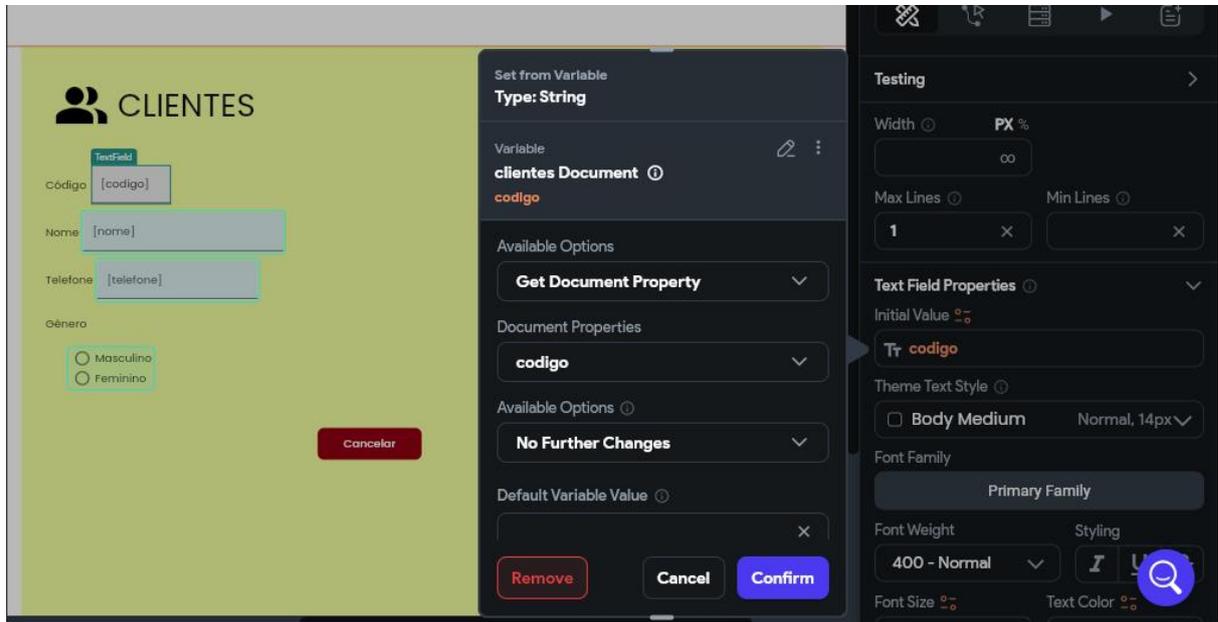
Figura 23: Pegando a referência de um documento



Fonte: o autor

Depois, é atribuído uma variável no valor inicial de cada campo do formulário. Essa variável puxa os valores do registro que será alterado na tabela. A imagem abaixo mostra um exemplo com campo código.

Figura 24: Atribuindo uma variável ao campo código.

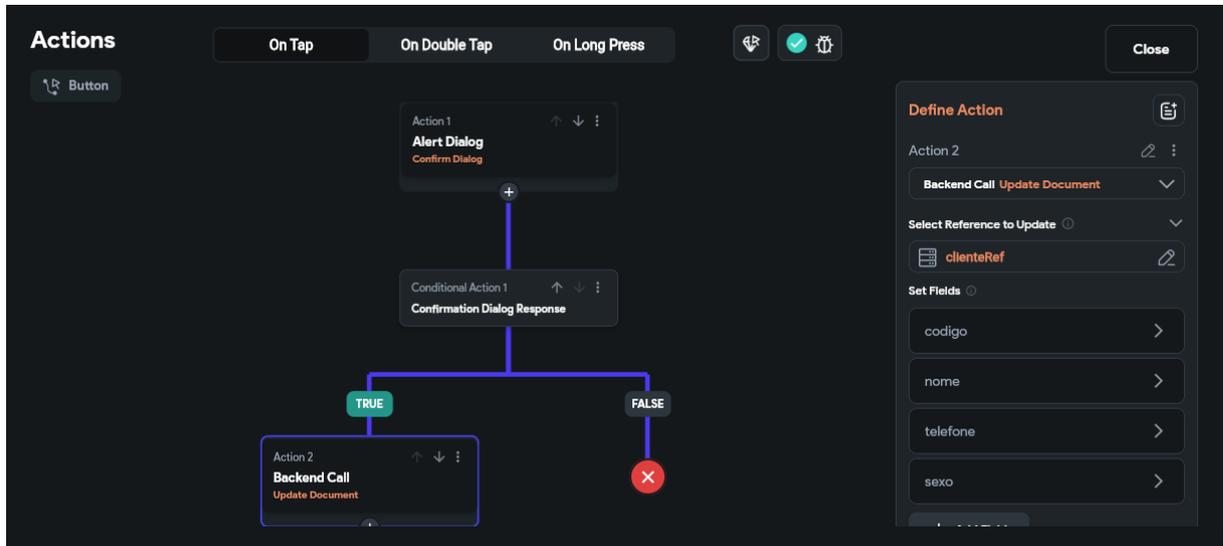


Fonte: o autor

Depois de realizar as alterações, o usuário deverá clicar no botão “Salvar Alterações” para salvá-las. Este botão é responsável por fazer um update na tabela clientes. Antes da alteração, aparece uma caixa de confirmação ao usuário. Caso ele con-

firme, serão salvas as alterações feitas nos campos, senão a árvore de ações é cancelada. A imagem abaixo mostra a árvore de ações do botão “Salvar Alterações”.

