



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO INTERDISCIPLINAR EM TECNOLOGIA DA INFORMAÇÃO

LUIS IZAÍAS VALENTIM DE AQUINO

**DESENVOLVIMENTO DE UM SISTEMA *DESKTOP* RESILIENTE PARA GESTÃO
FINANCEIRA COM CONCILIAÇÃO BANCÁRIA: ESTUDO DE CASO DO EMPÓRIO**

LENA

PAU DOS FERROS

2026

LUIS IZAÍAS VALENTIM DE AQUINO

**DESENVOLVIMENTO DE UM SISTEMA *DESKTOP* RESILIENTE PARA GESTÃO
FINANCEIRA COM CONCILIAÇÃO BANCÁRIA: ESTUDO DE CASO DO EMPÓRIO
LENA**

Monografia apresentada à Universidade Federal
Rural do Semi-Árido como requisito para obten-
ção do título de Bacharela em Interdisciplinar
em Tecnologia da Informação.

Orientador: Reudismam Rolim de Sousa,
Prof. Dr.

PAU DOS FERROS

2026

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

A657d Aquino, Luis Izaías Valentim de.
Desenvolvimento de um sistema desktop
resiliente para gestão financeira com conciliação
bancária: estudo de caso do Empório Lena / Luis
Izaías Valentim de Aquino. - 2026.
52 f. : il.

Orientador: Reudismam Rolim de Sousa.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Tecnologia da
Informação, 2026.

1. Gestão financeira. 2. Micro e pequenas
empresas. 3. Desenvolvimento desktop. 4. Automação
de processos. I. Sousa, Reudismam Rolim de,
orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).

Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

LUIS IZAÍAS VALENTIM DE AQUINO

**DESENVOLVIMENTO DE UM SISTEMA *DESKTOP* RESILIENTE PARA GESTÃO
FINANCEIRA COM CONCILIAÇÃO BANCÁRIA: ESTUDO DE CASO DO EMPÓRIO**

LENA

Monografia apresentada à Universidade Federal
Rural do Semi-Árido como requisito para obten-
ção do título de Bacharela em Interdisciplinar
em Tecnologia da Informação.

Defendida em: 01/07/2026

BANCA EXAMINADORA

Reudismam Rolim de Sousa,, Prof. Dr. ()
Presidente

Bruno Borges da Silva, Prof. Esp. (UFERSA)
Membro Examinador

Dyego Magno Oliveira Souza, Bel. (UFERSA)
Membro Examinador

Dedico este trabalho à minha querida avó, Maria José da Conceição (*In Memoriam*), que continua a me inspirar e a me dar forças, mesmo distante.

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Socorro e Clemilson, por me ensinarem o valor da perseverança, da responsabilidade e da dedicação. Sou grato por todo o apoio, incentivo e pelos ensinamentos que me tornaram uma pessoa forte, capaz de perseguir meus objetivos e enfrentar os desafios da vida.

Agradeço, também, ao meu orientador, Prof. Dr. Reudismam Rolim, pelo direcionamento, pela disponibilidade e pela prontidão em auxiliar e orientar durante todas as etapas deste trabalho. Sua contribuição foi fundamental para o desenvolvimento desta pesquisa. Estendo meus agradecimentos aos membros da banca examinadora, pelas valiosas considerações e contribuições para minha formação acadêmica e profissional.

À Universidade Federal Rural do Semi-Árido (UFERSA), expresso minha gratidão pela oportunidade de formação e por proporcionar um ambiente de aprendizado e crescimento. Agradeço igualmente a todos os docentes que fizeram parte da minha trajetória acadêmica e contribuíram para a construção dos conhecimentos que hoje carrego.

Por fim, agradeço aos amigos e às pessoas especiais que tornam minha caminhada mais leve e significativa. Em especial à Vitória Mikaelly e Andréia Matias e às amigas construídas durante minha passagem pela Universidade Federal do Rio Grande do Norte (UFRN): Gabriela Guedes, Dayanne Gabriella, Rayanne Ponciano, Vanessa Macena e Hiasmin Dantas, que compartilharam momentos importantes de aprendizado, companheirismo e crescimento.

No meu caminho pela UFERSA, também demonstro minha gratidão a João Gabriel, Janieli Tainar, Kayc Henderson, Letícia Vieira, Laura Beatriz, Diego Luccas e Raimundo Maia, pela amizade, apoio e pelos inúmeros momentos de convivência que tornaram esta jornada acadêmica mais agradável e enriquecedora.

A todos que, de alguma forma, contribuíram para minha formação e para a concretização deste trabalho, meu sincero agradecimento.

RESUMO

As Micro e Pequenas Empresas (MPEs) são fundamentais para o desenvolvimento econômico e social do Brasil, respondendo por cerca de 30% do PIB nacional. Todavia, a sustentabilidade desses negócios é constantemente ameaçada por fragilidades na gestão administrativa e financeira, as quais geram uma elevada taxa de mortalidade empresarial precoce. O presente trabalho apresenta a análise, especificação de requisitos e desenvolvimento de um Sistema de Gestão Financeira (SGF) customizado para o Empório Lena, uma microempresa especializada no processamento e venda de condimentos na região semiárida do Ceará. Atualmente, os controles operacionais e de fluxo de caixa da empresa são realizados manualmente via planilhas isoladas, tornando a conciliação bancária morosa e suscetível a falhas. Para mitigar esse problema, foram realizadas a identificação dos gargalos operacionais e oportunidades de automação financeira, documentação dos requisitos funcionais e não funcionais, modelagem arquitetural e implementação da interface *desktop*. Metodologicamente, adotou-se o *framework* ágil Scrum para gerenciamento e o modelo Volere para a especificação de requisitos. A pilha tecnológica selecionada baseia-se em Java 21 com Spring Boot no backend, estruturado sob o padrão de Arquitetura Hexagonal (Portas e Adaptadores), e banco de dados PostgreSQL. A interface desktop foi concebida utilizando o ecossistema Tauri (Rust + React) com persistência PostgreSQL embarcada, garantindo resiliência operacional offline e baixo consumo de recursos computacionais. A validação qualitativa da interface com os gestores evidenciou uma redução da carga cognitiva dos operadores, alta aderência do fluxo de telas aos processos da empresa e simplificação de operações recorrentes.

Palavras-chave: gestão financeira; micro e pequenas empresas; desenvolvimento desktop; automação de processos.

ABSTRACT

Micro and Small Enterprises (MSEs) are fundamental to the economic and social development of Brazil, accounting for approximately 30% of the national GDP. However, the sustainability of these businesses is constantly threatened by weaknesses in administrative and financial management, which generate a high rate of early business mortality. This work presents the analysis, requirements specification, and development of a customized Financial Management System (FMS) for Empório Lena, a micro-enterprise specializing in the processing and sale of condiments in the semi-arid region of Ceará. Currently, the company's operational and cash flow controls are performed manually using isolated spreadsheets, making bank reconciliation slow and prone to errors. To mitigate this problem, operational bottlenecks and opportunities for financial automation were identified, functional and non-functional requirements were documented, architectural modeling was performed, and the desktop interface was implemented. Methodologically, the Scrum agile framework was adopted for management and the Volere model for requirements specification. The selected technology stack is based on Java 21 with Spring Boot on the backend, structured under the Hexagonal Architecture pattern (Ports and Adapters), and a PostgreSQL database. The desktop interface was designed using the Tauri ecosystem (Rust + React) with embedded PostgreSQL persistence, ensuring offline operational resilience and low consumption of computational resources. Qualitative validation of the interface with managers showed a reduction in the cognitive load of operators, high adherence of the screen flow to the company's processes, and simplification of recurring operations.

Keywords: financial management; micro and small businesses; desktop development; process automation.

LISTA DE FIGURAS

Figura 1 – Tela de <i>login</i>	35
Figura 2 – Autenticação e validação em duas etapas	36
Figura 3 – Tela principal	37
Figura 4 – Extensão da tela principal	37
Figura 5 – Entrada de XML	38
Figura 6 – Ingestão de XML	39
Figura 7 – Demonstrativo de Resultados e Emissões	40
Figura 8 – Gestão de integrações	40
Figura 9 – Visão e <i>log</i> de acessos	41
Figura 10 – Faturas retidas	42
Figura 11 – Diagrama de Componentes	44
Figura 12 – Diagrama de Implantação	44

LISTA DE QUADROS

Quadro 1 – Requisito Funcional 01 – Extração de Dados Fiscais.	32
Quadro 2 – Requisito Funcional 02 – Validação Anti-Fraude de Cobranças.	32
Quadro 3 – Requisito Funcional 03 – Configuração de Contas Multi-Banco.	33
Quadro 4 – Requisito Funcional 04 – Iniciação de Pagamentos via <i>Open Finance</i>	33
Quadro 5 – Requisito Não-Funcional 01 – Autenticidade de Notificações (<i>Webhooks</i>). .	34
Quadro 6 – Requisito Não-Funcional 02 – Resiliência Operacional com Tauri.	34
Quadro 7 – Requisitos Funcionais (Parte 1: Autenticação, <i>Dashboard</i> e Faturamento). .	51
Quadro 8 – Requisitos Funcionais (Parte 2: Estoque, Relatórios, Integrações e Segurança).	52
Quadro 9 – Requisitos Não-Funcionais (Parte 1: Segurança e Rastreabilidade).	53
Quadro 10 – Requisitos Não-Funcionais (Parte 2: Arquitetura, Conectividade e Desempe- nho).	53

LISTA DE ABREVIATURAS E SIGLAS

ACID	<i>Atomicity, Consistency, Isolation, Durability</i>
API	<i>Application Programming Interface</i>
CMP	Custo Médio Ponderado
CMV	Custo de Mercadorias Vendidas
CNPJ	Cadastro Nacional da Pessoa Jurídica
DLQ	<i>Dead Letter Queue</i>
DRE	Demonstrativo de Resultado do Exercício
ERP	<i>Enterprise Resource Planning</i>
ITP	Iniciador de Transação de Pagamentos
JWT	<i>JSON Web Token</i>
KPI	<i>Key Performance Indicator</i>
MFA	<i>Multi-Factor Authentication</i>
MPE	Micro e Pequena Empresa
mTLS	<i>Mutual Transport Layer Security</i>
NF-e	Nota Fiscal Eletrônica
PDF	<i>Portable Document Format</i>
RBAC	<i>Role-Based Access Control</i>
RF	Requisito Funcional
RNF	Requisito Não-Funcional
SGF	Sistema de Gestão Financeira
SQL	<i>Structured Query Language</i>
SUS	<i>System Usability Scale</i>
TAM	<i>Technology Acceptance Model</i>
UI	<i>User Interface</i>
UX	<i>User Experience</i>
XML	<i>Extensible Markup Language</i>
XSD	<i>XML Schema Definition</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.2	Estrutura do Documento	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	O Ecossistema das Micro e Pequenas Empresas no Brasil	17
2.2	Gestão Financeira e Automação da Conciliação Bancária	17
2.3	Engenharia de Software e Metodologias Ágeis em Pequenos Projetos	18
2.4	Arquitetura Hexagonal (Portas e Adaptadores)	19
2.5	Arquitetura Orientada a APIs e Conectividade FinTech	19
2.6	Pilha Tecnológica: Java, Spring Boot e o Ecossistema Tauri	20
2.6.1	Desenvolvimento <i>Backend</i> e Persistência	20
2.6.2	Desenvolvimento <i>Desktop</i> com Tauri, Rust e React	20
2.6.3	Integração Bancária, <i>Webhooks</i> e Protocolos de Segurança (mTLS e Pix)	21
2.7	Desenvolvimento da Interface e Avaliação de Usabilidade	21
3	TRABALHOS RELACIONADOS	23
3.1	Análise de Soluções de Mercado e o Modelo SaaS	23
3.2	Contribuições da Literatura Acadêmica em Sistemas Customizados	23
3.3	Diferencial da Proposta	24
4	METODOLOGIA	25
4.1	Tipo de Pesquisa	25
4.2	Universo e Amostra	25
4.3	Instrumentos de Coleta de Dados	26
4.4	Método de Análise e Especificação Técnica	26
4.5	Integração de Ferramentas de Inteligência Artificial na Metodologia de Desenvolvimento	27

4.6	Aspectos de Confidencialidade e Proteção de Dados	27
5	PROPOSTA DE SOLUÇÃO	28
5.1	Estímulo Baseado na Literatura e Trabalhos Correlatos	28
5.2	Justificativa Tecnológica e Escolha da <i>Stack</i>	28
5.3	Escopo do Ciclo de Entrega Atual	29
6	RESULTADOS	31
6.1	Especificação de Requisitos no Modelo Volere	31
6.1.1	Requisitos Funcionais (RF)	31
6.1.2	Requisitos Não-Funcionais (RNF)	34
6.2	Interface do Sistema	35
6.2.1	<i>Login</i> e Tela Principal	35
6.2.2	Faturamento de Venda e Processamento de XML	37
6.2.3	Ingestão de XML de Compra e Gestão de Estoque	38
6.2.4	Demonstrativo de Resultados do Exercício (DRE) e Relatórios	39
6.2.5	Integrações Bancárias e <i>Open Finance</i>	40
6.2.6	Segurança, Controle de Acesso e Perfis	40
6.2.7	Fila de Faturas Retidas Localmente	41
6.2.8	Resultados de Usabilidade e Validação Qualitativa	42
6.3	Camada de Serviços Backend (Spring Boot)	43
6.3.1	Diagrama de Componentes	44
6.3.2	Diagrama de Implantação	44
7	CONCLUSÕES E TRABALHOS FUTUROS	45
7.1	Considerações Finais	45
7.2	Trabalhos Futuros	46
	REFERÊNCIAS	47
	APÊNDICE A ROTEIRO DE AVALIAÇÃO DE USABILIDADE E COLETA DE	
	<i>FEEDBACK</i> DA INTERFACE	49

1 INTRODUÇÃO

As Micro e Pequenas Empresas (MPEs) representam a espinha dorsal da economia nacional, atuando como motores de desenvolvimento regional e estabilidade social. Esse segmento responde por aproximadamente 30% do Produto Interno Bruto do Brasil, sendo responsável pela manutenção de cerca de 50% dos postos de trabalho formais. Para além dos indicadores macroeconômicos, existe um fator de identificação social, em que cerca de 70% da população brasileira demonstra preferência por consumir produtos e serviços de pequenos empreendedores, o que reforça a necessidade de ferramentas que garantam a sustentabilidade desses negócios (Enzzo *et al.*, 2024).

