ADS IFBA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FinTrack: um repositório inteligente para organização e recuperação de comprovantes de transações financeiras

Trabalho de Conclusão de Curso

Jeferson Leandro de Oliveira dos Reis Santana

José Dihego da Silva Oliveira
Orientador

Instituto Federal da Bahia - IFBA
Curso de Análise e Desenvolvimento de Sistemas
Campus Salvador

Salvador, Bahia, Brasil
14 de Julho de 2026

Sumário

1 Visão Geral **1**
1.1 Declaração do problema . . . 1
1.2 Proposta de solução de software . . . 4
1.3 Tecnologias da solução . . . 7
1.3.1 Frontend . . . 7
1.3.2 Integrações nativas Android . . . 8
1.3.3 Banco de dados . . . 8
1.3.4 Processamento de dados . . . 9
1.3.5 Inteligência artificial . . . 10
1.3.6 Segurança e criptografia . . . 11
1.3.7 Armazenamento em nuvem . . . 11
1.3.8 Versionamento e DevOps . . . 12
1.3.9 Modelagem e documentação . . . 13
1.3.10 Testes . . . 13
1.4 Trabalhos relacionados . . . 14
1.4.1 *CamScanner* . . . 15
1.4.2 *Google Drive* . . . 15
1.4.3 *Expensify* . . . 15
1.4.4 *Mobills* . . . 16
1.4.5 Comparação das ferramentas . . . 16

2 Requisitos **19**
2.1 Requisitos funcionais . . . 19
2.2 Requisitos não funcionais . . . 22

3 Design **25**
3.1 Projeto UML . . . 26
3.2 Visão arquitetural . . . 32
3.3 Modelo de banco de dados . . . 34

4 Testes de Software **36**
4.1 Projeto de testes . . . 37

5 Implantação **44**
5.1 Projeto de implantação . . . 45

6 Manual do Usuário **46**

7 Limitações **77**

8 Conclusão **81**

9 Agradecimentos **82**

10 Apêndices **88**

A Diagrama de Classes **88**

B Diagrama de Implantação **90**

C Modelo Lógico do Banco de Dados **92**

D Modelo Físico do Banco de Dados **94**

E Diagrama de Sequência — Registro no FinTrack **96**

F Diagrama de Sequência — Recuperação no FinTrack **98**

G Evidências de Integração Contínua **100**

1 Visão Geral

A necessidade de organizar, preservar e recuperar comprovantes de transações financeiras é recorrente e crescente; contudo, as soluções disponíveis para essa finalidade não foram concebidas especificamente para o contexto de transações financeiras realizadas cotidianamente por pessoas físicas no Brasil. As alternativas existentes atendem a propósitos distintos, como digitalização genérica de documentos, armazenamento em nuvem, gestão de despesas corporativas ou controle financeiro pessoal, e não atendem de forma integrada às necessidades de gestão ágil desses documentos no dia a dia.

Diante desse cenário, este trabalho tem como objetivo geral desenvolver o FinTrack, um aplicativo móvel Android concebido como um repositório inteligente de comprovantes de transações financeiras, orientado ao uso cotidiano. A solução utiliza técnicas de reconhecimento óptico de caracteres (OCR), processamento de linguagem natural (PLN) e enriquecimento de dados fiscais e cadastrais. Esses recursos viabilizam o escaneamento, a importação, a organização semântica e a recuperação eficiente desses documentos, incorporando princípios de privacidade e autonomia dos dados do usuário.

A solução desenvolvida tem como público-alvo pessoas físicas no contexto brasileiro que realizam transações financeiras cotidianamente e precisam armazenar, organizar e recuperar comprovantes físicos digitalizados ou arquivos de comprovantes nativamente digitais de forma prática. O foco principal recai sobre usuários com autonomia para gerenciar suas próprias compras, pagamentos e transferências, especialmente adultos, mas sem excluir usuários jovens que já realizam operações financeiras e acumulam notas fiscais, recibos ou arquivos de comprovantes nativamente digitais. Dessa forma, a solução se direciona ao uso pessoal e recorrente, e não a cenários corporativos de contabilidade ou auditoria.

1.1 Declaração do problema

A gestão de comprovantes de transações financeiras — tais como notas fiscais, recibos e comprovantes de pagamentos ou de transferências digitais — constitui uma atividade recorrente e relevante no cotidiano da vida adulta. Esses documentos estão associados a diversas finalidades, como controle de gastos, comprovação de despesas, processos de reembolso, cumprimento de obrigações fiscais e resolução de disputas comerciais, o que demanda sua adequada organização e preservação para posterior recuperação quando necessário.

Nesse cenário, o volume de comprovantes físicos e digitais gerados no cotidiano tem apresentado crescimento expressivo nos últimos anos. O Pix — sistema de pagamento instantâneo criado pelo Banco Central que permite a transferência de valores entre contas em segundos, a qualquer hora e dia [1] — ilustra bem essa tendência: um ano após seu lançamento, o Sistema de Pagamentos Instantâneos (SPI), infraestrutura por trás do Pix, liquidava 1,22 bilhão de transações interbancárias por mês, número que atingiu 5,71 bilhões em dezembro de 2024 [2]. Mais do que o volume absoluto, chama atenção o caráter estrutural desse crescimento — ainda que janeiro registre queda em relação a dezembro, seu volume supera o de novembro do ano anterior, evidenciando que o piso da curva se eleva continuamente, conforme ilustrado na Figura 1.

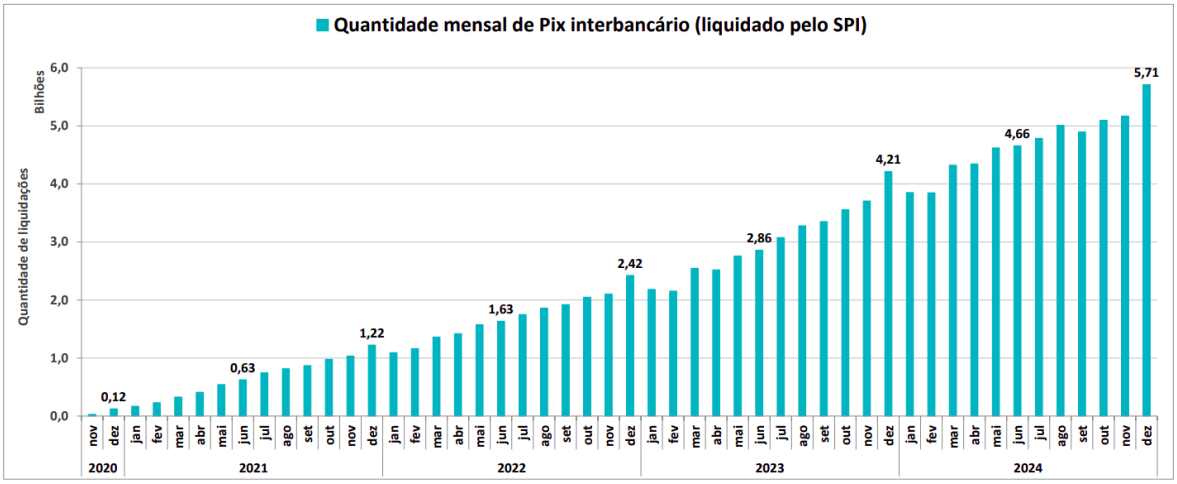


Figura 1: Evolução mensal dos Pix interbancários liquidados pelo SPI (2020–2024).
Fonte: Banco Central do Brasil [2].

Na prática, esse crescimento amplia a quantidade de transações que geram comprovantes no dia a dia. Segundo dados também divulgados pelo Banco Central, o Pix tornou-se, em 2024, o meio de pagamento mais difundido entre os brasileiros, sendo utilizado por 76,4% da população, à frente do dinheiro em espécie, utilizado por 68,9% [3], conforme ilustra a Figura 2. Essa substituição de pagamentos antes realizados em dinheiro é relevante porque o Pix não se limita a transferências entre pessoas, haja vista as suas diferentes modalidades, como pagamentos por QR Code, que permitem seu uso em compras em estabelecimentos comerciais presencialmente. Nesse sentido, transações antes feitas em dinheiro que, em muitos casos, não resultariam em um comprovante, passam a produzir registros digitais bancários e comprovantes impressos e eletrônicos.

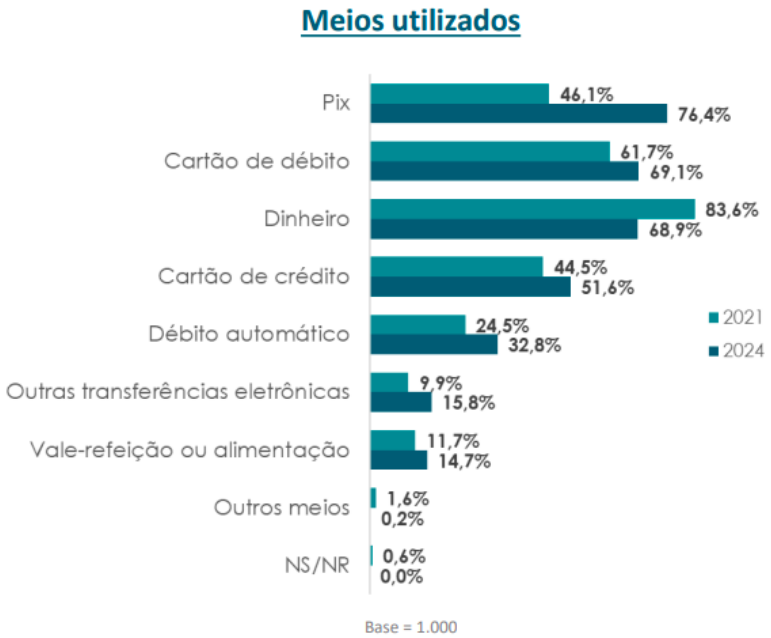


Figura 2: Métodos de pagamento utilizados pelos brasileiros (2021–2024).
Fonte: Banco Central do Brasil [3].

Assim, os comprovantes gerados por transações que passaram a ser realizadas via Pix somam-se àqueles que já eram produzidos por outras formas de pagamento digitalizadas, como cartão de crédito e débito, que também apresentaram crescimento entre 2021 e 2024, conforme também observado na Figura 2. Desse modo, o aumento da produção de comprovantes de transações financeiras não decorre apenas da consolidação do Pix, mas de sua combinação com meios de pagamento bancarizados que já integravam a rotina de consumo do brasileiro e que também vêm se expandindo, reforçando a tendência de acúmulo desses documentos.