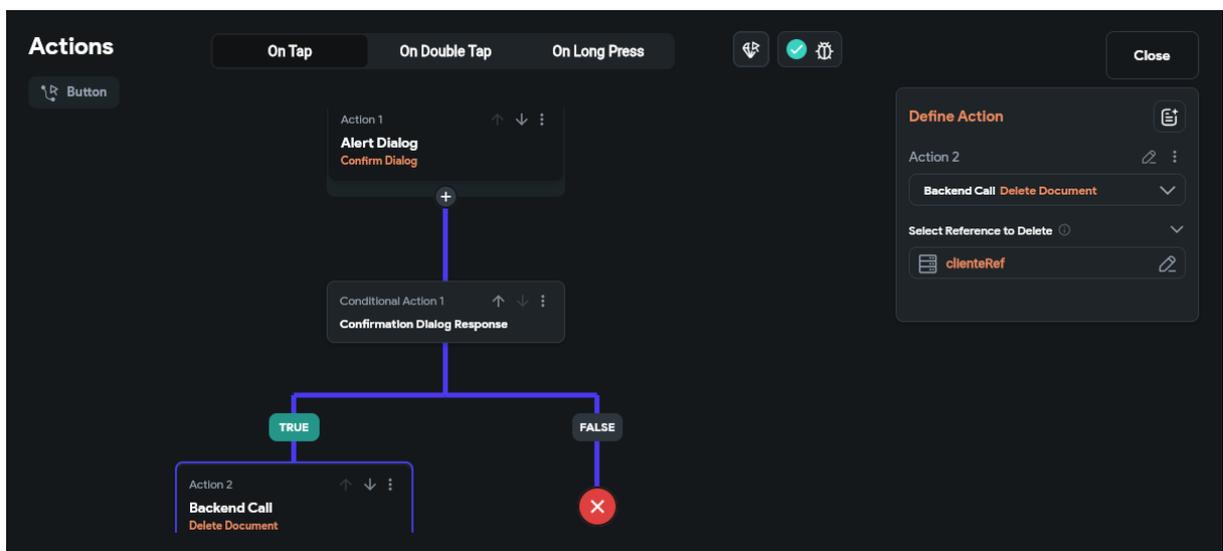
Figura 25: Árvore de ações do botão “Salvar Alterações”



Fonte: o autor

O botão “Excluir Cadastro” exclui o registro da tabela clientes permanentemente, por isso, assim como no botão “Salvar Alterações”, também é utilizado um caixa de confirmação. A imagem abaixo mostra a árvore de ações do botão “Excluir Cadastro”.

Figura 26: Árvore de ações do botão “Excluir Cadastro”

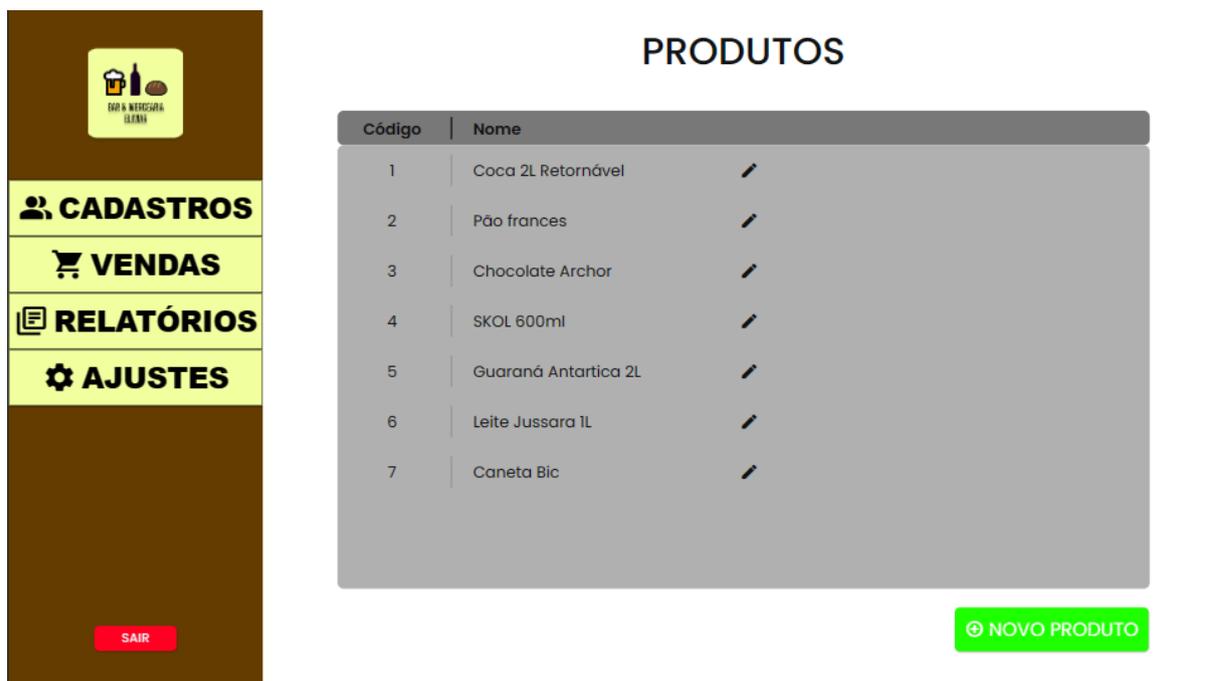


Fonte: o autor

5.2.6 Tela de cadastro de produtos

A tela de cadastro de produtos mostra todos os produtos cadastrados no sistema. Ela segue o mesmo modelo da tela de cadastro de clientes. Há uma lista de produtos que é vinculada com a tabela produtos e é possível cadastrar novos produtos pelo botão “NOVO PRODUTO” ou editar produtos existentes pelo ícone ao lado de cada registro.

Figura 27: Tela de cadastro de produtos.