Entretanto, a longevidade das MPEs é constantemente desafiada por fragilidades na gestão administrativa e financeira. A carência de planejamento e o controle ineficiente de recursos figuram entre os principais motivos do encerramento precoce de atividades. Estima-se que 25% das empresas brasileiras encerram suas operações antes de completarem dois anos de existência, sendo a ausência de uma gestão financeira profissionalizada a causa raiz dessa mortalidade precoce. O microempreendedor, muitas vezes focado na operação técnica de seu nicho, acaba negligenciando o fluxo de caixa, o que torna a saúde financeira da empresa vulnerável a variações de mercado (Machado, Santos e Souza, 2024).

A transformação digital é uma estratégia para mitigar as falhas de gestão financeira nos pequenos negócios. A implementação de sistemas de informação customizados permite a transição de um modelo de controle informal para uma gestão orientada a dados. A automação de processos não apenas reduz o esforço operacional, mas confere uma vantagem competitiva essencial, permitindo que a organização se adapte a novas demandas sem a necessidade de expansão imediata do quadro de funcionários (Tavares *et al.*, 2024).

Muitas MPEs, contudo, podem enfrentar dificuldades para adotar softwares de prateleira devido ao alto custo de licenciamento ou a funcionalidades de ferramentas genéricas que não se ajustam à realidade do pequeno negócio. A motivação deste trabalho reside em propor uma solução de software direcionada a uma MPE para preencher essa lacuna tecnológica, unindo práticas da engenharia de software a uma interface simplificada e funcional para o controle financeiro cotidiano. A integração de processos automáticos de informação é vital para garantir que os dados reflitam fielmente a situação patrimonial da empresa em tempo real de forma segura.

O foco deste trabalho é o acompanhamento inicial e modelagem de uma solução tec-

nológica para o Empório Lena, uma empresa especializada no processamento de condimentos localizada no interior do Ceará. Atualmente, a organização enfrenta gargalos operacionais recorrentes devido à ausência de uma ferramenta integrada que centralize suas movimentações financeiras. O controle manual, realizado via planilhas isoladas ou anotações físicas, torna o processo de conciliação bancária lento e passível de falhas, dificultando a análise precisa do capital de giro e das margens de lucro. A demora no acesso a serviços e operações bancárias também é evidenciada.

Para início do apoio à resolução dessa problemática, propõe-se a disponibilização de uma interface do Sistema de Gestão Financeira (SGF) customizado que atue como o núcleo administrativo da empresa. O diferencial tecnológico desta proposta é a integração inicial com a *Application Programming Interface* (API) do Banco Inter, permitindo que a geração de cobranças, leitura de extratos e visão do fluxo de caixa ocorram de forma automática dentro do sistema. Essa integração visa eliminar a redundância de dados e assegurar que o gestor e outros funcionários habilitados tenham um suporte tecnológico para decisões estratégicas, como o provisionamento de pagamentos e investimentos em expansão.

1.1 Objetivos

A atuação no presente trabalho visa realizar o acompanhamento do empreendimento e propor a modelagem de um sistema de gestão financeira customizado para o Empório Lena, focado na automação de processos e integração bancária. Para tanto, definem-se os seguintes objetivos específicos:

- Realizar o diagnóstico situacional dos fluxos financeiros atuais para identificar pontos de ineficiência;
- Elaborar o levantamento e a documentação de requisitos funcionais e não funcionais;
- Projetar a arquitetura do sistema com foco em escalabilidade e segurança;
- Desenvolver e validar a interface front-end do sistema por meio de demonstrações práticas e coleta de feedback qualitativo com os gestores do Empório Lena.

1.2 Estrutura do Documento

Este trabalho está organizado em sete capítulos fundamentais. O Capítulo 2 aborda a fundamentação teórica sobre microempreendedorismo e as tecnologias Web e *Desktop* utilizadas.

O Capítulo 3 analisa trabalhos relacionados, comparando soluções de gestão financeira existentes no mercado. No Capítulo 4, detalha-se a metodologia de desenvolvimento adotada. O Capítulo 5 descreve a proposta de solução com a definição de sua arquitetura. O Capítulo 6 apresenta os resultados obtidos, incluindo o resultado de requisitos e exposição da interface e arquitetura da aplicação. Por fim, o Capítulo 7 apresenta as considerações finais e as perspectivas de evolução futura.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo estabelece a base teórica e técnica necessária para sustentar a proposta do SGF. A discussão abrange desde a relevância socioeconômica das MPEs até as nuances arquiteturais de integração via APIs e o uso de *frameworks* para a aplicação *desktop*.

2.1 O Ecossistema das Micro e Pequenas Empresas no Brasil

As MPEs consolidaram-se como o motor da economia brasileira, representando um estrato fundamental para a dinamização regional. Segundo dados da Agência Sebrae analisados por Enzzo *et al.* (2024), esses negócios já respondem por aproximadamente 30% do Produto Interno Bruto nacional. No entanto, a sobrevivência desses empreendimentos está intrinsecamente ligada à sua capacidade de adotar estratégias proativas de gestão financeira, uma vez que a falta de capital de giro e o acesso limitado ao crédito ainda figuram como barreiras críticas para o crescimento sustentável (Machado, Santos e Souza, 2024).

A vulnerabilidade das MPEs é frequentemente exacerbada pela falta de conhecimento financeiro dos gestores e pela ausência de ferramentas que permitam uma análise de dados precisa. Conforme observado por Pires (2024), muitas microempresas enfrentam dificuldades severas na gestão de custos devido à desorganização e à falta de tempo, operando em um cenário onde a tomada de decisão é baseada em percepções subjetivas e não em indicadores reais de lucratividade e viabilidade. Essa lacuna tecnológica em regiões interioranas reforça a necessidade de sistemas que automatizem tarefas repetitivas, reduzindo a incidência de erros humanos e garantindo a integridade dos registros financeiros (Tavares *et al.*, 2024).

2.2 Gestão Financeira e Automação da Conciliação Bancária

A gestão financeira eficaz emerge como a chave para a expansão das organizações em ambientes competitivos. O controle do fluxo de caixa e a gestão de estoque, quando realizados manualmente, tornam-se processos morosos e propensos a falhas que podem comprometer a saúde financeira do negócio (Machado, Santos e Souza, 2024). A automação surge, portanto, como uma estratégia para superar esses desafios, permitindo que o gestor se concentre em atividades de maior valor agregado.

Um dos pilares dessa automação é a conciliação financeira. Tradicionalmente, este processo é realizado de forma manual, exigindo a conferência exaustiva de extratos bancários

contra registros internos, o que é ineficiente e suscetível a discrepâncias (Chiguano, 2025). Estudos recentes demonstram que a implementação de módulos de conciliação automatizada em sistemas financeiros integrados resulta em uma redução significativa de erros e melhora a rastreabilidade das operações, especialmente quando se lida com múltiplas instituições financeiras simultaneamente (Silva *et al.*, 2025). A transição de registros baseados em arquivos (como Excel ou CSV) para integrações baseadas em APIs representa uma evolução na confiabilidade dos dados e na agilidade dos reportes gerenciais (Ravalji e Jain, 2025).

No contexto de MPes, a apuração do Demonstrativo de Resultados do Exercício (DRE) sob o regime de competência é crucial para revelar a real rentabilidade econômica do negócio, diferentemente do regime de caixa, que se limita ao fluxo financeiro de entradas e saídas de recursos. No regime de competência, as receitas e despesas são registradas no momento de seu fato gerador, independentemente do pagamento ou recebimento. A precisão deste demonstrativo depende diretamente da correta mensuração dos estoques e da apuração do Custo de Mercadorias Vendidas (CMV). Para negócios agrícolas e de processamento de condimentos, a variação de preços de insumos exige um controle dinâmico. O recálculo instantâneo do Custo Médio Ponderado (CMP) a cada nova entrada de nota fiscal de compra permite valorizar o inventário com precisão científica. Essa atualização instantânea do CMP reflete-se imediatamente no custo unitário das matérias-primas consumidas no processo de produção, fornecendo uma base confiável para o cálculo do CMV e impedindo distorções na margem bruta do demonstrativo gerencial, o que otimiza as decisões estratégicas do microempreendedor.

2.3 Engenharia de Software e Metodologias Ágeis em Pequenos Projetos

O desenvolvimento de software para pequenas empresas e *startups* impõe desafios únicos de qualidade e produtividade. Em ambientes de alta demanda, processos rigorosos de engenharia de software são frequentemente negligenciados em favor da velocidade de entrega, o que pode resultar em produtos de baixa qualidade (Machado, Santos e Souza, 2024). Para mitigar esses riscos, a adoção de metodologias ágeis, como o Scrum, é recomendada, permitindo uma gestão de projetos flexível que se adapta às constantes mudanças de requisitos típicas do setor varejista (Machado, Santos e Souza, 2024).

A engenharia de software moderna para pequenos negócios deve focar na criação de artefatos de *design* que facilitem a manutenção futura. Conforme mapeado por Tavares *et al.* (2024), a utilização de ferramentas de gestão e estratégias bem definidas é essencial para garantir

a performance e a durabilidade do software. Isso inclui um levantamento de requisitos que considere não apenas as funcionalidades imediatas, mas também a escalabilidade da arquitetura diante do crescimento da empresa (Machado, Santos e Souza, 2024).

2.4 Arquitetura Hexagonal (Portas e Adaptadores)

A Arquitetura Hexagonal, também conhecida como padrão de Portas e Adaptadores, proposta por Cockburn (2005), é uma abordagem de design que visa isolar o núcleo da aplicação (regra de negócios e modelo de domínio) das tecnologias externas de infraestrutura, tais como bancos de dados, interfaces com o usuário, frameworks web e APIs bancárias. O núcleo de negócios é colocado no centro do "hexágono", sendo exposto ao exterior exclusivamente por intermédio de interfaces denominadas "portas"(*ports*), que podem ser de entrada (para comandos iniciados externamente) ou de saída (para solicitações feitas pelo sistema à infraestrutura). A conversão entre as requisições de agentes externos e a lógica interna do sistema é realizada por "adaptadores"(*adapters*). Adaptadores de entrada (*driving adapters*), como controladores REST ou interfaces gráficas, traduzem as intenções do usuário em chamadas do domínio. Adaptadores de saída (*driven adapters*) implementam as portas para acessar repositórios de dados ou APIs remotas. Esse desacoplamento rígido garante que mudanças tecnológicas ou substituições de provedores externos (como a migração de um banco de dados relacional ou alteração na API de um banco) ocorram sem afetar a lógica central do negócio, promovendo alta testabilidade, manutenibilidade e flexibilidade, essenciais no desenvolvimento de software contemporâneo.

2.5 Arquitetura Orientada a APIs e Conectividade FinTech

A integração de APIs revolucionou o cenário de transações digitais, permitindo uma conectividade sem precedentes entre sistemas heterogêneos. APIs RESTful, conhecidas por sua arquitetura leve e flexível, facilitam o intercâmbio de dados em tempo real entre sistemas de Planejamento de Recursos Empresariais (*Enterprise Resource Planning* - ERP) e instituições bancárias, promovendo a transparência e a segurança financeira (Ravalji e Jain, 2025). Esta arquitetura modular permite que processos de pagamento e contabilidade sejam decompostos em interfaces independentes, facilitando a adaptação do sistema a diferentes provedores bancários sem a necessidade de reestruturação completa do código (Ignev, 2025).

No contexto de tecnologias financeiras, as APIs servem como a base para a interope-

tabilidade e a escalabilidade. O uso de protocolos de autenticação segura, como OAuth 2.0, e o gerenciamento de tráfego via API Gateways garantem que a troca de dados financeiros sensíveis ocorra sob rigorosos padrões de segurança (Umavezi e Odukale, 2025). A automação da conciliação de transações via API não é apenas uma melhoria operacional, mas uma estratégia para fortalecer a governança corporativa em microempresas (Silva *et al.*, 2025).

2.6 Pilha Tecnológica: Java, Spring Boot e o Ecossistema Tauri

A escolha tecnológica para este projeto visa equilibrar robustez no processamento de dados com eficiência no lado do cliente (*frontend*).

2.6.1 Desenvolvimento *Backend* e Persistência

A utilização de Java com o *framework* Spring Boot justifica-se pela necessidade de uma infraestrutura estável e segura para o processamento de regras de negócio financeiras. O Spring Boot facilita a criação de microserviços e a implementação de APIs robustas que suportam o volume de transações bancárias automatizadas (Umavezi e Odukale, 2025). Complementarmente, o uso de sistemas de banco de dados relacionais garante a persistência segura e a integridade referencial necessária para auditorias e relatórios contábeis precisos (Enzzo *et al.*, 2024).

2.6.2 Desenvolvimento *Desktop* com Tauri, Rust e React

Para a interface *desktop*, o *framework* Tauri apresenta-se como uma alternativa superior ao tradicional Electron. Enquanto o Electron utiliza o motor Chromium, o que resulta em alto consumo de memória e binários extensos, o Tauri utiliza *webviews* nativas do sistema operacional e uma camada de *backend* escrita em Rust (Milichiewicz e Badurowicz, 2025). Esta arquitetura híbrida garante segurança de memória e um desempenho computacional significativamente maior, mantendo a flexibilidade do React para o desenvolvimento de interfaces reativas e modulares (Masuda, Kobayashi e Yatani, 2023).