Esse volume crescente, no entanto, não representa apenas uma questão de quantidade, pois cada um desses documentos pode precisar ser mantido e recuperado por diferentes períodos. Órgãos de defesa do consumidor, como o Procon, estabelecem orientações formais quanto aos prazos de guarda desses documentos, os quais podem se estender por períodos prolongados e variar de acordo com sua natureza e com exigências legais específicas [4]. Essa diversidade de prazos impõe ao consumidor a necessidade de manter comprovantes por intervalos distintos, contribuindo para o acúmulo progressivo e para o aumento da complexidade associada à sua gestão.

Entretanto, no uso cotidiano, a gestão desses documentos tende a ocorrer de maneira desestruturada e descentralizada, podendo seguir dois caminhos recorrentes quando de origem física. Em alguns casos, eles são simplesmente acumulados em carteiras, gavetas e pastas, tornando-se suscetíveis à perda, deterioração física, apagamento da tinta e extravio. Em outros casos, são digitalizados com a câmera do *smartphone* pessoal sem qualquer sistematização, permanecendo posteriormente dispersos na galeria de imagens do próprio dispositivo.

Mesmo quando de origem digital, esses documentos tendem a permanecer distribuídos entre extratos de aplicativos bancários, conversas em aplicativos de mensagens, e-mails ou serviços de armazenamento em nuvem não especializados. Além disso, quando esses documentos ficam restritos ao dispositivo sem uma política de cópia de segurança, a troca, perda ou falha do aparelho pode comprometer a preservação dos comprovantes.

Em todos os cenários, quando necessária, a recuperação desses documentos tende a ocorrer de forma onerosa e demorada, em razão da dispersão dos ambientes em que são mantidos e da ausência de mecanismos que facilitem sua organização e localização. Como consequência, o usuário frequentemente precisa realizar buscas manuais em múltiplos locais físicos e diretórios digitais, analisando individualmente cada documento até localizar o desejado ou uma informação específica em seu conteúdo.

Soma-se à dispersão uma limitação igualmente significativa na forma como esses comprovantes são estruturados nas práticas de gestão comumente adotadas, descritas anteriormente. Em geral, são armazenados fisicamente, com baixo ou nenhum nível de sistematização, ou mantidos como arquivos brutos — como imagens ou documentos digitais no formato Portable Document Format (PDF) — sem mecanismos de indexação semântica ou extração estruturada de dados.

Como resultado, informações relevantes — como valores, datas, estabelecimentos, categorias ou natureza da transação — não podem ser facilmente consultadas, agregadas ou analisadas, tampouco utilizadas como critérios de busca para localizar um documento especí-

fico de maneira rápida e eficiente. Adicionalmente, a ausência desses mecanismos também impede a realização de buscas contextuais, nas quais o indivíduo poderia localizar um comprovante com base no significado ou na finalidade associada a ele, ainda que os termos exatos presentes em seu conteúdo não sejam prontamente lembrados.

Essas limitações tornam-se particularmente relevantes no contexto da rotina cotidiana, marcada por múltiplas atividades e pela necessidade frequente de recuperar informações de maneira rápida, na qual é comum que o indivíduo se recorde do evento, da situação ou do propósito da transação, mas não dos detalhes formais registrados no comprovante, como o nome do estabelecimento, a data exata ou a descrição textual utilizada.

O uso de serviços genéricos de nuvem ou de processamento remoto também pode aumentar a exposição de dados financeiros pessoais, especialmente quando o usuário não tem clareza sobre onde os documentos são armazenados ou processados. Assim, além da organização e da recuperação dos comprovantes, a privacidade dos dados envolvidos também constitui uma dimensão relevante do problema.

Diante desse cenário, marcado pela crescente geração de comprovantes associados às atividades cotidianas e pela necessidade de mantê-los organizados e acessíveis para fins como controle de gastos, reembolso, cumprimento de obrigações fiscais e resolução de disputas comerciais, torna-se evidente a carência de mecanismos mais eficientes para sua gestão. A ausência de soluções especializadas para a gestão pessoal de comprovantes contribui para a manutenção das práticas desestruturadas e descentralizadas que dificultam sua recuperação quando necessário.

Nesse contexto, o presente trabalho justifica-se pela necessidade de propor uma solução de software orientada a suprir as lacunas identificadas nesse domínio, promovendo uma abordagem mais eficiente para o registro, organização e recuperação desses comprovantes de transações financeiras no cotidiano, com baixo esforço de interação e priorização do processamento no próprio dispositivo.

1.2 Proposta de solução de software

Este trabalho tem como objetivo geral o desenvolvimento do FinTrack, um aplicativo móvel Android concebido como um repositório inteligente de comprovantes de transações financeiras voltado ao uso no dia a dia. A solução se propõe a reduzir o esforço manual associado ao gerenciamento desses documentos, mitigando a sua dispersão e viabilizando sua organização estruturada e recuperação eficiente ao longo do tempo. Para alcançar esse objetivo geral, foram definidos os seguintes objetivos específicos:

1. identificar e analisar as práticas atuais de gestão de comprovantes de transações financeiras e as limitações das soluções existentes em termos de organização, persistência e recuperação;

2. especificar os requisitos funcionais e não funcionais da solução, considerando aspectos de usabilidade, privacidade, segurança e desempenho;

3. projetar a arquitetura do sistema, incluindo os mecanismos de escaneamento assistido de documentos, captura simples pela câmera, seleção de arquivos e importação de arquivos de comprovantes nativamente digitais via compartilhamento entre aplicativos, com minimização do esforço de interação do usuário;

4. implementar o processamento automatizado dos registros de comprovantes, incluindo extração textual com técnicas de OCR, enriquecimento de dados quando aplicável e estruturação das informações extraídas para viabilizar sua organização;

5. implementar mecanismos de busca contextual baseados em técnicas de PLN e similaridade semântica, permitindo a localização de registros de comprovantes a partir do significado ou da finalidade da transação, independentemente dos termos exatos presentes no documento;

6. implementar mecanismos opcionais de proteção de acesso local, como código numérico local, do inglês *Personal Identification Number* (PIN), biometria e bloqueio automático, alinhados aos princípios de privacidade e controle dos dados pelo usuário;

7. implementar recursos opcionais de *backup* criptografado e restauração em nuvem, permitindo a preservação e recuperação dos registros, dados extraídos, configurações relevantes e arquivos associados sob controle do usuário;

8. avaliar a viabilidade técnica da solução por meio de testes de software, discutindo limitações e perspectivas de evolução futura.

A aplicação é concebida sob uma abordagem *mobile-first*, alinhada à natureza cotidiana e dispersa das situações em que comprovantes são gerados — no comércio, em serviços, em transações instantâneas — e à ampla difusão de dispositivos móveis no contexto brasileiro, onde 88,9% da população com 10 anos ou mais já possui *smartphone* para uso pessoal [5]. Complementarmente, a solução adota o paradigma *offline-first*, no qual os arquivos associados aos comprovantes e seus dados extraídos são armazenados e processados prioritariamente no dispositivo do usuário. A conectividade é utilizada apenas em funcionalidades justificadas, como o *backup* criptografado em nuvem sob controle do usuário e consultas automáticas de enriquecimento fiscal ou cadastral a serviços públicos quando houver dados identificáveis e acesso à rede disponível.

O *offline-first* é uma decisão estratégica que garante maior disponibilidade e resiliência em cenários de conectividade limitada, além de melhorar a experiência do usuário ao eliminar a dependência de rede para operações críticas, como o registro, a organização e a recuperação dos comprovantes. Dessa forma, a aplicação preserva a continuidade de uso em diferentes contextos operacionais, mantendo as integrações externas como recursos complementares e não obrigatórios para o uso regular da solução.

Além disso, o FinTrack fundamenta-se no paradigma de interação de baixo esforço, no qual o registro visual de comprovantes físicos — como notas fiscais ou recibos de pagamentos em dinheiro ou cartão — constitui o principal ponto de interação do usuário com a aplicação, permitindo sua digitalização de forma simples e imediata por meio da câmera do dispositivo

móvel, com apoio de recursos de digitalização de documentos integrados à aplicação. De forma complementar, o sistema permite a importação de arquivos de comprovantes nativamente digitais — como notas fiscais eletrônicas, comprovantes de Pix, boletos, transferências bancárias ou compras *online* — recebidos por meio da funcionalidade de compartilhamento disponibilizada pelos aplicativos de origem, como aplicativos bancários, serviços de e-mail e gerenciadores de arquivos dos sistemas operacionais em geral.

Essa abordagem permite centralizar, de forma prática, esses documentos previamente dispersos, ao mesmo tempo em que reduz o esforço cognitivo e operacional associado ao seu registro. A utilização da solução torna-se, assim, tão simples quanto capturar uma imagem ou compartilhar um arquivo a partir de outro aplicativo, reaproveitando padrões de interação já consolidados no uso cotidiano de dispositivos móveis.

Para superar as limitações decorrentes da ausência de estrutura semântica no gerenciamento informal desses comprovantes, a aplicação realiza a extração, interpretação e organização assistida das informações neles contidas por meio de OCR e PLN. Os dados processados são apresentados ao usuário em uma prévia revisável antes da persistência definitiva do registro, permitindo confirmação, correção ou descarte das informações extraídas, inclusive em fluxos de importação em lote. Ao consolidar essas informações em um modelo estruturado, a solução viabiliza não apenas seu armazenamento organizado, mas também sua recuperação contextualizada e a agregação de dados, possibilitando a geração de relatórios financeiros e o acompanhamento financeiro ao longo do tempo.

No entanto, apesar do potencial de simplificação do processo de gestão desses comprovantes no cotidiano, a sua centralização em um ambiente ainda não familiar ao usuário pode suscitar preocupações relevantes relacionadas à privacidade, à segurança e ao controle dos dados. Tais preocupações podem reforçar a permanência das práticas informais e descentralizadas já destacadas, representando, portanto, uma barreira significativa à adoção da solução proposta.

