

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**DESENVOLVIMENTO DE SOFTWARE DE GERENCIAMENTO DE CONTAS DE
CLIENTES**

MATEUS DE SOUZA ALMEIDA

GOIÂNIA
2025

MATEUS DE SOUZA ALMEIDA

DESENVOLVIMENTO DE SOFTWARE DE GERENCIAMENTO DE CONTAS DE
CLIENTES

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a):

Prof. Me. André Luiz Alves

Banca examinadora:

Prof.^a Ma. Ana Flávia Marinho de Lima Garrote

Prof. Me. Ernesto Fonseca Veiga

GOIÂNIA
2025

MATEUS DE SOUZA ALMEIDA

DESENVOLVIMENTO DE SOFTWARE DE GERENCIAMENTO DE CONTAS DE
CLIENTES

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica e de Artes,
da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia
de Computação, em ____/____/____.

Orientador (a): Prof. Me. André Luiz Alves

Prof.^a Ma. Ana Flávia Marinho de Lima Garrote

Prof. Me. Ernesto Fonseca Veiga

GOIÂNIA
2025

RESUMO

O trabalho apresenta o projeto e desenvolvimento do sistema ContaFácil para gerenciar contas de clientes no açougue Casa das Carnes. O objetivo consiste em substituir o controle manual de vendas a prazo e reduzir a inadimplência, por meio de levantamento de requisitos junto ao proprietário, seguido da especificação, implementação e validação de uma aplicação *web*. Para tanto, foram conduzidas entrevistas, análises de processos e prototipação de telas, resultando na construção de funcionalidades que centralizam cadastro de clientes, registro de compras a prazo e gerenciamento de saldos pendentes. Os resultados demonstram que o sistema elimina registros em papel, minimiza erros de cálculo e fornece relatórios financeiros precisos, agilizando a cobrança e aprimorando principalmente a transparência financeira. O sistema promove maior eficiência operacional e pode ser adaptado para outras micro e pequenas empresas que adotem o “fiado” como prática de venda, evidenciando seu potencial de escalabilidade e melhoria na gestão de crédito.

Palavras chaves: Gerenciamento de Contas. Vendas a Prazo. Sistema *Web*. ContaFácil. Micro e Pequenas Empresas.

ABSTRACT

This work presents the design and development of the ContaFácil system to manage customer accounts at the butcher shop Casa das Carnes. The objective is to replace the manual control of credit sales and reduce default, through requirements gathering with the owner, followed by the specification, implementation, and validation of a web application. For this purpose, interviews, process analysis, and screen prototyping were conducted, resulting in the construction of functionalities that centralize customer registration, credit purchase records, and management of outstanding balances. The results demonstrate that the system eliminates paper records, minimizes calculation errors, and provides accurate financial reports, streamlining collection and mainly improving financial transparency. The system promotes greater operational efficiency and can be adapted for other micro and small businesses that adopt credit sales as a practice, highlighting its scalability potential and improvement in credit management.

Keywords: Account Management. Credit Sales. Web System. ContaFácil. Micro and Small Businesses.

SUMÁRIO

1	INTRODUÇÃO	7
2	ENGENHARIA DE SOFTWARE	9
2.1	Engenharia de Requisitos.....	12
2.1.1	Estudo da Viabilidade.....	14
2.1.2	Elicitação e Análise de Requisitos.....	15
2.1.3	Especificação de Requisitos.....	16
2.1.4	Validação de Requisitos.....	16
2.2	Projeto de Software.....	17
2.3	Construção de Software.....	18
2.4	Teste de Software.....	19
3	ESTUDO DE CASO	22
3.1	Desafios Enfrentados pelo Açougue.....	22
3.2	Benefícios Esperados com a Implementação do Sistema.....	23
4	PROPOSTA DE SOLUÇÃO	25
4.1	Detalhamento dos Requisitos.....	25
4.2	Prototipação e Interface.....	25
4.3	Arquitetura do Projeto.....	27
4.4	Banco de Dados.....	28
4.5	Back-end.....	29
4.6	Front-end.....	34
5	CONCLUSÃO	37
	REFERÊNCIAS	39
	APÊNDICE A - Documento de Requisitos do Sistema	42
	APÊNDICE B – Artefatos do Projeto	57
	APÊNDICE C – Casos de Testes	77

1 INTRODUÇÃO

A adoção de tecnologias digitais pelas micro e pequenas empresas tornou-se um fator determinante para a melhoria dos processos internos, otimização de recursos e manutenção da competitividade no contexto de mercado contemporâneo. Segundo Barros (2023), empresas que investem em digitalização de processos podem aumentar sua produtividade em até 30 %, uma vez que a automatização de tarefas repetitivas e o acesso remoto a informações estratégicas reduzem custos operacionais e retrabalho.

O Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE, 2023) reforça que a transformação digital não se resume à aquisição de ferramentas, mas implica mudanças nos processos organizacionais e capacitação de pessoas para utilização efetiva das soluções tecnológicas. Além disso, dados do Instituto Brasileiro de Geografia e Estatística (IBGE, 2023) apontam que mais de 87 % da população brasileira acessou a internet em 2022, sendo que 93,4 % desses usuários o fazem diariamente, o que reforça a necessidade de que mesmo negócios locais adotem recursos digitais para atender melhor o consumidor e gerenciar internamente suas operações.

Muitos estabelecimentos ainda utilizam métodos manuais para o controle financeiro, principalmente no que diz respeito à concessão de crédito informal, conhecido como “fiado”. Em geral, muitos comerciantes locais adotam essa prática para fidelizar clientes em situações de confiança mútua. Embora essa prática auxilie na fidelização e no fluxo de caixa em comunidades com acesso restrito ao crédito formal, traz consigo riscos significativos de inadimplência. O SEBRAE (2022) estima que cerca de 30% da população brasileira encontra-se inadimplente, o que impacta fortemente as finanças de pequenos comércios que mantêm contas a prazo sem controle adequado. Embora a tradição cultural do fiado se justifique por laços de vizinhança e histórico de relacionamento, a ausência de registros sistematizados potencializa perdas e dificulta a tomada de decisão econômica.

No caso específico do açougue Casa das Carnes, observa-se que o controle manual das vendas a prazo tem gerado discrepâncias nos registros, retrabalho no fechamento de contas e insatisfação tanto da equipe de atendimento quanto dos clientes, comprometendo o fluxo de caixa e a sustentabilidade do negócio. Diante disso, é necessário a criação de uma solução tecnológica capaz de suprir essas demandas específicas da empresa. Para tanto, serão

seguidos os passos tradicionais da Engenharia de *Software*, como levantamento de requisitos, projeto, implementação e testes, garantindo que este produto de software atenda às especificações do açougue.

Para alcançar esse propósito, estabeleceu-se como objetivo geral projetar, implementar e validar um sistema *web* que permita ao açougue Casa das Carnes gerenciar as contas dos clientes de forma digital, englobando funcionalidades de cadastro, consulta, registro de vendas a prazo, geração de relatórios financeiros e autenticação segura de usuários.

Este trabalho está organizado em cinco capítulos. O primeiro capítulo apresenta o contexto teórico e prático que justifica a criação de um sistema de gerenciamento de contas, além da definição dos objetivos e do escopo do projeto. No segundo capítulo, são revisados os principais fundamentos da engenharia de *software*, abrangendo processos de desenvolvimento, engenharia de requisitos e modelos de ciclo de vida, com ênfase na justificativa para a adoção do modelo incremental nesta proposta. O terceiro capítulo descreve as particularidades do açougue Casa das Carnes, os desafios enfrentados na gestão do fiado e os benefícios esperados com a automação desse processo. O quarto capítulo detalha a proposta de solução, apresentando a arquitetura adotada, a modelagem do banco de dados, o desenvolvimento das interfaces e as tecnologias utilizadas na construção do sistema. Por fim, o quinto capítulo apresenta a conclusão, comentando os objetivos alcançados, as lições aprendidas ao longo do desenvolvimento, as principais dificuldades superadas e as perspectivas de evolução e aprimoramento da solução desenvolvida.

Complementando o conteúdo principal, este trabalho conta ainda com três apêndices que reúnem a documentação técnica do projeto: o Apêndice A apresenta o documento de requisitos do sistema; o Apêndice B reúne os artefatos do projeto, como diagramas UML e protótipos de tela; e o Apêndice C documenta os casos de teste utilizados para validação do sistema.

2 ENGENHARIA DE SOFTWARE

O *software* é um conjunto de programas e dados que permitem a execução de tarefas em um computador, abrangendo também conteúdos interativos e informações digitais ou impressas (PRESSMAN; MAXIM, 2016). No entanto, quando falamos de engenharia de software, não se trata apenas do programa em si, mas de toda a documentação associada e dos dados de configuração necessários para garantir seu correto funcionamento. Um sistema de *software* desenvolvido profissionalmente é composto por múltiplos programas e arquivos de configuração, além de documentações técnicas e de usuário (SOMMERVILLE, 2011).

A Engenharia de Software reúne processos, métodos e ferramentas voltados à construção sistemática de soluções computacionais eficientes e confiáveis. Segundo Pressman e Maxim (2016, p. 14), "a engenharia de software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade". Ela busca estabelecer práticas e processos que assegurem a construção de software robusto, capaz de atender às demandas dos usuários de maneira eficiente e segura.

A criação de software requer um planejamento rigoroso e uma abordagem estruturada, uma vez que "software está profundamente incorporado em praticamente todos os aspectos de nossas vidas" e sua complexidade só tende a aumentar (PRESSMAN; MAXIM, 2016, n.p.). Para isso, a engenharia de software envolve diversas etapas, como levantamento de requisitos, análise, projeto, implementação, testes e manutenção. Esses processos garantem que o *software* seja desenvolvido com qualidade e possa evoluir ao longo do tempo.

O desenvolvimento de software é uma atividade complexa que envolve não apenas aspectos técnicos, mas também a interação constante entre pessoas, ferramentas e conhecimento. Nesse sentido, construir *software* é um processo de aprendizado social iterativo, no qual o conhecimento disperso e incompleto é gradualmente reunido e incorporado ao produto final. Conforme destacado por Howard Baetjer Jr. (1998, apud PRESSMAN; MAXIM, 2016), esse processo envolve o diálogo entre usuários, projetistas e ferramentas em constante evolução, permitindo que o conhecimento necessário ao sistema seja extraído e organizado durante o desenvolvimento.

Tecnicamente, um processo de software pode ser entendido como o conjunto de etapas organizadas que orientam o desenvolvimento, a manutenção e a evolução de um sistema, desde a concepção até sua entrega final. Segundo Pressman e Maxim (2016), o processo de *software* define a abordagem adotada durante o desenvolvimento e pode ser entendido como parte essencial da engenharia de *software*. Ele não apenas organiza o trabalho, mas também fornece estabilidade e controle a uma atividade que, sem um direcionamento claro, pode se tornar caótica.

Sommerville (2011) complementa essa visão ao definir processo de *software* como um conjunto de atividades relacionadas que resultam na produção de um produto de *software*. Essas atividades podem incluir o desenvolvimento do sistema do zero ou a modificação e integração de sistemas existentes, dependendo do contexto do projeto.

Já o modelo de processo de software atua como uma representação simplificada do processo de desenvolvimento. Ele serve como um guia prático para a execução disciplinada das atividades de engenharia de software. Como afirmam Pressman e Maxim (2016), o modelo de processo define o fluxo de atividades, o grau de iteração, os artefatos esperados e a organização geral do trabalho. Sommerville (2011) ressalta que cada modelo de processo fornece uma perspectiva parcial do desenvolvimento, sendo útil para destacar aspectos específicos e facilitar a compreensão e aplicação do processo.

Existem vários modelos genéricos cada um com suas vantagens e desvantagens. Segundo Sommerville (2011) os mais comuns são:

- O modelo cascata, em que todo o planejamento e programação das atividades é realizada antes de executá-la;
- O modelo incremental em que o sistema é entregue por partes;
- O modelo espiral que combina prevenção e adaptação a mudanças, gerenciando riscos durante o processo.

Não existe um modelo melhor ou pior, a escolha do modelo ideal depende das características e necessidades de cada projeto. Para o desenvolvimento deste projeto, foi adotado o modelo incremental, uma vez que os requisitos do sistema foram previamente definidos por meio de um levantamento estruturado.

Nesse modelo, o sistema é construído e entregue em partes funcionais, ou incrementos, onde cada módulo implementado já possui valor prático e pode ser validado de

forma isolada. Essa abordagem permite que o projeto avance de maneira organizada, facilitando a realização de testes, a identificação de ajustes necessários e a evolução contínua do sistema até sua versão final.

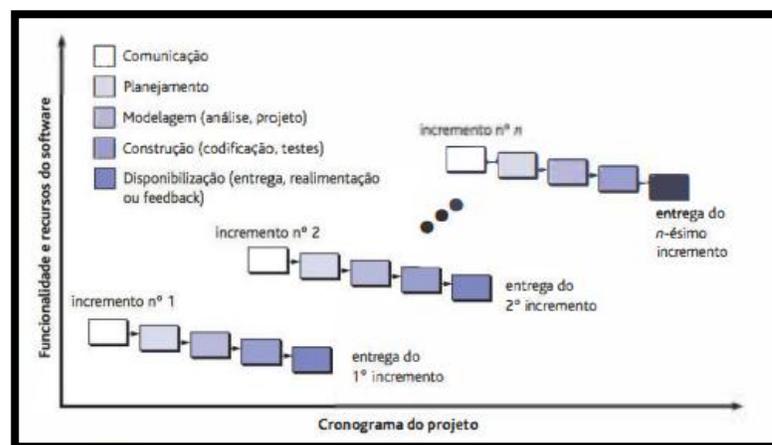
Segundo Sommerville (2011), “o desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido”. Essa abordagem intercala as atividades de especificação, desenvolvimento e validação, permitindo que o *feedback* do cliente seja considerado de forma contínua.

Além disso, o desenvolvimento incremental apresenta diversas vantagens como:

1. O custo de acomodar mudanças nos requisitos é reduzido, pois exige menos retrabalho documental.
2. É mais fácil obter *feedback* dos usuários por meio das versões funcionais entregues, em vez de apenas analisar documentos.
3. Permite a entrega de um sistema funcional em estágios iniciais, proporcionando valor ao cliente mesmo antes da finalização do projeto completo.

A Figura 1 ilustra o modelo incremental, demonstrando como as atividades de especificação, desenvolvimento e validação ocorrem de maneira interativa e simultânea durante os ciclos de construção do sistema. Essa abordagem facilita a organização das etapas de codificação, acelera entregas de partes utilizáveis do sistema e proporciona flexibilidade para eventuais ajustes sem comprometer o andamento do projeto como um todo.

Figura 1 - Modelo Incremental



Autor: Pressman e Maxim, 2016

2.1 Engenharia de Requisitos

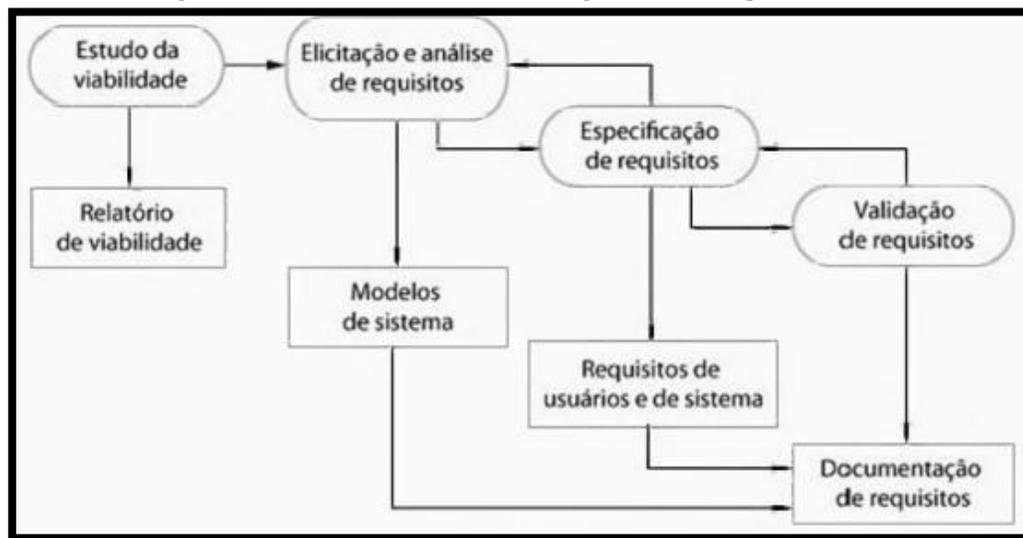
A engenharia de requisitos é uma etapa fundamental no desenvolvimento de *software*, responsável por definir e gerenciar as necessidades e expectativas dos usuários em relação a um sistema. Segundo Sommerville (2011), os requisitos representam descrições das funcionalidades do sistema, dos serviços que ele deve oferecer e das restrições que devem ser observadas.

Essa etapa tem um impacto significativo no desenvolvimento de *software*, afetando custo, cronograma, qualidade e riscos do projeto (YIN et al., 2022). Além disso, é um fator crítico para o sucesso dos projetos, pois um levantamento de requisitos bem estruturado reduz a margem de erro e evita custos desnecessários (HASTIE; WOJEWODA, 2015).

De acordo com Sommerville (2011), “o processo de engenharia de requisitos tem como objetivo produzir um documento de requisitos acordados que especifica um sistema que satisfaz os requisitos dos *stakeholders*”. A engenharia de requisitos envolve quatro atividades principais: estudo de viabilidade, elicitaco, especificaco e validaco dos requisitos.

A Figura 2 ilustra essas etapas de forma clara e sequencial, representando o fluxo tpico do processo de definico de requisitos em projetos de software.

Figura 2 - Atividades do Processo de Engenharia de Requisitos.



Fonte: SOMMERVILLE, 2011

Os requisitos identificados serão documentados de maneira padronizada. Essa documentação seguirá o modelo apresentado no Quadro 1, o que assegura uniformidade na especificação e facilita o entendimento, rastreabilidade e validação dos requisitos ao longo do ciclo de vida do *software*. A estrutura utilizada foi inspirada no padrão IEEE 830-1998, amplamente reconhecido para a elaboração de especificações de requisitos de software, promovendo clareza, organização e consistência na definição dos requisitos.

Quadro 1 - Exemplo de Quadro para Preenchimento dos Requisitos

ID: << Identificação do requisito >>
Nome: << Nome do requisito>>
Importância: << Informar a importância do requisito. Essencial, Desejável ou Opcional>>
Complexidade: << Identifica o requisito por grau de complexidade. Difícil, Médio e Fácil. >>
Prioridade: << Informar se os requisitos são essenciais, importantes ou complementares. >>
Origem do Requisito: << Refere-se à fonte de onde o requisito foi gerado. >>
Descrição: << Longa e detalhada especificação do requisito. >>
Justificativa: << Explicar a necessidade do requisito. >>
Critério de verificação: << Define o que precisa ser testado para confirmar que o requisito foi implementado corretamente. >>
Histórico: << Informar data de criação alteração ou exclusões. >>

Fonte: Autor.

Os requisitos de software podem ser classificados em dois tipos principais, os requisitos funcionais e requisitos não funcionais. Os requisitos funcionais definem as funcionalidades e comportamentos específicos que o sistema deve oferecer. Já os requisitos não funcionais descrevem características gerais do sistema, como desempenho, segurança e usabilidade.

i. Requisitos Funcionais

Os requisitos funcionais definem as funcionalidades que o sistema deve oferecer para atender às necessidades dos usuários. Conforme Sommerville (2011), esses requisitos podem

ser descritos de maneira abstrata para facilitar o entendimento dos usuários ou detalhados como requisitos do sistema, especificando funções, entradas, saídas e exceções.

Para garantir uma especificação eficaz, os requisitos funcionais devem atender a dois critérios essenciais:

- **Completeness:** Todos os serviços necessários devem ser especificados.
- **Consistência:** Não deve haver contradições entre os requisitos.

No entanto, atingir esses critérios em sistemas complexos pode ser desafiador devido à diversidade de *stakeholders* e suas necessidades distintas.

ii. Requisitos Não Funcionais

Os requisitos não funcionais definem características do sistema, como confiabilidade, tempo de resposta, segurança e restrições de implementação (SOMMERVILLE, 2011). Diferentemente dos requisitos funcionais, eles impactam toda a arquitetura do sistema e, quando não atendidos, podem comprometer a usabilidade e viabilidade do software.

Além disso, esses requisitos podem gerar novos requisitos funcionais ou restringir os já existentes (SOMMERVILLE, 2011). Sua correta especificação é fundamental para garantir um sistema eficiente e confiável.

Embora não se refiram diretamente às funcionalidades do *software*, eles desempenham um papel crucial na definição da qualidade e na experiência do usuário, além de garantir que o sistema opere dentro das restrições e exigências externas.

2.1.1 Estudo da Viabilidade

O estudo de viabilidade é a etapa inicial da Engenharia de Requisitos e tem como finalidade avaliar se as necessidades do usuário podem ser atendidas com as tecnologias disponíveis e se o projeto é viável economicamente dentro das restrições orçamentárias.

Segundo Sommerville (2011), o estudo de viabilidade busca identificar se o sistema proposto é viável dentro das restrições orçamentárias existentes e se ele representa uma solução rentável sob a perspectiva do negócio. O estudo deve ser realizado de forma rápida e com custos reduzidos, de modo a orientar a decisão sobre dar continuidade ou não ao desenvolvimento do sistema com uma análise mais aprofundada.

2.1.2 Elicitação e Análise de Requisitos

Essa atividade consiste na identificação e levantamento dos requisitos por meio da observação de sistemas existentes, interação com usuários e análise de tarefas. Essa fase pode envolver a criação de modelos ou protótipos para melhor compreensão do sistema (SOMMERVILLE, 2011).