Estudos comparativos indicam que aplicações desenvolvidas com Tauri possuem um tamanho de instalador reduzido e uma pegada de memória RAM consideravelmente menor em comparação com soluções similares, tornando-o ideal para hardware limitado frequentemente encontrado em pequenas empresas (Tarskich, 2023). A modularidade do *frontend* em React permite que a aplicação seja facilmente migrada de ambientes web para *desktop*, preservando a

lógica de negócio e a experiência do usuário (Milichiewicz e Badurowicz, 2025).

2.6.3 Integração Bancária, *Webhooks* e Protocolos de Segurança (mTLS e Pix)

A integração direta com instituições financeiras exige padrões rigorosos de segurança e comunicação assíncrona. O protocolo *Mutual Transport Layer Security* (mTLS) desempenha um papel central ao estabelecer uma autenticação mútua bidirecional baseada em certificados digitais X.509. Diferente do TLS convencional, onde apenas o cliente valida a identidade do servidor, o handshake mTLS exige que o servidor também solicite e valide o certificado apresentado pelo cliente. Esse processo garante que ambos os lados da comunicação tenham suas identidades criptograficamente atestadas por uma autoridade certificadora comum. Após a validação das assinaturas digitais, as chaves de sessão são trocadas, estabelecendo um canal criptograficamente seguro que impede interceptações do tipo *man-in-the-middle* ou falsificações de requisições.

Além da segurança no canal, o fluxo de conciliação depende do uso de *webhooks*, por meio dos quais a instituição financeira notifica ativamente o sistema em tempo real sobre a liquidação de pagamentos Pix. Essa abordagem orientada a eventos elimina a latência e o consumo de recursos associados a consultas periódicas por *polling*. Contudo, a recepção desses eventos exige validações adicionais de segurança para evitar injeções de falsas confirmações de pagamento. Para mitigar esse risco, o *endpoint* do *webhook* implementa a validação de assinaturas digitais aplicadas nos cabeçalhos das requisições (*HMAC* ou assinaturas assimétricas), garantindo a autoria e integridade do *payload*. A verificação do *hash* criptográfico e a restrição de tráfego a faixas de IPs homologadas garantem que apenas notificações legítimas e inalteradas do banco parceiro atualizem o banco de dados da aplicação.

2.7 Desenvolvimento da Interface e Avaliação de Usabilidade

A amostragem e usabilidade prévia de uma interface funcional provê detalhes essenciais no desenvolvimento de sistemas de informação, especialmente quando direcionados a usuários com pouca familiaridade tecnológica. O artefato simula de forma precisa o comportamento, a estética e a navegação da aplicação final, permitindo que os usuários reais interajam com a interface antes da codificação completa do *backend*. A avaliação de usabilidade qualitativa, por meio de sessões de demonstração e coleta de *feedback* semiestruturado, auxilia na identificação precoce de falhas de *layout*, termos ambíguos ou fluxos complexos, alinhando a solução às reais

necessidades operacionais do modelo de negócio.

3 TRABALHOS RELACIONADOS

A análise de trabalhos relacionados é fundamental para situar a proposta deste SGF no contexto das soluções tecnológicas contemporâneas, permitindo identificar as convergências e, primordialmente, as lacunas que justificam o desenvolvimento de uma ferramenta customizada.

3.1 Análise de Soluções de Mercado e o Modelo SaaS

O cenário atual de softwares de gestão financeira é dominado por sistemas de Planejamento de Recursos Empresariais (ERP) “de prateleira” e plataformas *Software as a Service* genéricas. Entre as soluções modernas, destaca-se o Pluma Finance, utilizado de forma preliminar pelo cliente apenas como teste, que oferece uma interface web robusta para o controle de fluxos de caixa e relatórios gerenciais. No entanto, ferramentas desse gênero impõem um modelo de operação que exige conectividade constante e alta largura de banda, o que se torna um ponto de falha em cenários de microempresas rurais, em que a estabilidade da rede é intermitente.

A análise crítica dessas ferramentas revela que, embora tecnicamente sofisticadas, elas frequentemente apresentam uma curva de aprendizado íngreme e um excesso de funcionalidades que não se alinham à realidade operacional de um empreendimento como o Empório Lena. A imposição de que a empresa adapte seus processos internos à lógica do software, em vez de esse recurso refletir a rotina do negócio, é uma limitação crônica (Machado, Santos e Souza, 2024). Para o microempreendedor, o custo de manutenção e a complexidade de sistemas multifuncionais acabam por desestimular a transição do controle informal para o digital, mantendo a vulnerabilidade administrativa (Santiago e Leal, 2024).

3.2 Contribuições da Literatura Acadêmica em Sistemas Customizados

A literatura acadêmica recente tem explorado extensivamente o impacto do desenvolvimento de software sob medida para o fortalecimento de MPEs. Estudos de caso demonstram que a prototipação de soluções tecnológicas específicas permite uma aproximação direta entre o desenvolvedor e o usuário final, resultando em ferramentas que atendem às “dores” reais do cotidiano empresarial (Machado, Santos e Souza, 2024). Esse alinhamento é essencial para que o sistema não seja visto como um fardo burocrático, mas como um facilitador da eficiência operacional (Tavares *et al.*, 2024).

Pesquisas realizadas em contextos de economia regional indicam que a transição de

controles financeiros informais para processos estruturados é mais eficaz quando a tecnologia é introduzida de forma modular e intuitiva (Pires, 2024). A automação de tarefas repetitivas, como a conciliação bancária, é apontada como um fator determinante para a redução de erros humanos e para a melhoria da integridade dos registros financeiros (Chiguano, 2025). Nesse sentido, o desenvolvimento customizado atua como uma ponte para a maturidade de gestão, permitindo que o microempreendedor tenha uma visão clara de sua liquidez e solvência (Enzzo *et al.*, 2024).

3.3 Diferencial da Proposta

Ao contrastar as soluções comerciais analisadas com os estudos acadêmicos revisados, identifica-se uma lacuna tecnológica específica: a carência de sistemas que unam robustez arquitetural com alta resiliência operacional para ambientes remotos. Enquanto softwares de mercado priorizam o modelo *web-only*, esta proposta adota o *framework* Tauri para garantir uma execução *desktop* local, o que assegura a continuidade das operações do Empório Lena mesmo diante de oscilações de internet típicas da zona rural (Tarskich, 2023).

Diferente dos ERPs genéricos, em que a integração bancária é frequentemente limitada ou exige módulos adicionais onerosos, a solução proposta automatiza a conciliação financeira de forma nativa e segura (Silva *et al.*, 2025). Esta abordagem modular não apenas simplifica o fluxo de dados entre o banco e a empresa, mas também nativamente consome menos recursos computacionais, características dificilmente encontradas de forma simultânea nas soluções prontas disponíveis no mercado (Milichiewicz e Badurowicz, 2025)(Ignev, 2025).

4 METODOLOGIA

A metodologia deste trabalho é compreendida como o estudo sistemático dos métodos e processos necessários para a consecução dos objetivos da investigação. Para a viabilização do sistema de gestão financeira proposto, delineou-se um percurso metodológico que integra o rigor da Engenharia de Software com práticas de pesquisa aplicada, garantindo que a solução técnica responda efetivamente às demandas do cenário real.

4.1 Tipo de Pesquisa

A presente pesquisa caracteriza-se como de natureza aplicada, apresentando uma abordagem qualitativa quanto ao problema. O foco central reside na análise, especificação e modelagem de uma solução de software customizada para a gestão financeira. Este tipo de investigação busca gerar conhecimentos para aplicação prática e solução de problemas específicos no contexto das MPes (Enzzo *et al.*, 2024).

O desenvolvimento seguiu os preceitos da Engenharia de Software voltada para pequenos negócios, em que a eficiência e a adaptabilidade do sistema são priorizadas para garantir a longevidade da organização (Machado, Santos e Souza, 2024). A pesquisa não se limita à codificação, mas abrange o ciclo completo de levantamento, análise de viabilidade e estruturação arquitetural de um artefato tecnológico.

4.2 Universo e Amostra

O universo da pesquisa compreende o setor de microempreendimentos que necessitam de controles financeiros robustos, porém acessíveis. A amostra foi delimitada a uma empresa específica, caracterizada como microempresa, situada em âmbito local. Esta delimitação justifica-se pela necessidade de observar processos administrativos em um ambiente controlado e real, em que as limitações de tempo e recursos do gestor são fatores determinantes para o sucesso da ferramenta de software (Pires, 2024). A análise focou nos desafios enfrentados por microempreendedores na gestão cotidiana de seus negócios (Santiago e Leal, 2024).

4.3 Instrumentos de Coleta de Dados

Para a extração das informações e necessidades do cliente, foram utilizados múltiplos instrumentos de coleta, garantindo a triangulação dos dados:

1. Entrevistas Presenciais: Realizaram-se sessões de entrevista com o gestor e usuários diretos, focando na observação das dores atuais, processos e limitações reais do cliente. Este método permitiu identificar gargalos operacionais que dificultam a gestão financeira eficiente (Machado, Santos e Souza, 2024).
2. Análise de Experiência Prévia: O cliente possui curta experiência com a plataforma *web* Pluma Finance. Esta vivência serviu como uma das maiores fontes para a extração e validação de requisitos, permitindo identificar quais funcionalidades eram satisfatórias e quais lacunas justificavam o desenvolvimento de uma solução *desktop* personalizada.
3. Observação Direta: Foram observados os fluxos de trabalho para entender como as informações sobre transações são processadas e como a automação de processos poderia reduzir erros de lançamentos manuais (Tavares *et al.*, 2024).

4.4 Método de Análise e Especificação Técnica

Os dados brutos coletados foram traduzidos para abordagens da Engenharia de Software por meio de métodos estruturados de análise e gestão:

- Metodologia Ágil Scrum: A pesquisa e o gerenciamento inicial do projeto utilizaram o *framework* Scrum. A utilização de ciclos iterativos permitiu que as necessidades do cliente fossem refinadas continuamente, garantindo que o desenvolvimento estivesse alinhado às expectativas de negócio (Machado, Santos e Souza, 2024).
- Modelo de Especificação de Requisitos da Volere: O método central adotado para a organização, estruturação e gestão formal dos requisitos extraídos é o Volere *Requirements Specification Template*. A adoção desse *framework* revelou-se altamente eficaz para a definição do ecossistema geral do projeto. O template foi crucial não apenas para mapear e documentar os requisitos funcionais e não-funcionais finais, mas também para estruturar as próprias fundações do projeto, englobando a definição clara dos objetivos do negócio, o alinhamento de expectativas com as partes interessadas (*stakeholders*), a consideração das restrições do ambiente operacional e a delimitação precisa do escopo a ser implementado no ciclo de entrega atual.

- Modelagem de Processos: Os requisitos foram convertidos em modelos de sistemas que priorizam a segurança financeira e a eficiência na conciliação de dados, utilizando arquiteturas modernas de integração via APIs (Silva *et al.*??, 2025, ??).

4.5 Integração de Ferramentas de Inteligência Artificial na Metodologia de Desenvolvimento

Em conformidade com a Política de Integridade na Atividade Científica, instituída pela Portaria nº 2.664/2026 (Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), 2026), declara-se o uso de ferramentas de Inteligência Artificial Generativa em etapas específicas deste trabalho. Tais tecnologias foram aplicadas no suporte à revisão de estilo, formatação textual e padronização das referências bibliográficas. Ademais, empregou-se a capacidade de agente autônomo do assistente Gemini (*Gemini Agent*) e a ferramenta *Chrome Auto Browse* (Automação do Chrome) para a execução de testes de software automatizados e interações na interface do usuário. A autoria intelectual, a validação de todas as informações geradas e a responsabilidade ética pelo manuscrito permanecem integralmente atribuídas aos autores humanos da pesquisa.

4.6 Aspectos de Confidencialidade e Proteção de Dados

Por questões de sigilo comercial e governança corporativa, os dados financeiros nominais, faturamentos reais e informações estratégicas do Empório Lena foram omitidos ou descaracterizados. As matrizes e roteiros apresentados nos apêndices refletem a estrutura metodológica e os fluxos sistêmicos projetados, operando com dados simulados para fins de validação acadêmica.

5 PROPOSTA DE SOLUÇÃO

O desenvolvimento do sistema de gestão financeira para o Empório Lena fundamenta-se na premissa de que a TI deve atuar como um agente de transformação e sustentabilidade para MPEs. Esta seção detalha as justificativas teóricas e tecnológicas que sustentam a arquitetura proposta, evidenciando sua adequação aos desafios específicos de um empreendimento rural.

5.1 Estímulo Baseado na Literatura e Trabalhos Correlatos

A literatura especializada aponta que a adoção de sistemas de gestão “de prateleira” por microempresas é frequentemente marcada por altas taxas de insucesso. Isso ocorre porque tais ferramentas, embora tecnicamente robustas, impõem uma curva de aprendizado íngreme e um excesso de funcionalidades que não guardam aderência aos processos locais e simplificados do pequeno negócio (Pires, 2024). A imposição de um modelo de gestão genérico força a empresa a adaptar seus processos ao software, o que resulta em ineficiência e resistência por parte dos gestores (Santiago e Leal, 2024).

Em contrapartida, a proposta de um software customizado apoia-se em evidências acadêmicas que associam soluções sob medida ao aumento da longevidade empresarial e à redução da informalidade gerencial (Machado, Santos e Souza, 2024). Ao focar nas “dores” reais do Empório Lena, o desenvolvimento personalizado permite a transição de um controle financeiro rudimentar para um processo estruturado sem descaracterizar a dinâmica do negócio (Enzzo *et al.*, 2024). Estudos de caso indicam que a informatização focalizada em requisitos específicos é capaz de mitigar a vulnerabilidade administrativa, proporcionando uma visão clara de liquidez e solvência (Tavares *et al.*, 2024).

5.2 Justificativa Tecnológica e Escolha da *Stack*

A seleção das tecnologias utilizadas nesta proposta foi norteada pelos princípios de robustez, segurança e, primordialmente, resiliência operacional para o contexto rural.