Nesse sentido, essas preocupações nortearam as decisões arquiteturais do FinTrack, as quais privilegiam o armazenamento e o processamento local (*on-device*) das informações diretamente no dispositivo do usuário — decisão que, além de garantir o funcionamento *offline* da solução, promove maior autonomia informacional e reduz riscos associados à exposição de dados sensíveis. Nessa mesma direção, o aplicativo não exige cadastro, conta externa ou autenticação prévia para as funcionalidades locais essenciais, dispensando o vínculo obrigatório com serviços externos e eliminando uma barreira de entrada desnecessária à sua finalidade. O acesso à aplicação e aos registros armazenados é protegido por mecanismos locais de segurança, como PIN ou biometria, e o uso de serviços de armazenamento em nuvem permanece opcional e sob controle do próprio usuário, sendo empregado exclusivamente como mecanismo de *backup* e recuperação dos dados.

Adicionalmente, o FinTrack adota uma abordagem *open source* como mecanismo de transparência e auditabilidade, permitindo que seu funcionamento possa ser inspecionado, validado e compreendido por seus usuários e pela comunidade. Essa característica contribui para o fortalecimento da confiança na utilização da aplicação e reforça o compromisso da

solução com a autonomia e a privacidade dos dados do usuário.

O público-alvo da solução é composto por pessoas físicas no contexto brasileiro que realizam transações financeiras recorrentes em seu cotidiano, acumulam comprovantes físicos digitalizados ou arquivos de comprovantes nativamente digitais e precisam recuperá-los posteriormente para fins pessoais, como controle de gastos, comprovação de despesas, garantias, reembolsos ou resolução de eventuais divergências comerciais. Dentro desse recorte, a solução se direciona a usuários de *smartphones* Android compatíveis, com familiaridade básica no uso de aplicativos móveis, capazes de capturar imagens de documentos ou importar arquivos, conceder permissões necessárias e revisar informações extraídas antes da persistência definitiva do registro.

Em termos de faixa etária, o foco principal recai sobre adultos com autonomia para gerenciar suas próprias compras, pagamentos e transferências, sem excluir usuários jovens que já realizam operações financeiras simples e acumulam notas fiscais, recibos ou arquivos de comprovantes nativamente digitais. O recorte proposto, portanto, não se baseia em renda, classe social ou meio de pagamento utilizado, mas na necessidade prática de registrar e recuperar comprovantes de maneira ágil.

Por essa mesma delimitação, não fazem parte do escopo principal organizações, departamentos financeiros, escritórios contábeis, sistemas corporativos de reembolso ou transações sem qualquer comprovante registrável, pois esses cenários exigiriam recursos adicionais de multiusuário, auditoria, integração contábil e governança documental.

1.3 Tecnologias da solução

As tecnologias e ferramentas adotadas para o desenvolvimento da solução foram selecionadas considerando não apenas os requisitos funcionais e não funcionais estabelecidos para o sistema, mas, sobretudo, de modo a viabilizar com maior facilidade a implementação das decisões estratégicas e arquiteturais previamente apresentadas. Assim, esta seção elenca essas tecnologias e ferramentas, bem como as respectivas justificativas que fundamentam sua escolha no contexto do FinTrack.

1.3.1 Frontend

A tecnologia a seguir é responsável pela construção da interface do usuário (UI) e pela experiência de interação do usuário (UX) com o FinTrack:

Flutter

O Flutter é um kit de desenvolvimento de software, do inglês *Software Development Kit* (SDK), de código aberto desenvolvido pela Google para a criação de aplicações móveis multiplataforma a partir de uma única base de código, utilizando a linguagem de programação Dart [6]. Embora o escopo deste trabalho seja Android, sua escolha justifica-se por oferecer alto desempenho no desenvolvimento e suporte robusto à integração com recursos nativos do dispositivo, como digitalização de documentos, captura de imagem, armazenamento local, autenticação local e processamento *on-device*, recursos-chave para a viabilidade do FinTrack.

Além disso, diferentemente de alternativas baseadas em camadas intermediárias de comunicação entre código interpretado e componentes nativos, o Flutter compila diretamente para código nativo, reduzindo *overhead* de execução e resultando em maior fluidez e responsividade da interface.

1.3.2 Integrações nativas Android

As tecnologias a seguir complementam a camada Flutter quando a aplicação precisa acionar diretamente recursos do sistema operacional Android:

Kotlin, Android SDK e MethodChannel

Embora a maior parte da aplicação seja desenvolvida em Flutter e Dart, alguns recursos do FinTrack dependem de integração direta com funcionalidades nativas do Android. Para isso, a solução utiliza código Kotlin em conjunto com o Android SDK e canais de plataforma do Flutter (*MethodChannel*), permitindo que a camada Dart acione funcionalidades do sistema operacional de forma controlada [7, 8]. Essa integração viabiliza fluxos como captura por câmera, seleção de arquivos, recebimento de arquivos compartilhados por outros aplicativos, compartilhamento e salvamento local de arquivos, autenticação por PIN e biometria, obtenção de informações do dispositivo e comunicação com tarefas nativas em segundo plano.

Essa camada nativa complementa o uso de pacotes Flutter quando há necessidade de maior controle sobre permissões, diretórios privados, interação com outros aplicativos e execução em segundo plano. No contexto do FinTrack, ela contribui para manter a experiência de uso integrada ao Android sem deslocar a regra de negócio principal para fora da aplicação Flutter.

1.3.3 Banco de dados

A persistência local dos dados do FinTrack é realizada pelas tecnologias descritas a seguir:

SQLite

O SQLite é um sistema gerenciador de banco de dados relacional leve e amplamente utilizado em aplicações móveis e sistemas embarcados [9]. Sua escolha fundamenta-se na capacidade de operar integralmente no dispositivo do usuário como parte integrante da aplicação, apresentando baixo consumo de recursos, alta confiabilidade e simplicidade operacional, sem a necessidade de configuração de servidores ou dependência de infraestrutura externa.

No contexto da solução proposta, o SQLite atua como o principal mecanismo de persistência, sendo responsável pelo armazenamento dos dados estruturados dos registros de comprovantes e demais informações geradas pela aplicação. Essa característica é fundamental para a implementação da abordagem *offline-first*, permitindo que todas as funcionalidades essenciais do sistema permaneçam disponíveis independentemente da conectividade com a internet.

Além disso, sua natureza transacional e o suporte a consultas na linguagem Structured Query Language (SQL) contribuem para a integridade e a consistência dos dados, aspectos

essenciais para o registro e a recuperação segura dos comprovantes. A adoção do SQLite também alinha-se às diretrizes arquiteturais de segurança e privacidade do FinTrack, que priorizam a autonomia do usuário sobre seus próprios dados, uma vez que todas as informações permanecem armazenadas diretamente no dispositivo.

Drift

O Drift é uma biblioteca de persistência para Flutter que atua como uma camada de abstração sobre o SQLite, permitindo a definição de esquemas de tabelas e consultas de forma tipada e segura em Dart [10]. Sua adoção no projeto reduz a necessidade de SQL manual nas operações comuns de persistência, mantendo o uso de comandos SQL customizados para estruturas especializadas, como índices, busca textual e armazenamento vetorial. Essa combinação reduz a incidência de erros em tempo de execução e facilita a manutenção do código ao longo do desenvolvimento.

Além da tipagem estática, o Drift oferece suporte a consultas reativas por meio de *streams*, onde qualquer alteração persistida no banco é automaticamente propagada para os componentes da interface que observam aquela consulta, sem necessidade de consultas explícitas adicionais. Essa característica é essencial para garantir uma experiência fluida ao usuário, pois operações de leitura e escrita ocorrem de forma assíncrona sem bloquear a interface. O Drift também gera automaticamente o código de mapeamento entre as tabelas do SQLite e as entidades Dart, eliminando a camada manual de serialização e desserialização de dados.

Busca textual e vetorial no SQLite

Além das tabelas relacionais convencionais, o FinTrack utiliza recursos especializados do ecossistema SQLite para viabilizar a recuperação eficiente dos registros de comprovantes. O Full-Text Search 5 (FTS5) é empregado para busca textual local sobre conteúdo extraído e dados estruturados [11], enquanto a extensão *sqlite-vector* permite persistir e consultar vetores de *embedding* associados aos registros [12]. Essa combinação sustenta a busca híbrida da aplicação, unindo correspondência textual e similaridade semântica diretamente no banco local.

1.3.4 Processamento de dados

Nesta seção concentram-se as tecnologias utilizadas na extração e no processamento automatizado dos arquivos e imagens de comprovantes:

Google ML Kit

O Google ML Kit é um conjunto de ferramentas de aprendizado de máquina desenvolvido pela Google e utilizado no FinTrack para apoiar a captura e a extração local de informações de arquivos e imagens de comprovantes [13]. Na solução, a aplicação utiliza o Document Scanner para digitalização assistida de comprovantes físicos, o Text Recognition para reconhecimento textual nas imagens processadas e o Barcode Scanning para leitura de códigos visuais, especialmente códigos de resposta rápida (QR Codes) presentes em documentos fiscais.

Sua escolha fundamenta-se na capacidade de execução *on-device*, permitindo que etapas sensíveis do fluxo de captura e interpretação ocorram de forma rápida e eficiente mesmo

offline. Diferentemente de soluções baseadas em interfaces externas de programação de aplicações, do inglês *Application Programming Interfaces* (APIs), que exigem envio de imagens ou textos para servidores remotos e dependem de conectividade com a internet, o ML Kit realiza essas operações diretamente no dispositivo. Com isso, reduz-se a latência percebida pelo usuário, preserva-se a privacidade dos dados financeiros e mantém-se a coerência da estratégia *offline-first* adotada pelo sistema.

DistilUSE Multilingual e execução local de modelos

O FinTrack utiliza o modelo `distiluse-base-multilingual-cased-v2-qint8` como componente de PLN para a geração de *embeddings* textuais, convertendo textos extraídos dos registros de comprovantes e consultas digitadas pelo usuário em vetores numéricos comparáveis semanticamente [14]. A execução ocorre localmente por meio do ONNX Runtime — ambiente de execução para modelos de aprendizado de máquina no formato Open Neural Network Exchange (ONNX) —, a partir de artefatos embarcados na própria aplicação, incluindo o modelo quantizado, vocabulário e arquivos auxiliares de tokenização [15]. Essa escolha preserva a premissa *on-device* da solução, evita o envio de conteúdo financeiro para APIs externas e viabiliza buscas por intenção em língua portuguesa, permitindo relacionar termos semanticamente próximos mesmo quando a consulta não reproduz literalmente o texto presente no comprovante.