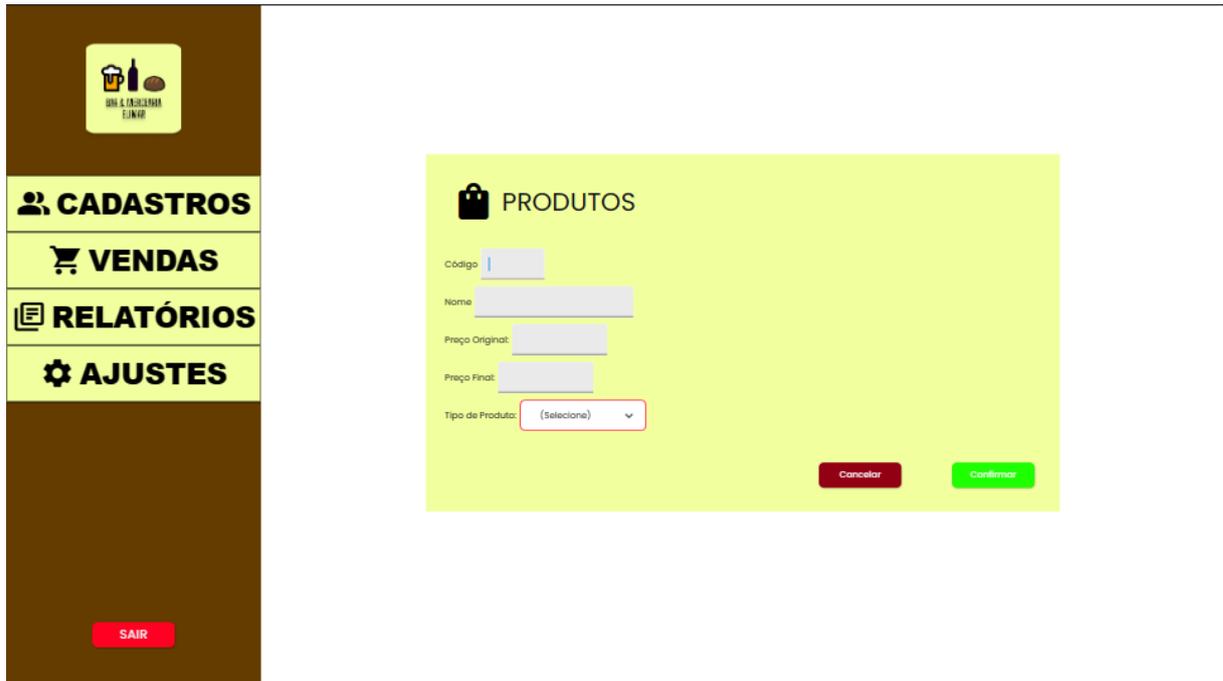


Fonte: o autor

5.2.7 Formulário de cadastro de produtos

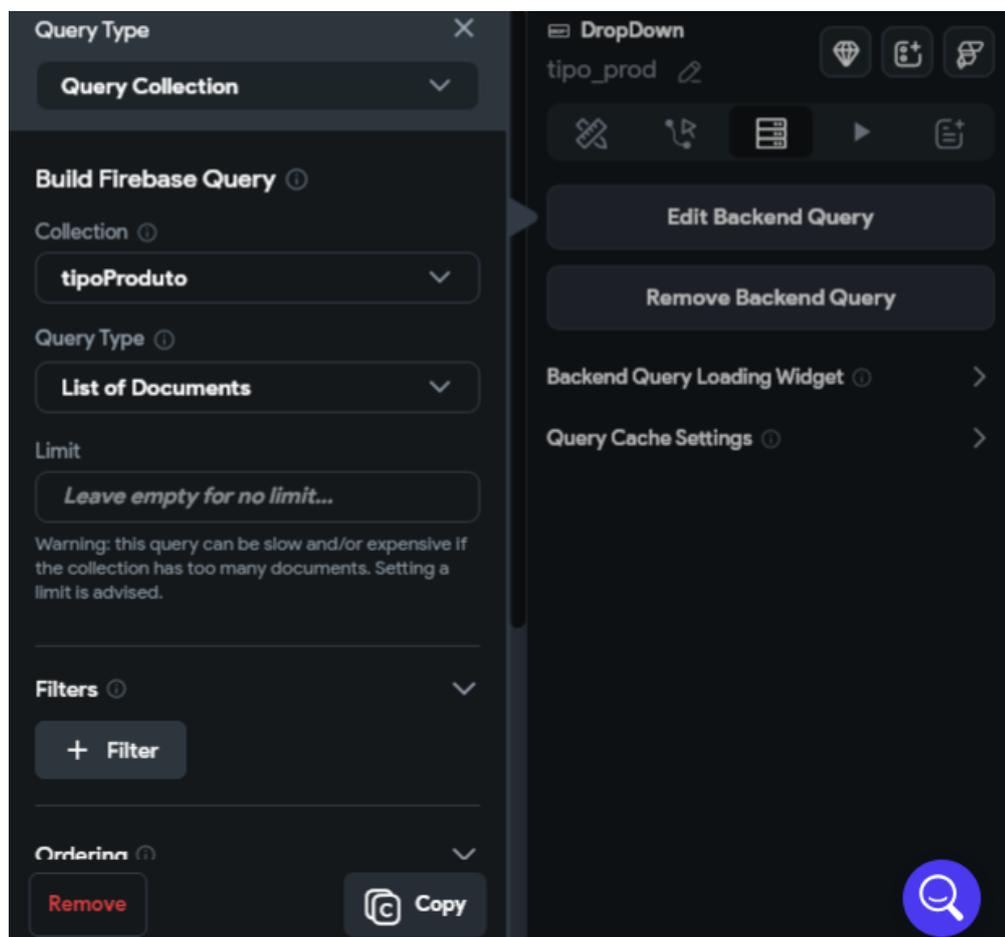
O formulário de cadastro de produtos também segue o mesmo modelo do formulário de clientes. O usuário deverá preencher os campos código, nome, preço original, preço final e tipo de produto. O código não poderá ser igual ao de um cadastro de produto existente. O campo tipo de produto é um DropDown e está ligado a tabela de tipos de produto e contém todos os nomes dos tipos de produto cadastrados no sistema. Esta ligação foi feita por duas etapas. Primeiro a tabela tipos de produto foi atribuída ao componente DropDown como uma lista de documentos.