- **Backend:** Para o desenvolvimento do backend, foram adotados a linguagem Java, versão 21, e o *framework* Spring Boot. A escolha do Java 21 justifica-se pela sua estabilidade e suporte a padrões modernos de concorrência. Aliado ao Spring Boot, o ecossistema permite a construção de uma API *RESTful* segura e escalável, facilitando a integração com serviços financeiros externos (Umavezi e Odukale, 2025). A arquitetura modular do

Spring Boot garante que regras de negócio complexas sejam tratadas de forma isolada, facilitando manutenções futuras (Ignev, 2025).

- Banco de Dados: Para garantir a integridade dos dados financeiros, foi usado o PostgreSQL. Sua conformidade com as propriedades *Atomicity*, *Consistency*, *Isolation*, *Durability* (ACID) assegura que as transações financeiras sejam registradas de forma imutável e segura, prevenindo perdas de informação em casos de falhas no sistema (Chiguano, 2025).
- Integração Bancária: A automação via API do Banco Inter visa eliminar a necessidade de lançamentos manuais de extratos e emissão de cobranças. A integração direta para conciliação bancária reduz drasticamente a incidência de erros humanos e garante que o fluxo de caixa reflita a realidade financeira da empresa em tempo real (Silva *et al.*??, 2025, ??).
- *Frontend/Desktop* (React + Tauri): No *frontend*, foram adotados o Tauri e o React. O uso do *framework* Tauri para empacotar a interface desenvolvida em React representa uma escolha arquitetural estratégica para o ambiente rural. Ao contrário do Electron, o Tauri utiliza *webviews* nativas e uma camada de *backend* em Rust, resultando em aplicações desktop leves com baixo consumo de memória RAM (Tarskich, 2023). Esta abordagem oferece resiliência a oscilações de internet, permitindo que o sistema funcione localmente nas máquinas do cliente com alta performance, garantindo a continuidade operacional mesmo sem conectividade estável (Milichiewicz e Badurowicz, 2025).

5.3 Escopo do Ciclo de Entrega Atual

Para o presente ciclo de desenvolvimento e avaliação acadêmica, o escopo da solução compreende os seguintes resultados:

- Artefatos de Requisitos Funcionais e não Funcionais.
- Serviço de *Backend*: Mapeamento das entidades do domínio financeiro e definição da arquitetura de serviços.
- Prototipação de Alta Fidelidade do *Frontend*: Desenvolvimento das interfaces de usuário utilizando React, focando na usabilidade e na experiência dos gestores, garantindo que o layout seja intuitivo e funcional.
- Estratégia de Implantação e Infraestrutura: Definição da arquitetura de implantação, contemplando o *backend* hospedado em ambiente de nuvem para a recepção de *webhooks* bancários e a aplicação desktop executada localmente nas máquinas do Empório Lena,

assegurando a integração fluida entre os componentes.

Cabe ressaltar que a delimitação do escopo supracitado foi diretamente influenciada por um incidente de segurança sofrido pela empresa parceira em meados de abril de 2026. A organização foi alvo de um ataque cibernético de *ransomware* (um *malware* de criptografia de dados acompanhado de extorsão financeira), o qual comprometeu o servidor local principal, inviabilizando completamente o sistema legado de pedidos e emissão de notas fiscais. Alinhada às boas práticas de segurança da informação, a administração do Empório Lena recusou o pagamento do resgate exigido pelos atacantes devido à ausência de garantias reais de recuperação dos arquivos. Consequentemente, as operações da empresa precisaram ser restabelecidas de forma parcial por meio da restauração de um *backup* histórico de dados.

Do ponto de vista de gestão do projeto de TCC, este incidente crítico provocou uma considerável janela de indisponibilidade técnica nas dependências da empresa, inviabilizando a infraestrutura necessária para a realização de testes de integração conjunta com a API de produção da instituição bancária no ciclo atual. Diante disso, o escopo de entrega do projeto consolidou-se na modelagem detalhada, na especificação arquitetural e no desenvolvimento da interface da aplicação.

Por outro lado, o evento forneceu uma validação empírica inquestionável quanto à necessidade imediata do SGF proposto. O incidente justificou o rigor arquitetural aplicado na concepção dos requisitos de segurança do sistema, tais como o controle de acesso baseado em papéis (RBAC), a autenticação mTLS nas transações bancárias, a assinatura digital para validação de *webhooks* e a criptografia simétrica com algoritmo AES para a guarda de credenciais no banco local.

6 RESULTADOS

O presente capítulo apresenta os aspectos de projeto e modelagem técnica concebidos para o SGF do Empório Lena. O detalhamento está estruturado em quatro divisões fundamentais: a especificação de requisitos no formato Volere, exposição da primeira versão da interface desenvolvida e avaliada pelos gestores, a arquitetura de software adotada no backend e os diagramas de modelagem que ilustram a dinâmica do sistema.

6.1 Especificação de Requisitos no Modelo Volere

A especificação de requisitos foi estruturada com base no modelo Volere, permitindo conectar cada funcionalidade aos critérios de aceitação e justificativas de negócio. O foco desta etapa de modelagem reside nas operações de faturamento, conciliação bancária automática, controle de acesso e segurança contra fraudes financeiras.

A fim de preservar a fluidez da leitura e evitar tabelas excessivamente longas no corpo deste capítulo, optou-se por apresentar nas seções a seguir apenas os requisitos macro formatados individualmente segundo o modelo Volere. Contudo, para fins de projeto e implementação detalhada da interface, realizou-se um mapeamento mais granular dos sub-requisitos operacionais. Esse detalhamento técnico complementar, com a especificação das ações e escopos (RF001 a RF027; RNF001 a RNF012), está consolidado em forma de quadros no Apêndice B deste trabalho.

6.1.1 Requisitos Funcionais (RF)

A especificação dos requisitos funcionais do sistema foi elaborada segundo o modelo Volere. O processo de captação de dados fiscais é detalhado no Quadro 1, enquanto as validações de segurança da emissão de cobranças constam no Quadro 2. A parametrização de contas e a iniciação de pagamentos Pix encontram-se descritas nos Quadros 3 e 4, respectivamente.

Quadro 1 – Requisito Funcional 01 – Extração de Dados Fiscais.

Identificador	RF01
Nome	Extração de Dados Fiscais
Tipo de Requisito	Requisito Funcional
Caso de Uso	UC01 (Importação de Documento Fiscal)
Descrição	O sistema deve processar arquivos no formato <i>Extensible Markup Language</i> (.xml) de notas fiscais eletrônicas (NF-e) geradas pelo faturamento do Empório Lena, extraindo de forma automática os dados cadastrais do cliente, valor total e itens comercializados.
Justificativa	Eliminar o preenchimento manual dos dados de cobrança, reduzindo a incidência de erros operacionais e economizando tempo administrativo.
Critério de Aceitação	A leitura de um XML válido de NF-e deve preencher a tela de visualização de cobrança em menos de 1 segundo, com equivalência absoluta (100%) de valores e Cadastro Nacional da Pessoa Jurídica (CNPJ)/CPF em relação ao arquivo de origem.
Prioridade	Alta

Fonte: Autoria própria (2026)

Quadro 2 – Requisito Funcional 02 – Validação Anti-Fraude de Cobranças.

Identificador	RF02
Nome	Validação Anti-Fraude de Cobranças
Tipo de Requisito	Requisito Funcional / Segurança
Caso de Uso	UC02 (Emissão de Títulos de Cobrança)
Descrição	O sistema deve confrontar as solicitações de emissão de boletos com o histórico de pagamentos anteriores do cliente destinatário. Caso o destinatário não possua registro ou apresente histórico de inconsistências, o sistema deve suspender o envio automático e exigir liberação gerencial.
Justificativa	Mitigar tentativas de golpes financeiros, faturamentos clonados ou spams gerados por acessos indevidos.
Critério de Aceitação	Bloqueio imediato de emissões automáticas direcionadas a CNPJs não constantes na base histórica ativa, solicitando confirmação de autenticação em duas etapas do administrador.
Prioridade	Alta

Fonte: Autoria própria (2026)

Quadro 3 – Requisito Funcional 03 – Configuração de Contas Multi-Banco.

Identificador	RF03
Nome	Configuração de Contas Multi-Banco
Tipo de Requisito	Requisito Funcional
Caso de Uso	UC03 (Manutenção de Contas Bancárias)
Descrição	O sistema deve prover interface para o cadastro de múltiplas contas bancárias, permitindo a parametrização de credenciais de APIs específicas (certificados mTLS, <i>tokens</i> e segredos) para cada instituição.
Justificativa	Garantir a flexibilidade de operação da empresa com diferentes parceiros financeiros.
Critério de Aceitação	Persistência isolada de credenciais com criptografia simétrica na base de dados e associação dinâmica da conta no momento da conciliação.
Prioridade	Média

Fonte: Autoria própria (2026)

Quadro 4 – Requisito Funcional 04 – Iniciação de Pagamentos via *Open Finance*.

Identificador	RF04
Nome	Iniciação de Pagamentos via <i>Open Finance</i>
Tipo de Requisito	Requisito Funcional
Caso de Uso	UC04 (Pagamento via <i>Open Finance</i>)
Descrição	O sistema deve permitir a iniciação de transações Pix diretamente a partir do ecossistema <i>Open Finance</i> regulado pelo Banco Central, utilizando os <i>tokens</i> de consentimento dos clientes do Empório Lena.
Justificativa	Agilizar o fluxo de recebimento sem a necessidade de emissão de boletos físicos ou leitura manual de códigos de barras por parte do cliente.
Critério de Aceitação	Redirecionamento seguro para a instituição do cliente, captura de retorno de <i>status</i> de transação pendente/concluída em tempo real.
Prioridade	Média

Fonte: Autoria própria (2026)

6.1.2 Requisitos Não-Funcionais (RNF)

Para assegurar o correto funcionamento da plataforma frente aos desafios de segurança e infraestrutura da zona rural, definiram-se os requisitos não-funcionais dispostos a seguir. O requisito RNF01, especificado no Quadro 5, trata da segurança dos *webhooks* de conciliação bancária, enquanto o RNF02, documentado no Quadro 6, estabelece as premissas de resiliência offline do aplicativo desktop.

Quadro 5 – Requisito Não-Funcional 01 – Autenticidade de Notificações (*Webhooks*).

Identificador	RNF01
Nome	Autenticidade de Notificações (<i>Webhooks</i>)
Tipo de Requisito	Segurança / Integridade
Descrição	O endpoint de recepção de <i>webhooks</i> para conciliação bancária deve validar a assinatura criptográfica e a validade do certificado digital de origem do remetente.
Justificativa	Evitar ataques de injeção de falsas confirmações de pagamento por agentes externos maliciosos.
Critério de Aceitação	Rejeição imediata com código de erro HTTP 403 para requisições cujo cabeçalho de assinatura não corresponda ao hash público esperado ou fora do range de IPs homologados.
Prioridade	Alta

Fonte: Autoria própria (2026)

Quadro 6 – Requisito Não-Funcional 02 – Resiliência Operacional com Tauri.

Identificador	RNF02
Nome	Resiliência Operacional com Tauri
Tipo de Requisito	Usabilidade / Confiabilidade
Descrição	O cliente desktop deve manter capacidade operacional para consulta de dados cadastrais locais e preparação de faturamento mesmo durante períodos de oscilação de conectividade com a internet.
Justificativa	Adequação às restrições de infraestrutura de rede típicas da zona rural na qual o Empório Lena opera.
Critério de Aceitação	Sincronização em segundo plano dos dados acumulados localmente em banco de dados embarcado assim que a conexão com o servidor em nuvem for restabelecida.
Prioridade	Alta

Fonte: Autoria própria (2026)

6.2 Interface do Sistema

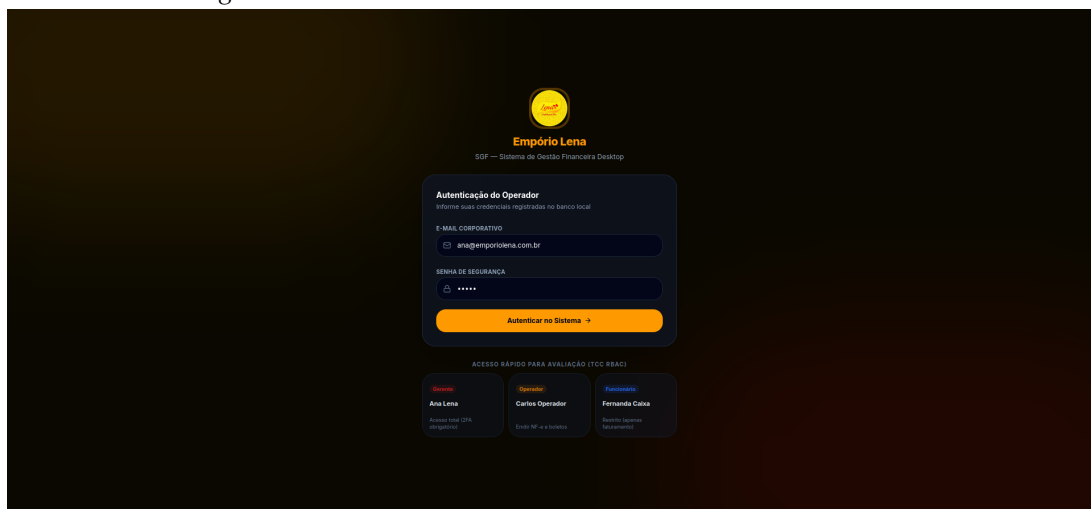
A interface de usuário (UI) foi projetada para mitigar a carga cognitiva dos operadores do empório e garantir alta responsividade em monitores de resoluções variadas, adotando uma identidade visual com cores neutras (escala de cinzas), detalhes em âmbar e sinalização cromática semântica como: verde para ingressos e transações liquidadas, amarelo para pendências e vermelho para saídas de capital ou alertas de segurança.