Diversas técnicas podem ser utilizadas para a elicitación de requisitos, como entrevistas, questionários, prototipagem e brainstorming. Conforme descrito no *SWEBOOK (Software Engineering Body of Knowledge)* (IEEE, 2014), essas técnicas fazem parte do conjunto de práticas recomendadas para identificação, análise e validação de requisitos, sendo escolhidas conforme o contexto e os stakeholders envolvidos. Neste projeto, foram aplicadas três abordagens principais para compreender as necessidades do sistema e garantir que ele atenda de forma eficaz às expectativas da empresa:

1. **Etnografia:** Foi realizada a observação direta do processo manual atual de registro de compras a prazo. Durante alguns dias, acompanhei o funcionário responsável pelo caixa enquanto ele registrava as transações em um bloco de notas. Essa imersão permitiu entender o fluxo de trabalho real, identificar dificuldades no processo e levantar desafios que poderiam passar despercebidos sem essa análise direta.
2. **Entrevistas:** Foram conduzidas conversas com o proprietário do açougue e o funcionário do caixa. Utilizamos um roteiro de perguntas para identificar dificuldades, necessidades e expectativas em relação ao novo sistema. Além disso, abrimos espaço para que eles compartilhassem experiências e sugestões, permitindo um entendimento mais aprofundado das melhorias necessárias.
3. **Brainstorming:** Foi realizada uma reunião colaborativa com o proprietário do açougue para discutir ideias e definir melhorias no sistema. Durante a conversa, exploramos possíveis funcionalidades, analisamos os principais desafios levantados nas etapas anteriores e priorizamos soluções que poderiam otimizar a operação de registro de compras a prazo.

Essas técnicas permitiram a elicitación de requisitos mais precisos e alinhados à realidade do negócio. Posteriormente, os requisitos foram organizados e formalizados no Documento de Requisitos do Sistema (conforme Apêndice A), o qual serviu como referência para as etapas seguintes do desenvolvimento e facilitou a implementação da solução.

2.1.3 Especificação de Requisitos

Após a etapa de elicitación, os requisitos levantados precisam ser organizados e documentados de maneira clara e compreensível. A especificação de requisitos consiste justamente nesse processo de documentação, sendo essencial para garantir o entendimento comum entre os desenvolvedores, usuários e demais *stakeholders* do projeto.

Segundo Sommerville (2011), essa especificação deve contemplar tanto os requisitos do usuário que descrevem, de forma mais geral, as necessidades e expectativas dos envolvidos quanto os requisitos do sistema, que fornecem uma descrição detalhada das funcionalidades esperadas, bem como das restrições técnicas que devem ser observadas.

No contexto deste trabalho, os requisitos identificados para o sistema proposto foram organizados em um documento estruturado de especificação de requisitos. Esse documento apresenta inicialmente uma introdução geral sobre a aplicação e o problema a ser resolvido, seguida pela definição dos objetivos do documento, do seu público-alvo, da lista das funcionalidades principais e incluídas informações de suporte, como termos e definições relevantes e o modelo utilizado para especificar os requisitos. Em seguida, são detalhados os requisitos funcionais e não funcionais, utilizando um modelo padronizado. Por fim, uma matriz de rastreabilidade foi elaborada para facilitar o acompanhamento da implementação de cada requisito ao longo do projeto.

A especificação completa pode ser consultada no Apêndice A e servirá como base para o desenvolvimento e validação do sistema, assegurando que ele atenda às necessidades levantadas durante as fases iniciais do projeto.

2.1.4 Validação de Requisitos

A validação de requisitos é a etapa responsável por garantir que os requisitos sejam coerentes, completos e realistas em relação às necessidades do sistema e às limitações do projeto. De acordo com Sommerville (2011), é nesse momento que se verifica se os requisitos realmente refletem as necessidades dos usuários e se são tecnicamente viáveis. Durante o processo de validação, é comum que erros ou inconsistências sejam identificados no documento de requisitos, o que exige revisões e correções antes que o desenvolvimento prossiga.

Essa etapa contribui diretamente para a qualidade do sistema, pois permite detectar e corrigir falhas ainda nas fases iniciais do projeto, reduzindo custos e retrabalho nas etapas posteriores.

2.2 Projeto de Software

O projeto de software é fundamental para que o desenvolvimento do software ocorra com qualidade. Ela começa após os requisitos terem sido analisados e modelados. Segundo Pressman e Maxim (2016, p. 227) “O projeto é a única maneira pela qual podemos transformar precisamente os requisitos dos envolvidos em um produto ou sistema de software finalizado”.

O Projeto Orientado a Objetos com *UML* organiza o software como um conjunto de objetos interagindo, cada um com responsabilidades e comportamentos bem definidos. Essa abordagem facilita a compreensão, manutenção e evolução dos sistemas, além de promover o reuso de código (SOMMERVILLE, 2011). A *UML (Unified Modeling Language)*, uma notação gráfica padronizada, é amplamente utilizada para modelar diferentes aspectos de um sistema orientado a objetos. A *UML* é essencial para planejar e visualizar sistemas de forma eficiente, utilizando diagramas que descrevem tanto a estrutura quanto o comportamento do sistema.

O uso de *UML* garante uma comunicação clara entre todos os envolvidos no projeto, como analistas, desenvolvedores e *stakeholders*. Para este projeto, foi utilizado um diagrama de classes *UML*, que apresenta as classes, seus métodos e atributos, conforme detalhado no Apêndice B deste trabalho. Nesse apêndice inclui:

- Diagrama de Classes;

- Diagrama de Relacionamento;
- Diagrama de Casos de Uso Geral;
- Projeto de Interface, com protótipos das telas do sistema.

O Diagrama de Classes é um dos principais diagramas na modelagem de sistemas orientados a objetos. Ele descreve a estrutura estática do sistema, representando as classes e os relacionamentos entre elas. Cada classe é definida por seus atributos e métodos. O Diagrama de Classes é crucial para entender como os objetos serão organizados dentro do sistema e como irão interagir (SOMMERVILLE, 2011).

Já o Diagrama de Relacionamento é utilizado para modelar a estrutura de dados do sistema, representando as entidades do domínio e os relacionamentos entre elas. Ele é fundamental para entender como os dados serão organizados e manipulados, sendo especialmente útil durante o processo de construção e manutenção do banco de dados (BELCIC e STRYKER, 2024).

O Diagrama de Casos de Uso serve para ilustrar as principais funcionalidades que o sistema deve oferecer e como diferentes atores interagem com essas funcionalidades. O objetivo principal deste diagrama é identificar os serviços que o sistema deve fornecer, compreendendo quem usará o sistema e suas expectativas (IBM, 2021).

Além dos diagramas, o apêndice B inclui também o Projeto de Interfaces ele se concentra na criação da camada visual e interativa do sistema. O objetivo principal é garantir que a interface seja eficiente, intuitiva e agradável para o usuário final. Para este projeto, as interfaces foram prototipadas utilizando a ferramenta Figma.

2.3 Construção de Software

O ContaFácil foi idealizado como uma aplicação *web*, acessada diretamente por navegadores, sem necessidade de instalação local. Essa abordagem foi escolhida por sua praticidade, baixo custo de manutenção e acessibilidade multiplataforma. Como o público-alvo são pequenos comerciantes que muitas vezes utilizam computadores compartilhados ou dispositivos móveis, a aplicação *web* elimina barreiras de compatibilidade, facilita atualizações centralizadas e permite o uso imediato em qualquer local com conexão à internet.

O desenvolvimento do ContaFácil envolveu a utilização de diversas ferramentas e tecnologias que, em conjunto, proporcionaram uma solução robusta e eficiente. As principais ferramentas e tecnologias empregadas incluem:

- **Java:** Linguagem de programação orientada a objetos amplamente usada em aplicações corporativas por sua portabilidade e robustez (Oracle, 2025).
- **Spring Boot:** Framework utilizado para o desenvolvimento do *backend*, facilitando a criação de aplicações Java prontas para produção com configuração mínima (Spring Boot, 2025).
- **PostgreSQL:** Banco de dados relacional de código aberto conhecido pela conformidade com ACID e confiabilidade (POSTGRESQL, 2025).
- **Padrão MVC:** O padrão adotado na maioria dos sistemas desenvolvidos para *web* (SOMMERVILLE, 2011).
- **HTML:** Linguagem de marcação padrão usada para estruturar páginas na *web* (W3C, 2025).
- **CSS:** Linguagem de estilo utilizada para definir a aparência visual de páginas HTML (W3C, 2025).
- **TypeScript:** Linguagem baseada em JavaScript que adiciona tipagem estática e melhora a produtividade no desenvolvimento (Microsoft, 2025).
- **Angular 17:** Framework para desenvolvimento do *front-end*, oferecendo melhorias de performance e estrutura modular (Google, 2025).
- **Bootstrap:** Biblioteca de componentes *front-end* que facilita o desenvolvimento de interfaces responsivas e consistentes (Bootstrap Team, 2025).
- **Git e GitHub:** Ferramentas para controle de versão e colaboração no desenvolvimento de *software* (Chacon e Straub, 2014).

2.4 Teste de Software

O teste de software é uma atividade essencial no processo de desenvolvimento, tendo como objetivo verificar se o programa faz o que se propõe a fazer e descobrir defeitos antes que ele entre em uso. Trata-se de uma etapa em que o *software* é executado com dados

fictícios, e os resultados obtidos são analisados em busca de erros, anomalias ou falhas relacionadas aos atributos funcionais e não funcionais do sistema (SOMMERVILLE, 2011).

Esse processo tem dois objetivos principais:

1. Demonstrar conformidade com os requisitos. No caso de softwares personalizados, isso implica a criação de ao menos um teste para cada requisito especificado. Já para softwares genéricos, é necessário contemplar testes para todas as funcionalidades e suas possíveis combinações a serem incluídas no produto final (SOMMERVILLE, 2011).
2. Detectar comportamentos incorretos ou indesejáveis. Isso envolve identificar falhas que causem panes, interações não previstas com outros sistemas, processamentos incorretos ou corrupção de dados (SOMMERVILLE, 2011).

Apesar de sua importância, o teste de software não é capaz de garantir a ausência total de defeitos. Sempre haverá a possibilidade de um cenário de teste não coberto revelar erros desconhecidos. Como destacado por Edsger Dijkstra, "os testes podem mostrar apenas a presença de erros, e não sua ausência" (DIJKSTRA et al., 1972 apud SOMMERVILLE, 2011).

O teste faz parte do processo mais amplo de verificação e validação (V&V), que frequentemente são confundidos, mas possuem propósitos distintos. Segundo Barry Boehm, "Validação: estamos construindo o produto certo?" e "Verificação: estamos construindo o produto da maneira certa?" (BOEHM, 1979 apud SOMMERVILLE, 2011). Esses processos são aplicados desde os estágios iniciais, a partir da definição dos requisitos, e acompanham todo o ciclo de desenvolvimento, buscando garantir que o software atenda às especificações e expectativas dos usuários.

No contexto deste trabalho, foi adotada a abordagem de teste funcional. Esse tipo de teste simula uma experiência completa no nível do usuário, utilizando ferramentas que reproduzem o comportamento real de um usuário humano. O foco dos testes funcionais está nos requisitos de negócio da aplicação, ou seja, eles avaliam se a funcionalidade está de acordo com o que foi especificado, considerando apenas as entradas fornecidas e as saídas produzidas, sem examinar os estados internos ou a lógica de implementação do sistema (PITTET, 2025).

Todos os casos de teste elaborados estão documentados no Apêndice C – Casos de Testes e foram organizados de forma estruturada, com o objetivo de garantir clareza, rastreabilidade e reprodutibilidade dos testes. Cada caso de teste foi composto pelos seguintes campos:

- Identificação: um código único que identifica o caso de teste.
- Objetivo: descrição clara do que o teste pretende verificar.
- Data e hora: registro do momento exato em que o teste foi realizado.
- Responsável: nome da pessoa que executou o teste.
- Procedimento inicial: ações iniciais para preparar ou acessar o objeto a ser testado.
- Passos: sequência detalhada de ações executadas durante o teste.
- Resultado esperado: o comportamento que o sistema deve apresentar ao fim do teste.
- Resultado real: o que de fato foi observado ao executar o teste.
- Evidências: capturas de tela, *logs* ou outros elementos que comprovem os resultados obtidos.

3 ESTUDO DE CASO

A venda a prazo, popularmente conhecida como "fiado", é uma prática tradicional em pequenos comércios, especialmente em bairros e comunidades onde a relação de confiança entre comerciante e cliente é valorizada. Apesar de sua popularidade, essa prática apresenta riscos significativos para a saúde financeira dos empreendimentos, principalmente quando realizada de forma informal.

Segundo o Blog do SEBRAE MS (2019), a venda a prazo é um dos grandes desafios enfrentados pelos micros e pequenos empreendedores, exigindo atenção especial à gestão financeira, à organização dos registros e ao controle do capital de giro. A ausência de mecanismos eficazes de controle pode gerar inadimplência, desorganização e comprometimento da sustentabilidade do negócio.

Com base nesse cenário, este capítulo apresenta o estudo de caso realizado no açougue Casa das Carnes, que serviu de base para o desenvolvimento da solução tecnológica proposta. A seguir, são descritos os desafios enfrentados pelo estabelecimento e os benefícios esperados com a aplicação do sistema ContaFácil.

3.1 Desafios Enfrentados pelo Açougue

Os principais desafios enfrentados por empresas que adotam as vendas a prazo estão diretamente ligados à falta de controle e à informalidade nos registros. Essa realidade foi observada no açougue Casa das Carnes, que foi utilizado como base para o presente estudo de caso.

No estabelecimento analisado, as vendas a prazo eram registradas manualmente, com anotações feitas em cadernos ou blocos. Esse método resultava em diversos problemas, como:

- **Inadimplência:** a ausência de garantias e o controle informal dos pagamentos a prazo elevam o risco de prejuízos (CREDIADO, 2024).
- **Descontrole financeiro:** a imprevisibilidade nos recebimentos dificulta o planejamento e compromete o capital de giro (QUALIFICA, 2018).

- **Perda de registros:** as anotações físicas eram frequentemente perdidas, danificadas ou esquecidas, gerando confusão;
- **Conflitos pessoais:** a cobrança direta pode gerar atritos nas relações com clientes, principalmente quando há vínculos de amizade ou parentesco (QUALIFICA, 2018).

Esse cenário dificultava a tomada de decisões e comprometia o funcionamento saudável do negócio. Além disso, impedia o comerciante de ter uma visão clara dos valores a receber e de quais clientes estavam inadimplentes.

Além dos desafios operacionais observados, é importante considerar a natureza do ambiente em que o açougue está inserido. O Casa das Carnes atua em um bairro residencial com forte presença de clientes fixos e relações de confiança, o que reforça o uso frequente da venda a prazo como mecanismo de fidelização. No entanto, essa prática, quando não gerida adequadamente, compromete a previsibilidade de receitas e a tomada de decisões estratégicas, como reposição de estoque, renegociação com fornecedores e controle de fluxo de caixa.

Outro fator relevante é o perfil do usuário do sistema: trata-se, em geral, de funcionários com pouca familiaridade com soluções tecnológicas, o que exige uma interface simples, acessível e orientada a tarefas. A informalidade nos registros anteriores também dificultava o acompanhamento das pendências dos clientes por diferentes funcionários, especialmente quando ocorriam trocas de turno ou ausência de quem havia realizado o lançamento da venda.

3.2 Benefícios Esperados com a Implementação do Sistema

Diante desses problemas, evidenciou-se a necessidade de uma solução tecnológica simples, acessível e adaptada à realidade de microempreendedores que ainda dependem das vendas a prazo como parte importante de sua estratégia comercial.

Com base nos desafios enfrentados pelo estabelecimento analisado, foi idealizada uma solução tecnológica voltada à organização das contas dos clientes. Essa solução deu origem ao sistema ContaFácil, uma ferramenta *web* desenvolvida com o objetivo de registrar

digitalmente as vendas a prazo, organizar contas de clientes e gerar relatórios financeiros de forma simples e acessível.

A implementação de um sistema informatizado no açougue Casa das Carnes visa proporcionar ganhos significativos em organização, agilidade e controle financeiro. Espera-se, inicialmente, a eliminação de registros físicos manuais, que são suscetíveis a extravios, rasuras e inconsistências. Com a digitalização das informações, a equipe poderá acessar rapidamente os dados de cada cliente, facilitando a cobrança, a renegociação de dívidas e a tomada de decisão baseada em relatórios confiáveis.

Outro benefício esperado é a redução de conflitos com os clientes, uma vez que o sistema permitirá registrar cada venda fiada de forma estruturada e com histórico acessível. Isso contribui para maior transparência e confiabilidade no relacionamento comercial, além de agilizar o atendimento no momento do pagamento ou consulta de débitos.

Adicionalmente, a adoção do sistema cria uma base sólida para que o açougue possa futuramente adotar práticas mais modernas de gestão, como envio de lembretes automáticos de cobrança, segmentação de clientes por perfil de compra, ou mesmo integração com soluções de pagamento digital. Ainda que essas funcionalidades estejam fora do escopo da versão inicial, a arquitetura modular do sistema permitirá sua evolução progressiva conforme as demandas do negócio.

4 PROPOSTA DE SOLUÇÃO

A partir da análise apresentada no capítulo anterior, foi concebida a proposta de uma aplicação *web* denominada ContaFácil, voltada à gestão de vendas a prazo realizadas em comércios de pequeno porte. O sistema foi projetado para permitir o controle eficaz das contas dos clientes, com funcionalidades simples, objetivas e acessíveis a usuários com pouca familiaridade com tecnologia.

A proposta visa substituir os métodos manuais de anotação por uma solução digital intuitiva, promovendo maior controle financeiro, segurança das informações e facilidade no acompanhamento das transações a prazo.

4.1 Detalhamento dos Requisitos

O levantamento de requisitos do sistema ContaFácil foi realizado por meio de três técnicas complementares: observação direta das rotinas do açougue Casa das Carnes, entrevistas semiestruturadas e *brainstorming*. Durante os dias de observação, identificaram-se gargalos no fluxo de caixa e nos processos manuais e pontos de falha na conferência de saldos. Nas reuniões de entrevista e *brainstorming*, foram elencados requisitos relacionados ao controle de crédito e notificações.

A especificação dos requisitos seguiu um modelo padronizado, contemplando identificador, descrição, pré e pós-condições, e critérios de aceitação. O documento completo de requisitos, com definição detalhada de cada item e seus atributos, está disponível no Apêndice A. Para validar esses requisitos, foi realizado reuniões com o proprietário, no qual os critérios de aceitação foram ajustados conforme o *feedback* obtido.

4.2 Prototipação e Interface

Todos os protótipos das telas do “ContaFácil” foram feitos utilizando a ferramenta Figma (conforme Apêndice B). O Figma é uma plataforma de design de interfaces e prototipação baseada em nuvem, que permite a criação colaborativa de *layouts* para aplicações *web* e mobile sem a necessidade de instalação de *software*, funcionando

diretamente no navegador ou em aplicativos *desktop* e *mobile* do próprio Figma. Com ferramentas de edição vetorial, recursos de prototipagem interativa e suporte a componentes reutilizáveis, o Figma agiliza o processo de *design*, teste de fluxos de navegação e compartilhamento de artefatos com desenvolvedores e *stakeholders*, tudo em um único ambiente acessível em diferentes plataformas (Figma, 2019).

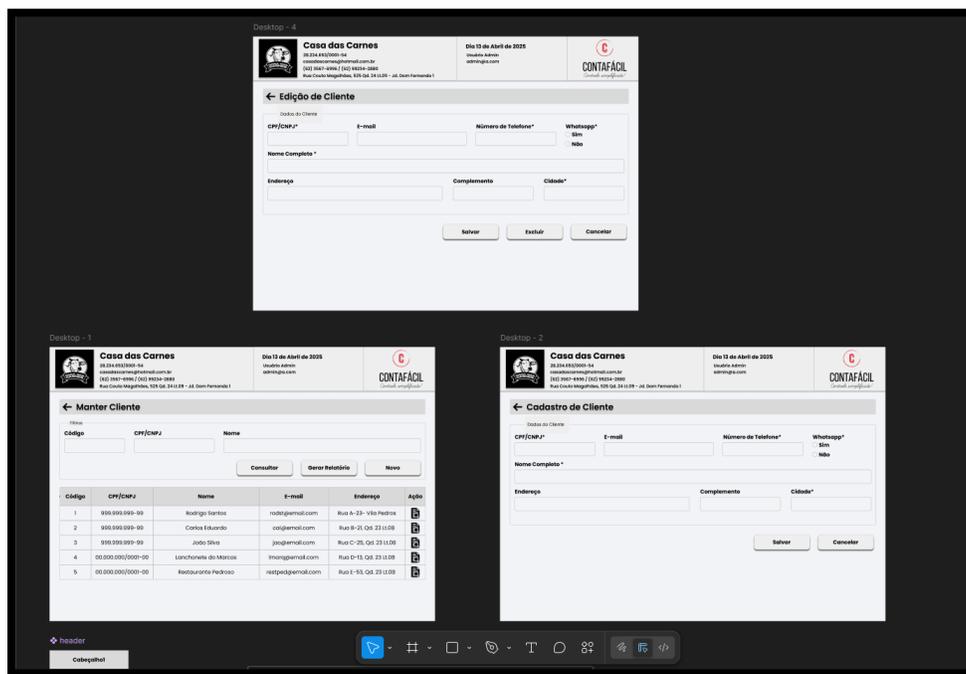
Dentre as vantagens do Figma está a integração nativa entre design e prototipação, reduz consideravelmente o retrabalho e acelera a entrega de design para desenvolvimento, enquanto o *Auto Layout* e as bibliotecas de componentes asseguram consistência visual e responsividade em diferentes resoluções, tornando o fluxo de criação mais rápido e organizado (Figma, 2019).