Pacote image

A biblioteca `image` é utilizada nas etapas de pré-processamento das imagens de comprovantes, permitindo gerar variantes temporárias com ajustes de qualidade antes da execução do OCR [16]. Esse processamento auxilia a melhoria da legibilidade para reconhecimento textual e permanece restrito ao dispositivo, em alinhamento com a estratégia *offline-first* da aplicação.

1.3.5 Inteligência artificial

Além dos modelos embarcados utilizados pela própria aplicação, ferramentas de inteligência artificial também foram empregadas como apoio ao processo de desenvolvimento:

OpenAI Codex

O OpenAI Codex é um agente de codificação voltado a tarefas de desenvolvimento de software, incluindo escrita, compreensão, revisão, depuração e automação de atividades de engenharia [17]. No contexto deste trabalho, seu uso ocorreu como ferramenta de apoio ao processo de implementação, refatoração, análise de consistência e revisão textual da aplicação e da documentação.

Esse apoio teve caráter auxiliar. As decisões de projeto, a definição dos requisitos, a validação da solução e a responsabilidade sobre o código e sobre o texto permaneceram vinculadas à condução deste trabalho. Assim, o Codex foi utilizado como copiloto de desenvolvimento, contribuindo para acelerar tarefas operacionais e ampliar a capacidade de inspeção sobre o código e os artefatos produzidos.

1.3.6 Segurança e criptografia

A proteção local de dados sensíveis e dos arquivos de *backup* é apoiada pelas tecnologias descritas a seguir:

Flutter Secure Storage e Android Keystore

O FinTrack utiliza o Flutter Secure Storage para armazenar segredos sensíveis, como a senha de *backup*, com apoio dos mecanismos seguros fornecidos pelo sistema operacional Android, incluindo o Android Keystore [18, 19]. Essa escolha evita a persistência de segredos em texto claro no banco de dados local e reforça a separação entre dados operacionais e credenciais ou chaves utilizadas em fluxos de segurança.

Pacote cryptography e algoritmos criptográficos

A criptografia dos arquivos de *backup* é implementada com a biblioteca `cryptography`, utilizando Advanced Encryption Standard em modo Galois/Counter Mode com chave de 256 bits (AES-GCM-256) para cifrar os dados e Password-Based Key Derivation Function 2 com HMAC-SHA256 (PBKDF2-HMAC-SHA256) para derivar a chave a partir da senha informada pelo usuário [20, 21]. Na implementação do FinTrack, a derivação utiliza *salt* criptográfico aleatório e 310.000 iterações.

O *salt* consiste em um valor aleatório associado ao processo de derivação da chave, impedindo que senhas iguais produzam necessariamente a mesma chave derivada e dificultando ataques baseados em tabelas pré-computadas [21]. A combinação com múltiplas iterações aumenta o custo computacional de ataques de força bruta contra arquivos de *backup* obtidos indevidamente.

1.3.7 Armazenamento em nuvem

A integração opcional com serviço de armazenamento em nuvem de terceiros no FinTrack é implementada a partir da ferramenta a seguir:

Google Sign-In/OAuth 2.0 e Google Drive API

O Google Drive é um serviço de armazenamento em nuvem amplamente utilizado para a guarda e sincronização de arquivos [22], sendo adotado na solução como mecanismo opcional de *backup* e recuperação dos registros de comprovantes, dados extraídos, configurações relevantes e arquivos associados previamente persistidos no dispositivo do usuário. O vínculo com a conta Google é realizado por meio do Google Sign-In e do fluxo OAuth 2.0, solicitando autorização específica para uso do espaço privado da aplicação no Drive [23]. Sua escolha fundamenta-se na não criação de uma infraestrutura própria de armazenamento, mas na qualificação do uso de uma infraestrutura de terceiro já familiar, ao passo que reduz custos e simplifica a manutenção do sistema.

No entanto, arquivos armazenados em serviços de nuvem de terceiros permanecem sujeitos às políticas e aos mecanismos técnicos do provedor utilizado. Portanto, no FinTrack, os arquivos de *backup* são criptografados localmente com AES-GCM-256, com chave derivada

por PBKDF2-HMAC-SHA256, antes de serem enviados. Desse modo, o que o Google Drive recebe são apenas dados cifrados e opacos, conferindo ao usuário uma camada adicional de privacidade, visto que, mesmo que os arquivos sejam acessados, o conteúdo dos registros, dados extraídos e arquivos associados permanecerá ilegível sem a chave de descriptografia, que nunca abandona o dispositivo.

A integração entre a aplicação e o serviço é realizada por meio da Google Drive API, que fornece interfaces padronizadas para autenticação, envio e recuperação de arquivos, permitindo que as operações de *backup* e restauração sejam executadas de forma automatizada e segura, respeitando as permissões definidas pelo próprio usuário [24].

1.3.8 Versionamento e DevOps

O controle da evolução do código-fonte e a automação dos processos de verificação de qualidade ao longo do desenvolvimento são suportados pelas ferramentas descritas a seguir:

Git

O Git é um sistema de controle de versão distribuído, de código aberto, licenciado sob a GNU General Public License v2.0 [25]. Desenvolvido originalmente para suportar o kernel Linux, foi concebido desde o início com foco em desempenho e eficiência, tendo se consolidado como padrão no desenvolvimento de software — 96% dos desenvolvedores profissionais o utilizam, segundo a Stack Overflow Developer Survey de 2022 [26].

No contexto do FinTrack, o Git é adotado como ferramenta padrão para controle de versão, oferecendo rastreabilidade das alterações e maior segurança na evolução do código ao longo do tempo. Seus mecanismos de ramificação (*branching*) e integração (*merging*) permitem o gerenciamento estruturado de diferentes funcionalidades e versões do sistema, contribuindo para a organização do desenvolvimento e para a redução de riscos associados a alterações no código-fonte.

Git Large File Storage

O Git Large File Storage (Git LFS) é uma extensão do Git voltada ao versionamento de arquivos grandes, substituindo esses arquivos no repositório por ponteiros textuais enquanto mantém seu conteúdo em armazenamento externo associado ao repositório [27]. No FinTrack, sua adoção é relevante em razão dos artefatos binários embarcados para execução local de PLN, especialmente o modelo ONNX utilizado na geração de *embeddings*. Dessa forma, o projeto preserva a rastreabilidade desses artefatos sem tornar o histórico Git excessivamente pesado para o versionamento do código-fonte e da documentação textual.

GitHub

O GitHub é uma plataforma de hospedagem de código baseada no Git, onde é possível armazenar, compartilhar e colaborar na escrita do código-fonte de uma aplicação [28]. Apesar da sua popularidade e adoção consolidada, sua escolha no FinTrack justifica-se, sobretudo, pelas funcionalidades oferecidas, como gerenciamento de tarefas, integração contínua (CI), automação de verificações e geração de artefatos. Além disso, sua ampla adoção e familiaridade

no meio acadêmico e profissional reduzem a curva de aprendizado e facilitam sua utilização no contexto da solução proposta.

GitHub Actions

Como extensão das funcionalidades oferecidas pelo GitHub, o GitHub Actions fornece recursos para simplificar a configuração e a manutenção de *pipelines* de integração contínua e automação da compilação, permitindo a execução automática de tarefas — como compilação, testes e geração de relatórios — quando ocorrem alterações relevantes no repositório do projeto [29]. No FinTrack, o fluxo automatizado é acionado em *commits* e *pull requests* que modificam o diretório da aplicação Flutter, executando formatação, análise estática, testes automatizados com cobertura e a compilação da solução, além de disponibilizar os resultados relevantes como artefatos da execução.

1.3.9 Modelagem e documentação

Esta seção apresenta a tecnologia utilizada para representar e comunicar a estrutura e o comportamento do sistema ao longo do seu ciclo de desenvolvimento:

PlantUML

O PlantUML é uma ferramenta para criação de diagramas a partir de uma linguagem textual, permitindo a definição de elementos arquiteturais e de modelagem de forma estruturada e versionável [30]. Sua adoção no projeto possibilita a integração da documentação ao código-fonte, permitindo o versionamento dos diagramas junto ao repositório do projeto, garantindo maior consistência, assegurando rastreabilidade das alterações e facilitando a manutenção da documentação conforme a evolução do sistema.

1.3.10 Testes

A verificação da corretude e da robustez da aplicação é suportada pelo conjunto de ferramentas descrito a seguir, adotado em conjunto com a abordagem de desenvolvimento orientado a testes detalhada na Seção 4.1.

Flutter Test

O Flutter Test é o conjunto de ferramentas nativas do Flutter para implementação de testes automatizados no *frontend*. Ele oferece suporte à criação de testes unitários, focados na lógica isolada, e de testes de *widgets*, voltados à consistência e à interação correta dos componentes visuais [31]. No FinTrack, sua adoção concentra-se nesses dois níveis automatizados, complementados por validações manuais dos fluxos integrados que dependem de recursos nativos do Android, serviços externos ou execução em dispositivos físicos reais. Dessa forma, a ferramenta contribui para verificar de maneira recorrente regras de negócio e comportamentos de interface, enquanto os cenários sistêmicos são avaliados por meio de validações práticas em dispositivos Android.

LCOV

O LCOV é um formato e conjunto de ferramentas utilizado para registrar e apresentar informações de cobertura de código obtidas durante a execução de testes automatizados [32]. No FinTrack, os relatórios em formato LCOV são gerados a partir da execução do Flutter Test com cobertura, permitindo acompanhar quais trechos da aplicação são exercitados pela suíte unitária e de *widgets*. Na integração contínua, o arquivo `coverage/lcov.info` também é convertido em relatório HTML, sendo disponibilizado como artefato do fluxo automatizado. Essa informação apoia a identificação de áreas pouco cobertas e complementa a estratégia de verificação automatizada adotada no projeto.

Mockito

O Mockito é uma biblioteca de testes utilizada para criação de dublês de teste (*mocks*) na linguagem Dart, permitindo simular o comportamento de dependências externas durante a execução de testes automatizados [33]. Sua adoção no projeto fundamenta-se na necessidade de isolar unidades de software — especialmente serviços e repositórios — de componentes cuja execução real envolveria acesso a recursos externos, como banco de dados, digitalização de documentos, captura de imagem, sistema de arquivos, OCR, consultas automáticas de enriquecimento fiscal ou cadastral e serviços de processamento de dados.