A seguir, são detalhadas e expostas as principais telas concebidas, as quais foram submetidas à avaliação qualitativa dos gestores para validação visual e também do fluxo operacional da aplicação.

6.2.1 Login e Tela Principal

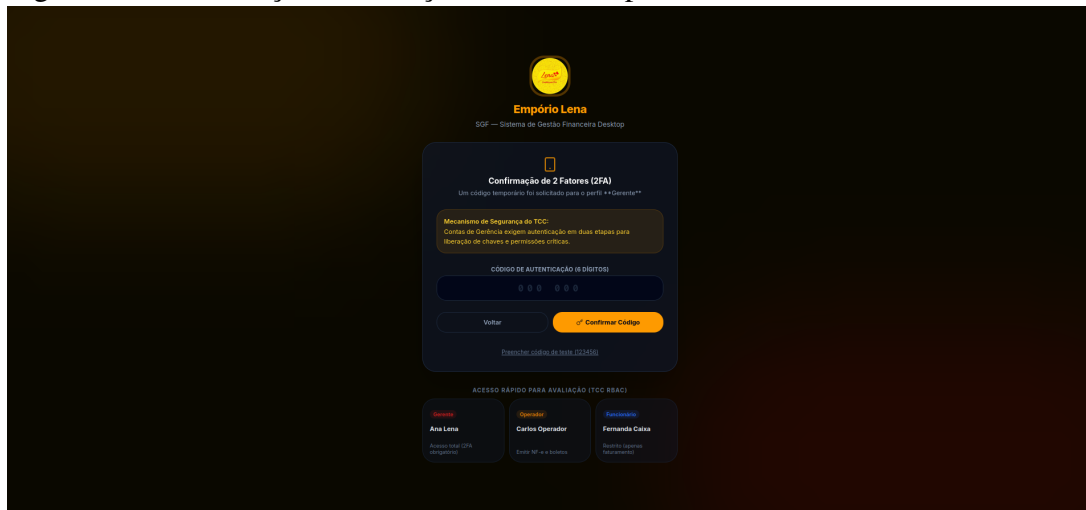
O acesso à plataforma ocorre por meio da interface de *login*, ilustrada na Figura 1. Para mitigar riscos de segurança, o sistema exige uma validação adicional em duas etapas, representada na Figura 2. No sistema apresentado, constavam três perfis de usuários estáticos, dos quais cada um possuía um escopo de atividade e acessos diferentes no sistema. O usuário que iniciar a sessão será direcionado à Tela Principal.

Figura 1 – Tela de *login*



Fonte: Autoria própria (2026)

Figura 2 – Autenticação e validação em duas etapas

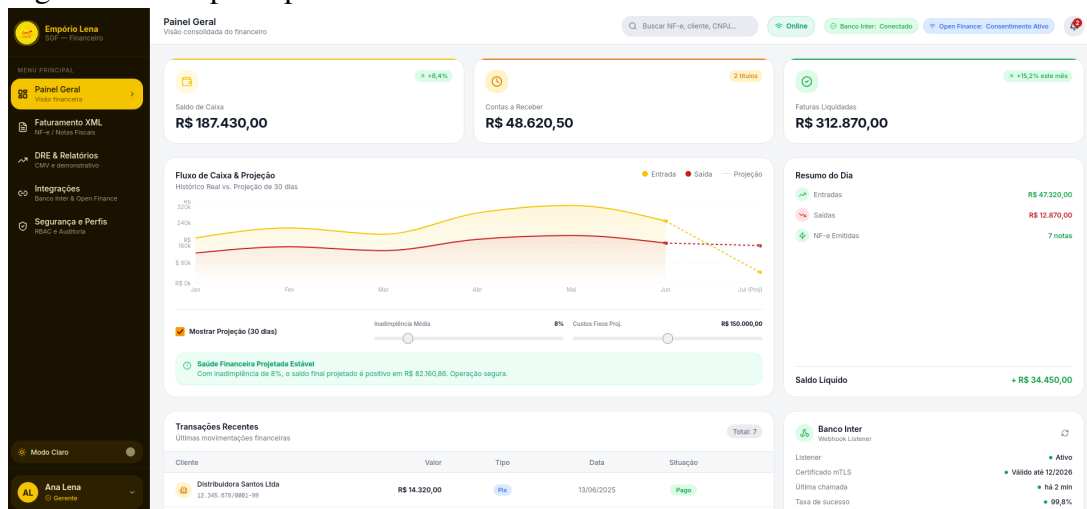


Fonte: Autoria própria (2026)

A tela principal da aplicação, ou Dashboard, apresentada na Figura 3 e cuja extensão é mostrada na Figura 4, atua como console de controle unificado da saúde financeira do empório. A interface é subdividida em três quadrantes principais, estruturados da seguinte forma:

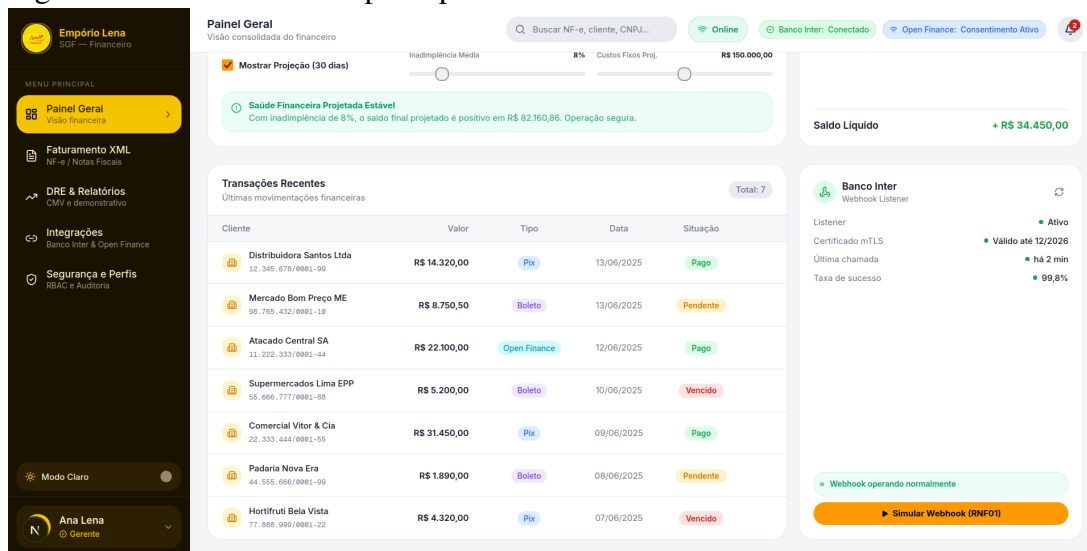
- **Indicadores de Desempenho:** Exibição resumida dos saldos em caixa, volume total de contas a receber e faturas liquidadas no período vigente. A construção visual utiliza cartões com gradientes sutis e variações percentuais em relação ao período anterior.
- **Gráfico de Fluxo de Caixa e Projeção:** Apresenta graficamente a relação histórica de entradas e saídas de caixa. Integra um simulador de projeção financeira de 30 dias que permite aos gestores ajustar, via controles deslizantes (*ranges*), a taxa média de inadimplência estimada e o custo fixo operacional. O componente recalcula o saldo final projetado em tempo real, gerando alertas textuais de “Caixa Crítico” caso a curva sinalize solvência negativa, viabilizando decisões preventivas.
- **Console de Monitoramento do Webhook (RNF01):** Exibe o *status* do ouvinte (*listener*) de notificações bancárias. Contém uma interface de simulação que demonstra, em formato de console de depuração (*log*), o fluxo de recebimento de um evento de pagamento enviado pela API do Banco Inter.

Figura 3 – Tela principal



Fonte: Autoria própria (2026)

Figura 4 – Extensão da tela principal



Fonte: Autoria própria (2026)

6.2.2 Faturamento de Venda e Processamento de XML

Esta interface operacionaliza o faturamento a partir de NF-e de saída, cuja área de importação e processamento está ilustrada na Figura 5. A tela está estruturada em dois estados principais:

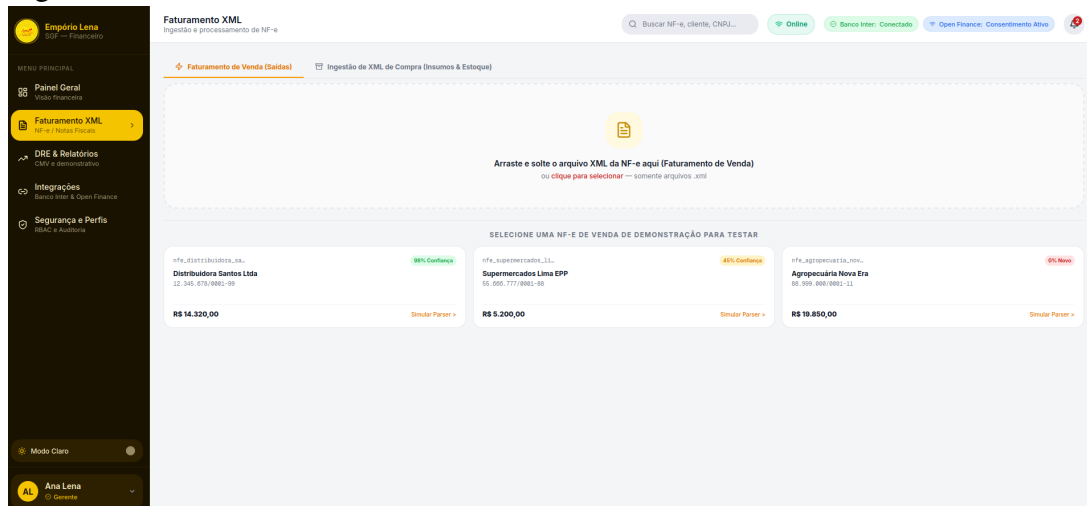
- Área de Importação Assistida (Drag-and-Drop): Uma zona de descarte de arquivos configurada para receber exclusivamente extensões .xml.
- Painel de Dados Extraídos e Configuração de Cobrança: Apresenta as informações recuperadas do documento fiscal e também permite configurar a data de vencimento da fatura e o

canal de pagamento (Pix, Boletão ou *Open Finance*).

Associado a esta tela, o sistema exibe o nível de risco do destinatário com base em seu histórico cadastral e de pagamentos:

- **Confiança Elevada:** CNPJs ativos com histórico de pontualidade.
- **Risco Moderado:** Clientes com atrasos pretéritos recorrentes, recomendando-se prazos de vencimento reduzidos.
- **Novo Destinatário ou Risco Crítico:** Aciona o bloqueio antifraude preventivo. A interface suspende o faturamento automático e abre um modal de autenticação que exige a introdução de uma credencial de autenticação multifatorial (*Multi-Factor Authentication - MFA*) do Gerente para autorizar a operação.

Figura 5 – Entrada de XML



Fonte: Autoria própria (2026)

6.2.3 Ingestão de XML de Compra e Gestão de Estoque

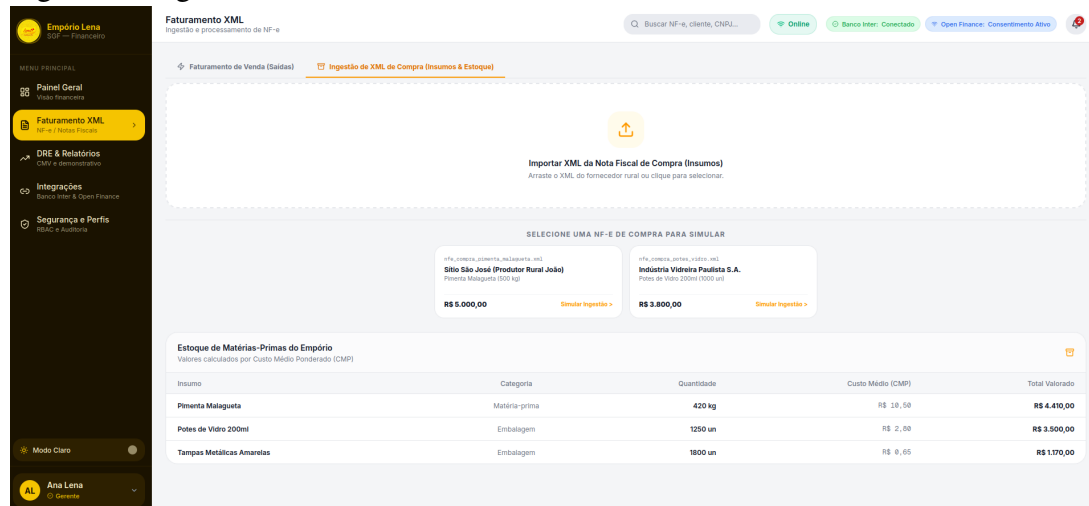
Voltada ao gerenciamento de fornecedores (insumos agrícolas e embalagens), esta aba operacionaliza a entrada de notas fiscais, conforme ilustrado na Figura 6. O fluxo realiza a extração dos itens contidos na NF-e de entrada (identificando ingredientes como pimenta malagueta ou recipientes de vidro), exibindo suas quantidades e valores unitários.

A interface simula a integração com o módulo de estoque no backend, efetuando o recálculo imediato do Custo Médio Ponderado (CMP) dos insumos conforme a fórmula matemática padrão:

$$\text{CMP} = \frac{(Q_{\text{anterior}} \times \text{CMP}_{\text{anterior}}) + (Q_{\text{adquirido}} \times P_{\text{aquisição}})}{Q_{\text{anterior}} + Q_{\text{adquirido}}} \quad (6.1)$$

Em que Q representa as quantidades de itens em estoque e P representa o preço unitário de aquisição. O valor atualizado do estoque e o novo CMP são consolidados e exibidos visualmente na tabela de inventário da aplicação.