Figura 28: Formulário de produtos



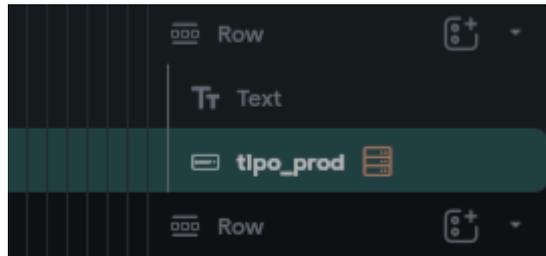
Fonte: o autor

Figura 29: Configuração do formulário



Fonte: o autor

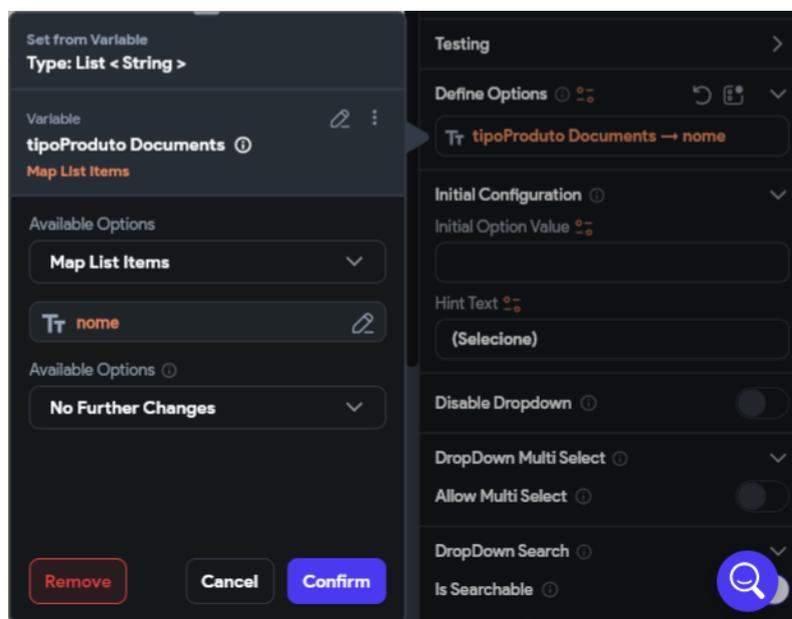
Figura 30: Atribuindo a tabela tipos de produto ao componente



Fonte: o autor

E por fim, é definido o campo nome da tabela tipos de produto nas opções definitivas do DropDown.

Figura 31: definido alguns campos



Fonte: o autor

5.2.8 Editar cadastro de produtos

Essa tela segue o mesmo padrão do formulário editar cadastro de clientes. As do registro informações não puxadas por meio de um parâmetro dado pelo IconButton na tela de cadastro de produtos e então alocadas em seus respectivos campos.

Os botões “Excluir Cadastro” e “Salvar Alterações” também tem a mesma função que os do formulário editar cadastro de cliente, mas nesse caso fazem alterações na tabela de produtos.

Figura 32: Tela editar cadastro de produtos

PRODUTOS

Código: 5

Nome: Guaraná Antarctica 2L

Preço Original: 6

Preço Final: 7.2

Tipo de Produto: Bebidas

Cancelar Excluir Cadastro Salvar Alterações

Fonte: o autor

5.2.9 Tela de cadastro de tipos de produtos

Assim como o modelo de tela de cadastro de clientes e produtos, ela também mostra todos os registros do sistema só que agora dos tipos de produtos. Os tipos de produto são categorias em que um produto pode ser inserido, por exemplo, produtos de limpeza, alimentícios, bebidas e entre outros.

Figura 33: Tela de cadastro de tipos de produto

TIPOS DE PRODUTO

Código	Nome
1	Bebidas
2	Materiais Escolares
3	Produtos Alimentícios
4	Porções
5	Bebidas Alcoolicas
6	Utensílios