Diferentemente da execução direta de dependências reais, que pode introduzir variabilidade e aumentar o custo computacional dos testes, o uso de *mocks* permite validar apenas a lógica da unidade em teste, garantindo maior previsibilidade e confiabilidade na verificação do comportamento do sistema. Como consequência, a utilização do Mockito contribui para a melhoria da qualidade do código, permitindo a construção de testes mais rápidos, isolados e reprodutíveis, reforçando a confiabilidade do processo de desenvolvimento orientado a testes adotado no projeto.

1.4 Trabalhos relacionados

Atualmente, é possível identificar diferentes categorias de aplicação com propósitos específicos no domínio da gestão de comprovantes de transações financeiras, como ferramentas de digitalização de documentos, plataformas de armazenamento em nuvem, sistemas de gestão de despesas corporativas e aplicativos de controle financeiro pessoal. Cada uma dessas categorias atende a demandas particulares relacionadas à prestação de contas organizacional, à geração de métricas financeiras, ao escaneamento genérico de documentos ou ao armazenamento de arquivos digitais.

No entanto, embora desempenhem papéis relevantes em seus respectivos contextos dentro do domínio de interesse do FinTrack, essas soluções não foram projetadas com foco específico na gestão de comprovantes de transações financeiras no uso pessoal cotidiano. Diante desse cenário, esta seção apresenta soluções de software representativas dessas diferentes categorias, analisando suas funcionalidades e limitações com o objetivo de evidenciar lacunas funcionais relacionadas à gestão de comprovantes de transações financeiras no dia a dia.

1.4.1 *CamScanner*

O CamScanner é um aplicativo móvel, disponível também para iOS [34], voltado à digitalização de documentos físicos por meio da câmera do próprio dispositivo [35], que tem como principal objetivo a conversão rápida de conteúdos em formato digital. A aplicação permite capturar imagens, aplicar correções automáticas, converter arquivos em formato PDF e armazená-los localmente ou em nuvem, oferecendo suporte ao gerenciamento básico de documentos em geral.

Entretanto, embora ofereça recursos de gerenciamento de arquivos, o CamScanner não realiza busca contextual voltada a comprovantes financeiros, por exemplo, com base na inferência do tipo de despesa, nem oferece organização financeira automática baseada em valor, data, estabelecimento, natureza da transação e categoria. Além disso, quando o usuário opta pela sincronização remota, a infraestrutura de nuvem utilizada é própria da plataforma, o que reduz o controle direto sobre o armazenamento de suas informações e arquivos digitalizados.

1.4.2 *Google Drive*

O Google Drive, conforme descrito anteriormente, é um serviço voltado ao armazenamento e à sincronização de arquivos digitais em nuvem da Google, cuja aplicação possui versão para web e dispositivos móveis, incluindo iOS [36], sendo amplamente utilizado como um repositório generalista para guarda e compartilhamento de arquivos. A Conta Google oferece 15 GB de armazenamento, compartilhados entre Google Drive, Gmail e Google Photos [37]. Além disso, o Drive permite organizar manualmente arquivos em diretórios, acessá-los a partir de diferentes dispositivos conectados à internet, digitalizar documentos pelo aplicativo móvel e salvá-los como arquivos pesquisáveis [38]. Assim como o CamScanner, também permite realizar buscas com base em termos presentes no nome ou no conteúdo textual dos documentos.

Todavia, sua digitalização é generalista e não especializada no fluxo de gestão de comprovantes financeiros. O Google Drive não dispõe de prévia revisável de dados extraídos, categorização financeira, relatórios baseados nos comprovantes ou busca semântica local orientada ao significado da transação. Além disso, o acesso pleno às funcionalidades depende de conectividade com a internet, o que limita o uso da solução em situações cotidianas em que o acesso à rede é restrito.

1.4.3 *Expensify*

O Expensify é uma aplicação, com versão para web e para dispositivos móveis, incluindo iOS e Android, voltada à gestão de despesas corporativas e processos de reembolso financeiro [39, 40], sendo amplamente utilizada em ambientes empresariais para o registro e o controle de gastos. A aplicação permite registrar comprovantes, escanear recibos com o SmartScan, encaminhar recibos por e-mail para leitura automática, organizar despesas em relatórios financeiros e exportar os dados associados às transações [41]. Além disso, oferece suporte ao acompanhamento de despesas em contextos organizacionais e a funcionalidades de captura e registro de despesas mesmo quando a sincronização com a plataforma não ocorre imediatamente.

Entretanto, sua proposta central está associada ao fluxo de gerenciamento de gastos, reembolso, sincronização com plataforma em nuvem, equipes, políticas corporativas e integrações contábeis, e não à manutenção de um repositório pessoal *offline-first* de comprovantes sob controle local do usuário. Assim, embora possua recursos robustos de captura e automação de despesas, o comprovante permanece subordinado ao registro de despesa e ao ecossistema da plataforma, em vez de constituir o objeto central de uma organização pessoal local, revisável e orientada à recuperação contextual.

1.4.4 *Mobills*

O Mobills é uma aplicação voltada à gestão financeira pessoal, com versões para dispositivos móveis e web, e recursos para o registro e o acompanhamento de receitas e despesas no cotidiano do usuário [42, 43]. Ele permite categorizar gastos, estabelecer orçamentos mensais e visualizar relatórios financeiros, oferecendo recursos que auxiliam no planejamento e no controle das finanças individuais.

Contudo, embora o Mobills permita o acesso às informações financeiras em modo *offline*, o registro de comprovantes depende de conexão com a rede. Além disso, esses documentos são tratados como anexos de um registro de despesa ou receita [44], de modo que a aplicação não disponibiliza mecanismos específicos para a busca direta desses comprovantes.

1.4.5 Comparação das ferramentas

Embora as soluções apresentadas nas subseções anteriores ofereçam recursos úteis ao gerenciamento de documentos em geral, elas possuem lacunas que comprometem sua efetividade para a resolução do problema abordado neste trabalho, uma vez que não foram projetadas especificamente para facilitar a gestão de comprovantes de transações financeiras no dia a dia.

O CamScanner pode ser utilizado para digitalizar comprovantes físicos com eficiência, mas não disponibiliza uma estrutura de organização financeira automática nem realiza busca contextual voltada ao significado da transação. O Google Drive pode ser utilizado para a preservação em nuvem de arquivos de comprovantes digitalizados ou nativamente digitais, além de possuir digitalização básica de documentos pelo aplicativo móvel; porém, não oferece fluxo especializado de revisão de dados extraídos, categorização financeira, relatórios ou busca semântica local para registros de comprovantes. Já o Expensify e o Mobills, embora voltados à gestão financeira, subordinam os comprovantes a registros de despesa ou a fluxos de reembolso e controle financeiro mais amplos, sem tratá-los como objeto central de um repositório pessoal *offline-first*.

De modo transversal, a recuperação de informações e comprovantes nessas soluções permaneceria dependente da sua organização manual ou da lembrança de detalhes formais do documento por parte do usuário — limitação diretamente oposta à abordagem de baixo esforço proposta pelo FinTrack. Igualmente, no que tange à autonomia sobre os dados, observa-se que essas ferramentas dependem, em diferentes graus, de contas, sincronização ou armazenamento em infraestruturas de terceiros para acesso, preservação ou ampliação do

uso dos documentos, os quais podem conter informações financeiras e pessoais sensíveis.

No que se refere ao design, nenhuma delas foi concebida para proporcionar agilidade no processo de gestão pessoal de comprovantes financeiros, isto é, registrar, organizar e recuperar com o mínimo de esforço e o menor número de etapas possível. Por terem sido projetadas para finalidades mais amplas, aplicações como o Expensify e o Mobills tendem a apresentar interfaces cujas funcionalidades relacionadas aos comprovantes estão dispersas ou subordinadas a outros fluxos.

Essa questão do design se reflete em aspectos concretos, como a disposição dos elementos na tela, a localização dos pontos de interação e a forma como as informações são apresentadas ao usuário. O FinTrack, por sua vez, é concebido desde o início sob a premissa de priorização da agilidade e da simplicidade no processo de gestão de comprovantes de transações financeiras, orientando cada decisão de interface para reduzir o esforço cognitivo e operacional associado ao registro, à organização e à recuperação desses documentos.

No que se refere à barreira de entrada para uso das soluções, todas as aplicações analisadas, exceto pelo CamScanner, exigem autenticação prévia por meio de *login* — seja por cadastro próprio na plataforma ou por acesso via conta Google —, o que representa uma fricção inicial ao uso independentemente da finalidade com que o usuário acessa a aplicação. Nas plataformas de gestão financeira, como Expensify e Mobills, essa barreira é ampliada: por tratarem os comprovantes como anexos de registros de despesa e não como objeto central do sistema, o processo de registro exige o preenchimento manual de informações adicionais — como categorias, valores, descrições ou dados institucionais — que vão além do documento em si. Esse conjunto de exigências impacta diretamente na agilidade do processo de gestão, que é precisamente o foco central da solução proposta neste trabalho.

Nesse sentido, o FinTrack adota uma abordagem distinta: por operar localmente no dispositivo do usuário e não depender de infraestrutura própria de armazenamento, não há necessidade de cadastro ou autenticação prévia para uso da aplicação. O acesso ao Google é solicitado exclusivamente nos casos em que o usuário opta por realizar *backup* em nuvem, mantendo essa etapa como uma decisão voluntária e pontual, dissociada do fluxo principal de uso. Como medida de segurança local, a aplicação adota mecanismos de proteção contra acesso inadvertido aos registros armazenados, sem que isso implique a criação de uma conta ou o vínculo obrigatório com serviços externos.

Por outro lado, quanto à abrangência de plataformas, todas as soluções analisadas possuem versões para iOS, enquanto algumas também oferecem acesso pela web. No escopo deste trabalho, o FinTrack é direcionado a dispositivos Android, o que delimita sua abrangência inicial em relação às alternativas analisadas. No entanto, essa restrição tem impacto parcialmente mitigado no contexto brasileiro, uma vez que o Android representa a maior parte da participação observada de sistemas operacionais móveis e tablets no país. Segundo dados da StatCounter Global Stats, em janeiro de 2026 o Android correspondia a 81,83% desse mercado no Brasil, enquanto o iOS representava apenas 17,94% [45].