Figura 6 – Ingestão de XML



Fonte: Autoria própria (2026)

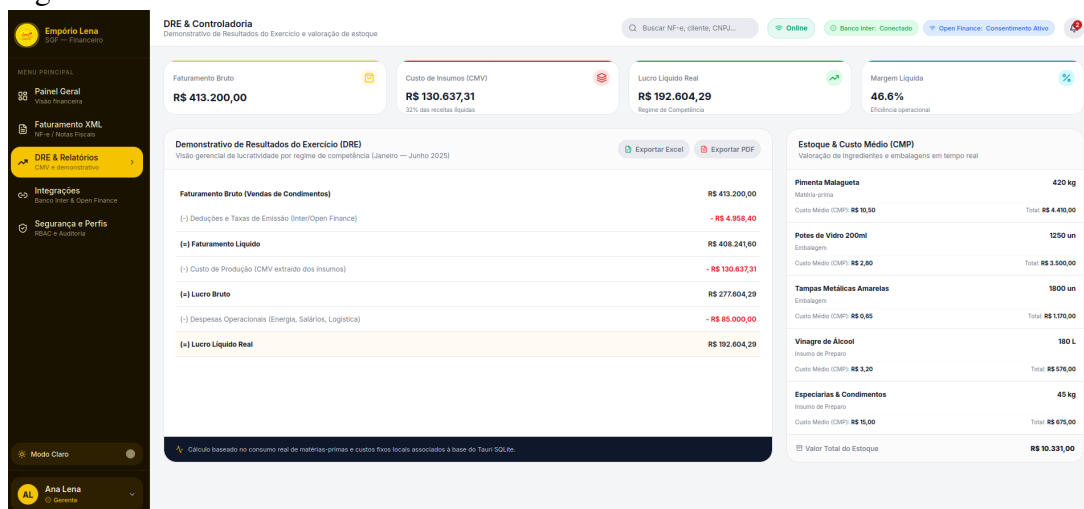
6.2.4 Demonstrativo de Resultados do Exercício (DRE) e Relatórios

Viabiliza o acompanhamento econômico do Empório Lena sob o regime de competência, conforme apresentado na Figura 7. O principal diferencial reside na apuração do Custo de Mercadorias Vendidas (CMV), calculado dinamicamente a partir do consumo histórico de matérias-primas (como pimenta malagueta e condimentos) e embalagens (como potes de vidro e tampas metálicas), cujos custos unitários são atualizados pelo Custo Médio Ponderado (CMP) processado em tempo real no inventário.

A interface exibe quatro painéis superiores com os seguintes consolidadores econômicos:

- **Faturamento Bruto:** Soma acumulada das vendas de condimentos realizadas no período.
- **Custo de Insumos (CMV):** Representação do custo real dos insumos consumidos na fabricação, parametrizada heurísticamente em 32% das receitas líquidas conforme dados históricos de CMP do estoque de condimentos.
- **Lucro Líquido Real:** Resultado obtido após a dedução de taxas, impostos e custos operacionais fixos.
- **Margem Líquida:** Indicador de eficiência operacional, expresso pela razão percentual entre o lucro líquido e a receita bruta.

Figura 7 – Demonstrativo de Resultados e Emissões

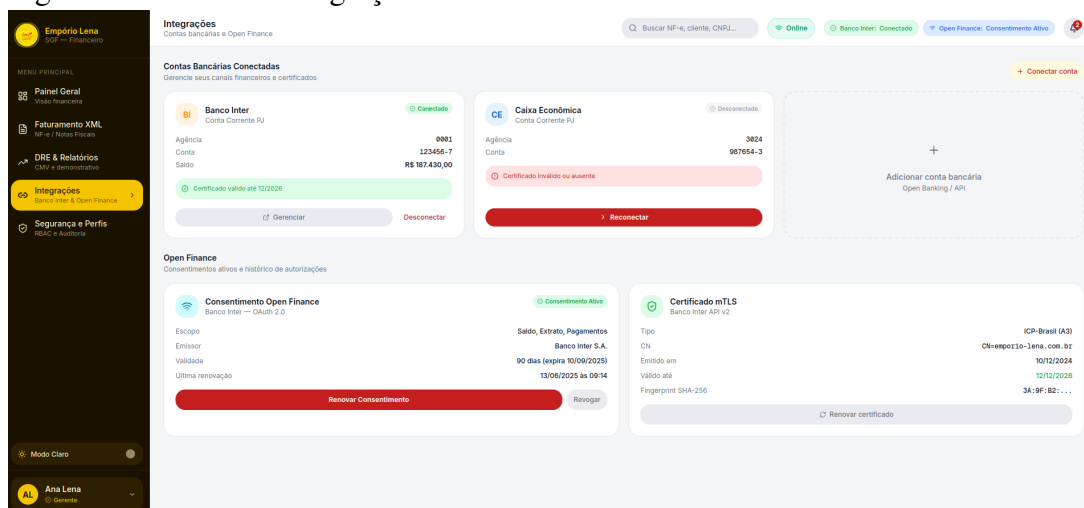


Fonte: Autoria própria (2026)

6.2.5 Integrações Bancárias e Open Finance

O sistema disponibiliza interfaces de configuração destinadas às conexões com APIs financeiras, mostradas na Figura 8. Nestas telas, o gerente cadastra credenciais, *tokens* de autorização e chaves criptográficas para o Banco Inter e para o ecossistema do *Open Finance* nacional.

Figura 8 – Gestão de integrações



Fonte: Autoria própria (2026)

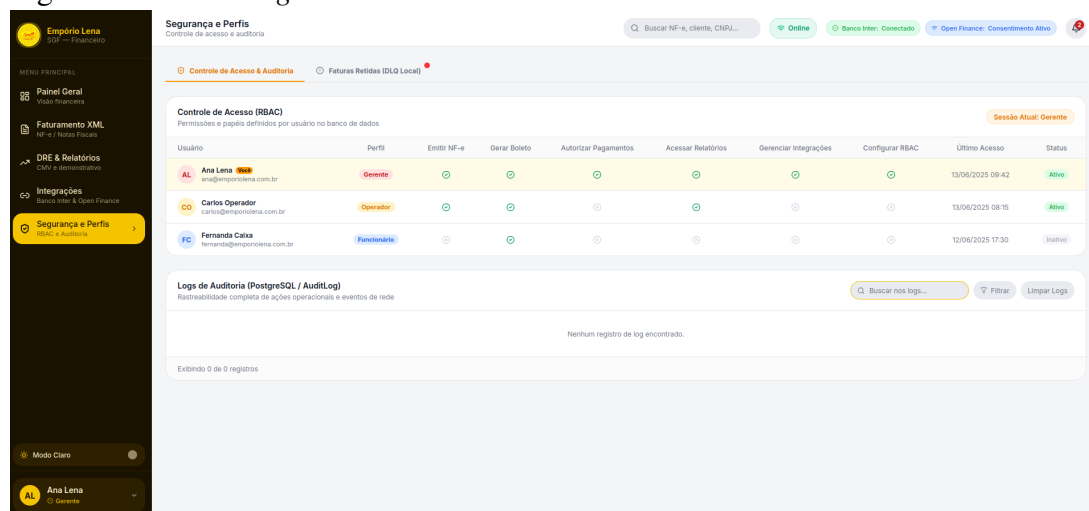
6.2.6 Segurança, Controle de Acesso e Perfis

A tela de Segurança e Perfis, ativada a partir do atalho de mesmo nome na Sidebar e ilustrada na Figura 9, foi desenvolvida como console administrativo do sistema para demonstração

dos mecanismos de segurança lógica, perfis de usuários e rastreabilidade da aplicação. A interface organiza-se em duas abas principais:

- **Controle de Acesso Baseado em Papéis:** Exibe a matriz de permissões do sistema no formato de tabela comparativa. A interface mapeia três perfis de usuários principais (Gerente, Operador e Funcionário) associados a permissões granulares de execução. A interface destaca visualmente a conta conectada na sessão corrente, permitindo a validação experimental de restrições de permissões para cada perfil.
- **Logs de Auditoria (PostgreSQL / AuditLog):** Console de monitoramento contínuo que expõe a rastreabilidade das ações efetuadas na plataforma. Cada registro de *log* será persistido com propriedades estruturadas da operação finalizada por um usuário. O console oferece mecanismos integrados para filtragem rápida, busca textual de eventos e funcionalidade para expurgo manual dos logs de teste.

Figura 9 – Visão e *log* de acessos



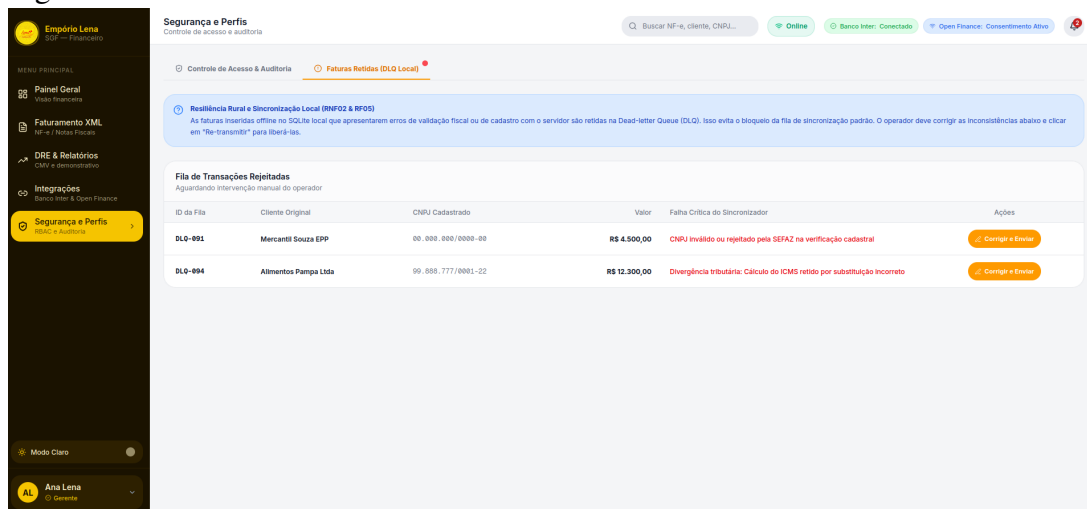
Fonte: Autoria própria (2026)

6.2.7 Fila de Faturas Retidas Localmente

Com o propósito de complementar a resiliência operacional (RNF02), a aba de Faturas Retidas, ilustrada na Figura 10, gerencia de forma isolada as notas fiscais importadas offline que apresentaram rejeições fiscais ou cadastrais na tentativa de sincronização automática com o servidor em nuvem. Este mecanismo evita que transações defeituosas (com erros como CNPJ inválido ou substituição tributária incorreta de ICMS) travem a fila geral de transmissão.

A interface exibe a tabela Fila de Transações Rejeitadas, na qual cada item presente representa uma transação que o gestor terá que analisar, corrigir e reenviar manualmente.

Figura 10 – Faturas retidas



Segurança e Perfis
Controle de acesso e auditoria

🔍 Buscar NF-e, cliente, CNPJ... Online Banco Inter: Conectado Open Finance: Consentimento Ativo

🔗 Controle de Acesso & Auditoria Faturas Retidas (DLO Local)

Resiliência Rural e Sincronização Local (RNF02 & RNF05)
As faturas inseridas offline no SQUE local que apresentarem erros de validação fiscal ou de cadastro com o servidor são retidas na Dead-letter Queue (DLQ). Isso evita o bloqueio da fila de sincronização padrão. O operador deve corrigir as inconsistências abaixo e clicar em "Re-transmitir" para liberá-las.

Fila de Transações Rejeitadas
Aguardando intervenção manual do operador

ID da Fila	Cliente Original	CNPJ Cadastrado	Valor	Falha Crítica do Sincronizador	Ações
DLO-491	Mercantil Souza EPP	00.800.800/0000-00	R\$ 4.500,00	CNPJ inválido ou rejeitado pela SEFAZ na verificação cadastral	Corrigir e Reenviar
DLO-494	Alimentos Pampa Ltda	99.800.777/0001-22	R\$ 12.300,00	Divergência tributária: Cálculo do ICMIS retido por substituição incorreto	Corrigir e Reenviar

Modo Claro AL Ana Lena AL

Fonte: Autoria própria (2026)

6.2.8 Resultados de Usabilidade e Validação Qualitativa

Com base no Roteiro de Avaliação de Usabilidade e Coleta de Feedback apresentado no Apêndice A, foram conduzidas sessões de demonstração prática e entrevistas semiestruturadas com os gestores e usuários do Empório Lena para validação da interface de alta fidelidade desenvolvida em React. As sessões práticas confirmaram a aderência visual da interface no momento em que foi acessada via navegador.

Durante a avaliação, os gestores elogiaram o padrão de *design* adotado, destacando que a paleta de cores neutras. A usabilidade geral do sistema foi classificada pelos participantes como simples e fluida, indicando baixa curva de aprendizado para a execução das rotinas diárias de importação de documentos fiscais e conciliação bancária.

Contudo, sob uma perspectiva de crítica acadêmica e aprimoramento de engenharia de fatores humanos, os usuários pontuaram a ocorrência de uma sobrecarga visual na Tela Principal (Dashboard). A exibição simultânea de múltiplos indicadores econômicos, gráficos de projeção com controles deslizantes e o console de depuração em tempo real do *webhook* gerou fadiga informativa. Como recomendação de melhoria para iterações futuras, os avaliadores sugeriram a simplificação do layout do painel, agrupando ou ocultando elementos de monitoramento secundários. Por fim, confirmou-se que todas as funcionalidades simuladas durante o roteiro de testes operaram de forma estável e atenderam estritamente ao escopo funcional previamente acordado entre as partes interessadas.

6.3 Camada de Serviços Backend (Spring Boot)

O desenvolvimento do backend foi projetado sob os princípios da Arquitetura Hexagonal (ou padrão de Portas e Adaptadores). Este padrão isola as regras de negócio centrais (o núcleo do domínio financeiro) de tecnologias externas e APIs de terceiros, como mostrado no Diagrama de Componentes da Figura 11 e no Diagrama de Implantação da Figura 12.