A utilização dessa ferramenta permitiu desenhar todas as interfaces (login, cadastro de clientes, cadastro de contas, painel financeiro etc.) antes da implementação de código, conforme registrado no Apêndice B. Com isso, conseguimos antecipar e eliminar diversos pontos de fricção na navegação, como fluxos excessivamente longos ou botões mal posicionados, antes que se tornassem problemas de código, garantindo que cada interação fluísse de forma natural. Esse processo incluiu a apresentação dos protótipos para o proprietário do açougue Casa das Carnes, o que possibilitou coletar *feedback* concreto sobre nomenclaturas, agrupamento de campos e ordem das telas, alinhando o produto às expectativas reais de uso.

Com as telas validadas, o retrabalho foi drasticamente reduzido: ajustes de layout e fluxo que teriam demandado horas de recodificação foram feitos rapidamente no protótipo, sem impacto na base de código já desenvolvida. Além disso, a definição de bibliotecas de componentes e estilos no próprio Figma assegurou a padronização visual e comportamental, desde cores e tipografia até estados de botões e mensagens de erro, promovendo coesão em todas as páginas e simplificando tanto a manutenção futura quanto a colaboração entre design e desenvolvimento.

A Figura 3 apresenta os protótipos das interfaces da funcionalidade "Manter Cliente" da aplicação ContaFácil. Outros protótipos de interface, assim como o link para acessá-los diretamente na ferramenta de prototipação, estão disponíveis no Capítulo 2 do Apêndice B – Artefatos do Projeto.

Figura 3 – Protótipo com Figma



Fonte: Autor

4.3 Arquitetura do Projeto

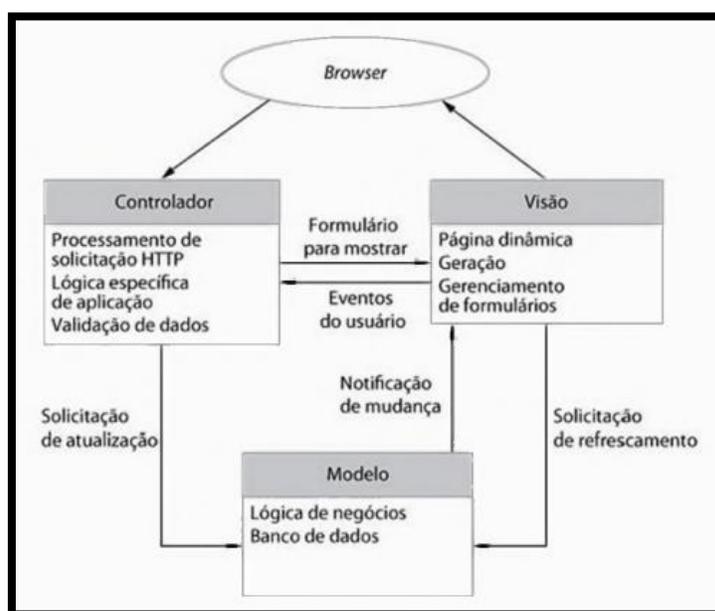
A arquitetura adotada na construção do ContaFácil foi o padrão *MVC (Model-View-Controller)*, uma abordagem amplamente utilizada em aplicações *web* por oferecer uma separação clara entre os componentes responsáveis pela lógica de negócios, interface do usuário e controle de fluxo da aplicação.

Segundo Sommerville (2011) o padrão MVC divide a aplicação em três componentes principais:

1. Modelo (*Model*): Responsável por gerenciar os dados da aplicação, bem como a lógica de negócios e o acesso ao banco de dados.
2. Visão (*View*): Trata da apresentação dos dados ao usuário, ou seja, como as informações serão exibidas na interface.
3. Controlador (*Controller*): Atua como intermediário entre o modelo e a visão, sendo responsável por interpretar as interações do usuário (como cliques e preenchimento de formulários) e atualizar os dados ou a visualização conforme necessário.

A Figura 4 ilustra de forma clara a dinâmica dessa arquitetura em aplicações *web*, mostrando como o navegador se comunica com os três componentes principais para realizar a renderização da interface, processar eventos do usuário e manter os dados atualizados.

Figura 4 - Arquitetura de aplicações *web* usando o padrão MVC



Fonte: SOMERVILLE, 2011

A adoção do padrão MVC permite um desenvolvimento mais organizado e modular, facilitando a manutenção, os testes e futuras evoluções do sistema. Uma das principais vantagens dessa abordagem é a possibilidade de alterar a visualização dos dados sem afetar diretamente a lógica de negócios ou vice-versa, além de permitir múltiplas formas de visualização de um mesmo conjunto de informações (SOMMERVILLE, 2011).

4.4 Banco de Dados

O desenvolvimento do ContaFácil utilizou o sistema de gerenciamento de banco de dados relacional PostgreSQL. A escolha por essa tecnologia deve-se, em parte, ao conhecimento prévio adquirido durante as disciplinas do curso de graduação, onde o

PostgreSQL foi utilizado em atividades práticas. Essa familiaridade proporcionou maior agilidade na modelagem e implementação do banco de dados da aplicação.

Além disso, o PostgreSQL é reconhecido por sua robustez, conformidade com os padrões SQL e suporte completo às propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), essenciais para garantir a integridade e confiabilidade dos dados em sistemas transacionais (POSTGRESQL, 2025). Sua arquitetura extensível permite a criação de tipos de dados personalizados, funções definidas pelo usuário e outros recursos que facilitam a adaptação do banco de dados às necessidades específicas da aplicação (POSTGRESQL, 2025).

O diagrama de entidade-relacionamento que detalha as tabelas cliente, conta e compra, bem como seus relacionamentos e restrições de integridade, encontra-se no capítulo 4 do Apêndice B deste trabalho. Nesse modelo, cada cliente pode manter várias contas (relação 1:N) e cada conta pode agrupar diversas compras (relação 1:N). Essa modelagem foi adotada para espelhar a realidade do açougue, em que um cliente pode acumular múltiplas dívidas (contas) e cada conta registra as compras a prazo correspondentes. Chaves estrangeiras adequadas foram definidas para manter a integridade referencial, garantindo que contas e compras permaneçam vinculadas a clientes e contas válidos, respectivamente.

4.5 Back-end

No ContaFácil, o *back-end* é responsável por implementar toda a lógica de negócios, orquestrar o acesso aos dados e expor serviços que podem ser consumidos pelo front-end ou por clientes externos via *API (Application Programming Interface)*. Para isso, foram adotadas tecnologias como *Java, API REST, Spring Boot, Spring Security e JSON Web Tokens (JWT)*.

Java é uma linguagem de programação orientada a objetos, compilada para *bytecode* e executada na *Java Virtual Machine (JVM)*, o que assegura portabilidade entre diferentes sistemas operacionais e plataformas (Oracle, 2025). Ele foi escolhido para o desenvolvimento do *back-end*, por ser uma linguagem madura, multiplataforma e de forte tipagem, com uma grande comunidade ativa e ecossistema de bibliotecas corporativas.

O Spring é um framework modular para aplicações Java que simplifica aspectos como injeção de dependência e programação orientada a aspectos (Spring Boot, 2025). *O Spring Boot*, por sua vez, adiciona configuração automática (“*convention over configuration*”) e servidor *web* embutido (Tomcat), reduzindo significativamente o esforço inicial (Spring Boot, 2025). No ContaFácil, o Spring Boot gerencia *beans* de serviços e repositórios, configura o contêiner *web* e integra o Spring Data JPA para acesso aos dados.

A figura 5 apresenta a classe principal da aplicação, anotada com `@SpringBootApplication`, responsável por inicializar o contexto do Spring e disparar o servidor Tomcat embutido por meio de “`SpringApplication.run()`”.

Figura 5 – Classe Principal do Back-end

```
1 package com.example.conta_facil_back_end;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ContaFacilBackEndApplication {
8
9     public static void main(final String[] args) {
10         SpringApplication.run(ContaFacilBackEndApplication.class, args);
11     }
12 }
13 }
```

Fonte: Autor

Para simplificar o mapeamento objeto-relacional, o *Spring Data JPA* abstrai grande parte das operações *CRUD* (*Create, Read, Update e Delete*) e dá suporte a consultas derivadas de nomes de métodos. Sua integração transparente com o *Hibernate* permite manter o foco na modelagem de entidades e regras de domínio, sem sacrificar a flexibilidade em consultas complexas.

A arquitetura *REST* (*Representational State Transfer*) define princípios para criação de serviços *web* escaláveis e independentes de estado, baseados em recursos acessíveis por *URLs* e operações *HTTP* padronizadas (*GET, POST, PUT, DELETE*) (MDN, 2025). No ContaFácil, foi adotada uma *API REST* para disponibilizar *endpoints* que permitem criar,

consultar, atualizar e remover registros de clientes, contas e compras, facilitando a integração com o front-end em Angular.

A figura 6 exemplifica um dos *controllers REST* do ContaFácil, onde vemos:

- A anotação “`@RestController`”, que instrui o Spring a tratar a classe como um ponto de recepção de requisições *HTTP* e a serializar automaticamente as respostas em *JSON*.
- O mapeamento “`@RequestMapping("/api/clientes")`” na classe, definindo a rota base para operações sobre clientes.
- O método anotado com “`@PostMapping`”, responsável por receber um objeto cliente via corpo da requisição e delegar sua persistência ao serviço.

Figura 6 – Classe Controller REST da Entidade Cliente

```
1 package com.example.conta_facil_back_end.mantercliente;
2
3 > import ...
17
18 @RestController  ⚡ Mateus Almeida
19 @RequestMapping(⊕"/api/cliente")
20 public class ClienteAPI {
21
22     private final ClienteService clienteService; 7 usages
23
24     public ClienteAPI(final ClienteService clienteService) { ⚡ Mateus Almeida
25         this.clienteService = clienteService;
26     }
27
28     @PostMapping(⊕"/novo") ⚡ Mateus Almeida
29     public ResponseEntity<Void> criarCliente(@RequestBody final ClienteDTO clienteDTO) {
30         this.clienteService.inserirCliente(clienteDTO);
31         return new ResponseEntity<>(HttpStatus.OK);
32     }
33
```

Fonte: Autor

Para proteger os recursos expostos, o ContaFácil integra o *Spring Security*, que fornece mecanismos de autenticação e autorização, além de proteção contra ataques comuns como *CSRF* (*Cross-Site Request Forgery*) e força bruta de credenciais (OWASP, 2025). A autenticação é feita via formulário de login que emite tokens *JWT* para acesso subsequente.

A estratégia adotada elimina sessões de servidor (“*stateless*”), permitindo que apenas requisições com *JWT* válidos sejam atendidas, e disponibiliza livre acesso apenas ao *endpoint* de autenticação.

A Figura 7 mostra a classe responsável por configurar a segurança da aplicação utilizando o *Spring Security*, onde são definidos: a desativação do *CSRF*, a política de criação de sessão como *STATELESS*, a configuração de *CORS*, a permissão pública ao caminho “/auth/login” e a exigência de autenticação para todas as outras requisições, além da injeção do filtro de segurança que intercepta e valida o token *JWT* antes do processamento padrão do *Spring Security*.

Figura 7 – Classe SecurityConfig

```

1 package com.example.conta_facil_back_end.infra.security;
2
3 > import ...
4
5 @Configuration
6 @EnableWebSecurity
7 public class SecurityConfig {
8
9     private final CustomUserDetailsService userDetailsService;
10
11     final 2 usages
12     SecurityFilter securityFilter;
13
14     public SecurityConfig( new *
15         final CustomUserDetailsService userDetailsService,
16         final SecurityFilter securityFilter) {
17
18         this.userDetailsService = userDetailsService;
19         this.securityFilter = securityFilter;
20     }
21
22 @Bean
23 public SecurityFilterChain securityFilterChain(final HttpSecurity http, final CorsConfigurationSource corsConfigSource) throws Exception {
24     http
25         .csrf( CsrfConfigurer<HttpSecurity> csrf -> csrf.disable())
26         .sessionManagement( SessionManagementConfigurer<HttpSecurity> session -> session.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
27         .cors( CorsConfigurer<HttpSecurity> cors -> cors.configurationSource(corsConfigSource))
28         .authorizeHttpRequests( AuthorizationManagerRequestMat... authorize -> authorize
29             .requestMatchers(HttpMethod.POST, @"/auth/login").permitAll()
30             .anyRequest().authenticated()
31         )
32         .addFilterBefore(this.securityFilter, UsernamePasswordAuthenticationFilter.class);
33     return http.build();
34 }
35
36 @Bean
37 public AuthenticationManager authenticationManager(final AuthenticationConfiguration authenticationConfiguration) throws Exception {
38     return authenticationConfiguration.getAuthenticationManager();
39 }
40
41 }

```

Fonte: Autor

Os tokens *JWT* são um padrão aberto (RFC 7519) para transmissão segura de informações em formato *JSON*, permitindo comunicação sem estado e escalabilidade horizontal (JWT.IO, 2025). No ContaFácil, após autenticação, o servidor emite um *JWT*

contendo as credenciais e permissões do usuário, que é enviado no cabeçalho *Authorization* em cada requisição subsequente.

A Figura 8 mostra a classe *TokenService*, responsável pela geração e validação dos tokens *JWT* no ContaFácil. Nela, é injetada a chave secreta configurada no arquivo de propriedades da aplicação, utilizada para assinar e verificar a autenticidade dos tokens. O método “generateToken” cria o token usando o algoritmo HMAC256, definindo o emissor, o assunto e a data de expiração. E o método “validateToken” reconstrói o mesmo algoritmo, verifica a assinatura digital e retorna o assunto do token caso seja válido.

Figura 8 – Classe TokenService

```

15
16 @Service 5 usages  ↳ Mateus de Souza Almeida +1
17 public class TokenService {
18
19     @Value("${api_security.jwt_secret}")
20     private String secret;
21
22
23     public String generateToken(final Usuario usuario) { 2 usages  ↳ Mateus de Souza Almeida +1
24         try {
25             final Algorithm algorithm = Algorithm.HMAC256(this.secret);
26
27             return JWT.create()
28                 .withIssuer("login-auth-api")
29                 .withSubject(usuario.getEmail())
30                 .withExpiresAt(this.generateExpirationDate())
31                 .sign(algorithm);
32         } catch (final JWTCreationException e) {
33             throw new RuntimeException("Erro ao gerar token");
34         }
35     }
36
37     private Instant generateExpirationDate() { return LocalDateTime.now().plusHours(2).toInstant(ZoneOffset.of(-03:00)); }
38
39
40
41     public String validateToken(final String token) { 1 usage  ↳ Mateus de Souza Almeida
42         try {
43             final Algorithm algorithm = Algorithm.HMAC256(this.secret);
44             return JWT.require(algorithm).verification()
45                 .withIssuer("login-auth-api")
46                 .build().JWTVerifier
47                 .verify(token).DecodedJWT
48                 .getSubject();
49         } catch (final JWTVerificationException e) {
50             return null;
51         }
52     }
53
54
55 }

```

Fonte: Autor

O ContaFácil realiza o tratamento de dados pessoais como nome, CPF ou CNPJ, telefone, e-mail e endereço dos clientes, necessários para o controle das contas a prazo. O sistema aplica medidas de segurança como autenticação via JWT, controle de acesso com Spring Security e armazenamento criptografado de senhas. O tratamento desses dados se apoia no legítimo interesse do estabelecimento, voltado à organização financeira e ao controle de contas dos clientes. Embora a gestão de perfis ainda não estejam implementados, essas funcionalidades estão previstas como possíveis evoluções futuras do sistema.

4.6 Front-end

No desenvolvimento front-end do sistema ContaFácil, foram utilizadas tecnologias modernas que asseguram semântica, estilo, tipagem e desempenho, resultando em uma interface acessível, responsiva e de fácil manutenção. Entre as tecnologias empregadas destacam-se: *HTML*, *CSS*, *Bootstrap*, *TypeScript* e *Angular 17*.

O *HTML* foi utilizado para estruturar o conteúdo das páginas. Para estilizar e posicionar os elementos da interface, empregamos o *CSS3*, complementado pelo *framework* *Bootstrap 5*. O *CSS3* permite a criação de *layouts* responsivos por meio de *media queries* e *grid layouts* (MDN, 2025). Já o *Bootstrap* oferece um sistema de grid baseado em *flexbox* e componentes pré-estilizados, agilizando o desenvolvimento de interfaces consistentes e adaptáveis a diferentes tamanhos de tela. Essa abordagem de design responsivo é muito importante, pois o monitor utilizado no açougue possui um tamanho diferente da que foi usada durante o desenvolvimento, e garantir que a interface mantenha coerência e usabilidade em diferentes tamanhos de tela evita distorções no layout e facilita a operação em qualquer ambiente (COROMINAS, 2024).

Para ilustrar a aplicação prática de *HTML* e *Bootstrap*, a Figura 9 apresenta um trecho do código *HTML* referente a interface Manter Cliente. Nesse exemplo, utilizam-se classes do *Bootstrap* como “*row*”, “*col*” e “*btn btn-primary*” para garantir responsividade e alinhamento consistente dos elementos.

Figura 9 – Código HTML de Manter Cliente

```
1 <app-cabecalho></app-cabecalho>
2
3 <div class="principal">
4   <app-faixa-titulo titulo="Manter Cliente" icone="seta"></app-faixa-titulo>
5
6   <fieldset>
7     <legend class="legenda-filtro">Filtros</legend>
8
9     <form [formGroup]="filtroCliente">
10      <div class="row">
11        <div class="form-group col-sm-2 col-md-2 col-lg-2">
12          <label for="codigo">Código</label>
13
14          <input formControlName="codigo"
15                type="text"
16                class="form-control"
17                id="codigo"
18                mask="0*"
19                maxLength="10">
20        </div>
21
22        <div class="form-group col-sm-3 col-md-3 col-lg-3">
23          <label for="cpfCnpj">CPF / CNPJ</label>
24
25          <input formControlName="cpfCnpj"
26                type="text"
27                class="form-control"
28                id="cpfCnpj"
29                mask="000.000.000-00||00.000.000/0000-00">
30        </div>
31      </div>
32    </form>
33  </fieldset>
34 </div>
```

Fonte: Autor

Para aprimorar a robustez e a escalabilidade do código, adotei o *TypeScript*. O *TypeScript* adiciona tipagem estática opcional ao JavaScript, permitindo a detecção de erros em tempo de compilação e facilitando a refatoração do código (MICROSOFT, 2025). Além disso, sua sintaxe orientada a objetos e recursos avançados contribuem para um desenvolvimento mais organizado e eficiente.

Integrando essas tecnologias, o *framework Angular* na versão 17 foi escolhido para a construção da aplicação. O *Angular* oferece uma arquitetura baseada em componentes, injeção de dependência e ferramentas poderosas para o desenvolvimento de aplicações *web* modernas. A versão 17 introduz melhorias significativas em desempenho e usabilidade, como renderização adiada e otimizações de *runtime*, que resultam em aplicações mais rápidas e responsivas (GECHEV, 2023).

A Figura 10 apresenta a implementação do serviço “ClienteService” em TypeScript, evidenciando como o *Angular 17* integra tipagem estática e consumo de *API REST*.