NOVO PRODUTO

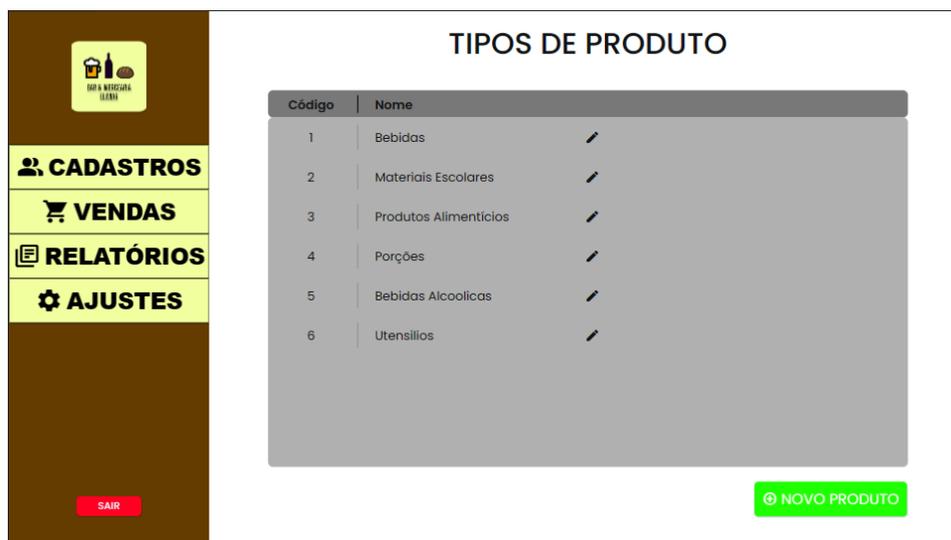
Fonte: o autor

5.2.8 Formulário de cadastro de tipos de produtos

Esse formulário também segue o mesmo padrão de cadastro dos clientes e produtos e possui apenas dois campos. O campo código e o campo nome sendo ambos obrigatórios e únicos.

Os tipos de produtos são usados no cadastro de novos produtos, por isso devem ser registrados no sistema antecipadamente.

Figura 34: Tela do formulário de cadastro de tipos de produto



Fonte: o autor

5.2.9 Editar cadastro de tipos de produto

Essa tela também segue o mesmo padrão do formulário editar cadastro de clientes.

As do registro informações não puxadas por meio de um parâmetro dado pelo IconButton na tela de cadastro de tipos de produto e então alocadas em seus respectivos campos.

Os botões “Excluir Cadastro” e “Salvar Alterações” também tem a mesma função que o formulário editar cadastro de cliente e produto, mas nesse caso fazem alterações na tabela de tipos de produto.

Figura 35: Tela editar cadastro de tipos de produto.



Fonte: o autor

5.2.10 Tela de vendas

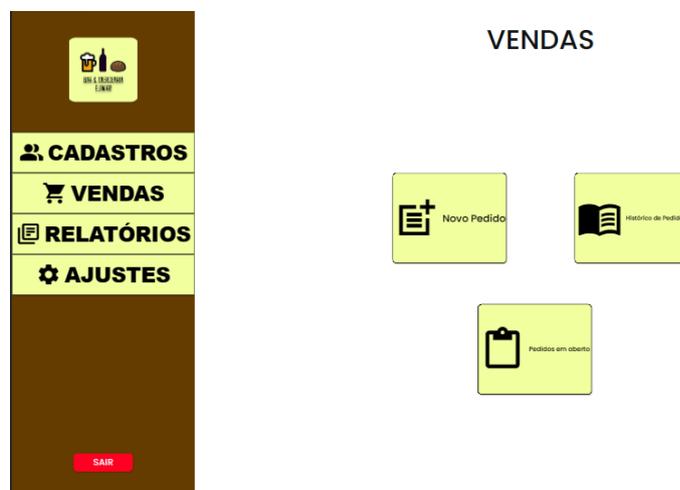
Na tela de vendas estão presentes 3 botões, o botão “Novo Pedido” onde poderá ser cadastrado novos pedidos.

O botão “Pedidos em aberto” onde estará listado os pedidos que ainda precisam ser concluídos.

E por fim, o botão “Histórico de pedidos” onde é apresentado os pedidos já finalizados em ordem do mais recente para o mais antigo.

Foram atribuídos a todos os botões uma ação que os faz navegar para as suas respectivas páginas com um clique.

Figura 36: Tela de vendas



Fonte: o autor

5.2.11 Formulário de cadastro de um novo pedido

É possível acessar essa tela através do botão “Novo Pedido” da tela de vendas. Nela está presente um formulário que contém as informações iniciais do pedido. Os campos a serem preenchidos são o de código do pedido, cliente, data e forma de pagamento.

O campo cliente é um Dropdown que está ligado a tabela de clientes e contém todos os clientes cadastrados no sistema. Já o campo data tem como valor padrão a data e o horário total. Após o pedido ser salvo, ele aparecerá na tela pedidos em aberto.

Figura 37: Formulário de pedido

NOVO PEDIDO

Código do Pedido:

CLIENTE:

Data: 2023-10-22 10:43:10.081

Forma de Pagamento:

Fonte: o autor

5.2.12 Pedidos em aberto

Nessa tela estão listados todos os pedidos em aberto e pode ser acessada também pela tela de vendas.

Os pedidos em aberto poderão ser acessados pelo ícone presente ao lado de cada registro para que possa ser adicionado os produtos que o cliente irá comprar.

Figura 38: Tela pedidos em aberto

PEDIDOS EM ABERTO

Código	Cliente	Data	
4	Arthur Alves	2023-10-22 10:43:10.081	

Fonte: o autor

5.2.13 Tela de carrinho

É acessada através da tela pedidos em aberto e é aqui que o usuário poderá editar as informações do pedido e adicionar os produtos que o cliente irá comprar. Na parte de carrinho estarão os campos referentes as informações dos produtos, como quantidade, produto e preço. O campo quantidade é um CountController, o campo produto é um DropDown que está ligado a tabela produto e contém todos os nomes dos produtos cadastrados no sistema.

Por fim, o campo preço está ligado ao campo preco_final da tabela produtos e atualiza simultaneamente o seu valor conforme o usuário seleciona um produto. Abaixo da parte de carrinho mostra os produtos adicionados com a quantidade, nome do produto, preço e subtotal. Cada produto poderá ser excluído pelo ícone vermelho ao lado de cada subtotal. O campo total mostra a soma de todos os subtotais, sendo atualizado toda vez que o usuário adiciona produtos ao carrinho.