Por fim, mas não menos importante, todas as soluções analisadas anteriormente são distribuídas como software proprietário, sem adotar uma abordagem *open source*, o que reduz

a transparência sobre o funcionamento interno dos sistemas e limita as possibilidades de auditoria independente — aspecto particularmente relevante no contexto do armazenamento de documentos dessa natureza. O FinTrack, por sua vez, é uma aplicação *open source*, o que permite a inspeção do código-fonte e a validação independente da solução proposta.

A Tabela 1 sintetiza as principais funcionalidades e características observadas nas soluções analisadas em comparação ao FinTrack, evidenciando as convergências e distinções relevantes para o contexto de gestão de comprovantes de transações financeiras no cotidiano do usuário.

Tabela 1: Comparação entre soluções existentes e o FinTrack quanto à gestão de comprovantes financeiros

Funcionalidade / Característica	**Cam Scanner**	**Google Drive**	**Expensify**	**Mobills**	**FinTrack**
Digitalização simplificada de comprovantes físicos	✓	✓	✓		✓
Recebimento/importação de arquivos	✓	✓	✓	✓	✓
Busca por termos em nome e conteúdo do arquivo	✓	✓			✓
Sincronização contínua entre dispositivos	✓	✓	✓	✓	
Enriquecimento fiscal/cadastral automático quando aplicável					✓
Foco em repositório pessoal *offline-first* de comprovantes financeiros					✓
Geração de relatórios financeiros			✓	✓	✓
Disponibilidade em iOS	✓	✓	✓	✓	
Busca contextual por significado da transação					✓
Fluxos corporativos de reembolso e prestação de contas			✓		
Armazenamento local como base principal dos registros e arquivos associados	✓				✓
Backup criptografado opcional sob controle do usuário					✓
Compartilhamento colaborativo ou multiusuário	✓	✓	✓		

Funcionalidade / Característica	Cam Scanner	Google Drive	Expensify	Mobills	FinTrack
Prévia revisável antes da persistência definitiva do registro					✓
Sincronização ou armazenamento remoto como base operacional	✓	✓	✓	✓	
Fluxo principal de gestão de comprovantes disponível *offline*	✓				✓
Autonomia local sobre os dados operacionais					✓
Abordagem *open source*					✓

Durante o levantamento de soluções disponíveis, também foram identificadas aplicações homônimas ou com denominações próximas a FinTrack na Google Play Store, como *FinTrack: Track, Plan, Prosper* [46], *FinTrack - Personal Finance* [47], *FinTrack* da Abzer Mobility Team [48] e *Fintrack: Expense Tracker* [49], em geral voltadas ao controle financeiro pessoal, ao registro de despesas ou ao acompanhamento de orçamento. Entretanto, essas soluções não foram selecionadas como trabalhos relacionados principais por apresentarem baixa adoção aparente, com faixas públicas de instalação reduzidas, e por não evidenciarem, em suas descrições, o mesmo foco da solução proposta na gestão de comprovantes de transações financeiras. Dessa forma, a comparação prioriza aplicações mais consolidadas ou mais representativas das categorias funcionais relacionadas ao problema tratado neste trabalho.

2 Requisitos

Esta seção apresenta os requisitos do FinTrack organizados em requisitos funcionais (RFs) e requisitos não funcionais (RNFs). Esses requisitos orientam as decisões de design, a definição dos casos de teste, o projeto de implantação e a validação das funcionalidades descritas no manual do usuário.

2.1 Requisitos funcionais

A Tabela 2 apresenta os requisitos funcionais do sistema, definidos para atender às lacunas identificadas nas soluções atuais de gerenciamento de registros de comprovantes de transações financeiras de uso pessoal. Cada requisito funcional é numerado para facilitar sua referência e acompanhamento ao longo do desenvolvimento do projeto.

Tabela 2: Requisitos funcionais do sistema FinTrack

Nº	Requisito Funcional
RF01	O sistema deve suportar a configuração de um mecanismo de autenticação local — como PIN ou biometria — para impedir o acesso inadvertido aos registros de comprovantes, arquivos associados e dados extraídos do usuário.
RF02	O sistema deve suportar o registro de arquivos ou imagens de comprovantes por meio do scanner de documentos do dispositivo móvel, mantendo também alternativas de captura de imagem e importação de arquivos quando aplicáveis.
RF03	O sistema deve suportar a importação individual ou em lote de arquivos de comprovante por meio do seletor de arquivos ou da funcionalidade de compartilhamento dos demais aplicativos, gerando registros revisáveis antes da persistência definitiva.
RF04	O sistema deve permitir a visualização dos registros de comprovantes armazenados na aplicação para acompanhamento das transações cadastradas.
RF05	O sistema deve extrair, enriquecer quando aplicável e organizar automaticamente as informações contidas nos arquivos ou imagens de comprovante registrados, a fim de reduzir a necessidade de preenchimento manual e viabilizar consulta, filtragem e busca posteriores.
RF06	O sistema deve permitir a ordenação dos registros de comprovantes por critérios como data, valor ou estabelecimento para facilitar sua localização.
RF07	O sistema deve permitir a filtragem dos registros de comprovantes por categorias, períodos, tipos de comprovante ou natureza financeira para restringir os resultados exibidos.
RF08	O sistema deve suportar a pesquisa de registros de comprovantes utilizando termos presentes no nome do arquivo associado, no conteúdo textual extraído ou nos campos estruturados, para localizar rapidamente um registro específico cujos detalhes formais o usuário recorde.
RF09	O sistema deve suportar a busca contextual de registros de comprovantes utilizando palavras relacionadas ao significado ou à finalidade da despesa — como “combustível”, “mercado” ou “consulta” — mesmo que esses termos exatos não constem no conteúdo do documento, para localizar registros a partir do contexto da transação.
RF10	O sistema deve permitir a visualização dos detalhes de um registro de comprovante para consultar as informações associadas ao arquivo ou imagem processada.
RF11	O sistema deve suportar a edição das informações associadas a um registro de comprovante para corrigir dados quando necessário.

Nº	Requisito Funcional
RF12	O sistema deve suportar a exclusão de um registro de comprovante armazenado no sistema, incluindo seus dados extraídos e arquivo associado, quando não for mais necessário.
RF13	O sistema deve permitir a exportação, o salvamento no dispositivo ou o compartilhamento do arquivo associado a um registro de comprovante, seja ele oriundo de um comprovante nativamente digital ou de um documento físico digitalizado, para reutilização quando necessário.
RF14	O sistema deve disponibilizar relatórios financeiros a partir dos registros de comprovantes armazenados para acompanhamento de receitas, despesas, saldo e agrupamentos ao longo do tempo.
RF15	O sistema deve permitir ao usuário vincular sua conta Google à aplicação para habilitar o uso do Google Drive como destino opcional de *backup* dos dados da aplicação.
RF16	O sistema deve permitir a execução manual e automática de *backup* criptografado dos registros de comprovantes, dados extraídos, configurações relevantes e arquivos associados no Google Drive, quando habilitado pelo usuário e quando houver conta Google vinculada, senha de *backup* configurada e disponibilidade da operação em nuvem, para reduzir o risco de perda dos dados em caso de troca ou perda do dispositivo.
RF17	O sistema deve permitir a restauração dos registros de comprovantes, dados extraídos, configurações relevantes e arquivos associados a partir do *backup* em nuvem em um novo dispositivo.
RF18	O sistema deve permitir ao usuário desvincular sua conta Google da aplicação e revogar o acesso concedido ao *backup* em nuvem a qualquer momento, a fim de manter controle sobre as informações e permissões concedidas à aplicação.
RF19	O sistema deve informar ao usuário, por meio da interface da aplicação e do histórico de operações, quando uma operação de *backup* ou restauração em nuvem não puder ser concluída, apresentando o estado da operação e, quando disponível, a descrição do erro ocorrido.
RF20	O sistema deve informar o usuário, pela interface da aplicação, quando o espaço local ocupado pelos arquivos associados aos registros atingir ou ultrapassar o limite de armazenamento configurado no FinTrack, a fim de possibilitar o gerenciamento dos dados e evitar o esgotamento do espaço reservado à aplicação.
RF21	O sistema deve permitir ao usuário ativar, desativar e configurar o intervalo do *backup* automático dos dados da aplicação em nuvem, quando houver conta Google vinculada e senha de *backup* definida.
RF22	O sistema deve apresentar uma prévia revisável dos dados extraídos antes da persistência definitiva do registro, permitindo ao usuário confirmar, corrigir ou descartar as informações processadas.

Nº	Requisito Funcional
RF23	O sistema deve permitir a revisão em lote de arquivos de comprovante importados simultaneamente, possibilitando salvar, corrigir, reprocessar ou descartar itens individualmente antes da persistência definitiva.
RF24	O sistema deve consultar e reaproveitar dados fiscais ou cadastrais, quando houver QR Code, chave de acesso ou Cadastro Nacional da Pessoa Jurídica (CNPJ) identificado, utilizando cache local e consultas remotas quando houver conectividade disponível, para enriquecer as informações extraídas dos registros de comprovantes.
RF25	O sistema deve permitir ao usuário criar, editar, excluir, visualizar e reordenar categorias, bem como associá-las aos registros de comprovantes para apoiar a organização e a geração de relatórios.
RF26	O sistema deve disponibilizar configurações de uso, incluindo tema visual, limite de armazenamento, tutorial inicial, senha de *backup* e preferências de autenticação local.
RF27	O sistema deve permitir ao usuário reportar problemas a partir da própria aplicação, reunindo informações diagnósticas necessárias para auxiliar a análise de falhas.
RF28	O sistema deve permitir ao usuário remover o *backup* armazenado em nuvem e limpar o histórico local de operações de *backup*, sem excluir os registros e arquivos associados mantidos no dispositivo.
RF29	O sistema deve permitir a seleção de múltiplos registros de comprovantes para executar ações em lote, como compartilhamento, salvamento no dispositivo e exclusão dos registros selecionados.
RF30	O sistema deve permitir a retomada de importações em lote pendentes quando houver interrupção do processamento, preservando os itens já processados para continuidade ou revisão posterior.
RF31	O sistema deve bloquear novamente o acesso à aplicação quando ela retornar do segundo plano após o intervalo de bloqueio automático configurado pelo usuário.

2.2 Requisitos não funcionais

A Tabela 3 apresenta os requisitos não funcionais, que estabelecem as restrições de qualidade, desempenho, segurança e portabilidade que o sistema deve atender, independentemente das funcionalidades implementadas. Cada requisito não funcional é numerado para facilitar sua referência e acompanhamento ao longo do desenvolvimento do projeto.