Figura 10 – Classe ClienteService

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpHeaders } from '@angular/common/http';
3 import { Cliente } from '../models/Cliente';
4
5 @Injectable({ Show usages: ± Mateus Almeida
6   providedIn: 'root'
7 })
8 export class ClienteService {
9   private baseUrl : string = 'http://localhost:8081/api/cliente';
10
11   constructor(private http: HttpClient) { no usages: ± Mateus Almeida
12   }
13
14   private getAuthHeaders(): HttpHeaders { Show usages: ± Mateus Almeida
15     const token : string | null = sessionStorage.getItem('auth-token');
16     return new HttpHeaders({
17       'Authorization': `Bearer ${token}`
18     });
19   }
20
21   salvarCliente(cliente: Cliente) : Observable<Cliente> { Show usages: ± Mateus Almeida
22     const headers : HttpHeaders = this.getAuthHeaders();
23     return this.http.post<Cliente>(this.baseUrl + "/novo", cliente, { headers });
24   }
25
26   consultarCliente(filtroCliente: any) : Observable<any> { Show usages: ± Mateus Almeida
27     const headers : HttpHeaders = this.getAuthHeaders();
28     return this.http.get<any>(`${this.baseUrl}/consultar`, {
29       headers,
30       params: filtroCliente
31     });
32   }
33 }
```

Fonte: Autor

5 CONCLUSÃO

O presente trabalho abordou de forma detalhada o desenvolvimento e a implementação do ContaFácil, um sistema de gerenciamento de contas de clientes para o açougue Casa das Carnes, integrando conceitos e práticas da Engenharia de Software. Esse projeto exigiu a aplicação dos conhecimentos adquiridos ao longo do curso de Engenharia de Computação, desde a definição precisa de requisitos até o desenho da arquitetura, a implementação de funcionalidades e a entrega final de um produto de software totalmente funcional. A documentação desenvolvida orientou cada etapa, garantindo que todas as necessidades reais do açougue fossem contempladas.

Durante a implementação, foram desenvolvidas funcionalidades essenciais para substituir o controle manual de contas a prazo: cadastro de clientes, registro de vendas fiadas, monitoramento de contas em aberto, pagas e vencidas, geração de relatórios financeiros e painel de consulta de movimentações. A integração de *Spring Security* com *JWT* garantiu autenticação e segurança dos dados, enquanto o front-end, desenvolvido em Angular assegurou um sistema prático e fácil de usar. Nos testes, a identificação de inconsistências levou a ajustes finos em várias camadas do sistema, reforçando a importância de cenários de validação abrangentes e de um ciclo iterativo de correção e reteste.

Este trabalho teve relevância não apenas na resolução de um problema de organização financeira, mas também como uma oportunidade de aprendizado prático em desenvolvimento de software. A aplicação dos conceitos adquiridos ao longo do curso de Engenharia de Computação, desde a elaboração de documentação de requisitos até a utilização de tecnologias de desenvolvimento, possibilitou vivenciar todo o ciclo de vida de um projeto, aprimorando competências de análise, projeto, codificação, teste e entrega.

Mais do que atender às necessidades imediatas da empresa, este sistema formou um produto de software modular e escalável, pronto para evoluir. Como trabalhos futuros, destaca-se a possibilidade de estender o ContaFácil para outros pequenos comércios que operam com vendas a prazo, incorporando funcionalidades como perfis de acesso diferenciados, personalização de dados do estabelecimento, envio automático de notificações e integração com meios de pagamento digital. Ademais, outras evoluções potenciais incluem integração com o WhatsApp para envio automatizado de cobranças, emissão de boletos

bancários via APIs de serviços financeiros, implementação de um módulo de controle de estoque integrado e suporte multiclientes.

Além dessas possibilidades, é fundamental realizar uma análise de conformidade com a Lei Geral de Proteção de Dados (LGPD), visando garantir que o tratamento das informações dos clientes estejam alinhado às exigências legais de privacidade e segurança de dados. Por fim, vale ressaltar que o sistema implementa autenticação básica de usuários via *JWT*, mas recursos avançados de gestão de usuários, como criação de múltiplos perfis e atribuição de permissões específicas, não fazem parte do escopo deste projeto, permanecendo como possíveis evoluções futuras. Dessa forma, o produto está preparado para atender novas demandas e reforçar a organização financeira de empreendedores de diversos segmentos.

REFERÊNCIAS

- BARROS, Hildenê Moreira. TRANSFORMAÇÃO DIGITAL PARA MEI, MICRO E PEQUENAS EMPRESAS: O PAPEL DAS FERRAMENTAS DIGITAIS PARA A INOVAÇÃO E O DESENVOLVIMENTO DO NEGÓCIO. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, [S. l.], v. 9, n. 9, p. 4529–4545, 2023. DOI: 10.51891/rease.v9i9.11570. Disponível em: <<https://periodicorease.pro.br/rease/article/view/11570>>. Acesso em: 2 jun. 2025.
- BELCIC, Ivan; STRYKER, Cole. O que é diagrama de entidade e relacionamento?. IBM, 2024. Disponível em: <<https://www.ibm.com/br-pt/think/topics/entity-relationship-diagram>>. Acesso em: 12 maio 2025.
- BOOTSTRAP TEAM. Bootstrap Documentation. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 09 mai. 2025.
- CHACON, Scott; STRAUB, Ben. Pro Git, Second Edition. Apress, 2014.
- COROMINAS, Julia. Responsive Design: The Key to a Perfect User Experience on All Devices. ADSMURAI, 2024. Disponível em: <<https://www.adsmurai.com/en/articles/responsive-design-the-key-to-a-perfect-user-experience-on-all-devices>>. Acesso em: 12 maio 2025.
- CREDIADO. Como controlar vendas fiado? 2024. Disponível em:<<https://blog.crediado.com/controlar-vendas-fiado/>>. Acesso em: 19 abr. 2025.
- FIGMA. “What is Figma?”. Figma Learn, 2019. Disponível em: <<https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>>. Acesso em: 12 maio 2025.
- GECHEV, Minko. Introducing Angular v17. Medium, 2023. Disponível em: <<https://medium.com/angular-blog/introducing-angular-v17-4d7033312e4b>>. Acesso em: 12 maio 2025.
- GOOGLE. Angular, 2025. Disponível em: <<https://angular.dev/overview>>. Acesso em: 12 maio 2025.
- HASTIE, S.; WOJEWODA, S. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, 2015. Disponível em: <<https://www.infoq.com/articles/standish-chaos-2015/>>. Acesso em: 22 fev. 2025.
- IBGE. 161,6 milhões de pessoas com 10 anos ou mais de idade utilizaram a Internet no país, em 2022. Agência de Notícias IBGE, Brasília, 9 nov. 2023. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/38307-161-6-milhoes-de-pessoas-com-10-anos-ou-mais-de-idade-utilizaram-a-internet-no-pais-em-2022>>. Acesso em: 2 jun. 2025.

- IBM. Diagramas de Caso de Uso. IBM, 2021. Disponível em: < <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case> >. Acesso em: 12 maio 2025.
- IEEE. Guide to the Software Engineering Body of Knowledge (SWEBOK V3.0). 2014. Disponível em: < <https://www.computer.org/education/bodies-of-knowledge/software-engineering> >. Acesso em: 12 jun. 2025.
- JWT.IO. Introduction to JSON Web Tokens. Disponível em: < <https://jwt.io/introduction/> >. Acesso em: 12 maio 2025.
- MDN. REST (Representational State Transfer). Disponível em: < <https://developer.mozilla.org/en-US/docs/Glossary/REST> >. Acesso em: 12 maio 2025.
- MICROSOFT. TypeScript Documentation. Disponível em: < <https://www.typescriptlang.org/> >. Acesso em: 09 mai. 2025.
- ORACLE. The Java Programming Language. Disponível em:< <https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html> > Acesso em: 09 mai. 2025.
- OWASP. Cross-Site Request Forgery (CSRF). Disponível em: < <https://owasp.org/www-community/attacks/csrf> >. Acesso em: 12 maio 2025.
- PITTET, STEN. Diferentes tipos de testes de software. Atlassian, 2025. Disponível em: < <https://www.atlassian.com/br/continuous-delivery/software-testing/types-of-software-testing> >. Acesso em: 09 mai. 2025.
- POSTGRESQL. About PostgreSQL, 2025. Disponível em: < <https://www.postgresql.org/docs/current/> >. Acesso em: 09 mai. 2025.
- PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.
- QUALIFICA. Não vendemos fiado: adote esse posicionamento sem perder o cliente. 2018. Disponível em: < <https://www.qualifica.com.br/blog/nao-vendemos-fiado-adote-esse-posicionamento-sem-perder-o-cliente> >. Acesso em: 19 abr. 2025.
- SEBRAE. Como a digitalização de processos impacta os resultados da empresa? Portal Sebrae, 2023. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/ufs/pe/artigos/como-a-digitalizacao-de-processos-impacta-os-resultados-da-empresa,833be1541664a810VgnVCM1000001b00320aRCRD>>. Acesso em: 2 jun. 2025.
- SEBRAE. Inadimplência: como resolver? Portal Sebrae, 2022. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/artigos/inadimplencia-como-resolver,204a341340461810VgnVCM100000d701210aRCRD>>. Acesso em: 2 jun. 2025.

SOMMERVILLE, I. Engenharia de Software, 9ª ed., Pearson Education do Brasil, 2011.

SPRING BOOT Documentation. Spring, 2025. Disponível em: < <https://spring.io/projects/spring-boot> > Acesso em: 09 mai. 2025.

SPRING DATA. Spring Data JPA – Reference Documentation. Disponível em: < <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> >. Acesso em: 12 maio 2025.

Venda a prazo: um dos grandes desafios na gestão dos micros e pequenos empreendedores. Blog do Sebrae MS, 2019. Disponível em:< <https://sebrae.ms/gestao-financeira/venda-a-prazo-um-dos-grandes-desafios-na-gestao-dos-micro-e-pequenos-empresendedores/> > Acesso em: 19 abr. 2025.

W3C. CSS Introduction. W3Schools, 2025. Disponível em: < https://www.w3schools.com/css/css_intro.asp >. Acesso em: 09 mai. 2025.

W3C. HTML Introduction. Disponível em: < https://www.w3schools.com/html/html_intro.asp >. Acesso em: 09 mai. 2025.

YIN, L. et al. Requirement-driven engineering change management in product design. Computers and Industrial Engineering, v. 168, 2022. DOI: 10.1016/j.cie.2022.108053.

APÊNDICE A - Documento de Requisitos do Sistema

**DOCUMENTO DE REQUISITOS -
CONTA FÁCIL**

VERSÃO 1.1

1. Introdução

A gestão eficiente das contas de clientes é essencial para o sucesso financeiro de micro e pequenas empresas. Muitas delas enfrentam dificuldades ao controlar suas contas, geralmente feitas de forma manual, o que pode resultar em erros, esquecimentos e desafios na hora de acompanhar pendências financeiras. Um controle desorganizado pode prejudicar a saúde financeira do negócio e dificultar a tomada de decisões.

O ContaFácil, sistema de gestão das contas de clientes proposto, visa automatizar esse processo. Ele oferece uma solução digital que organiza, centraliza e facilita o controle das transações a prazo. Com isso, o sistema traz mais segurança, eficiência e clareza no gerenciamento das contas, reduzindo riscos e melhorando o controle financeiro das empresas.

A venda a prazo é uma prática comum em pequenos comércios, como açougues, mercearias, padarias e lojas de bairro. O ContaFácil foi criado para resolver um problema real enfrentado pelo açougue Casa das Carnes, mas os mesmos desafios também estão presentes em outros tipos de estabelecimentos. Por isso, a solução pode ser facilmente aplicada a diferentes negócios que precisam organizar melhor as contas de seus clientes e ter mais controle sobre suas finanças.

1.1. Objetivos

O objetivo deste documento é definir os requisitos e especificações para o ContaFácil, sistema de gestão de contas de clientes destinado ao açougue Casa das Carnes. O produto é um software que tem como finalidade substituir o registro manual de compras feitas a prazo, proporcionando maior segurança, organização e eficiência no controle das contas dos clientes.

1.2. Público-alvo

O sistema será utilizado pelos funcionários da empresa, que trabalham no caixa, para registrar e gerenciar as compras realizadas a prazo pelos clientes.

2. Principais Funcionalidades do Sistema

O software será uma aplicação *web* e deverá ser desenvolvido com foco em usabilidade, desempenho e segurança. Ele deverá incluir as seguintes funcionalidades essenciais:

- Cadastro de clientes;
- Cadastro de contas;
- Cadastro de compras;
- Filtros de busca conforme a necessidade do cliente;
- Impressão de relatórios;

3. Informações de Suporte

3.1 Termos e Definições

CSU – Casos de uso;

RFUN – Requisitos funcionais;

RNF – Requisitos não funcionais;

RFUT – Requisitos futuros;

3.2 Modelo de Especificação de requisitos

A seguir são apresentados os campos do Modelo de Especificação de Requisitos utilizado nesta documentação.

ID: Cada requisito deve possuir um identificador único, isso facilita o referenciamento de cada requisito. Será o tipo seguido por uma numeração sequencial.

Nome: Curta descrição do requisito.

Importância: É definida assinalando uma das opções a seguir:

- **Essencial:** requisito imprescindível, sem o qual o sistema não entra em funcionamento ou não possui um mínimo das características desejáveis;

- **Desejável:** requisito sem o qual o sistema entra em funcionamento, porém de forma não satisfatória;
- **Opcional:** requisitos opcionais, sem os quais o sistema entra em funcionamento de forma satisfatória;

Complexidade: Identifica o requisito por grau de complexidade. Esse será um requisito que pode levar mais tempo de conclusão. A mensuração é feita na escolha de uma das opções: Difícil, Médio e Fácil.

Prioridade: A mensuração é feita na escolha de uma das opções:

- **Alto:** Requisitos essenciais e indispensáveis para o funcionamento do sistema. Sem eles, o sistema não poderá ser utilizado.
- **Médio:** Requisitos importantes, cuja ausência pode comprometer algumas funcionalidades, mas ainda permitirá que o sistema funcione, embora de forma limitada.
- **Baixo:** Requisitos complementares que agregam valor ao sistema, mas sua implementação pode ser feita posteriormente sem comprometer o funcionamento principal.

Origem do requisito: refere-se à fonte de onde o requisito foi gerado. Ela indica quem ou o que causou a necessidade do requisito no projeto.

Descrição: mostra uma longa e detalhada especificação do requisito.

Justificativa: Fundamental para explicar a necessidade deste requisito, ajudando a garantir que ele esteja alinhado com o escopo do projeto.

Critério de verificação: Um requisito é considerado verificável se existir um processo custo-efetivo que permita confirmar se o produto de software atende a esse requisito.

Histórico: Os requisitos podem ser acompanhados por sua data de criação, alterações ou exclusões.

4. Requisitos do Software

Nesta seção, serão apresentados os requisitos funcionais e não funcionais do sistema, detalhando suas funcionalidades essenciais e características esperadas para garantir um desempenho adequado e atender às necessidades dos usuários.

4.1 Requisitos Funcionais (RFUN)

O quadro 2 mostra os requisitos funcionais do sistema.

Quadro 2 - Requisitos Funcionais

Ref.	Descrição
RFUN-01	Manter clientes
RFUN-02	Manter contas
RFUN-03	Manter compras
RFUN-04	Calcular automaticamente o total de compras
RFUN-05	Disponibilizar filtros de busca em todas as interfaces
RFUN-06	Gerar relatórios em todas as interfaces
RFUN-07	Controlar contas abertas e encerradas
RFUN-08	Disponibilizar Painel Financeiro

Fonte: Elaborado pelo Autor.

ID: RFUN-01
Nome: Manter Clientes
Importância: Essencial
Complexidade: Médio
Prioridade: Alto
Origem do Requisito: Etnografia e Entrevista
<p>Descrição: O sistema deve permitir o cadastro, edição, consulta e exclusão de clientes, assegurando a integridade e unicidade das informações.</p> <p>Cada cliente deve possuir:</p> <ul style="list-style-type: none"> • Um identificador único no sistema; • Dados obrigatórios como: nome, CPF ou CNPJ, telefone, endereço e e-mail. <p>Deve ser disponibilizada uma tela específica para realizar todas essas operações relacionadas aos clientes.</p> <p>Regras de Negócio:</p> <ul style="list-style-type: none"> • Não é permitido cadastrar dois clientes com o mesmo CPF ou CNPJ. • Não é permitido excluir um cliente que esteja vinculado a uma conta.
Justificativa: Essencial para o funcionamento do sistema, pois os clientes são a base da operação do açougue.

Critério de verificação: O usuário deve conseguir cadastrar o cliente sem erros no sistema.

Histórico:

- Data de criação: 19/02/2025.
- Ajustes no requisito: 12/04/2025.
- Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-02

Nome: Manter Contas

Importância: Essencial

Complexidade: Médio

Prioridade: Alto

Origem do Requisito: Etnografia e Brainstorming

Descrição: O sistema deve permitir o cadastro automático de uma conta para o cliente sempre que for realizada uma nova compra. Cada cliente poderá ter apenas uma conta em aberto por vez.

Também deve ser possível criar uma conta manualmente através da interface "Manter Conta", desde que o cliente não possua uma conta em aberto. Caso essa condição seja atendida, o sistema deve permitir a criação da nova conta.

Além disso, o sistema deve possibilitar a consulta, edição e cancelamento da conta.

Cancelamento de Conta