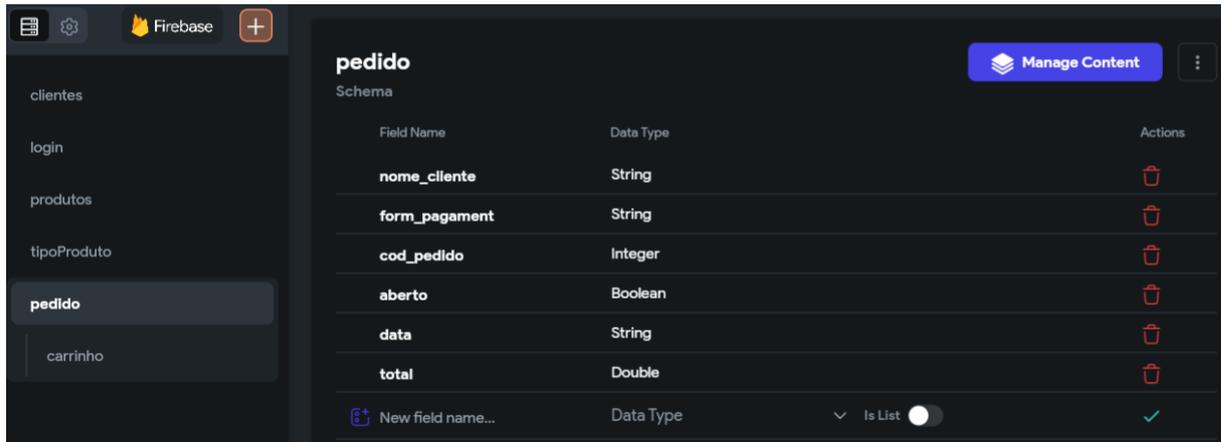
Figura 39: Tela de carrinho



Fonte: o autor

É importante ressaltar que a tabela carrinho utiliza o recurso de “subcollection”. Segundo a documentação oficial da FireBase, “subcollections” são coleções armazenadas em um documento. Eles são usados para organizar dados hierarquicamente e para criar uma estrutura aninhada. No caso do sistema, cada registro da tabela de pedidos pode conter uma tabela carrinho que, por sua vez pode conter vários registros relacionados aos produtos. Este recurso é exclusivo do Cloud FireStore e não é presente na estrutura dos bancos SQL.

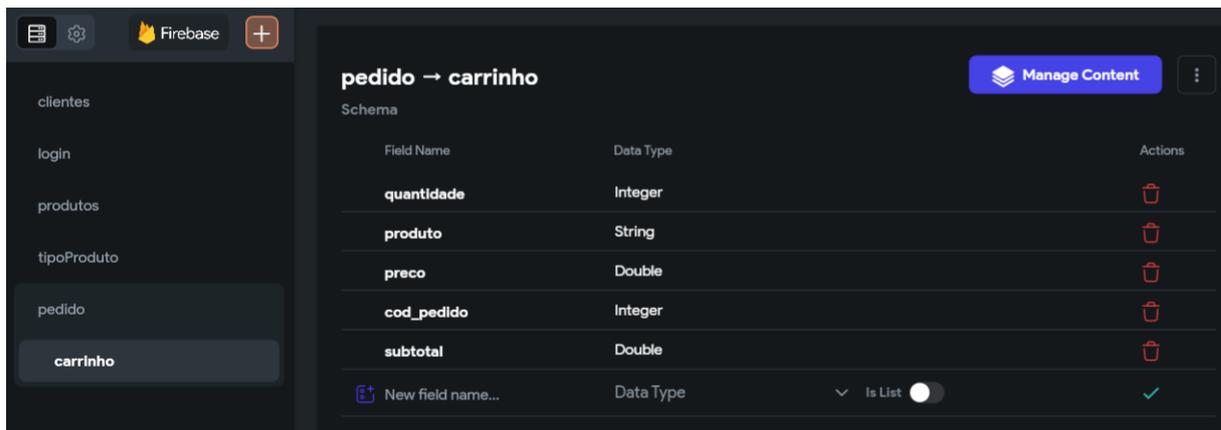
Figura 40: Esquema da tabela pedido



Field Name	Data Type	Actions
nome_cliente	String	
form_pagament	String	
cod_pedido	Integer	
aberto	Boolean	
data	String	
total	Double	
New field name...	Data Type	

Fonte: o autor

Figura 41: Esquema da tabela carrinho



Field Name	Data Type	Actions
quantidade	Integer	
produto	String	
preco	Double	
cod_pedido	Integer	
subtotal	Double	
New field name...	Data Type	

Fonte: o autor

5.2.14 Histórico de pedidos

Essa tela mostra, por meio de um filtro, o histórico de pedidos que já foram finalizados. Os pedidos por padrão são listados na ordem do mais recente para o mais antigo através de uma ordenação.