Tabela 3: Requisitos não funcionais do sistema FinTrack

Nº	Requisito Não Funcional
RNF01	O sistema deve priorizar o armazenamento e o processamento local dos registros de comprovantes, arquivos associados e dados extraídos, sem depender de servidores próprios ou de processamento remoto obrigatório. Transmissões a serviços externos devem ocorrer apenas em situações delimitadas e justificadas, como *backup* em nuvem autorizado pelo usuário ou consultas automáticas de enriquecimento fiscal ou cadastral a serviços públicos quando houver dados identificáveis e conectividade, com controle dos endereços consultados e persistência local dos resultados quando aplicável.
RNF02	O sistema deve utilizar comunicação segura por meio do protocolo Hypertext Transfer Protocol Secure (HTTPS) nas operações externas controladas pela aplicação, incluindo *backup*, restauração, autenticação com serviços Google e consultas cadastrais à BrasilAPI. Consultas fiscais derivadas de URLs presentes em QR Codes identificados em arquivos ou imagens de comprovante devem aceitar apenas endereços HTTP ou HTTPS associados a domínios fiscais reconhecidos ou governamentais.
RNF03	O sistema deve garantir que, quando a proteção de acesso local estiver habilitada, os dados armazenados localmente sejam acessíveis pela interface da aplicação apenas mediante autenticação configurada pelo usuário.
RNF04	O sistema deve apresentar tempo de resposta inferior a 2 segundos para a exibição da lista de registros de comprovantes armazenados localmente.
RNF05	O sistema deve concluir o processamento inicial de cada imagem de comprovante em até 10 segundos em dispositivo compatível com os requisitos mínimos de implantação, mantendo o usuário informado por meio de estado de progresso enquanto executa pré-processamento, OCR, extração estruturada e enriquecimentos locais ou oportunísticos. A interface deve adotar um tempo limite operacional superior ao tempo-alvo, permitindo tratar documentos mais custosos sem encerramento prematuro do fluxo.
RNF06	O sistema deve suportar o armazenamento de, no mínimo, 600 registros de comprovantes sem degradação perceptível de desempenho.
RNF07	O sistema deve ser compatível com dispositivos móveis que utilizem o sistema operacional Android na versão 8.0, correspondente ao nível de API 26, ou superior.
RNF08	O sistema deve apresentar interface responsiva, adaptada a diferentes tamanhos e densidades de tela de dispositivos móveis Android, garantindo que todos os elementos interativos permaneçam acessíveis e legíveis independentemente da resolução do dispositivo.

Nº	Requisito Não Funcional
RNF09	O sistema deve registrar internamente falhas de *backup* ou restauração em nuvem e erros críticos de operação, incluindo data, hora e descrição do erro, para fins de diagnóstico, suporte e manutenção da aplicação.
RNF10	O sistema deve permanecer funcional para operações essenciais de registro, consulta, revisão, organização e recuperação local de registros de comprovantes mesmo na ausência de conexão com a internet, ressalvadas as funcionalidades de *backup* em nuvem e as consultas automáticas de enriquecimento fiscal ou cadastral.
RNF11	O pacote de instalação da aplicação não deve exceder 210 MB, excluindo os arquivos e registros armazenados pelo usuário, considerando os artefatos locais de PLN embarcados na aplicação e de forma a viabilizar a instalação em dispositivos com capacidade de armazenamento interno reduzida.
RNF12	O sistema deve informar ao usuário, em local de fácil acesso na interface, o espaço total ocupado pelos arquivos associados aos registros armazenados localmente no dispositivo, permitindo que o usuário tome decisões conscientes sobre a gestão do seu armazenamento.
RNF13	O sistema deve adotar estratégias de controle do armazenamento ocupado pelos arquivos associados aos registros, incluindo o uso de formatos de imagem compactados quando gerados pelo scanner de documentos e o pré-processamento de variantes temporárias para OCR, sem comprometer a legibilidade necessária à consulta posterior.
RNF14	O processo de registro de um arquivo ou imagem de comprovante — desde a abertura do scanner de documentos ou seleção do arquivo até a confirmação do armazenamento — deve ser concluído em no máximo quatro etapas de interação, reforçando a abordagem de baixo esforço que orienta o design da aplicação.
RNF15	O processo de importação de um arquivo de comprovante por meio da funcionalidade de compartilhamento de outros aplicativos deve ser concluído em no máximo três etapas de interação a partir do momento em que o usuário aciona o compartilhamento no aplicativo de origem.
RNF16	O sistema deve exibir mensagens de erro compreensíveis ao usuário, sem exposição de detalhes técnicos internos, sempre que uma operação não puder ser concluída com sucesso.
RNF17	A integração com o Google Drive deve utilizar o escopo de acesso mínimo necessário (*drive.appdata* ou equivalente), restringindo o acesso da aplicação exclusivamente aos arquivos por ela criados, sem permissão de leitura ou modificação de outros arquivos presentes na conta do usuário.

Nº	Requisito Não Funcional
RNF18	Os arquivos de *backup* enviados ao Google Drive devem ser criptografados localmente no dispositivo antes da transmissão, de forma que o conteúdo dos registros, dados extraídos e arquivos associados não esteja acessível em texto claro no ambiente de armazenamento em nuvem.
RNF19	O sistema deve solicitar a reautenticação do usuário com a conta Google sempre que o *token* de acesso expirar ou for revogado, sem armazenar credenciais de forma persistente no dispositivo fora dos mecanismos seguros fornecidos pelo sistema operacional.
RNF20	O sistema deve armazenar segredos sensíveis, como a senha de *backup*, em mecanismo seguro fornecido pelo sistema operacional, sem persistir esses dados em texto claro no banco de dados local.
RNF21	Os arquivos associados aos registros de comprovantes armazenados pela aplicação, incluindo imagens capturadas e documentos digitais importados, devem ser mantidos em diretório privado da aplicação no dispositivo, evitando exposição direta na galeria ou em áreas públicas de armazenamento.
RNF22	O relatório de problema enviado pelo usuário deve limitar-se ao relato informado, dados técnicos básicos do aplicativo e do dispositivo e registros recentes de erro, sem incluir comprovantes, dados financeiros, imagens, conteúdo extraído por OCR ou outros dados sensíveis do usuário.
RNF23	O processo de restauração de *backup* deve validar a versão, a estrutura e o limite de armazenamento do pacote antes de substituir os dados locais, utilizando área temporária e mecanismo de reversão para evitar que falhas durante a restauração deixem registros e arquivos associados em estado inconsistente.

3 Design

O design do FinTrack é orientado por uma premissa central que permeia todas as decisões subsequentes: as funcionalidades essenciais de escaneamento, processamento, registro e recuperação de comprovantes são executadas no próprio dispositivo do usuário. Os arquivos ou imagens associados aos registros, bem como os dados extraídos durante o processamento, também são mantidos localmente, sem dependência de infraestrutura externa para o funcionamento regular da aplicação.

Nesse sentido, esta seção documenta o design do sistema a partir de três perspectivas complementares. A Seção 3.1 apresenta os diagramas que formalizam a estrutura estática e o comportamento dinâmico do sistema, utilizando a Unified Modeling Language (UML). A Seção 3.2 detalha as decisões arquiteturais utilizando a notação C4 Model — uma abordagem de modelagem arquitetural organizada em níveis progressivos de abstração [50] —, descrevendo o sistema por meio dos níveis de contexto e contêineres. E a Seção 3.3 apresenta a modelagem do banco de dados local, tanto em nível lógico quanto físico.

3.1 Projeto UML

Na primeira perspectiva de documentação do design, os diagramas de casos de uso delimitam o escopo funcional da aplicação do ponto de vista do usuário e dos comportamentos internos associados aos fluxos principais. O diagrama de classes estabelece a estrutura estática do sistema — suas entidades, contratos e dependências — e os diagramas de sequência conectam as duas perspectivas anteriores, mostrando como os componentes estruturais colaboram para realizar dois fluxos centrais da aplicação: o registro de arquivos ou imagens de comprovante e a recuperação dos registros salvos.

Diagrama de casos de uso

Os casos de uso do FinTrack foram organizados em blocos menores para facilitar a leitura e reduzir a sobreposição visual entre relações. O primeiro bloco, visualizado na Figura 3, cobre os fluxos de escaneamento de documento físico, importação individual ou em lote de arquivos de comprovante, processamento de prévia, revisão de lote, enriquecimento fiscal ou cadastral, confirmação e descarte do registro. Nesse bloco, a BrasilAPI e os serviços fiscais das Secretarias da Fazenda (SEFAZ) e do gov.br aparecem como atores externos porque participam, respectivamente, da consulta cadastral por CNPJ e da consulta fiscal por QR Code ou chave de acesso.

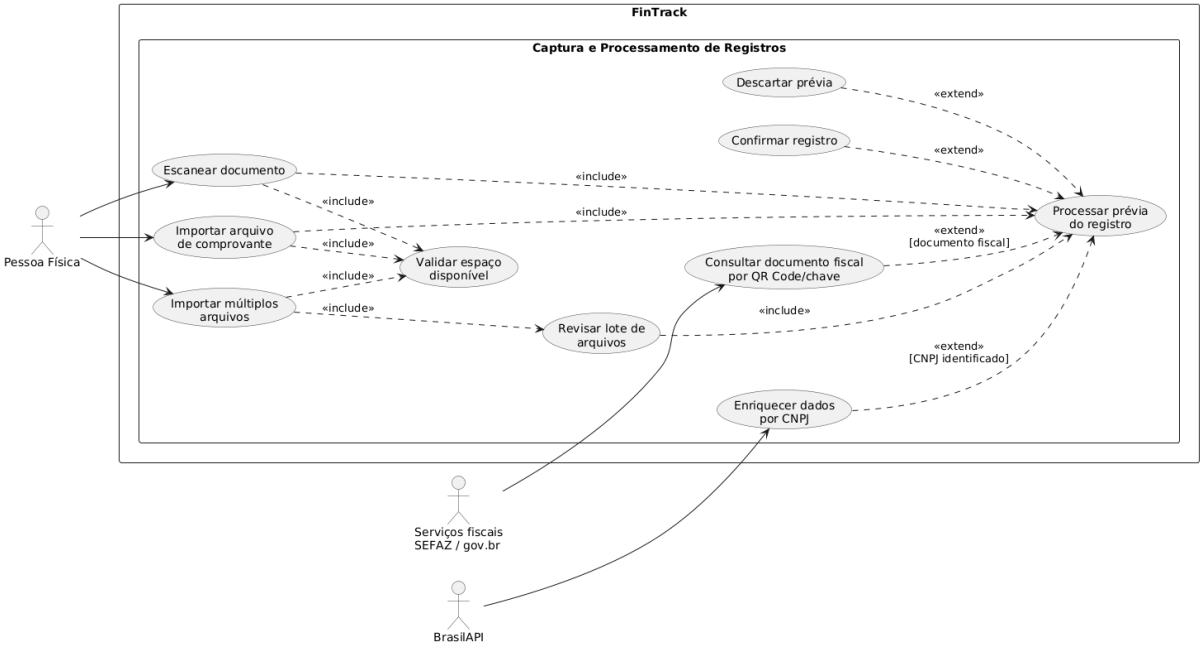


Figura 3: Diagrama de Casos de Uso do FinTrack — Escaneamento, Importação e Processamento

O processamento automático — OCR, extração de dados estruturados, geração de *embeddings* e classificação por categoria — não é um caso de uso iniciado diretamente pelo usuário, mas sim um comportamento interno decorrente dos casos de uso de escaneamento e importação. O usuário interage com o resultado desse processamento ao visualizar, confirmar ou corrigir as informações extraídas, e posteriormente ao recuperar o registro salvo.