- Para cancelar uma conta, o usuário deve acessar a edição do registro e alterar o status para "CANCELADO".
- Nesse caso, o campo "data fim" deve ser preenchido obrigatoriamente.
- Também será necessário informar uma justificativa, que só será exibida quando o status "CANCELADO" estiver selecionado.
- Não é permitido cancelar uma conta com status "ENCERRADO".

Vencimento da Conta

- Toda conta criada terá uma data de vencimento definida de forma automática, será de 30 dias a partir da data de abertura da conta.

Regras de Status

- O único status que pode ser selecionado manualmente pelo usuário é o "CANCELADO".
- Os demais status são gerenciados automaticamente pelo sistema:

<ul style="list-style-type: none"> ○ Ao criar a conta, o status inicial será "ATIVO". ○ Quando uma data fim for informada (sem status cancelado), a conta será automaticamente marcada como "ENCERRADO".
<p>Justificativa: Essencial para o funcionamento do sistema, pois as funcionalidades principais giram em torno das contas.</p>
<p>Critério de verificação: O usuário deve conseguir criar uma conta, ao cadastrar uma nova compra, editar a compra e cancelar a compra.</p>
<p>Histórico:</p> <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 12/04/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-03
Nome: Manter Compras
Importância: Essencial
Complexidade: Médio
Prioridade: Alto
Origem do Requisito: Etnografia e Entrevista
<p>Descrição: O sistema deve permitir o registro de compras associadas a uma conta específica, além de possibilitar a consulta, edição e exclusão dessas compras.</p> <p>Cadastro de Compra: Durante o cadastro de uma compra, os seguintes campos devem ser fornecidos obrigatoriamente:</p> <ul style="list-style-type: none"> • Cliente (associado à conta) • Valor da compra • Data da compra <p>Também deve existir um campo de descrição, que é opcional, para inserção de informações adicionais.</p> <p>Verificação de Conta em Aberto</p> <ul style="list-style-type: none"> • Ao selecionar um cliente, o sistema deve buscar automaticamente se há contas em aberto para esse cliente. • Se houver, os dados da conta em aberto eles devem ser exibidos em um <i>fieldset</i> na parte inferior da interface. • Se não houver conta em aberto, o sistema deve exibir a seguinte mensagem:

<ul style="list-style-type: none"> • "Não existe uma conta em aberto para esse cliente." • Nessa situação, o sistema deverá criar automaticamente uma conta para o cliente ao salvar o cadastro da compra. <p>Regra de Bloqueio</p> <ul style="list-style-type: none"> • Se o cliente possuir uma conta vencida, o sistema não deve permitir o registro de uma nova compra até que a situação seja regularizada.
Justificativa: Fundamental para o controle das contas.
Critério de verificação: O usuário deve conseguir registrar compras e o sistema deve associá-las corretamente a conta.
<p>Histórico:</p> <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 12/04/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-04
Nome: Calcular automaticamente o total das compras
Importância: Essencial
Complexidade: Fácil
Prioridade: Alto
Origem do Requisito: Entrevista
<p>Descrição: O sistema deve calcular automaticamente o valor total da conta com base no valor da compra que está sendo efetuada e das compras que já estão vinculadas a conta.</p> <p>Funcionamento na Interface:</p> <ul style="list-style-type: none"> • O cálculo deve ser realizado automaticamente ao retirar o foco (evento de <i>blur</i>) do campo valor durante o preenchimento do formulário de cadastro de compra. • O valor total da compra deve ser exibido de forma clara no <i>fieldset</i> de dados da conta, localizado na parte inferior da interface.
Justificativa: Facilita o fechamento das contas e reduz erros humanos.
Critério de verificação: O total da compra deve ser calculado corretamente e exibido na interface.

Histórico:

- Data de criação: 19/02/2025.
- Ajustes no requisito: 12/04/2025.
- Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-05

Nome: Disponibilizar filtros de busca em todas as telas

Importância: Essencial

Complexidade: Fácil

Prioridade: Médio

Origem do Requisito: Entrevista

Descrição: Todas as interfaces de consulta do sistema devem disponibilizar filtros para facilitar a busca de informações.

Justificativa: Fundamental para o controle das contas.

Critério de verificação: O usuário deve conseguir realizar uma consulta em qualquer tela utilizando os filtros disponíveis.

Histórico:

- Data de criação: 19/02/2025.
- Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-06

Nome: Gerar relatórios em todas as telas

Importância: Desejável

Complexidade: Fácil

Prioridade: Médio

Origem do Requisito: Brainstorming

Descrição: O sistema deve permitir exportar relatório de consulta em todas as interfaces do sistema.
Justificativa: Facilita a conferência
Critério de verificação: O usuário deve conseguir gerar e imprimir um registro de compra corretamente.
Histórico: <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 12/04/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-07
Nome: Controlar contas abertas e encerradas
Importância: Essencial
Complexidade: Médio
Prioridade: Alto
Origem do Requisito: Brainstorming
Descrição: O sistema deve manter um controle sobre contas ativas, canceladas e encerradas, deve existir um status para cada conta e cada conta deve estar vinculada a apenas um cliente dentro do mês e ano.
Justificativa: Garante a organização e rastreabilidade das contas, evitando inconsistências e possibilitando um melhor controle financeiro.
Critério de verificação: O sistema deve impedir que um cliente tenha mais de uma conta aberta no mesmo mês e ano. As contas devem possuir um status bem definido (ATIVO, ENCERRADO, CANCELADO)
Histórico: <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 12/04/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RFUN-08
Nome: Disponibilizar Painel Financeiro
Importância: Essencial
Complexidade: Difícil
Prioridade: Médio
Origem do Requisito: Brainstorming
Descrição: O sistema deve fornecer uma interface contendo informações relevantes sobre as contas e compras dos clientes, incluindo: quantidade total e valores das contas que ainda não foram pagas, lista de contas que ultrapassaram o prazo de pagamento, os clientes que mais compraram e dos que menos compraram em um determinado período. O filtro de período deve ser composto por dois campos: data de início e fim.
Justificativa: A implementação desse requisito melhora a gestão e a tomada de decisão dentro do sistema, permitindo que os administradores acompanhem rapidamente a situação das contas. Com dados visuais e rankings, fica mais fácil identificar padrões e tomar decisões, como cobrar contas vencidas ou oferecer benefícios para clientes que compram com frequência.
Critério de verificação: O sistema deve exibir um resumo das contas abertas e vencidas, incluindo valores e quantidades. Além de apresentar um ranking de clientes, mostrando os que mais e menos compraram
Histórico: <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

4.2 Requisitos Não Funcionais (RNF)

O quadro 3 mostra os requisitos não funcionais do sistema.

Quadro 3 - Requisitos Não Funcionais

Ref.	Descrição
RNF-01	Aplicação <i>web</i>
RNF-02	Oferecer uma interface intuitiva
RNF-03	Otimizar Funções
RNF-04	Garantir segurança de dados

Fonte: Elaborado pelo Autor.

ID: RNF-01
Nome: Aplicação <i>Web</i>
Importância: Essencial
Complexidade: Médio
Prioridade: Alto
Origem do Requisito: Entrevista
Descrição: O sistema será desenvolvido como uma aplicação <i>web</i> , acessível via navegador.
Justificativa: Uma aplicação <i>web</i> permite maior acessibilidade e facilidade de uso, eliminando a necessidade de instalação em dispositivos dos usuários.
Critério de verificação: O sistema deve ser acessível em um navegador <i>web</i> .
Histórico: <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RNF-02
Nome: Oferecer uma interface intuitiva
Importância: Desejável
Complexidade: Médio
Prioridade: Alto
Origem do Requisito: Entrevista
Descrição: A interface do sistema deve ser fácil de usar, mesmo para pessoas com pouca experiência em tecnologia. Para isso, devem ser seguidas as seguintes práticas: <ol style="list-style-type: none"> 1. Rótulos Claros: Todos os campos, botões e ícones devem ter nomes claros, sem abreviações ou termos técnicos confusos. 2. Fluxos Simples: A navegação deve ser direta e organizada de forma lógica, para que os usuários encontrem o que precisam rapidamente. 3. Feedback Visual: O sistema deve mostrar mensagens de erro ou sucesso de forma clara, destacando os campos com problemas.

<p>4. Responsividade: A interface deve funcionar bem em dispositivos como celulares, tablets e computadores, ajustando-se ao tamanho da tela.</p> <p>5. Testes de Usabilidade: Testes serão feitos com usuários sem experiência em tecnologia para garantir que a interface seja fácil de usar.</p>
<p>Justificativa: Facilitar o uso do sistema pelo caixa independentemente de sua familiaridade com tecnologia.</p>
<p>Critério de verificação: Um usuário com pouca experiência em tecnologia deve conseguir utilizar as funções sem auxílio.</p>
<p>Histórico:</p> <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RNF-03
<p>Nome: Otimizar Funções</p>
<p>Importância: Essencial</p>
<p>Complexidade: Médio</p>
<p>Prioridade: Alto</p>
<p>Origem do Requisito: Entrevista</p>
<p>Descrição: O sistema deve ser rápido e eficiente, com as funções sendo realizadas sem demora. Para garantir isso:</p> <ol style="list-style-type: none"> 1. Redução de Tempo: O sistema deve responder rapidamente às ações do usuário, carregando dados de forma eficiente e apenas quando necessário. 2. Consultas Eficientes: O banco de dados deve ser otimizado para retornar resultados rapidamente, com consultas e processos eficientes. 3. Navegação Ágil: A navegação entre telas e funcionalidades deve ser rápida e sem atrasos. 4. Monitoramento de Performance: O desempenho do sistema será monitorado para identificar e corrigir possíveis lentidões. 5. Experiência Fluida: O sistema deve responder de forma instantânea às ações do usuário, sem interrupções visíveis durante o uso.
<p>Justificativa: Essencial para proporcionar uma experiência fluida e eficaz ao usuário.</p>
<p>Critério de verificação:</p> <ul style="list-style-type: none"> • Ao executar uma função do sistema ela deve ser finalizada rapidamente. • A exibição das informações na tela deve ser feita rapidamente.

<ul style="list-style-type: none"> • A navegação pelas interfaces deve ser fluida.
Histórico: <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

ID: RNF-04
Nome: Garantir segurança de dados
Importância: Essencial
Complexidade: Difícil
Prioridade: Alto
Origem do Requisito: Entrevista
Descrição: O sistema deve garantir que apenas usuários autorizados possam acessar as telas internas, utilizando autenticação e criptografia de senhas.
Justificativa: Fundamental para proteger informações sensíveis e garantir a integridade do sistema.
Critério de verificação: O sistema deve exigir autenticação e armazenar senhas criptografadas.
Histórico: <ul style="list-style-type: none"> • Data de criação: 19/02/2025. • Ajustes no requisito: 20/04/2025.

Fonte: Elaborado pelo Autor.

4.3 Requisitos Futuros (RFUT)

RFUT 01 – Notificações automáticas para lembrar clientes sobre o vencimento das contas.

RFUT 02 – Integração com sistemas de pagamento digital para facilitar a quitação das contas.

5. Matriz de Rastreabilidade

A tabela 1 é uma matriz de rastreabilidade, ela serve para mapear e acompanhar a relação entre os requisitos do sistema.

Tabela 1 - Matriz de Rastreabilidade dos Requisitos

Requisito	RFUN-01	RFUN-02	RFUN-03	RFUN-04	RFUN-05	RFUN-06	RFUN-07	RFUN-08
RFUN-01	-	X	X					
RFUN-02		-	X				X	
RFUN-03		X	-	X		X		
RFUN-04				-				
RFUN-05					-			X
RFUN-06			X			-		
RFUN-07		X					-	
RFUN-08					X			-

Fonte: Elaborado pelo Autor.

Os requisitos não funcionais não foram incluídos diretamente na matriz de rastreabilidade, pois se aplicam de forma transversal a todos os requisitos funcionais. Isso significa que características como desempenho, usabilidade e segurança impactam o sistema como um todo, sendo consideradas em cada funcionalidade desenvolvida.

APÊNDICE B – Artefatos do Projeto

1. Casos de Uso

1.1. Atores

Funcionário do caixa – Registra compras e gerência contas dos clientes e pode visualizar relatórios.

1.2. Lista de casos de uso

O quadro 4 mostra os casos de uso de software identificados.

Quadro 4 - Casos de Uso do Software

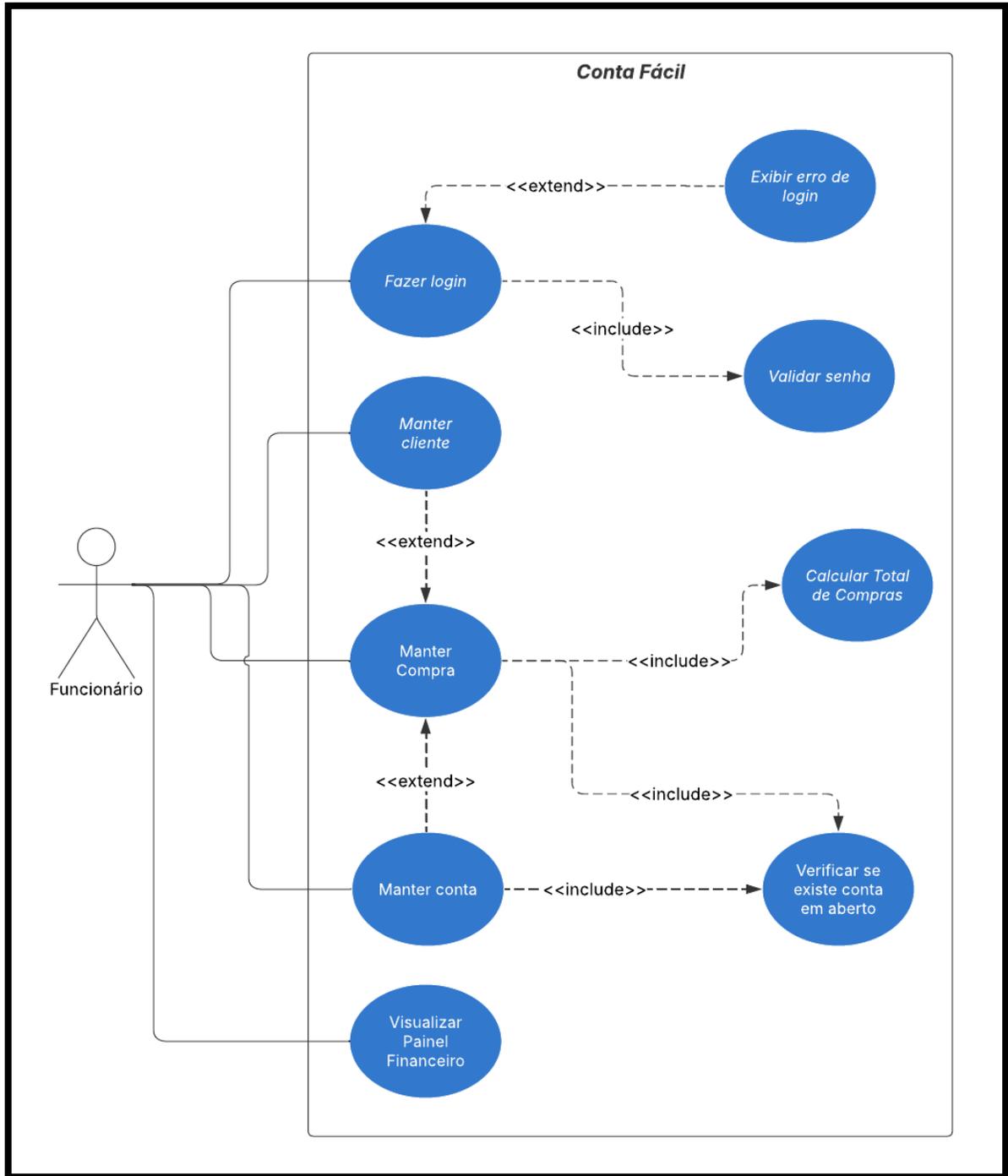
Ref.	Descrição	Atores
CSU1	Fazer login	Funcionário
CSU2	Manter cliente	Funcionário
CSU3	Manter compra	Funcionário
CSU4	Manter conta	Funcionário
CSU5	Calcular Total de Compras	Funcionário
CSU6	Visualizar Painel Financeiro	Funcionário

Fonte: Elaborado pelo Autor.

1.3. Diagrama de Casos de Uso

A figura 11 mostra o diagrama dos casos de uso do sistema.

Figura 11 - Diagrama de Caso de Uso Geral



Fonte: Elaborado pelo Autor.

1.4. Descrição de Casos de Uso Detalhado

Identificação: CSU1

Nome: Fazer Login
Atores: Funcionário
Acionador: URL da aplicação (www.contafacilweb.com.br)
Pré-condições: O usuário deve estar cadastrado no sistema.
Pós-condições: O usuário é autenticado e tem acesso ao sistema.
Sequência típica de eventos
<ol style="list-style-type: none"> 1. O sistema exibe os campos de e-mail e senha. 2. O funcionário informa suas credenciais e solicita login 3. O sistema verifica se as credenciais são válidas. 4. O sistema gera um token para o usuário e armazena ele no <i>sessionStorage</i>. 5. O sistema faz a autenticação do usuário, mostra a mensagem “Bem-vindo, Usuário!” e redireciona para o painel principal.
Sequência de exceção
4.1 Credenciais inválidas
<ol style="list-style-type: none"> 1. O sistema exibe a seguinte mensagem: “Usuário ou senha incorretos! Tente novamente”

Fonte: Elaborado pelo Autor.

Identificação: CSU2
Nome: Manter Cliente
Atores: Funcionário
Requisitos Relacionados: RFUN-01, RFUN-05 e RFUN-06.
Acionador: Botão manter clientes do painel principal
Pré-condições: O funcionário deve estar autenticado no sistema.
Pós-condições: <ul style="list-style-type: none"> • O cliente é cadastrado no sistema e pode realizar compras a prazo. • O sistema armazena corretamente as informações dos clientes • Em caso de edição, as informações do cliente são atualizadas corretamente. • Em caso de exclusão, o cliente é removido do sistema (se aplicável). • Se o relatório for gerado, o arquivo .pdf dele será baixado.
Sequência típica de eventos
<ol style="list-style-type: none"> 1. Sistema exibe a interface de consulta com: <ul style="list-style-type: none"> ○ Tabela de clientes (com coluna "Ações" para editar). ○ Filtros não obrigatórios (Código, CPF/CNPJ e Nome). ○ Botão para cadastrar novo cliente. ○ Botão Consultar (aplica filtros). ○ Botão Gerar Relatório (exporta dados da consulta) 2. Funcionário escolhe uma opção. 3. Sistema exibe formulário correspondente (cadastro/edição). 4. Funcionário preenche/modifica dados (nome, telefone, endereço etc.). 5. Sistema valida os dados. 6. Se válidos:

<ul style="list-style-type: none"> ○ Sistema armazena/atualiza informações. ○ Sistema exibe mensagem de sucesso ("Cliente salvo com sucesso").
Sequência de Alternativa
<p>2.1 Caso o funcionário clique em novo</p> <ol style="list-style-type: none"> 1. Sistema exibe formulário em branco com os campos: <ul style="list-style-type: none"> ○ CPF/CNPJ (Obrigatório). ○ E-mail (Opcional). ○ Número de Telefone (Obrigatório). ○ Whatsapp (Obrigatório). ○ Nome Completo (Obrigatório). ○ Endereço (Obrigatório). ○ Complemento (Opcional). ○ Cidade (Obrigatório). 2. Funcionário pode: <ul style="list-style-type: none"> ○ Preencher dados e clicar em "Salvar". (Volta para validação. Sequência Típica, passo 5). ○ Clicar em "Cancelar" <ul style="list-style-type: none"> • Sistema descarta alterações e volta para a tela de consulta.
<p>2.2 Caso o funcionário clique em Consultar</p> <ol style="list-style-type: none"> 1. O Sistema realiza a consulta aplicando os filtros. 2. Se não houver resultados o sistema mostra a mensagem "Não há dados disponíveis para exibir."
<p>2.3 Caso o funcionário clique em Gerar Relatório</p> <ol style="list-style-type: none"> 1. O sistema exporta um PDF contendo os dados da consulta de cliente.
<p>2.4 Caso o funcionário clique no ícone de edição</p> <ol style="list-style-type: none"> 1. Sistema exibe formulário preenchido com dados do cliente. 2. Funcionário pode: <ul style="list-style-type: none"> ○ Modificar dados e clicar em "Salvar". ○ Clicar em "Excluir Cliente" (ver Sequência Alternativa 2.5). ○ Clicar em "Cancelar". <ul style="list-style-type: none"> • Sistema descarta alterações e volta para a tela de consulta.
<p>2.5 Excluir Cliente.</p> <ol style="list-style-type: none"> 1. Sistema solicita confirmação ("Deseja realmente excluir este cliente?"). 2. Se confirmado: <ul style="list-style-type: none"> ○ Sistema remove cliente do banco de dados. ○ Sistema exibe mensagem ("Cliente excluído com sucesso") e retorna à lista inicial.