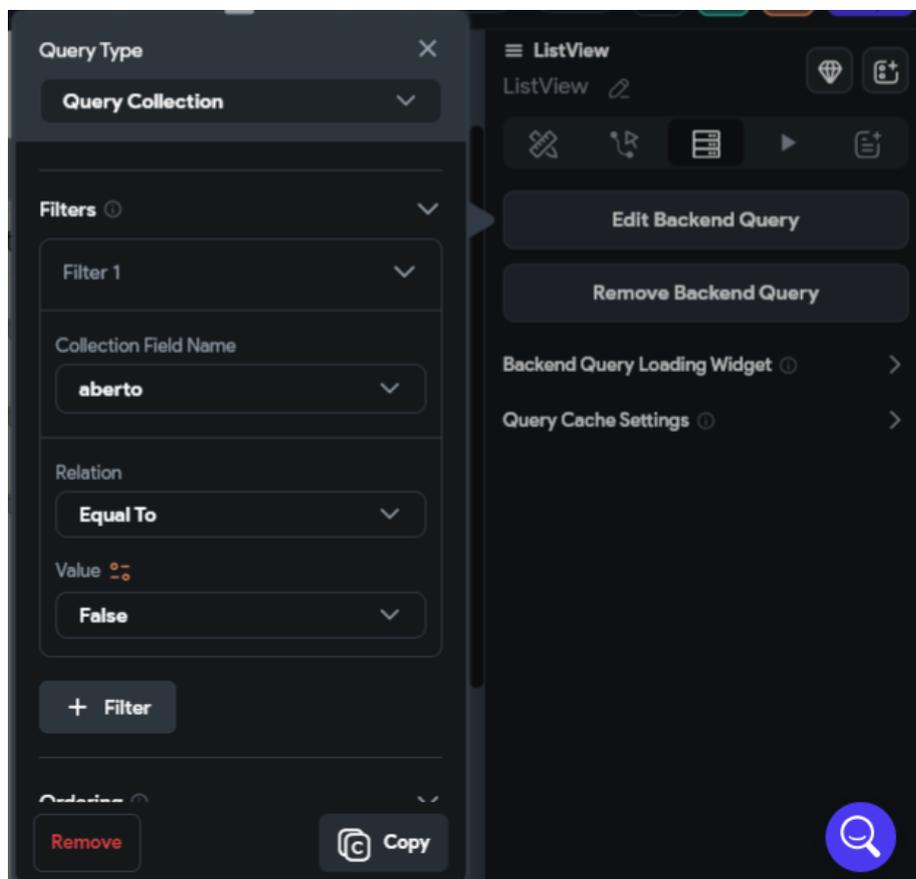
Ela segue a mesmo modelo de listagem através de um ListView das telas de cadastro de clientes, produtos e tipos de produto. Ao lado do total de cada pedido tem um ícone que leva a página de detalhes dos pedidos.