Desta forma, a integração com diferentes instituições bancárias e *Open Finance* é realizada por meio de acoplamento fraco, no qual o domínio de faturamento consome uma interface definida como porta, cujas implementações concretas atuam como adaptadores externos.

A modularização do backend divide-se em componentes e padrões de projeto essenciais:

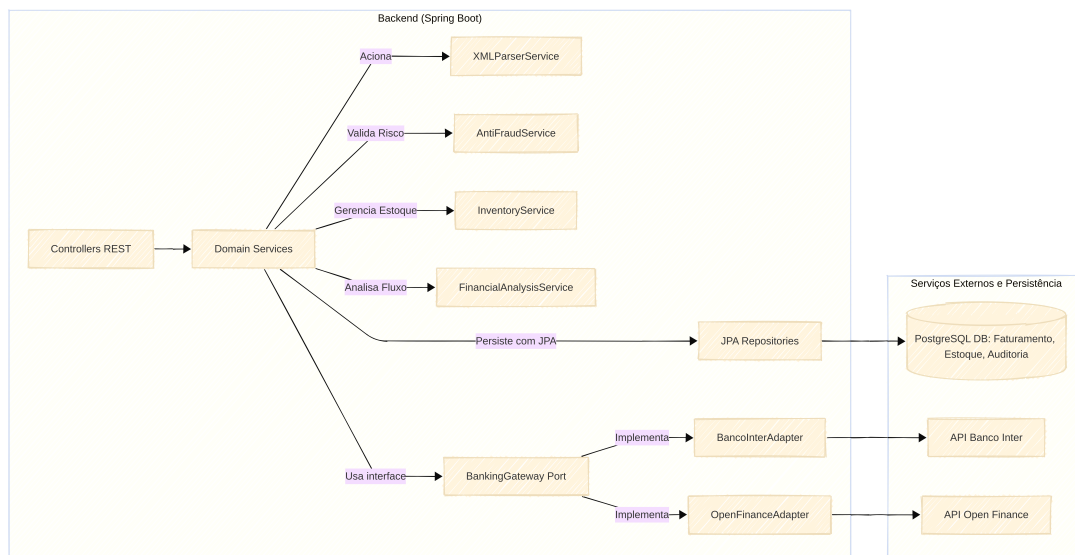
- **BankingGateway**: Porta (*Strategy*). Interface Java que padroniza os métodos de comunicação com as instituições de pagamento (emissão, consulta de *status*, conciliação e cancelamento).
- **BancoInterAdapter**: Adaptador (*Adapter*). Implementa a interface **BankingGateway**, traduzindo as chamadas de método genéricas para requisições HTTPS utilizando o protocolo HTTP/2, *tokens* OAuth2 temporários e autenticação mTLS.
- **OpenFinanceAdapter**: Adaptador (*Adapter*). Responsável pela comunicação com APIs de consentimento do *Open Finance* brasileiro, lidando com redirecionamentos e iniciação de pagamentos.
- **XMLParserService**: Serviço. Utiliza a especificação DOM do Java para análise estruturada do arquivo XML de NF-e importado (seja nota fiscal de saídas de venda ou entradas de fornecedores). O serviço valida o esquema do XML (*XML Schema Definition* - XSD) e extrai os blocos estruturados correspondentes para conversão em DTOs específicos.
- **AntiFraudService**: Serviço. Componente lógico que executa validações heurísticas antes do envio de cobranças. Ele analisa dados históricos de recebíveis, calculando o indicador de risco do cliente. Além disso, intercepta chamadas de *webhooks* de pagamento, validando assinaturas criptográficas de notificação por meio de chaves públicas dos emissores.
- **InventoryService**: Serviço. Componente transacional encarregado do processamento de entradas de estoque. Ao receber o DTO de uma nota fiscal de fornecedor analisada pelo **XMLParserService**, este serviço atualiza o saldo físico dos insumos agrícolas e embalagens no banco de dados relacional e recalcula de forma ponderada o Custo Médio Ponderado (CMP) unitário de cada item afetado.

- **FinancialAnalysisService:** Serviço. Centraliza a consolidação das informações contábeis para geração da Demonstração do Resultado do Exercício (DRE) e realiza o cálculo assíncrono dos diagnósticos financeiros exibidos pelo Diretor Financeiro Virtual (Virtual CFO), simulando cenários baseados na taxa de inadimplência média e custos fixos.

6.3.1 Diagrama de Componentes

O diagrama de componentes ilustra a organização dos módulos do sistema, evidenciando o isolamento das regras de domínio em relação aos adaptadores externos de banco de dados e APIs bancárias.

Figura 11 – Diagrama de Componentes

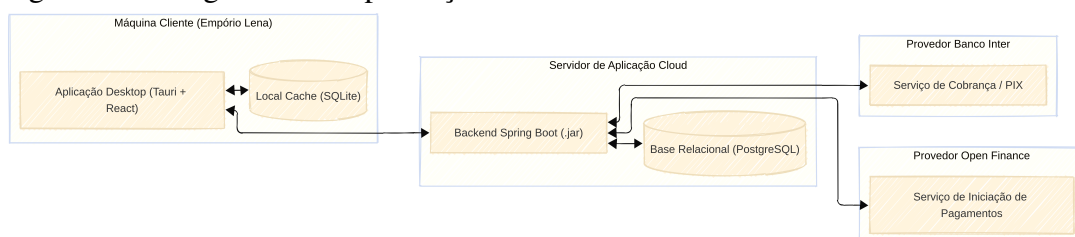


Fonte: Autoria própria (2026)

6.3.2 Diagrama de Implantação

O diagrama de implantação ilustra a distribuição física dos componentes nos nós de execução, destacando a comunicação segura pela rede.

Figura 12 – Diagrama de Implantação



Fonte: Autoria própria (2026)

7 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as considerações finais acerca do processo de modelagem e implementação da interface do SGF customizado para o Empório Lena, bem como as limitações identificadas e as propostas de evolução técnica para a implementação completa da ferramenta.

7.1 Considerações Finais

A realização deste trabalho permitiu consolidar a modelagem e a especificação de uma solução tecnológica sob medida, estruturada especificamente para responder às fragilidades administrativas e de infraestrutura de uma microempresa no ramo de processamento de condimentos. A investigação partiu de um diagnóstico situacional detalhado, no qual se evidenciou a ineficiência e a vulnerabilidade operacional dos controles manuais e planilhas desconexas anteriormente utilizados na empresa.

Os objetivos específicos definidos foram plenamente alcançados. O levantamento e a especificação de requisitos baseados no modelo Volere desempenharam um papel primordial na estruturação do ecossistema do projeto, indo além do simples mapeamento de funcionalidades para alinhar os objetivos estratégicos de negócio e as necessidades das partes interessadas.

A validação da interface do sistema, desenvolvida em React e demonstrada no momento via navegador, obteve retornos qualitativos favoráveis por parte da gestão do Empório Lena. A interface foi caracterizada como simples e de usabilidade fluida, sendo elogiada pelo uso de uma paleta neutra em escala de cinzas e detalhes em âmbar com sinalização semântica cromática. No entanto, em termos de refinamento ergonômico, os usuários indicaram uma sobrecarga de informações na tela principal (Dashboard) devido ao excesso de indicadores exibidos simultaneamente, gerando fadiga informativa que demandará ajustes no layout. A simulação das demais funcionalidades, como o processamento de XML e a gestão de faturas retidas, operou de modo robusto dentro do escopo planejado.

Por fim, o ataque de *ransomware* sofrido pelo Empório Lena em meados de abril de 2026 alterou de forma profunda a percepção de valor sobre o projeto. A ocorrência demonstrou empiricamente os perigos decorrentes da falta de redundância e de proteção adequada nos fluxos operacionais diários. Diante dessa realidade, o sistema proposto deixou de ser visto meramente como uma ferramenta de automação administrativa e conciliação bancária, passando a configurar-se como um ativo tecnológico essencial de segurança da informação e resiliência corporativa. A

modelagem de mecanismos de proteção contra fraudes externas e ameaças internas, o isolamento em banco local criptografado em AES e a continuidade operacional passaram a ser pontos ainda maiores de atenção.

7.2 Trabalhos Futuros

Como o escopo atual concentrou-se na modelagem arquitetural e no desenvolvimento da interface da aplicação, a continuidade deste trabalho pressupõe a execução das seguintes etapas de engenharia e pesquisa:

- **Simplificação de Interface do Dashboard:** Reestruturar o layout da Tela Principal para atenuar a densidade de componentes gráficos e indicadores informados de forma simultânea, visando reduzir a fadiga informativa identificada nas sessões de avaliação de usabilidade qualitativa. Quando finalizados os ajustes visuais, a aplicação React estará habilitada ao empacotamento realizado pelo Tauri.
- **Políticas de Backup e Segurança Local:** Desenvolver rotinas robustas de *backup* local automatizado e políticas de segurança na ponta operacional (cliente desktop em Tauri) para mitigar o impacto de potenciais novas infecções por *ransomware* ou outros incidentes de perda de dados;
- **Codificação do Backend e Persistência:** Implementar os serviços propostos em Java 21 com Spring Boot e configurar o esquema do banco de dados relacional PostgreSQL, incluindo o log de auditoria transacional;
- **Homologação e Segurança Bancária:** Configurar os certificados de segurança X.509 e estabelecer a comunicação bidirecional *Mutual Transport Layer Security* (mTLS) com a API do Banco Inter em ambiente de produção, garantindo o armazenamento de forma segura de credenciais na aplicação desktop;
- **Sincronização Assíncrona Offline:** Implementar a lógica de sincronização da fila de faturas retidas localmente (banco de dados PostgreSQL embarcado no Tauri) com o banco de dados em nuvem, tratando possíveis concorrências e conflitos de estado no restabelecimento da conexão;
- **Validação Quantitativa de Usabilidade:** Realizar testes empíricos de usabilidade pós-implementação com aplicação de questionários padronizados, como o *System Usability Scale* (SUS) ou baseando-se no modelo *Technology Acceptance Model* (TAM), para avaliar estatisticamente a facilidade de uso e a intenção de adoção contínua da ferramenta.

REFERÊNCIAS

- CHIGUANO, N. E. C. **Sistema de Gestión Financiera para la Asociación de Profesores de la Escuela Politécnica Nacional**: componente de conciliación. Dissertação (Mestrado) — Escuela Politécnica Nacional, Quito, Equador, 2025. Disponível em: <https://bibdigital.epn.edu.ec/handle/15000/26880>. Acesso em: 23 maio 2026.
- COCKBURN, A. **Hexagonal Architecture (Ports and Adapters)**. 2005. Disponível em: <https://alistair.cockburn.us/hexagonal-architecture/>. Acesso em: 5 jun. 2026.
- Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). **Portaria CNPq nº 2.664**: Política de Integridade na Atividade Científica do CNPq. Brasília: CNPq, 2026. Disponível em: http://memoria2.cnpq.br/web/guest/view/-/journal_content/56_INSTANCE_0oED/10157/23142775. Acesso em: 9 jul. 2026.
- ENZZO, C. M.; FAUSTO, N. S.; RIBEIRO, G. d. C. *et al.* **E-commerce invertido**: valorizando microempreendedores. São Paulo: Universidade Paulista (UNIP), 2024. Disponível em: <https://repositorio.unip.br/wp-content/uploads/taianacan-items/99630/179222/Grupo-1-E-commerce-Invertido.pdf>. Acesso em: 13 maio 2026.
- IGNEV, I. Modular architecture for financial process automation: from Excel/ini to API and ERP integrations. In: **Proceedings of the IEEE International Conference on Artificial Intelligence (ICAI)**. Boston, EUA: [s.n.], 2025. p. 408–411. Disponível em: <https://doi.org/10.1109/ICAI67591.2025.11324882>. Acesso em: 24 maio 2026.
- MACHADO, K. A.; SANTOS, K. A. d. O.; SOUZA, L. S. d. **Sistema de Gerenciamento de Projetos para Pequenas Empresas**. São Paulo: Faculdade de Tecnologia de São Paulo (FATEC), 2024. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/28481>. Acesso em: 13 maio 2026.
- MASUDA, T.; KOBAYASHI, M.; YATANI, K. Synapse: interactive support on photoemission spectroscopy measurement and analysis for non-expert users. **Journal of Synchrotron Radiation**, v. 30, n. 6, p. 1127–1134, 2023. Disponível em: <https://journals.iucr.org/s/issues/2023/06/00/ve5171/index.html>. Acesso em: 14 maio 2026.
- MILICHIEWICZ, R.; BADUROWICZ, M. Comparative analysis of cross-platform application development tools in terms of operating system integration. **Journal of Computer Sciences Institute**, v. 36, p. 357–364, 2025. Disponível em: <https://ph.pollub.pl/index.php/jcsi/article/view/7831>. Acesso em: 2 jun. 2026.
- PIRES, S. P. Estudo sobre a gestão financeira em micro e pequenas empresas na quarta colônia. **Saber Humano**, p. 394–421, 2024. Disponível em: <https://saberhumano.emnuvens.com.br/sh/article/view/685>. Acesso em: 12 maio 2026.
- RAVALJI, V. Y.; JAIN, S. Automating financial reconciliation through restful apis. **Journal of Quantum Science and Technology**, v. 2, n. 1, 2025. Disponível em: <https://jqst.org/index.php/j/article/view/151>. Acesso em: 19 maio 2026.
- SANTIAGO, A. R. d. P. G.; LEAL, L. V. Desafios do microempreendedor individual florianense com relação à gestão financeira de seu negócio. **Management in Perspective**, v. 5, n. 1, p. 44–71, 2024. Disponível em: <https://seer.ufu.br/index.php/RevistaMiP/article/view/71860>. Acesso em: 22 maio 2026.