Sequência de exceção
<p>5.1 Campos Obrigatórios Não Preenchidos</p> <ol style="list-style-type: none"> 1. Funcionário tenta salvar sem preencher campos obrigatórios. 2. Sistema identifica os campos vazios/inválidos e: <ul style="list-style-type: none"> ○ Destaca os campos em vermelho. ○ Mostra uma mensagem global no topo do formulário: "Preencha todos os campos obrigatórios."

6.1 Cliente com CPF/CNPJ Já Cadastrado

1. Funcionário tenta cadastrar um cliente com CPF/CNPJ já existente no sistema.
2. Sistema detecta a duplicidade e:
 - Mostra uma mensagem global de alerta: "Erro ao salvar cliente."

2.5.1 Cliente tem vínculo com uma conta

- Sistema mostra mensagem "Erro ao excluir cliente."

Fonte: Elaborado pelo Autor.

Identificação: CSU3
Nome: Manter Compra
Atores: Funcionário
Requisitos Relacionados: RFUN-03, RFUN-05 e RFUN-06.
Acionador: Botão "Manter Compra" do painel principal
Pré-condições: O cliente deve estar cadastrado no sistema.
Pós-condições: <ul style="list-style-type: none"> • A compra é registrada e associada a conta de um cliente no sistema. • Em caso de edição, as informações do cliente são atualizadas corretamente. • Em caso de exclusão, o cliente é removido do sistema (se aplicável). • Se o relatório for gerado, o arquivo .pdf dele será baixado.
Sequência típica de eventos
<ol style="list-style-type: none"> 1. Sistema exibe a interface de consulta com: <ul style="list-style-type: none"> ○ Tabela de compras e opções (com coluna "Ações" para editar). ○ Filtros não obrigatórios (Código, CPF/CNPJ, Nome e Data Compra). ○ Botão para cadastrar nova compra. ○ Botão Consultar (aplica filtros). ○ Botão Gerar Relatório (exporta dados da consulta) 2. Funcionário escolhe uma opção. 3. Sistema exibe formulário correspondente (cadastro/edição). 4. Funcionário preenche/modifica dados. 5. Sistema valida os dados. 6. Se válidos: <ul style="list-style-type: none"> ○ Sistema armazena/atualiza informações. ○ Sistema exibe mensagem de sucesso ("Compra salva com sucesso").
Sequência alternativa
<ol style="list-style-type: none"> 2.1 Caso o usuário clique em novo <ol style="list-style-type: none"> 1. Sistema exibe formulário em branco com os campos: <ul style="list-style-type: none"> ○ Cliente (Obrigatório). ○ Valor (Obrigatório). ○ Data (Obrigatório). ○ Descrição (Opcional). 2. Funcionário pode:

<ul style="list-style-type: none"> ○ Preencher dados e clicar em "Salvar". (Volta para validação. Sequência Típica, passo 5). ○ Clicar em "Cancelar" <ul style="list-style-type: none"> • Sistema descarta alterações e volta para a tela de consulta.
<p>2.2 Caso o funcionário clique no ícone de edição</p> <ol style="list-style-type: none"> 1. Sistema exibe formulário preenchido com dados da conta e os botões: <ul style="list-style-type: none"> ○ Modificar os dados e clicar em "Salvar". ○ Clicar em "Cancelar". ○ "Excluir Compra". <ul style="list-style-type: none"> • Sistema descarta alterações e volta para a tela de consulta
<p>2.3 Caso o funcionário clique em excluir</p> <ol style="list-style-type: none"> 1. Sistema solicita confirmação ("Deseja realmente excluir esta compra?") 2. Se confirmado: <ul style="list-style-type: none"> ○ Sistema remove compra do banco de dados ○ Sistema exibe a mensagem "Compra excluída!"
<p>2.4 Caso o funcionário clique em Gerar Relatório</p> <ol style="list-style-type: none"> 1. O sistema exporta um PDF contendo os dados da consulta de compra.
<p>2.5 Caso o funcionário clique em Consultar</p> <ol style="list-style-type: none"> 1. O Sistema realiza a consulta aplicando os filtros. 2. Se não houver resultados o sistema mostra a mensagem "Não há dados disponíveis para exibir."
Sequência de exceção
<p>5.1 Cliente com conta vencida.</p> <ol style="list-style-type: none"> 1. Sistema mostra mensagem "Este cliente tem uma conta vencida. Verifique!"
<p>5.2 Campos Obrigatórios Não Preenchidos</p> <ol style="list-style-type: none"> 1. Funcionário tenta salvar sem preencher os campos obrigatórios. 2. Sistema identifica os campos vazios/inválidos e: <ul style="list-style-type: none"> ○ Destaca os campos em vermelho. ○ Mostra uma mensagem global no topo do formulário: "Preencha todos os campos obrigatórios."

Fonte: Elaborado pelo Autor.

Identificação: CSU4
Nome: Manter Conta
Atores: Funcionário
Requisitos Relacionados: RFUN-02, RFUN-05, RFUN-06 e RFUN-07.
Acionador: Botão manter conta do painel principal
Pré-condições: O cliente deve estar cadastrado no sistema.
Pós-condições: <ul style="list-style-type: none"> • Se uma conta for cadastrada, ela estará disponível para novas compras. • Se uma conta for editada, suas informações serão atualizadas corretamente. • Se uma conta for cancelada, ela será marcada como cancelada e não poderá ser utilizada para novas compras. • Se o relatório for gerado, o arquivo .pdf dele será baixado.

Sequência típica de eventos
<ol style="list-style-type: none"> 1. Sistema exibe a interface de consulta com: <ul style="list-style-type: none"> ○ Tabela de contas (com coluna "Ações" para editar). ○ Filtros não obrigatórios (Código, CPF/CNPJ, Cliente, Data Início e Data Fim). ○ Botões de ação: <ul style="list-style-type: none"> ▪ Nova Conta (cadastro). ▪ Consultar (aplica filtros). ▪ Gerar Relatório (exporta dados da consulta). 2. Funcionário escolhe uma opção. 3. Sistema exibe formulário correspondente (cadastro/edição). 4. Funcionário preenche/modifica dados (cliente, data início, data fim etc.). 5. Sistema valida os dados. 7. Se válidos: <ul style="list-style-type: none"> ○ Sistema armazena/atualiza informações. ○ Sistema exibe mensagem de sucesso ("Conta salva com sucesso").
Sequência alternativa
<p>2.1 Caso o funcionário clique em novo</p> <ol style="list-style-type: none"> 1. Sistema exibe formulário com campos: <ul style="list-style-type: none"> ○ Cliente (Obrigatório). ○ Data de Vencimento preenchido com 30 dias a partir da data de início. ○ Status preenchido como ATIVO (Obrigatório). ○ Data Início ○ Data Fim 2. Usuário preenche os dados e clica em "Salvar". (Volta para validação. Sequência Típica, passo 5).
<p>2.2 Caso o funcionário clique em consultar</p> <ol style="list-style-type: none"> 1. O Sistema realiza a consulta aplicando os filtros. 2. Sistema atualiza a tabela com resultados. <ul style="list-style-type: none"> ○ Se não houver resultados o sistema mostra a mensagem “Não há dados disponíveis para exibir.”
<p>2.3 Caso o funcionário clique em Gerar Relatório</p> <ul style="list-style-type: none"> ○ O sistema exporta um PDF contendo os dados da consulta da conta.
<p>2.4 Caso o funcionário clique no ícone de edição</p> <ol style="list-style-type: none"> 3. Sistema exibe formulário preenchido com dados da conta. 4. Funcionário pode: <ul style="list-style-type: none"> ○ Modificar dados e clicar em "Salvar". (Volta para validação. Sequência Típica, passo 5). ○ Clicar em “Cancelar”. <ul style="list-style-type: none"> • Sistema descarta alterações e volta para a tela de consulta.
Sequência de exceção
<p>5.1 Campos Obrigatórios Não Preenchidos</p> <ol style="list-style-type: none"> 1. Funcionário tenta salvar sem preencher campos obrigatórios. 2. Sistema identifica os campos vazios/inválidos e: <ul style="list-style-type: none"> ○ Destaca os campos em vermelho.

<ul style="list-style-type: none"> ○ Mostra uma mensagem global no topo do formulário: "Preencha todos os campos obrigatórios."
5.2 Cliente Já Possui Conta Ativa <ol style="list-style-type: none"> 1. O usuário tenta inserir um cliente que já tem um conta com status ATIVO 2. O sistema mostra a mensagem “Este cliente já possui uma conta ativa”.
5.3 Cliente com conta vencida que não foi pago. <ol style="list-style-type: none"> 1. Sistema mostra mensagem “Este cliente tem uma conta vencida. Verifique!”

Fonte: Elaborado pelo Autor.

Identificação: CSU5
Nome: Calcular Total de Compras
Requisitos Relacionados: RFUN-03, RFUN-4
Atores: Funcionário
Acionador: Perda de foco do campo valor da compra
Pré-condições: <ul style="list-style-type: none"> • O funcionário deve estar autenticado no sistema. • Deve existir pelo menos uma compra vinculada a conta • O usuário deve estar registrando ou editando uma compra.
Pós-condições: O total é atualizado dinamicamente conforme as compras são adicionados, canceladas ou alteradas.
Sequência típica de eventos
<ol style="list-style-type: none"> 1. O sistema faz o cálculo do total de todas as compras vinculada a conta instantaneamente. 2. O funcionário preenche o campo valor 3. O sistema exibe o valor calculado no campo “Valor Total” localizado no <i>fieldset</i> da “Conta Vinculada”.

Fonte: Elaborado pelo Autor.

Identificação: CSU6
Nome: Visualizar Painel Financeiro
Requisitos Relacionados: RFUN-08 e RFUN-05
Atores: Funcionário
Acionador: Botão Painel Financeiro no painel principal
Pré-condições:
Pós-condições: <ul style="list-style-type: none"> • O funcionário pode visualizar informações essenciais para tomada de decisão • O sistema exibe métricas sobre contas e compras
Sequência típica de eventos
<ol style="list-style-type: none"> 1. O sistema exibe a interface do painel com informações como: total de contas abertas, contas vencidas e valores totais. 2. O funcionário seleciona um período para análise (Data Início e Data Fim). 3. O sistema atualiza os resultados com base no período escolhido.

Sequência de exceção
<p>1.1 Nenhuma Conta Cadastrada</p> <ol style="list-style-type: none"> 1. O sistema verifica que não há dados disponíveis. 2. O sistema exibe a mensagem “Não há registros para exibição”.

Fonte: Elaborado pelo Autor.

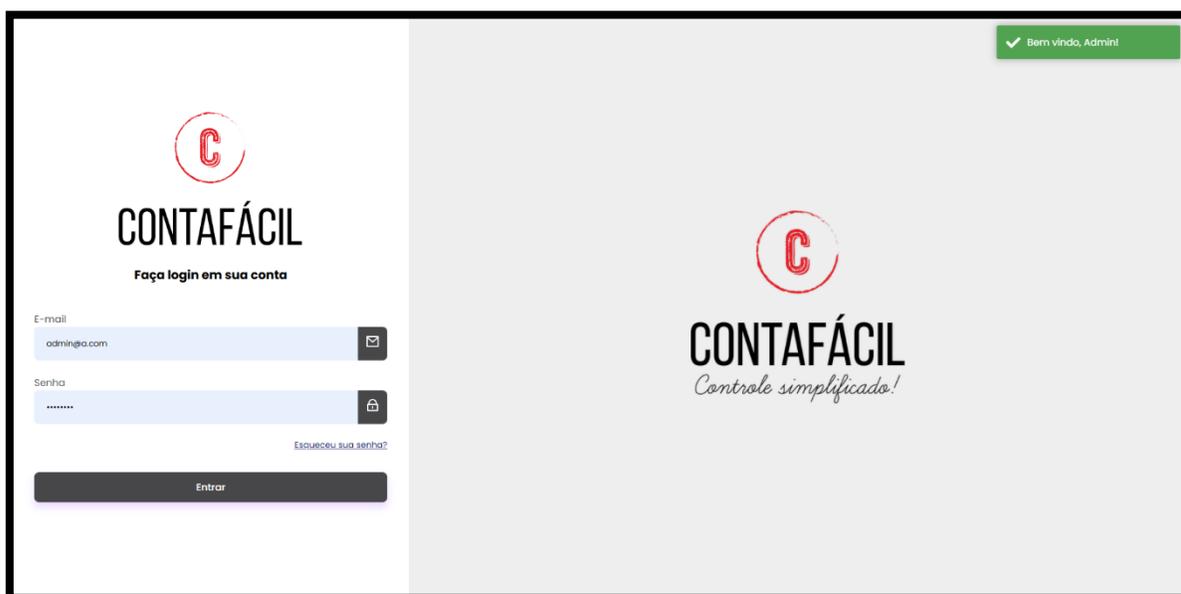
2. Projeto de Interfaces

Este capítulo apresenta os protótipos das interfaces correspondentes aos casos de uso da aplicação. Toda a interface visual foi desenvolvida utilizando a ferramenta Figma. As telas também podem ser acessadas diretamente por meio do seguinte link: <https://www.figma.com/design/zugcmml4tBCgrDOTkwPfJL/Conta-F%C3%A1cil---Prot%C3%B3tipo-das-Telas?node-id=0-1&t=B7xQ4FTEEm1ciRZe-1>.

2.1 Interface de Login

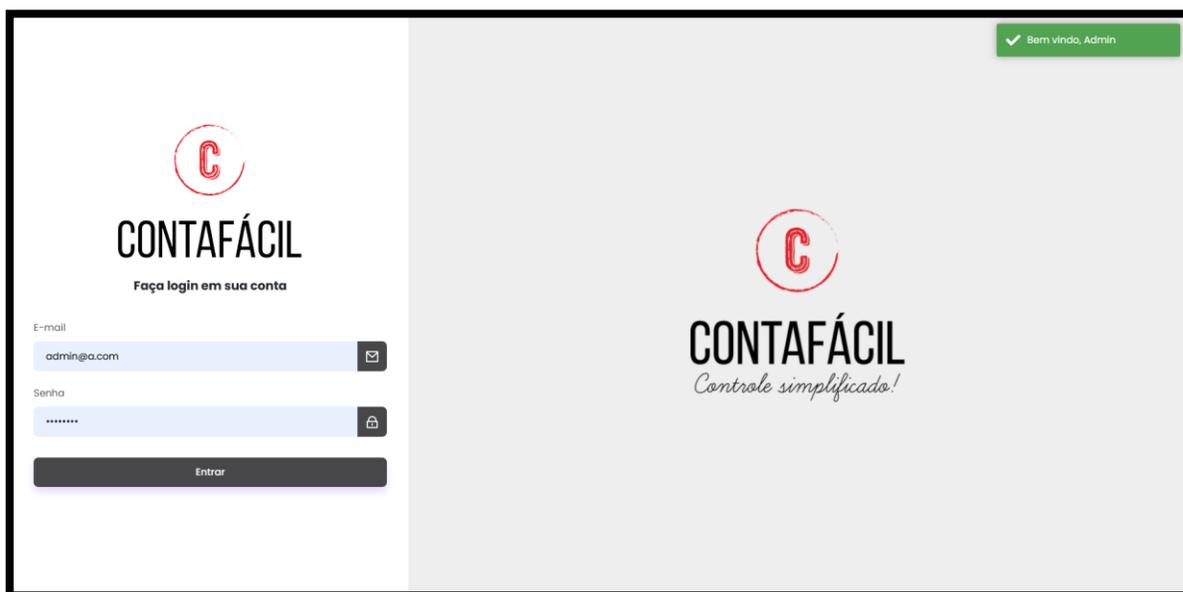
O caso de uso relacionado é o CSU1.

Figura 12 – Interface de Login com Mensagem de Sucesso



Fonte: Elaborado pelo Autor.

Figura 13 – Interface de Login com Mensagem de Erro

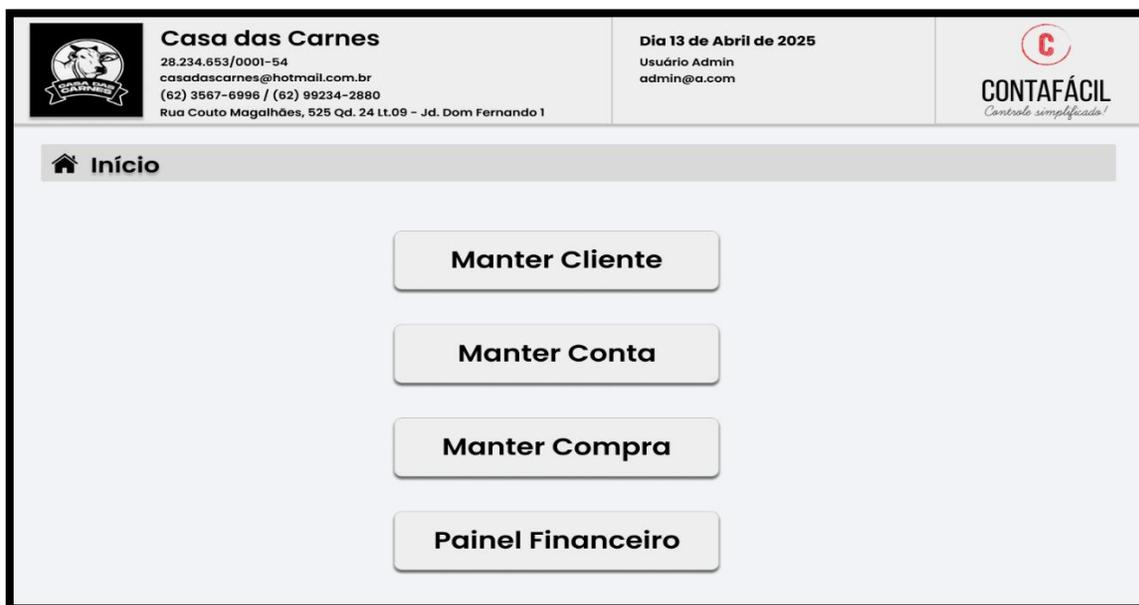


Fonte: Elaborado pelo Autor.

2.2 Painel principal

O caso de uso relacionado a essa interface é o CSU1.

Figura 14 – Painel Principal



Fonte: Elaborado pelo Autor.

2.3 Interface Manter Cliente

O caso de uso relacionado a essa interface é o CSU2.

Figura 15 – Interface de Manter Cliente (Consulta)

Código	CPF/CNPJ	Nome	E-mail	Endereço	Ação
1	999.999.999-99	Rodrigo Santos	rodst@email.com	Rua A-23- Vila Pedros	
2	999.999.999-99	Carlos Eduardo	cal@email.com	Rua B-21, Qd. 23 Lt.08	
3	999.999.999-99	João Silva	jao@email.com	Rua C-25, Qd. 23 Lt.08	
4	00.000.000/0001-00	Lanchonete do Marcos	lmarq@email.com	Rua D-13, Qd. 23 Lt.08	
5	00.000.000/0001-00	Restaurante Pedroso	restped@email.com	Rua E-53, Qd. 23 Lt.08	

Fonte: Elaborado pelo Autor.

Figura 16 – Interface de Manter Cliente (Cadastro)



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando 1

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com



CONTAFÁCIL
Controle simplificado!

← Manter Compra

Filtros

Código

CPF / CNPJ do Cliente

Nome do Cliente

Data da Compra

Consultar
Gerar Relatório
Novo

Código	Conta	Data da Compra	Cliente	Valor	Ação
1	5	12/12/2024	999.999.999-99 - Rodrigo Santos	158,55	
2	5	21/12/2024	999.999.999-99 - Rodrigo Santos	85,88	
3	3	11/01/2025	000.000.000-00 - João Silva	255,74	
4	11	01/03/2025	00.000.000/0001-00 - Lanchonete do Marcos	38,99	
5	18	18/02/2025	00.000.000/0001-00 - Restaurante Pedroso	1253,65	

Fonte: Elaborado pelo Autor.

Figura 19 – Interface de Manter Compra (Cadastro)



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando 1

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com



CONTAFÁCIL
Controle simplificado!

← Cadastro de Compra

Dados da Compra

Cliente*

Valor*

Data*

Descrição

Conta Vinculada

Código: 1

Status: ATIVO

Data Início: 01/04/2025

Data Fim:

Valor Total Da Conta: 500,17

Compras Efetuadas

Código	Data da Compra	Valor
5	01/01/2025	158,55
4	11/01/2025	85,88

Salvar
Cancelar

Fonte: Elaborado pelo Autor.

Figura 20 - Interface de Manter Compra (Cadastro de Compra sem Conta Vinculada)

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Cadastro de Compra

Dados da Compra

Cliente* 232.815.430-14 - João Silva **Valor*** 100,00 **Data*** 21/01/2025

Descrição

Conta

Atenção
Não existe uma conta em aberto para esse cliente.

Salvar Cancelar

Fonte: Elaborado pelo Autor.

Figura 21 - Interface de Manter Compra (Edição)

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Edição de Compra

Dados da Compra

Cliente* CPF / CNPJ - Nome **Valor*** 0,00 **Data*** DD/MM/AAAA

Descrição

Conta Vinculada

Código: 1
Status: ATIVO
Data Início: 01/04/2025
Data Fim:

Compras Efetuadas

Código	Data da Compra	Valor
5	01/01/2025	158,55
4	11/01/2025	85,88
7	21/01/2025	255,74
Valor Total		500,17

Salvar Excluir Cancelar

Fonte: Elaborado pelo Autor.

2.5 Interface Manter Conta

O caso de uso relacionados a essa interface e o CSU4.

Figura 22 – Interface de Manter Conta (Consulta)

Código	Status	Data Início	Data Fim	Cliente	Total	Ação
1	ENCERRADO	01/12/2024	27/12/2024	999.999.999-99 - Rodrigo Santos	233,55	
2	ATIVO	12/04/2025		999.999.999-99 - Rodrigo Santos	145,21	
3	ATIVO	03/04/2025		000.000.000-00 - João Silva	35,66	
4	CANCELADO	10/03/2025	11/03/2025	00.000.000/0001-00 - Lanchonete do Marcos	22,57	
5	ENCERRADO	01/02/2025	28/02/2025	00.000.000/0001-00 - Restaurante Pedroso	2562,35	

Fonte: Elaborado pelo Autor.

Figura 23 – Interface de Manter Conta (Cadastro)

Atenção
Não existem compras vinculadas a esta conta.

Fonte: Elaborado pelo Autor.

Figura 24 – Interface de Manter Conta (Edição)



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com



CONTA FÁCIL
Controle simplificado!

← Edição de Conta

Dados da Conta

Cliente* **Status***

Data Início* **Data Fim**

Compras Efetuadas

Código	Data da Compra	Valor
5	01/01/2025	158,55
4	11/01/2025	85,88
7	21/01/2025	255,74

Valor Total
500,17

Fonte: Elaborado pelo Autor.

Figura 25 – Interface de Manter Conta (Cancelamento)



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com



CONTAFÁCIL
Controle simplificado!

← Edição de Conta

Dados da Conta

Cliente*

Status*

Data Início*

Data Fim

Compras Efetuadas

Codigo	Data da Compra	Valor
5	01/01/2025	158,55
4	11/01/2025	85,88
7	21/01/2025	255,74

Valor Total

Cancelamento

Justificativa*

Fonte: Elaborado pelo Autor.

2.6 Painel Financeiro

O caso de uso relacionado a essa interface é o CSU7.

Figura 26 – Interface Painel Financeiro



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando 1

Dia 13 de Abril de 2025
Usuário Admin
admin@a.com



CONTAFÁCIL
Controle simplificado!

← **Painel Financeiro**

Filtros

Data Início

Data Fim

Contas em Aberto

R\$ 15.200,00

8 contas

Contas Vencidas

R\$ 1.200,00

4 contas

Ranking de Clientes

Mais Compraram	Menos Compraram
João Almeida	José Silva
Carlos Souza	Ronaldo Fernandes
Ana Pereira	Neymar Santos
Marcos Melo	Rubens Castro
Almir Santos	Fernando Fernandes

Contas Vencidas

Cliente	Conta	Vencimento
João Almeida	12	10/12/2024
Carlos Souza	22	31/01/2025
Ana Pereira	14	12/12/2012
Marcos Melo	13	28/11/2024
Almir Santos	7	29/11/2024

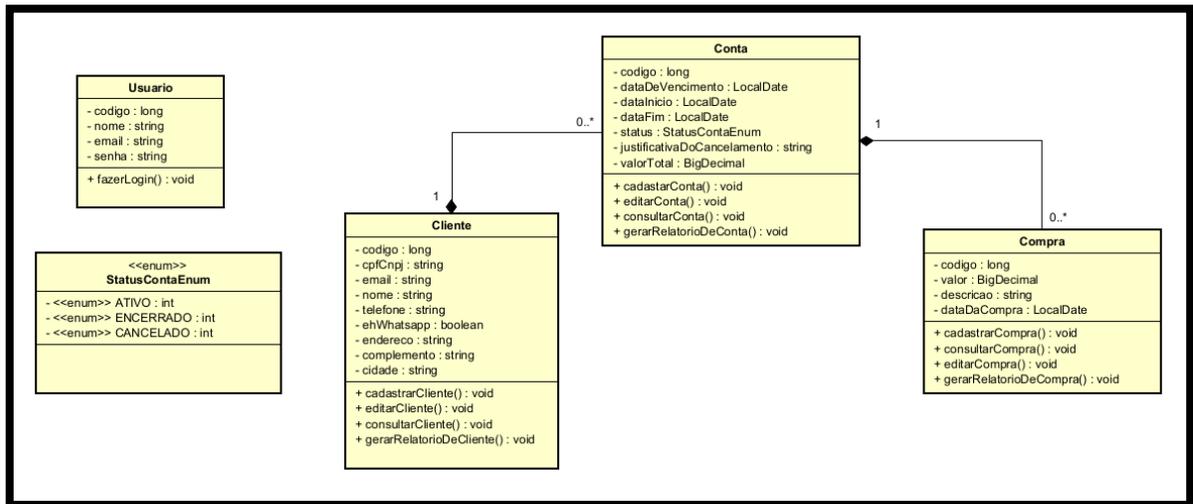
Fonte: Elaborado pelo Autor.

3. Diagrama UML

A Figura 27 apresenta o diagrama de classes da aplicação, permitindo uma visão geral da estrutura do sistema. Através dele, é possível identificar as principais classes, seus atributos, métodos e os relacionamentos entre elas.

O modelo contempla entidades como Cliente, Conta, Compra e Usuario, além do uso do enumerador StatusContaEnum para representar os estados possíveis de uma conta. Essa representação facilita a compreensão da lógica de negócio e contribui para uma codificação mais organizada e eficiente, servindo como base para o desenvolvimento e manutenção do sistema.

Figura 27- Diagrama de Classes



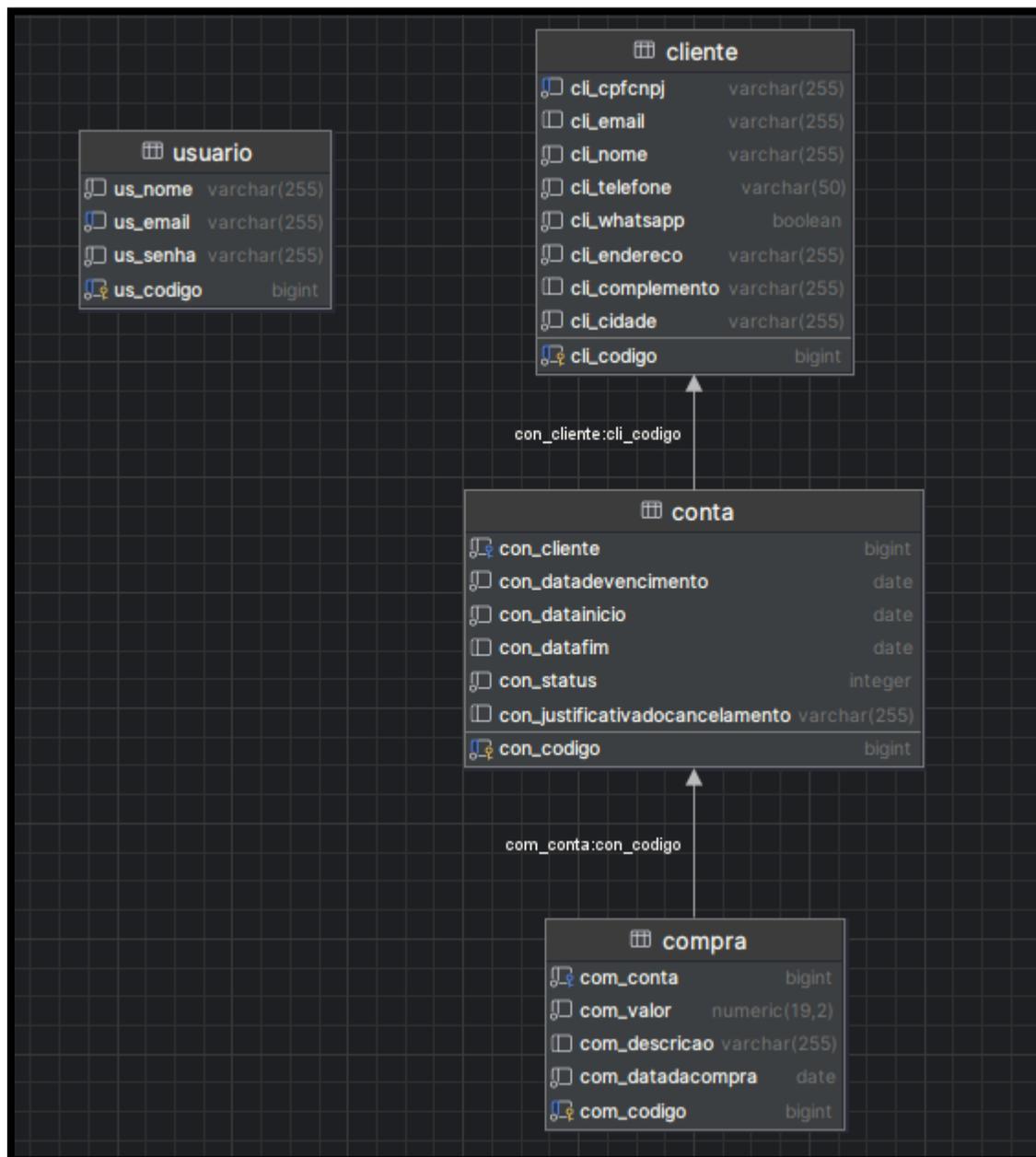
Fonte: Autor

4. Diagrama Relacionamento

A Figura 28 representa o diagrama de relacionamento entre as entidades do banco de dados. Por meio dessa visualização, é possível compreender de forma clara como as tabelas

estão estruturadas e interligadas. O diagrama evidencia os relacionamentos entre as entidades cliente, conta, compra e usuário, destacando as chaves primárias e estrangeiras utilizadas.

Figura 28 – Diagrama de Relacionamento



Fonte: Autor

APÊNDICE C – Casos de Testes

Identificação	CT01 – Login
Objetivo	Realiza a tentativa de login do funcionário
Data – Hora	01/06/2025 – 14:11
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Acessar a URL da aplicação (www.contafacilweb.com.br)