Figura 42: Tela de histórico de pedidos



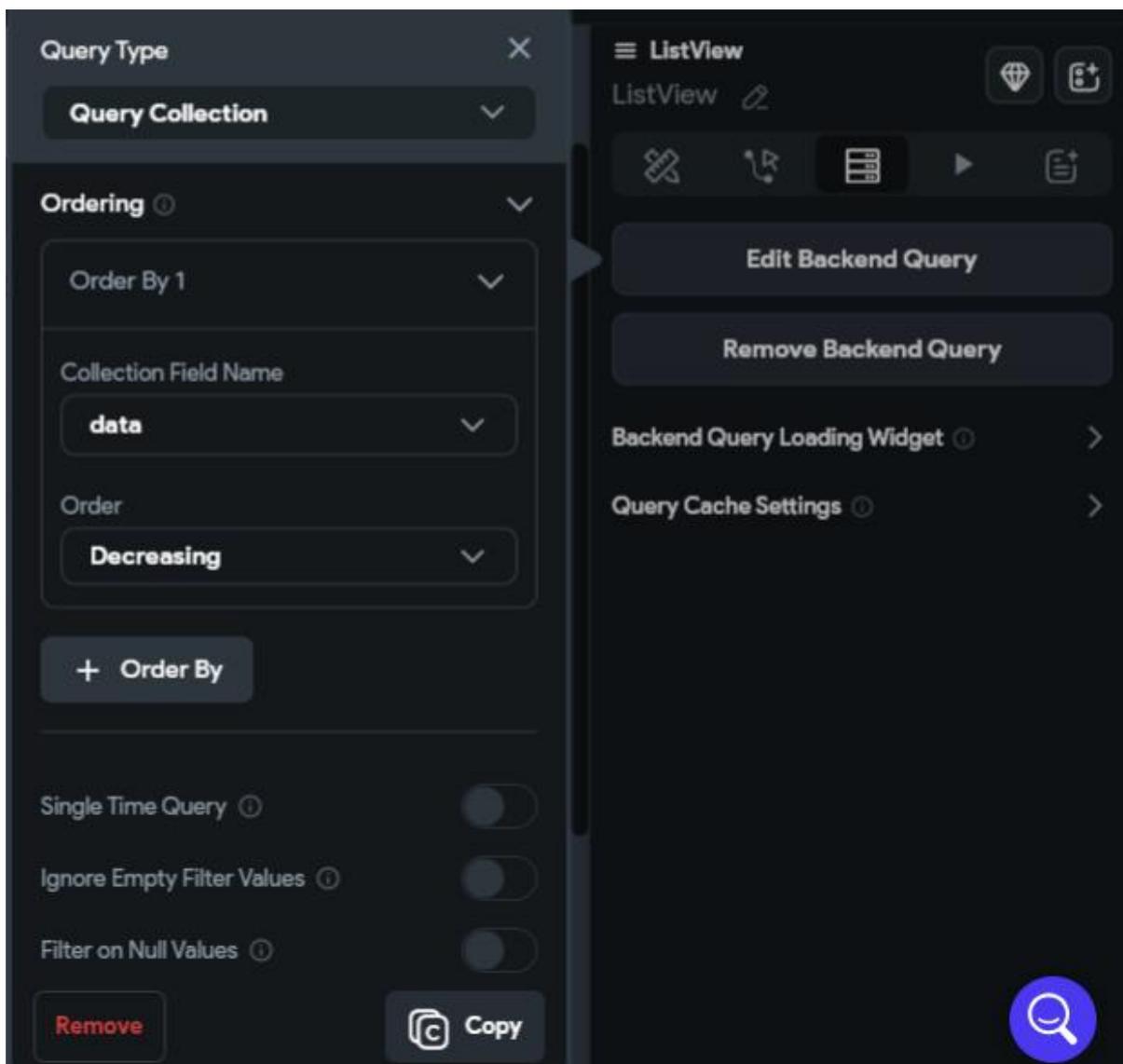
Fonte: o autor

Figura 43: Filtro utilizado no histórico



Fonte: o autor

Figura 44: Ordenação utilizada no histórico



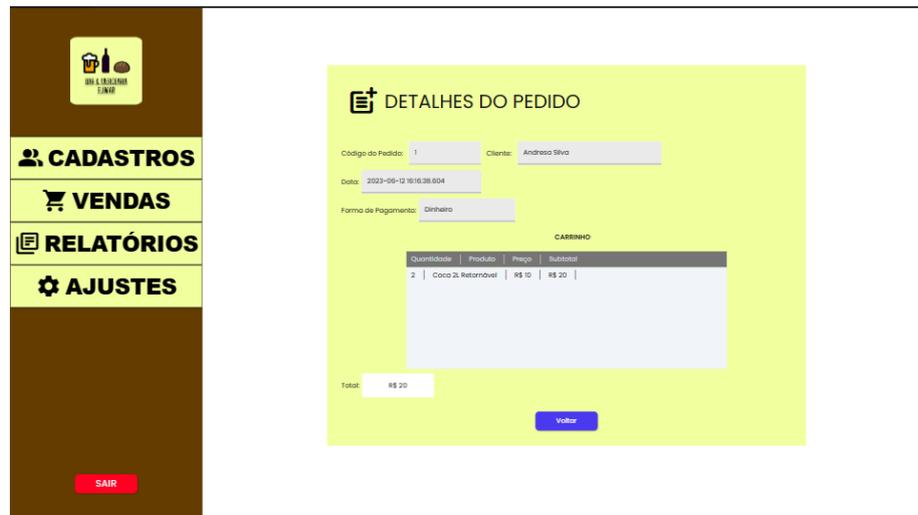
Fonte: o autor

5.2.15 – Detalhes dos pedidos

É acessada por meio do histórico de pedidos e mostra as informações do pedido finalizado como código do pedido, nome do cliente, data e hora, forma de pagamento, produtos pedidos e total.

As informações não poderão ser editadas.

Figura 45: Tela de detalhes do pedido.



Fonte: o autor

6 Resultados e Discussão

A solução proposta foi o desenvolvimento de um software de gestão personalizado, projetado especificamente para atender às necessidades do estabelecimento Bar & Merceria Elimar.

Esta solução alcançou o objetivo de organizar as vendas e apresentar informações sobre os produtos vendidos. Através do processo de organização de cadastros de clientes e a geração de dashboards e relatórios através de um sistema, o que contribuirá no momento da tomada de decisão.

Como o principal problema encontrado foi a falta de informação com relação as vendas, pois não era possível saber com clareza quais produtos eram ou não vendidos. Com isso, a solução proposta também ajudará na organização deste processo.

As principais funcionalidades do sistema são: Cadastrar clientes; cadastrar produtos; registrar vendas; registrar histórico de vendas; gerar relatórios; e gerar dashboards.

Cientes do contexto atual, onde os processos exigem qualidade e agilidade, este projeto desenvolveu uma solução sistêmica que aborda de maneira eficaz o desafio da gestão de um estabelecimento (Bar & Merceria), proporcionando uma solução transparente e organizada.

Considerações finais

Ao longo do desenvolvimento desta solução sistêmica, desde agosto de 2022, buscamos atender aos objetivos iniciais estabelecidos para o desenvolvimento do sistema de gestão de um estabelecimento (Bar & Mercearia).

O principal desafio foi a complexidade dos processos e a organização dos dados sobre estoque e controle de vendas, pelo fato do comércio não apresentar uma forma coerente de registro de dados e uma aparente desorganização em seus processos. Outro desafio a ser mencionado, foi desenvolver o sistema de uma forma que não atrapalhasse a flexibilidade das vendas do comerciante.

Visando a liberação do uso desta solução, toda a infraestrutura de desenvolvimento foi implementada e as tecnologias escolhidas provaram ser sólidas e eficazes. Essas tecnologias ofereceram uma série de recursos e facilidades modernas e escaláveis que contribuíram para o sucesso do desenvolvimento de um Sistema de Gestão Integrada.

No futuro pretendemos desenvolver algumas novas funcionalidades, valorizando a fidelização dos clientes.

Referências

CANGUÇU, Raphael. **Requisitos funcionais e requisitos não funcionais, o que são?** Fevereiro 2021.

BR Modelo Web - Ferramenta online para modelagem de banco de dados. Disponível em: <https://www.brmodeloweb.com/lang/pt-br/index.html>. Acesso em: 22 out. 2023.

DART PROGRAMMING language. Outubro 2023.

DRAW.IO editor. Disponível em: <https://www.drawio.com/doc/getting-started-editor>. Acesso em: 22 out. 2023.

FIREBASE authentication. Outubro 2023.

FIRESTORE, firebase. Outubro 2023.

FLUTTERFLOW - Build apps for any screen. Outubro 2023.

LUCIDCHART. Diagramming, data visualization and real-time collaboration. 2023.

MEYER, Maximiliano. O que é excel? 27 jul. 2018. Disponível em: <https://www.aprenderexcel.com.br/2013/tutoriais/o-que-e-excel>. Acesso em: 22 out. 2023.

NOLEDO, Cairo. **Requisitos não funcionais**: o guia completo! Abril 2022.

PIRES, Raphael. **5W2H**: o que é esse método de definição de atividades e como usar. Maio 2019.

REIS, Tiago. **Business Model Canvas**: o que é esse modelo de plano de negócios? Abril 2019.

SCHEMES, Taynara. **Regras de negócio em TI**: conheça esse importante conceito! Dezembro 2020.