SILVA, A. F. C.; NEGREIROS, A. C. S. V.; LUCENA, A. D.; SENA, D. C. Integração de sistemas erp: otimização de API para eficiência e segurança financeira na conciliação de cartões de crédito e débito. In: **Anais do Encontro Nacional de Engenharia de Produção (ENEGEP)**. Rio de Janeiro: [s.n.], 2025. Disponível em: https://doi.org/10.14488/enegep2025_tn_wg_429_2108_49939. Acesso em: 17 maio 2026.

TARSKICH, I. **Wie performt Tauri im Vergleich to Electronjs?** Analyse der Leistung und der Erfahrungen von Entwicklern bei der Erstellung von Desktop-Anwendungen unter Verwendung von Webtechnologien. Dissertação (Mestrado) — FH Joanneum, Graz, Áustria, 2023. Disponível em: <https://resolver.obvsg.at/urn:nbn:at:at-fhj:1-20795>. Acesso em: 2 jun. 2026.

TAVARES, L. L.; PEREIRA, E. M. C.; ROCHA, J. I. B. *et al.* **Plataforma automatizada de informação sobre licitações:** desenvolvimento e implementação no contexto do estado da Paraíba. João Pessoa: Instituto Federal da Paraíba (IFPB), 2024. Disponível em: <https://repositorio.ifpb.edu.br/handle/177683/4178>. Acesso em: 16 maio 2026.

UMAVEZI, J. U.; ODUKALE, B. API development and modelling: streamlining financial technology integration for seamless digital transactions. **International Journal of Research Publication and Reviews**, v. 6, n. 1, p. 1556–1573, 2025. Disponível em: <https://doi.org/10.55248/gengpi.6.0125.0328>. Acesso em: 28 maio 2026.

APÊNDICE A – ROTEIRO DE AVALIAÇÃO DE USABILIDADE E COLETA DE FEEDBACK DA INTERFACE

Este apêndice apresenta o roteiro de demonstração interativa e o conjunto de questões semiestruturadas aplicadas aos gestores e funcionários do Empório Lena para a validação qualitativa da interface desenvolvida.

Tópicos da Demonstração Prática

Durante a sessão, os participantes interagiram com a interface executando as seguintes ações simuladas:

1. Realização de *login* e validação em duas etapas (MFA);
2. Navegação e interpretação do Gráfico de Projeção de Fluxo de Caixa no *Dashboard*;
3. Importação assistida de Nota Fiscal Eletrônica (NF-e) via arrastar e soltar (XML de venda);
4. Ingestão de XML de compra de insumos e visualização do recálculo automático do Custo Médio Ponderado (CMP);
5. Consulta e exportação do Demonstrativo de Resultados do Exercício (DRE);
6. Configuração de conexões de APIs de bancos e análise dos logs de auditoria de acessos.

Perguntas da Entrevista Semiestruturada

Após o uso assistido, os participantes responderam às seguintes perguntas norteadoras:

- Aderência Visual: O esquema de cores em escala de cinzas e detalhes em âmbar facilitou a leitura das informações contábeis? A sinalização em verde, amarelo e vermelho foi intuitiva para identificar o *status* das transações?
- Carga Cognitiva: Algum termo técnico utilizado na interface (como CMP, CMV ou DRE) gerou dúvida durante o uso? Se sim, como você sugeriria simplificar?
- Facilidade de Aprendizado: O processo de importação de arquivos XML de notas fiscais facilitou a alimentação de dados em relação à planilha Excel utilizada anteriormente?
- Resiliência e Erros: A visualização da fila de faturas retidas localmente ajudou a compreender quais notas fiscais apresentaram problemas e necessitam de correção manual?
- Críticas e Sugestões: Quais funcionalidades apresentadas você considera mais importantes para o cotidiano do Empório Lena e o que ainda sente falta na interface atual?

APÊNDICE B – DETALHAMENTO GRANULAR DOS REQUISITOS DO SISTEMA

Este apêndice apresenta a matriz de requisitos funcionais e não-funcionais detalhados de forma granular, servindo de base para o projeto de usabilidade, navegabilidade e comportamento da interface do Sistema de Gestão Financeira.

Requisitos Funcionais Granulares

Os Requisitos Funcionais (RF) definem o comportamento operacional da aplicação, mapeando detalhadamente cada recurso presente nas telas da interface. O Quadro 7 detalha os requisitos voltados às etapas de autenticação, painel financeiro consolidado e fluxo de faturamento fiscal.

O Quadro 8 apresenta a segunda parte dos requisitos funcionais, abrangendo o processamento offline de faturamento, a tela de DRE, a configuração de integrações bancárias e as permissões de segurança.

Requisitos Não-Funcionais Granulares

Os Requisitos Não-Funcionais (RNF) especificam restrições técnicas, requisitos de desempenho, de segurança e de arquitetura que garantem o correto funcionamento do ecossistema de software. O Quadro 9 consolida os critérios técnicos voltados à segurança e à criptografia.

O Quadro 10 detalha as restrições de arquitetura, resiliência de rede e desempenho, garantindo usabilidade e tolerância a falhas na zona rural.

Quadro 7 – Requisitos Funcionais (Parte 1: Autenticação, *Dashboard* e Faturamento).

Código	Nome	Detalhes	Atores	Importância
RF001	Autenticação de Usuário	O sistema deve validar as credenciais de e-mail e senha do usuário, gerando o token de sessão JWT correspondente para acesso.	Usuário	ALTA
RF002	Seletor de Perfis no <i>Login</i>	O formulário de <i>login</i> deve exibir botões de simulação para entrada rápida como Gerente, Operador ou Funcionário.	Usuário	MÉDIA
RF003	Exibição de KPIs do <i>Dashboard</i>	O painel geral deve exibir cartões consolidados contendo Saldo em Caixa, Contas a Receber, Faturamento Mensal e Contas a Pagar.	Gestor, Operador	ALTA
RF004	Gráfico Interativo de Caixa	O <i>dashboard</i> deve gerar gráficos interativos de fluxo de caixa histórico (entradas <i>versus</i> saídas) atualizados em tempo real.	Gestor, Operador	MÉDIA
RF005	Simulação com Sliders Financeiros	O <i>dashboard</i> deve disponibilizar controles deslizantes para simular o impacto de taxas de inadimplência e custos fixos na receita.	Gestor, Operador	MÉDIA
RF006	Alertas de Projeção de Caixa	O sistema deve recalcular o fluxo de caixa futuro com base nos sliders e emitir avisos textuais de alerta caso o saldo seja crítico.	Gestor, Operador	MÉDIA
RF007	Terminal de Log do <i>Webhook</i>	O <i>dashboard</i> deve expor um console que simula o recebimento do <i>webhook</i> de pagamento do Banco Inter e detalha suas etapas de processamento.	Gestor, Operador	MÉDIA
RF008	<i>Upload</i> e Arrastar-e-Soltar de XML	A interface de faturamento deve dispor de área interativa para arrastar ou selecionar arquivos XML de notas fiscais (NF-e).	Gestor, Operador	ALTA
RF009	Validação Estrutural de XML	O sistema deve realizar a validação do esquema do XML importado e recusar arquivos corrompidos ou em formatos incompatíveis.	Gestor, Operador	ALTA
RF010	Preenchimento Automático da NF-e	O sistema deve extrair Razão Social, CNPJ, itens de venda, impostos e valor total do XML e preencher os campos do formulário.	Gestor, Operador	ALTA
RF011	Análise de Risco Antifraude	O sistema deve comparar o CNPJ da nota fiscal importada com a base histórica do empório e pontuar o nível de risco de inadimplência.	Gestor, Operador	ALTA
RF012	Desbloqueio de Risco via MFA	O sistema deve impedir a emissão automática de notas sob risco e exigir a validação de código MFA temporário do perfil Gerente.	Gerente	ALTA

Fonte: Autoria própria (2026)

Quadro 8 – Requisitos Funcionais (Parte 2: Estoque, Relatórios, Integrações e Segurança).

Código	Nome	Detalhes	Atores	Importância
RF013	Faturamento Offline	O sistema deve habilitar a finalização local do faturamento quando detectar indisponibilidade de internet, guardando dados localmente.	Gestor, Operador	ALTA
RF014	Demonstrativo DRE Gerencial	O sistema deve exibir o demonstrativo DRE detalhando Faturamento Bruto, Deduções, CMV, Lucro Bruto, Despesas e Lucro Líquido Real.	Gestor, Operador	MÉDIA
RF015	Cálculo de CMV baseado no CMP	O DRE deve calcular de forma automatizada o custo das mercadorias vendidas usando o Custo Médio Ponderado dos insumos de estoque.	Gestor, Operador	ALTA
RF016	Exportação de Relatórios	O DRE deve disponibilizar funções para exportação dos dados contábeis em planilha Excel (XLSX) e em formato PDF.	Gestor, Operador	MÉDIA
RF017	Filtro de Competência de DRE	A tela de DRE deve disponibilizar seletores de datas para definir o intervalo temporal da apuração das notas fiscais.	Gestor, Operador	MÉDIA
RF018	Configuração de Integrações	O sistema deve dispor de interface para parametrização de chaves e URLs de integração do Banco Inter e do Open Finance.	Gerente	MÉDIA
RF019	Upload de Certificados Digitais	O sistema deve permitir o <i>upload</i> seguro de arquivos de chave privada e certificado público A1 da empresa para conexões bancárias.	Gerente	MÉDIA
RF020	Simulação do Open Finance	O sistema deve gerar interface de simulação de iniciação de pagamento via Pix sob o fluxo do ecossistema Open Finance nacional.	Gestor, Operador	MÉDIA
RF021	Matriz de Permissões RBAC	A interface de segurança deve expor uma tabela mapeando permissões de uso do sistema entre os perfis de Gerente, Operador e Funcionário.	Gerente	ALTA
RF022	Escopo de Acesso do Gerente	O perfil Gerente deve possuir acesso total, incluindo faturamento, relatórios de DRE, gerenciamento de mTLS e configuração de matriz RBAC.	Gerente	ALTA
RF023	Escopo de Acesso do Operador	O perfil Operador deve estar limitado a emitir notas e consultar DRE, sendo impedido de configurar integrações bancárias e segurança.	Operador	ALTA
RF024	Escopo de Acesso do Funcionário	O perfil Funcionário deve ficar restrito à emissão de notas fiscais, sendo bloqueado de consultar DRE, gerenciar credenciais ou auditar logs.	Funcionário	ALTA
RF025	Visualização de Logs de Auditoria	O sistema deve exibir logs contendo carimbo de data/hora, módulo afetado, ação, usuário/perfil executor e severidade do evento.	Gerente	ALTA
RF026	Exposição de Faturas Retidas (DLQ)	A aba de segurança deve exibir faturas locais cuja sincronização automática falhou devido a inconsistências cadastrais ou tributárias.	Gestor, Operador	ALTA
RF027	Edição e Reenvio Manual da DLQ	O operador deve poder retificar dados cadastrais (Razão Social, CNPJ e valor) diretamente no modal da DLQ e acionar a retransmissão.	Gestor, Operador	ALTA

Fonte: Autoria própria (2026)

Quadro 9 – Requisitos Não-Funcionais (Parte 1: Segurança e Rastreabilidade).

Código	Nome	Detalhes	Atores	Importância
RNF001	Autenticação mTLS	As conexões diretas entre o <i>backend</i> Spring Boot e a API do Banco Inter devem exigir autenticação mTLS mútua por canal criptografado.	Sistema	ALTA
RNF002	Validação de <i>Webhooks</i>	O ouvinte de <i>webhooks</i> bancários deve validar a assinatura criptográfica dos payloads recebidos para mitigar ataques de injeção.	Sistema	ALTA
RNF003	Criptografia de Credenciais	Os segredos e chaves das instituições financeiras configurados no banco de dados devem ser mantidos sob criptografia simétrica forte AES.	Sistema	ALTA
RNF004	Expiração Segura de JWT	Os <i>tokens</i> de acesso JWT do usuário devem possuir ciclo de vida curto, com tempo de expiração programável e armazenamento seguro na sessão.	Sistema	ALTA
RNF005	Logs de Auditoria do Banco	O sistema deve registrar eventos críticos em banco de dados de maneira ordenada por timestamps e impossibilitar edições ou exclusões diretas.	Sistema	ALTA
RNF006	Isolamento Transacional na DLQ	Notas fiscais offline que falharem na sincronização por inconsistências cadastrais não devem bloquear a fila de envio principal do cache.	Sistema	ALTA

Fonte: Autoria própria (2026)

Quadro 10 – Requisitos Não-Funcionais (Parte 2: Arquitetura, Conectividade e Desempenho).

Código	Nome	Detalhes	Atores	Importância
RNF007	Banco de Dados Embarcado (Tauri)	A persistência local em caso de indisponibilidade de internet deve ser processada em banco de dados PostgreSQL nativo executado no Tauri.	Sistema	ALTA
RNF008	Algoritmo de Sincronismo (Smart Sync)	A sincronização de dados retidos offline deve ser ativada de forma automática pelo cliente <i>desktop</i> React assim que a internet retornar.	Sistema	ALTA
RNF009	Controle de Retentativas de Sincronia	O Smart Sync deve aplicar algoritmo de backoff exponencial progressivo (2s, 4s, 8s, 16s...) para evitar saturação do canal de internet rural.	Sistema	ALTA
RNF010	Propriedades ACID em Operações	A persistência e o recálculo de inventário e CMV no <i>backend</i> PostgreSQL central devem ser transacionais e em conformidade com o padrão ACID.	Sistema	ALTA
RNF011	Desempenho de UI e Multithreading	As operações de ping de gateway e de leitura de arquivos XML volumosos devem rodar em threads secundárias, sem travar a interface.	Sistema	ALTA
RNF012	Abstração de Gateways Bancários	A arquitetura de comunicação bancária do <i>backend</i> deve empregar interfaces baseadas no padrão Strategy para facilitar a troca de provedores.	Sistema	MÉDIA

Fonte: Autoria própria (2026)