Passos	<ol style="list-style-type: none"> 1. Digitar no campo “E-mail”. 2. Digitar no campo “Senha”. 3. Clicar no botão Entrar.
Resultado Esperado	O sistema deve validar o e-mail e a senha informados, verificando se o usuário está cadastrado no banco de dados. Caso as credenciais estejam corretas, o <i>backend</i> deverá retornar um token de autenticação. Esse token será armazenado no <i>sessionStorage</i> do navegador e, em seguida, o usuário será redirecionado automaticamente para o painel principal da aplicação.
Resultado Real	O sistema efetuou a autenticação do usuário com sucesso, armazenou o token na variável de seção e redirecionou o usuário para o painel principal.
Evidências	Captura de tela com mensagem de sucesso após realizar o login e o token na variável de seção (figura 29).

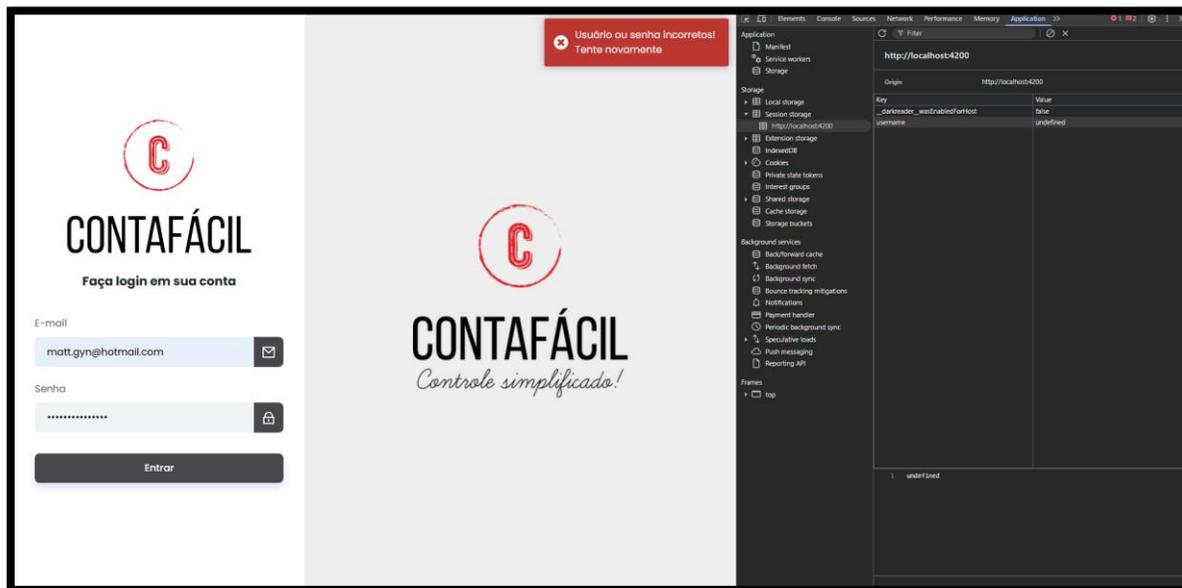
Figura 29 – Evidência do Teste CT01

The screenshot displays the Contafácil application interface and its developer tools. The application header includes the logo for 'Casa das Carnes' and contact information. A green notification banner reads 'Bem vindo, Mateus de Souza Almeida!'. The main content area features a 'Início' section with four buttons: 'Manter Cliente', 'Manter Conta', 'Manter Compra', and 'Painel Financeiro'. The developer tools on the right show the 'Application' tab with 'Session storage' expanded, displaying a token stored in the 'with-token' key.

Fonte: Autor

Identificação	CT02 – Login com Credenciais Inválidas
Objetivo	Verificar a resposta do sistema ao informar credenciais inválidas
Data – Hora	01/06/2025 – 14:17
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Acessar a URL da aplicação (www.contafacilweb.com.br)
Passos	<ol style="list-style-type: none"> 1. Informar um e-mail inválido ou inexistente 2. Informar uma senha incorreta 3. Clicar no botão “Entrar”
Resultado Esperado	O sistema exibe a mensagem: “Usuário ou senha incorretos! Tente novamente” e o usuário permanece na tela de login.
Resultado Real	O sistema exibiu a mensagem de erro corretamente e manteve o usuário na tela de login.
Evidências	Captura da tela com mensagem de erro retornado pelo <i>backend</i> e sem informação do token (figura 30).

Figura 30 – Evidência do Teste CT02



Fonte: Autor

Identificação	CT03 – Cadastrar Cliente
Objetivo	Realizar o cadastro de um cliente
Data – Hora	01/06/2025 – 14:21
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo”
Passos	<ol style="list-style-type: none"> 1. Digitar no campo “CPF/CNPJ” 2. Digitar no campo “E-mail” 3. Digitar no campo “Número de Telefone” 4. Selecionar no campo “Whatsapp” 5. Digitar no campo “Nome Completo”

	6. Digitar no campo “Endereço” 7. Digitar no campo “Complemento” 8. Digitar no campo “Cidade” 9. Clicar no botão “Salvar”
Resultado Esperado	O sistema salva os dados no banco e mostra a mensagem "Cliente salvo com sucesso".
Resultado Real	O sistema realizou o cadastro do cliente corretamente.
Evidências	Captura de tela mostrando a mensagem de sucesso (figura 31).

Figura 31 – Evidência do Teste CT3

Fonte: Autor

Identificação	CT04 – Editar Cliente
Objetivo	Alterar os dados de um cliente
Data – Hora	01/06/2025 – 14:26
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no ícone de edição na tabela de clientes
Passos	1. Modificar o campo “E-mail” 2. Modificar o campo “Número de Telefone” 3. Clicar no botão “Salvar”
Resultado Esperado	O sistema atualiza os dados no banco e exibe a mensagem “Cliente salvo com sucesso.”
Resultado Real	O sistema alterou os dados do cliente com sucesso e mostrou a mensagem.
Evidências	Captura de tela mostrando a mensagem de sucesso (figura 32).

Figura 32 – Evidência do Teste CT04

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 1 de junho de 2025
Usuário Admin
admin@a.com

✓ Cliente salvo com sucesso!

CONTAFÁCIL
Central simplificada

← Edição de Cliente

Dados do Cliente

CPF / CNPJ* 710.306.210-20 E-mail matt.gyn@hotmail.com Número de Telefone* (62) 99331-3212 É Whatsapp* Sim Não

Nome Completo* Mateus de Souza Almeida

Endereço* Rua Couto Magalhães, 525 Complemento Qd. 24, Lt.09 Cidade* Goiânia

Salvar Excluir Cancelar

Fonte: Autor

Identificação	CT05 – Excluir Cliente
Objetivo	Remover um cliente do sistema
Data – Hora	01/06/2025 – 14:30
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no ícone de edição de um cliente e em seguida no botão “Excluir Cliente”
Passos	<ol style="list-style-type: none"> 1. Clicar no botão “Excluir Cliente” 2. Confirmar a exclusão na janela “Deseja realmente excluir este cliente?”
Resultado Esperado	O sistema remove o cliente, mostra a mensagem “Cliente excluído com sucesso” e redireciona o usuário para a tela de consulta.
Resultado Real	O sistema excluiu o cliente do banco e mostrou a mensagem.
Evidências	Captura de tela mostrando a mensagem de sucesso (figura 33).

Figura 33 – Evidência do Teste CT05

Fonte: Autor

Identificação	CT06 – Consultar Cliente com Filtros
Objetivo	Filtrar clientes cadastrados usando um ou mais filtros
Data – Hora	01/06/2025 – 14:34
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Manter Cliente” no painel principal
Passos	<ol style="list-style-type: none"> 1. Digitar no campo “CPF/CNPJ” 2. Clicar no botão “Consultar”
Resultado Esperado	O sistema exibe os resultados conforme o filtro aplicado.
Resultado Real	O sistema realizou a consulta utilizando o filtro preenchido e exibiu o resultado correspondente.
Evidências	Captura de tela com o resultado da consulta(figura 34).

Figura 34 – Evidência do Teste CT06

Código	CPF/CNPJ	Nome	E-Mail	Endereço	Ação
27	230.518.540-50	Gabriela Geraldina	gg@gmail.com	Rua castro	

Fonte: Autor

Identificação	CT07 – Consultar Cliente Sem Resultados
Objetivo	Verificar a resposta do sistema quando a consulta não retorna dados
Data – Hora	01/06/2025 – 14:36
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Acessar a tela principal de consulta de clientes
Passos	<ol style="list-style-type: none"> 1. Digitar um CPF/CNPJ inexistente 2. Clicar no botão “Consultar”
Resultado Esperado	O sistema exibe a mensagem “Não há dados disponíveis para exibir.”
Resultado Real	O sistema exibiu a mensagem corretamente.
Evidências	Captura de tela mostrando a mensagem (figura 35).

Figura 35 – Evidência do Teste CT07

The screenshot shows the 'Manter Cliente' interface. At the top left, there is a logo for 'Casa das Carnes' with contact information. The top right shows the date 'Dia 1 de junho de 2025' and user information 'Usuário Admin' and 'admin@ca.com'. The main header is '← Manter Cliente'. Below it, there is a 'Filtros' section with three input fields: 'Codigo', 'CPF / CNPJ' (with the value '00.000.000/0000-00'), and 'Nome'. To the right of these fields are three buttons: 'Consultar', 'Gerar Relatório', and 'Novo'. The main content area below the filters is empty except for the message 'Não há dados disponíveis para exibir.'

Fonte: Autor

Identificação	CT08 – Gerar Relatório de Clientes
Objetivo	Gerar e exportar um relatório PDF com os dados da consulta
Data – Hora	01/06/2025 – 14:38
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Acessar a tela principal de consulta de clientes
Passos	<ol style="list-style-type: none"> 1. Aplicar um filtro ou não 2. Clicar no botão “Gerar Relatório”
Resultado Esperado	O sistema gera e baixa um arquivo PDF com os dados da consulta.
Resultado Real	O sistema exportou um pdf com o relatório dos clientes
Evidências	Captura de tela mostrando o relatório gerado (figura 36).

Figura 36 – Evidência do Teste CT08

RELATÓRIO DE CLIENTES				
Código	CPF/CNPJ	Nome	E-Mail	Endereço
14	33597634000122	ROBERTO OTAIR FERREIRA SW	rob@gmail.com	Rua Couto Magalhães, 525
17	33597634000123	Alberto Silva	sou-00@hotmail.com	adsadasdasd
18	01393953000120	Irineu Castro	qwb@gmail.com	Rua C234
20	26736085080	Fernando Pereira	fer@gmail.com	rua 8
21	69930932003	Pereira Fernandes	per@gmail.com	Rua 76
22	06866975026	Bira do Posto		Rua do posto
23	26125855082	Cristiano Lopes		Rua LP21
24	03272086005	Alice Fernandes		Rua da esuqina
25	77468354065	Hamburgueria Do Silvera	hsbg@gmail.com	Rua GG-34
26	12315236037	Pedro Pedrosa		Rua PP23
27	23051854050	Gabriela Geraldina	gg@gmail.com	Rua castro

Fonte: Autor

Identificação	CT09 – Cadastro de Cliente com Campos Obrigatórios Vazios
Objetivo	Verificar o comportamento do sistema ao tentar salvar um cliente com campos obrigatórios em branco
Data – Hora	01/06/2025 – 14:40
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo”
Passos	<ol style="list-style-type: none"> 1. Deixar o campo “Nome Completo” vazio 2. Preencher os demais campos obrigatórios 3. Clicar no botão “Salvar”
Resultado Esperado	O sistema destaca o campo vazio em vermelho e exibe a mensagem “Preencha todos os campos obrigatórios.”
Resultado Real	O sistema destacou os campos e exibiu a mensagem corretamente.
Evidências	Captura de tela mostrando a mensagem (figura 37).

Figura 37 – Evidência do Teste CT09

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

localhost:4200 diz
Preencha todos os campos obrigatórios.

Dia 3 de junho de 2025
Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Cadastro de Cliente

Dados do Cliente

CPF / CNPJ* 33.597.634/0001-22 E-mail Número de Telefone* (11) 1111-1111 É Whatsapp* Sim Não

Nome Completo* O Nome é obrigatório.

Endereço* Rua Couto Magalhães, 525 Complemento Cidade* Goiânia

Salvar Cancelar

Fonte: Autor

Identificação	CT10 – Cadastro de Cliente com CPF/CNPJ Duplicado
Objetivo	Testar a tentativa de cadastrar um cliente com CPF/CNPJ já existente
Data – Hora	01/06/2025 – 14:42
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo”
Passos	<ol style="list-style-type: none"> 1. Digitar CPF/CNPJ de um cliente já existente 2. Preencher os demais campos 3. Clicar no botão “Salvar”
Resultado Esperado	O sistema exibe a mensagem “Erro ao salvar cliente.”
Resultado Real	O sistema exibiu a mensagem corretamente
Evidências	Captura de tela mostrando a mensagem de erro (figura 38).

Figura 38 – Evidência do Teste CT10

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025 **Erro ao salvar cliente.**

Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Cadastro de Cliente

Dados do Cliente

CPF / CNPJ* 33.597.634/0001-22 E-mail Número de Telefone* (62) 99331-3212 É Whatsapp* Sim Não

Nome Completo* teste

Endereço* Rua Couto Magalhães, 525 Complemento Qd. 24, Lt.09 Cidade* Goiânia

Salvar Cancelar

Fonte: Autor

Identificação	CT11 – Exclusão de Cliente Vinculado a Conta
Objetivo	Tentar excluir um cliente que possui vínculo com uma conta
Data – Hora	01/06/2025 – 14:45
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no ícone de edição de um cliente com vínculo e depois em “Excluir Cliente”
Passos	1. Clicar no botão “Excluir Cliente” 2. Confirmar a exclusão
Resultado Esperado	O sistema exibe a mensagem “Erro ao excluir cliente.”
Resultado Real	O sistema mostrou a mensagem corretamente.
Evidências	Captura de tela mostrando a mensagem de erro (figura 39).

Figura 39 – Evidência do Teste CT11

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025 Erro ao excluir cliente.
Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Edição de Cliente

Dados do Cliente

CPF / CNPJ* 33.597.634/0001-22 E-mail rob@gmail.com Número de Telefone* (62) 99331-3212 É Whatsapp* Sim Não

Nome Completo* ROBERTO OTAIR FERREIRA SW

Endereço* Rua Couto Magalhães, 525 Complemento Qd. 24, Lt.09 Cidade* Goiânia

Salvar Excluir Cancelar

Fonte: Autor

Identificação	CT12 – Editar Compra
Objetivo	Alterar os dados de uma compra já cadastrada
Data – Hora	01/06/2025 – 14:47
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no ícone de edição de uma compra
Passos	1. Alterar o campo “Data” 2. Alterar o campo “Descrição” 3. Clicar no botão “Salvar”
Resultado Esperado	O sistema salva as alterações, mostra a mensagem “Cliente salvo com sucesso” e redireciona o usuário para a tela anterior.
Resultado Real	O sistema salvou as alterações no banco e exibiu a mensagem de sucesso e redirecionou o usuário para a tela anterior.
Evidências	Captura de tela mostrando a mensagem de sucesso (figura 40).

Figura 40 – Evidência do Teste CT12

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com

✓ Compra alterada com sucesso!
CONTAFÁCIL
Controle simplificado!

← Manter Compra

Filtros
Código CPF / CNPJ Nome

Consultar Gerar Relatório Novo

Código	Conta	Data da Compra	Cliente	Valor	Ação
7	4	2025-06-01	33597634000122 - ROBERTO OTAIR FERREIRA SW	R\$12.21	[Ícone]
8	11	2025-06-01	33597634000123 - Alberto Silva	R\$15.15	[Ícone]
9	15	2025-06-01	03272086005 - Alice Fernandes	R\$85.78	[Ícone]
10	13	2025-06-01	12315236037 - Pedro Pedrosa	R\$154.23	[Ícone]
6	4	2025-05-27	33597634000122 - ROBERTO OTAIR FERREIRA SW	R\$34.47	[Ícone]

Fonte: Autor

Identificação	CT13 – Excluir Compra
Objetivo	Remover uma compra do sistema
Data – Hora	01/06/2025 – 14:49
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no ícone de edição e, em seguida, em “Excluir Compra”
Passos	<ol style="list-style-type: none"> 1. Clicar em “Excluir Compra” 2. Confirmar a exclusão quando o sistema solicitar
Resultado Esperado	O sistema remove a compra, exibe a mensagem “Compra excluída!” e redireciona o usuário para a tela de consulta.
Resultado Real	O sistema excluiu a compra, mostrou a mensagem de sucesso e redirecionou o usuário para a tela de consulta.
Evidências	Captura de tela mostrando a mensagem de sucesso após a exclusão do registro (figura 41).

Figura 41 – Evidência do Teste CT13

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025
Usuário Admin
admin@ca.com

✓ Compra excluída com sucesso!
CONTAFÁCIL
Controle simplificado!

← Manter Compra

Filtros
Código CPF / CNPJ Nome

Consultar Gerar Relatório Novo

Não há dados disponíveis para exibir.

Fonte: Autor

Identificação	CT14 – Consultar Compras com Filtro
Objetivo	Consultar compras cadastradas usando filtros
Data – Hora	01/06/2025 – 14:52
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Manter Compra” no painel principal
Passos	1. Preencher o campo “CPF/CNPJ” 2. Clicar no botão “Consultar”
Resultado Esperado	O sistema exibe apenas as compras que correspondem aos filtros aplicados.
Resultado Real	O sistema exibiu os resultados de acordo com os filtros preenchidos.
Evidências	Captura de tela mostrando os filtros aplicados e o resultado obtido (figura 42).

Figura 42 – Evidência do Teste CT14

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025
Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Manter Compra

Filtros

Código CPF / CNPJ Nome

Consultar **Gerar Relatório** **Novo**

Código	Conta	Data da Compra	Cliente	Valor	Ação
6	4	2025-05-27	33597634000122 - ROBERTO OTAIR FERREIRA SW	R\$334.23	
7	4	2025-06-01	33597634000122 - ROBERTO OTAIR FERREIRA SW	R\$12.21	

Fonte: Autor

Identificação	CT15 – Gerar Relatório de Compras
Objetivo	Exportar os dados das compras para um arquivo PDF
Data – Hora	01/06/2025 – 14:54
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Manter Compra” no painel principal
Passos	1. Clicar no botão “Gerar Relatório”
Resultado Esperado	O sistema exporta um arquivo PDF com os dados da consulta.
Resultado Real	O sistema exportou o relatório em formato pdf com os dados da consulta.
Evidências	Captura de tela mostrando o relatório gerado (figura 43).

Figura 43 – Evidência do Teste CT15

RELATÓRIO DE COMPRAS				
Código	Conta	Data da Compra	Cliente	Valor
7	4	31/05/2025	ROBERTO OTAIR FERREIRA SW	R\$ 12,21
8	11	31/05/2025	Alberto Silva	R\$ 15,15
9	15	31/05/2025	Alice Fernandes	R\$ 85,78
10	13	31/05/2025	Pedro Pedrosa	R\$ 154,23
6	4	26/05/2025	ROBERTO OTAIR FERREIRA SW	R\$ 34,47

Fonte: Autor

Identificação	CT16 – Cadastro com Cliente com Conta Vencida
Objetivo	Testar a tentativa de cadastrar compra para cliente com conta vencida
Data – Hora	01/06/2025 – 14:56
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo”
Passos	<ol style="list-style-type: none"> 2. Selecionar um cliente com conta vencida 3. Digitar o campo “Valor” 4. Preencher o campo “Data” 5. Clicar em “Salvar”
Resultado Esperado	O sistema exibe a mensagem “Este cliente possui uma conta vencida. Verifique!”.
Resultado Real	O sistema mostrou a mensagem corretamente.
Evidências	Captura de tela mostrando a mensagem (figura 44).

Figura 44 – Evidência do teste CT16



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025

Usuário Admin
admin@a.com

Este cliente possui uma conta vencida. Verifique!



CONTAFÁCIL
Controle simplificado!

← **Cadastro de Compra**

Dados da Conta

Cliente* Valor* Data*

Descrição

Conta Vinculada

Atenção

Não existe uma conta em aberto para esse cliente.

Fonte: Autor

Identificação	CT17 – Cadastro com Campos Obrigatórios Não Preenchidos
Objetivo	Verificar o comportamento do sistema ao tentar cadastrar compra sem preencher os campos obrigatórios
Data – Hora	01/06/2025 – 15:00
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo”
Passos	<ol style="list-style-type: none"> 1. Deixar o campo “Valor” em branco 2. Deixar o campo “Data” em branco 3. Clicar no botão “Salvar”
Resultado Esperado	O sistema destaca os campos obrigatórios em vermelho e exibe a mensagem “Preencha todos os campos obrigatórios.”
Resultado Real	O sistema destacou os campos obrigatórios e exibiu a mensagem corretamente.
Evidências	Captura de tela mostrando os campos e a mensagem.

Identificação	CT18 – Cadastrar Compra
Objetivo	Realizar o cadastro de uma compra válida associada a um cliente
Data – Hora	01/06/2025 – 15:03
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Manter Compra” no painel principal e depois no botão “Nova Compra”
Passos	<ol style="list-style-type: none"> 1. Selecionar cliente no campo ‘Cliente’

	<ol style="list-style-type: none"> 2. Digitar em 'Valor': 250,00 3. Digitar em 'Data': 07/05/2025 4. Digitar em 'Descrição': Compra de materiais de escritório 5. Clicar no botão 'Salvar'
Resultado Esperado	O sistema salva os dados no banco e mostra a mensagem "Compra salva com sucesso"
Resultado Real	O sistema realizou o cadastro da compra e mostrou a mensagem de sucesso
Evidências	Captura de tela mostrando a mensagem de sucesso após o cadastro e mostrando a conta na tela de consulta (figura 45).

Figura 45 – Evidência do Teste CT18

Código	Conta	Data da Compra	Cliente	Valor	Ação
7	4	2025-06-01	33597634000122 - ROBERTO OTAIR FERREIRA SW	RS12,21	
8	11	2025-06-01	33597634000123 - Alberto Silva	RS15,15	
9	15	2025-06-01	03272086005 - Alice Fernandes	RS85,78	
10	13	2025-06-01	12315236037 - Pedro Pedrosa	RS154,23	
6	4	2025-05-27	33597634000122 - ROBERTO OTAIR FERREIRA SW	RS34,47	
11	16	2025-06-03	33597634000122 - ROBERTO OTAIR FERREIRA SW	RS250,00	

Fonte: Autor

Identificação	CT19 – Cadastrar Conta
Objetivo	Realizar o cadastro de uma nova conta válida para um cliente sem conta ativa
Data – Hora	01/06/2025 – 15:06
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Acessar o painel principal e clicar no botão “Manter Conta”, em seguida clicar em “Novo”
Passos	<ol style="list-style-type: none"> 1. Selecionar cliente no campo ‘Cliente’ 2. Digitar em ‘Data Início’: 07/06/2025 3. Verificar que o campo ‘Data de Vencimento’ foi preenchido automaticamente com 07/06/2025 4. Verificar que o campo ‘Status’ está preenchido como ATIVO 5. Clicar no botão ‘Salvar’
Resultado Esperado	O sistema armazena os dados da conta, exibe a mensagem "Conta salva com sucesso" e retorna para a tela de consulta com a nova conta listada.

Resultado Real	O sistema realizou o cadastro da conta com sucesso e mostrou a mensagem.
Evidências	Captura de tela mostrando a mensagem de sucesso após clicar em salvar (figura 46).

Figura 46 – Evidência do Teste CT19

Casa das Carnes
28.234.853/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6998 / (62) 99234-2880
Rua Cauto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com

Conta salva com sucesso!

CONTAFÁCIL
Controle simplificado!

← Manter Conta

Filtros

Código CPF / CNPJ Nome do Cliente

Data Início Data Fim

dt/mm/aaaa dt/mm/aaaa

Consultar Gerar Relatório Novo

Código	Status	Data início	Data Fim	Cliente	Ação
11	CANCELADO	2025-06-01		33597634000123 - Alberto Silva	
12	ATIVO	2025-06-01		23051854050 - Gabriela Geraldina	
14	ATIVO	2025-06-01		77468354065 - Hamburgueria Do Silvera	
15	ATIVO	2025-06-01		03272086005 - Alice Fernandes	
13	ATIVO	2025-02-01		12315236037 - Pedro Pedrosa	
4	ENCERRADO	2025-05-25	2025-06-10	33597634000122 - ROBERTO OTAIR FERREIRA SW	
16	ATIVO	2025-06-03		33597634000122 - ROBERTO OTAIR FERREIRA SW	
17	ATIVO	2025-06-03		69930932003 - Pereira Fernandes	

Fonte: Autor

Identificação	CT20 – Editar Conta
Objetivo	Validar a edição de uma conta com dados válidos
Data – Hora	01/06/2025 – 15:08
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta" no painel principal e clicar no ícone de edição de uma conta existente
Passos	<ol style="list-style-type: none"> 1. Modificar campo 'Data Fim': 10/06/2025 2. Clicar no botão 'Salvar'
Resultado Esperado	O sistema atualiza os dados da conta, exibe a mensagem "Conta salva com sucesso" e retorna para a tela de consulta.
Resultado Real	O sistema fez a alteração da conta no banco.
Evidências	Captura de tela mostrando o antes e depois da conta (figura 47 e 48).

Figura 47 – Evidência do teste CT20 (antes)



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6986 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com

Conta salva com sucesso!



← **Manter Conta**

Filtros

Código CPF / CNPJ Nome do Cliente

Data Início Data Fim

Código	Status	Data início	Data Fim	Cliente	Ação
11	CANCELADO	2025-06-01		33597634000123 - Alberto Silva	
12	ATIVO	2025-06-01		23051854050 - Gabriela Geraldina	
14	ATIVO	2025-06-01		77468354065 - Hamburgueria Da Silveira	
15	ATIVO	2025-06-01		03272086005 - Alice Fernandes	
13	ATIVO	2025-02-01		12315236037 - Pedro Pedrosa	
4	ENCERRADO	2025-05-25	2025-06-10	33597634000122 - ROBERTO OTAIR FERREIRA SW	
16	ATIVO	2025-06-03		33597634000122 - ROBERTO OTAIR FERREIRA SW	
17	ATIVO	2025-06-03		69930932003 - Pereira Fernandes	

Fonte: Autor

Figura 48 – Evidência do Teste CT20 (depois)



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6986 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com



← **Manter Conta**

Filtros

Código CPF / CNPJ Nome do Cliente

Data Início Data Fim

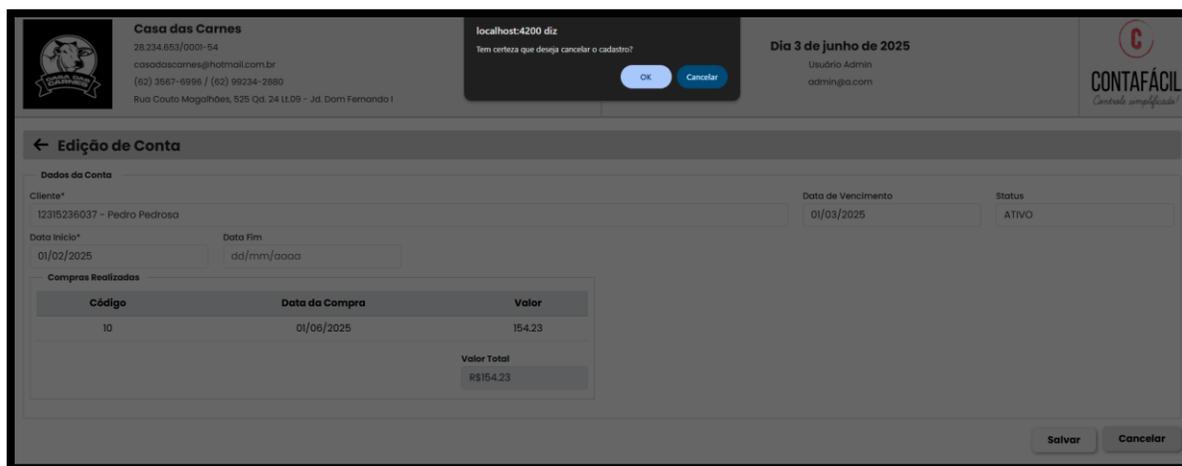
Código	Status	Data início	Data Fim	Cliente	Ação
11	CANCELADO	2025-06-01		33597634000123 - Alberto Silva	
12	ATIVO	2025-06-01		23051854050 - Gabriela Geraldina	
14	ATIVO	2025-06-01		77468354065 - Hamburgueria Da Silveira	
15	ATIVO	2025-06-01		03272086005 - Alice Fernandes	
13	ATIVO	2025-02-01		12315236037 - Pedro Pedrosa	
4	ENCERRADO	2025-05-25	2025-06-10	33597634000122 - ROBERTO OTAIR FERREIRA SW	
16	ATIVO	2025-06-03		33597634000122 - ROBERTO OTAIR FERREIRA SW	
17	ATIVO	2025-06-03	2025-06-03	69930932003 - Pereira Fernandes	

Fonte: Autor

Identificação	CT21 – Cancelar Edição de Conta
Objetivo	Verificar se o sistema descarta alterações ao cancelar a edição
Data – Hora	01/06/2025 – 15:10
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta" e clicar no ícone de edição de uma conta
Passos	<ul style="list-style-type: none"> Modificar campo 'Data Fim': 10/06/2025 Clicar no botão 'Cancelar'

Resultado Esperado	O sistema descarta as alterações e retorna para a tela de consulta sem modificar a conta.
Resultado Real	O sistema descarta as informações alteradas.
Evidências	Capturas de tela mostrando a mensagem de confirmação para cancelar o cadastro (figura 49).

Figura 49 – Evidência do Teste CT21



Fonte: Autor

Identificação	CT22 – Consultar Contas com Filtros Preenchidos
Objetivo	Validar a aplicação de filtros na consulta de contas
Data – Hora	01/06/2025 – 15:11
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta" no painel principal
Passos	<ol style="list-style-type: none"> 1. Digitar em 'Código' 2. Digitar em 'CPF/CNPJ' 3. Digitar em 'Cliente' 4. Digitar em 'Data Início' 5. Clicar no botão 'Consultar'
Resultado Esperado	Sistema exibe na tabela os registros que atendem aos filtros preenchidos.
Resultado Real	O sistema exibiu o resultado da consulta com os filtros corretamente.
Evidências	Captura de tela mostrando o resultado da consultas com os filtros (figura 50).

Figura 50 – Evidência do Teste CT22

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(02) 3567-6998 / (02) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com

CONTAFÁCIL
Controle simplificado

← Manter Conta

Filtros

Código: 12 CPF / CNPJ: 230.518.540-50 Nome do Cliente: Gabriela Geraldina

Data início: 01/05/2025 Data fim: da/mm/aaaa

Consultar Gerar Relatório Novo

Código	Status	Data início	Data fim	Cliente	Ação
12	ATIVO	2025-06-01		23051854050 - Gabriela Geraldina	

Fonte: Autor

Identificação	CT23 – Consultar Contas Sem Retorno
Objetivo	Validar a mensagem de retorno vazio na consulta
Data – Hora	01/06/2025 – 15:13
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta"
Passos	<ol style="list-style-type: none"> 1. Preencher filtros com dados que não correspondem a nenhuma conta. 2. Clicar no botão 'Consultar'.
Resultado Esperado	O sistema exibe a mensagem “Não há dados disponíveis para exibir.”
Resultado Real	O sistema mostrou a mensagem corretamente.
Evidências	Captura de tela mostrando a mensagem (figura 51).

Figura 51 – Evidência do Teste CT23

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(02) 3567-6998 / (02) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com

CONTAFÁCIL
Controle simplificado

← Manter Conta

Filtros

Código: 2131232 CPF / CNPJ: Nome do Cliente:

Data início: da/mm/aaaa Data fim: da/mm/aaaa

Consultar Gerar Relatório Novo

Não há dados disponíveis para exibir.

Fonte: Autor

Identificação	CT24 – Gerar Relatório de Contas
Objetivo	Validar exportação do relatório de contas em PDF
Data – Hora	01/06/2025 – 15:14
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta"
Passos	<ol style="list-style-type: none"> 1. Preencher filtros conforme necessário 2. Clicar no botão ‘Gerar Relatório’
Resultado Esperado	Sistema gera e realiza o download de um arquivo .pdf com os dados da consulta
Resultado Real	O sistema exportou o relatório em formato pdf com os dados corretamente.
Evidências	Captura de tela do relatório gerado (figura 52).

Figura 52 – Evidência do Teste CT24

RELATÓRIO DE CONTAS				
Código	Status	Data início	Data Fim	Cliente
11	CANCELADO	31/05/2025		Alberto Silva
12	ATIVO	31/05/2025		Gabriela Geraldina
14	ATIVO	31/05/2025		Hamburgueria Do Silvera
15	ATIVO	31/05/2025		Alice Fernandes
13	ATIVO	31/01/2025		Pedro Pedrosa
4	ENCERRADO	24/05/2025	09/06/2025	ROBERTO OTAIR FERREIRA SW
16	ATIVO	02/06/2025		ROBERTO OTAIR FERREIRA SW

Fonte: Autor

Identificação	CT25 – Cadastro com Campos Obrigatórios Não Preenchidos
Objetivo	Verificar a resposta do sistema ao tentar salvar uma conta sem preencher campos obrigatórios
Data – Hora	01/06/2025 – 15:15
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta" e clicar em “Nova Conta”
Passos	<ol style="list-style-type: none"> 1. Deixar o campo ‘Data Início’ em branco 2. Clicar no botão ‘Salvar’
Resultado Esperado	Sistema destaca os campos obrigatórios em vermelho e exibe a mensagem “Preencha todos os campos obrigatórios.”
Resultado Real	O sistema destacou o campo obrigatório que não foi preenchido e exibiu a mensagem.
Evidências	Captura de tela mostrando as mensagens (figura 53).

Figura 53 – Evidência do Teste CT25

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

localhost:4200 diz
Preencha todos os campos obrigatórios.

← Cadastro de Conta

Dados da Conta

Cliente* 33597634000122 - ROBERTO OTAIR FERREIRA SW Data de Vencimento 02/07/2025 Status

Data Início* dd/mm/aaaa Data Fim dd/mm/aaaa

A Data Início é obrigatória.

Compras

Atenção
Não existem compras vinculadas a esta conta.

Salvar Cancelar

Fonte: Autor

Identificação	CT26 – Cadastro com Cliente Já Possuindo Conta Ativa
Objetivo	Garantir que o sistema não permita cadastro duplicado de conta ativa para o mesmo cliente
Data – Hora	01/06/2025 – 15:17
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo” na interface Manter Conta
Passos	<ol style="list-style-type: none"> 1. Selecionar cliente com conta ativa já cadastrada 2. Preencher outros dados normalmente 3. Clicar em ‘Salvar’
Resultado Esperado	Sistema exibe a mensagem: “Este cliente já possui uma conta ativa. Verifique!”
Resultado Real	Ao tentar salvar o sistema exibe a mensagem corretamente
Evidências	Captura de tela (figura 54) mostrando a mensagem de erro.

Figura 54 – Evidência do teste CT26

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025
Usuário Admin
admin@a.com

Erro: Este cliente já possui uma conta ativa. Verifique!

CONTAFÁCIL
Controle simplificado!

← Cadastro de Conta

Dados da Conta

Cliente* 33597634000122 - ROBERTO OTAIR FERREIRA SW Data de Vencimento 02/07/2025 Status

Data Início* 02/06/2025 Data Fim dd/mm/aaaa

Compras

Atenção
Não existem compras vinculadas a esta conta.

Salvar Cancelar

Fonte: Autor

Identificação	CT27 – Cadastro com Cliente que Possui Conta Vencida Não Paga
Objetivo	Verificar alerta ao cadastrar conta para cliente com conta vencida não paga.
Data – Hora	01/06/2025 – 15:19
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão “Novo” na interface Manter Conta
Passos	<ol style="list-style-type: none"> 1. Selecionar cliente com conta vencida e não paga 2. Preencher dados restantes 3. Clicar em ‘Salvar’
Resultado Esperado	Sistema exibe a mensagem: “Este cliente possui uma conta vencida. Verifique!”
Resultado Real	O sistema exibe a mensagem corretamente.
Evidências	Captura de tela (figura 55) com a mensagem.

Figura 55 – Evidência do teste CT27

Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 2 de junho de 2025
Usuário Admin
admin@a.com

CONTAFÁCIL
Controle simplificado!

← Cadastro de Conta

Dados da Conta

Cliente* 12315236037 - Pedro Pedrosa Data de Vencimento 02/07/2025 Status

Data Início* 02/06/2025 Data Fim dd/mm/aaaa

Compras

Atenção
Não existem compras vinculadas a esta conta.

Salvar Cancelar

Fonte: Autor

Identificação	CT28 – Calcular Total de Compras
Objetivo	Verificar se o sistema calcula corretamente o valor total das compras vinculadas à conta após o preenchimento do campo de valor de uma nova compra.
Data – Hora	01/06/2025 – 15:22
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Manter Conta" no painel principal e selecionar uma conta existente para edição.
Passos	<ol style="list-style-type: none"> 1. Clicar no botão "Editar" na linha correspondente à conta desejada. 2. Clicar no botão "Nova Compra". 3. Digitar em 'Valor': "200,00". 4. Perder o foco do campo 'Valor'.
Resultado Esperado	O campo "Valor Total" é atualizado automaticamente com o somatório das compras vinculadas à conta, incluindo a nova compra registrada.
Resultado Real	Ao remover o foco do campo Valor a soma das compras anteriores e da nova compra é efetuada e a previa é exibida no campo Valor Total.
Evidências	Captura de tela (figura 56) com o somatório do valor.

Figura 56 – Evidência do teste CT28



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 Lt.09 - Jd. Dom Fernando I

Dia 1 de junho de 2025

Usuário Admin
admin@a.com



CONTAFÁCIL
Controle simplificado!

← Cadastro de Compra

Dados da Conta

Cliente* Valor* Data*

Descrição

Conta Vinculada

Código: 4
Status: ATIVO
Data Início: 2025-05-25
Data Fim:

Compras Efetuadas

Código	Data da Compra	Valor
6	27/05/2025	334.23
4	26/05/2025	231.21
7	01/06/2025	12.21
Valor Total		R\$777.65

Fonte: Autor

Identificação	CT29 – Visualizar Painel Financeiro
Objetivo	Verificar se o funcionário consegue visualizar o painel financeiro com as métricas corretas e se o sistema reage adequadamente à seleção de período.
Data – Hora	01/06/2025 – 15:25
Responsável	Mateus de Souza Almeida
Procedimento Inicial	Clicar no botão "Painel Financeiro" no painel principal.
Passos	<ol style="list-style-type: none"> 1. Digitar no campo "Data Início" o período desejado. 2. Digitar no campo "Data Fim" o período desejado. 3. Clicar no botão "Consultar" para atualizar os resultados.
Resultado Esperado	O painel exibe corretamente as informações financeiras atualizadas com base no período selecionado, como total de contas abertas, contas vencidas e valores totais.
Resultado Real	Os quadros do painel financeiro foram exibidos, apresentando as informações corretas
Evidências	Captura da tela (figura 57) mostrando os dados do painel.

Figura 57 – Evidência do teste CT29



Casa das Carnes
28.234.653/0001-54
casadascarnes@hotmail.com.br
(62) 3567-6996 / (62) 99234-2880
Rua Couto Magalhães, 525 Qd. 24 11.09 - Jd. Dom Fernando I

Dia 3 de junho de 2025
Usuário Admin
admin@ca.com



CONTAFÁCIL
Controle simplificado!

← Painel Financeiro

Filtros

Data Início: Data Fim: Consultar

Contas em Aberto
R\$397.61 4 contas

Contas Vencidas
R\$154.23 1 contas

Ranking de Clientes

Mais Compraram	Menos Compraram
ROBERTO OTAIR FERREIRA SW	Alberto Silva
Pedro Pedrosa	Alice Fernandes
Alice Fernandes	Pedro Pedrosa
Alberto Silva	ROBERTO OTAIR FERREIRA SW

Contas Vencidas

Cliente	Conta	Vencimento
Pedro Pedrosa	13	01/03/2025

Fonte: Autor



**PUC
GOIÁS**

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Mateus de Souza Almeida
do Curso de Engenharia de Computação, matrícula 2019.1.0033.0099-3,
telefone: 62993313212 e-mail Matt.gyn@hotmail.com, na qualidade de titular dos
direitos autorais, em consonância com a Lei n° 9.610/98 (Lei dos Direitos do autor),
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o
Trabalho de Conclusão de Curso intitulado
DESENVOLVIMENTO DE APLICAÇÃO PARA GERENCIAMENTO DE CONTAS DE
CLIENTES, gratuitamente, sem ressarcimento dos direitos autorais, por 5
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 03 de abril de 2025.

Assinatura do(s) autor(es):  Documento assinado digitalmente
MATEUS DE SOUZA ALMEIDA
Data: 03/04/2025 23:05:56-0300
Verifique em <https://validar.iti.gov.br>

Nome completo do autor: Mateus de Souza Almeida

Assinatura do professor-orientador:  Documento assinado digitalmente
ANDRE LUIZ ALVES
Data: 03/04/2025 23:36:42-0300
Verifique em <https://validar.iti.gov.br>

Nome completo do professor-orientador: André Luiz Alves