O segundo bloco, visualizado na Figura 4, apresenta os casos de uso relacionados à gestão e recuperação dos registros de comprovantes, incluindo visualização, edição, exclusão, busca textual e semântica, filtros, ordenação e ações individuais ou em lote sobre os arquivos associados aos registros.

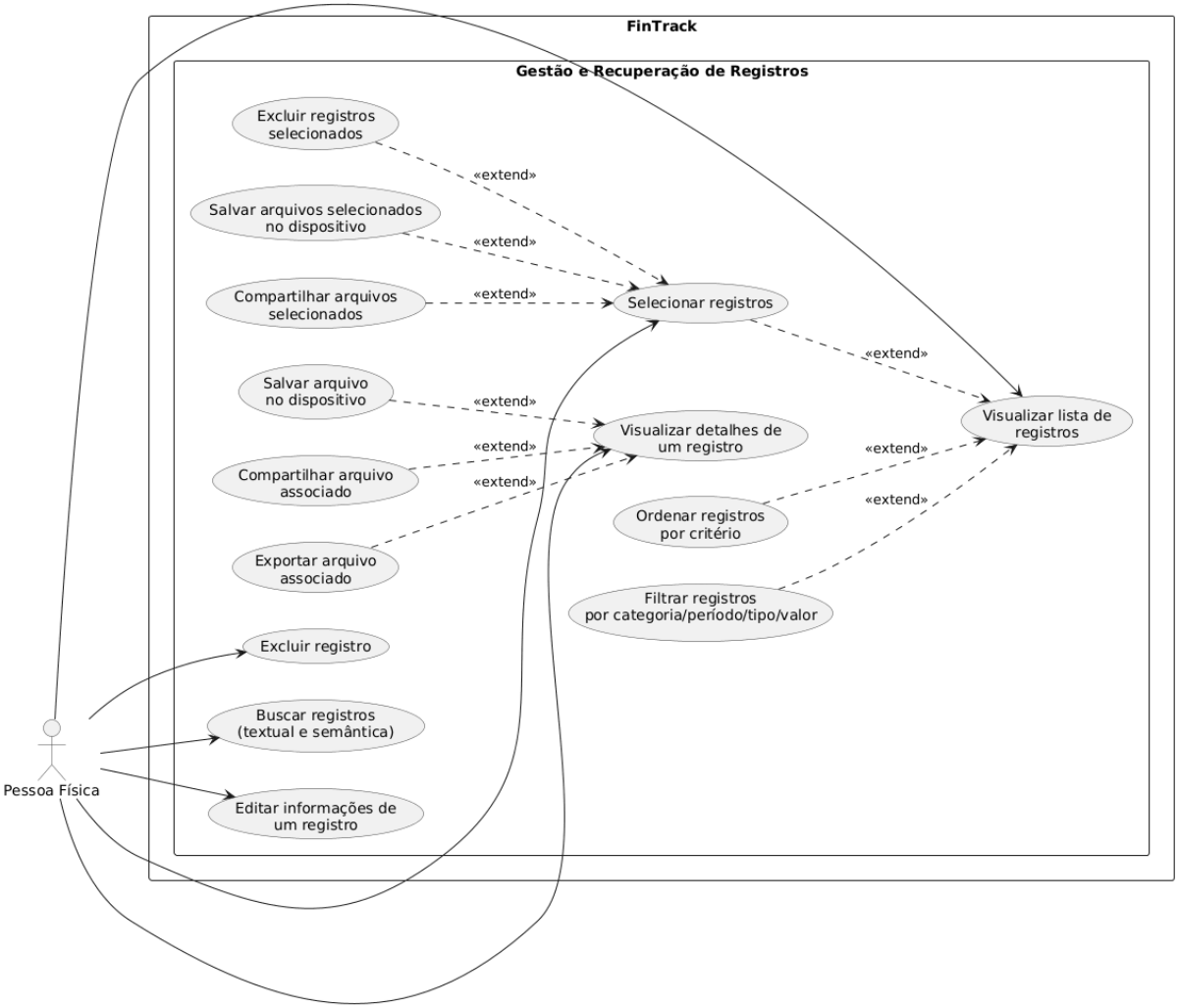


Figura 4: Diagrama de Casos de Uso do FinTrack — Gestão e Recuperação

O terceiro bloco, visualizado na Figura 5, concentra os casos de uso associados à gestão de categorias, contemplando visualização, busca, criação, edição, exclusão, reordenação e visualização dos comprovantes associados a uma categoria. Essas categorias organizam os registros de comprovantes, qualificam filtros e apoiam a geração de relatórios.

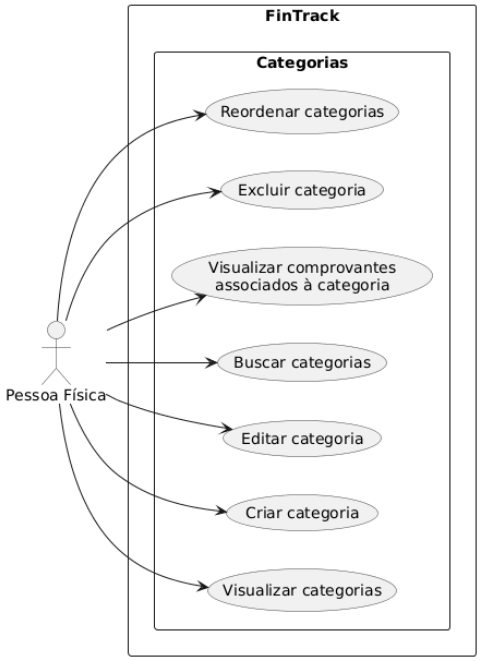


Figura 5: Diagrama de Casos de Uso do FinTrack — Categorias

O quarto bloco, visualizado na Figura 6, reúne as configurações de uso, segurança local e experiência, abrangendo refazer tutorial, autenticação local, intervalo de bloqueio automático quando a proteção local está ativa, tema visual, limite de armazenamento, acesso à tela Sobre o FinTrack, FAQ, reporte de problema e saída do aplicativo.

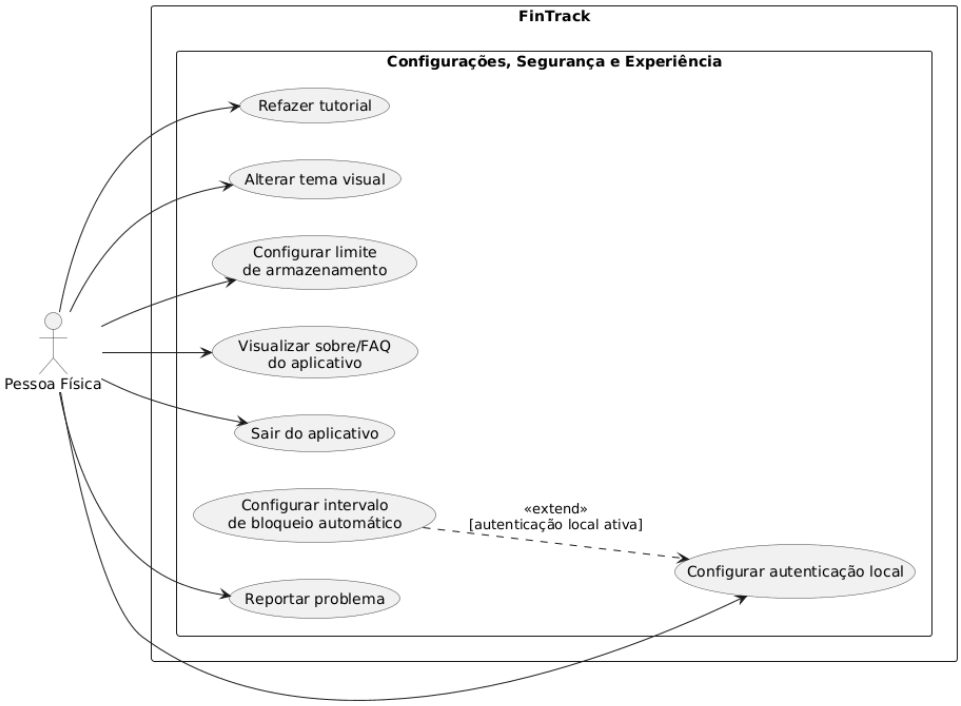


Figura 6: Diagrama de Casos de Uso do FinTrack — Configurações, Segurança e Experiência

O quinto bloco, visualizado na Figura 7, cobre as funcionalidades complementares de *backup* em nuvem, que são opcionais e dependentes de conectividade. Na solução implementada, o armazenamento em nuvem é realizado por meio do Google Drive; por isso, o bloco inclui vínculo e desvínculo da conta Google, criação do *backup*, restauração, exclusão do *backup* remoto, limpeza do histórico local, visualização de detalhes do histórico, configuração, alteração e remoção da senha de *backup*, além da configuração do *backup* automático e de seu intervalo. Os serviços Google aparecem como ator externo por representarem a autenticação/autorização e o armazenamento em nuvem utilizados nesse fluxo.

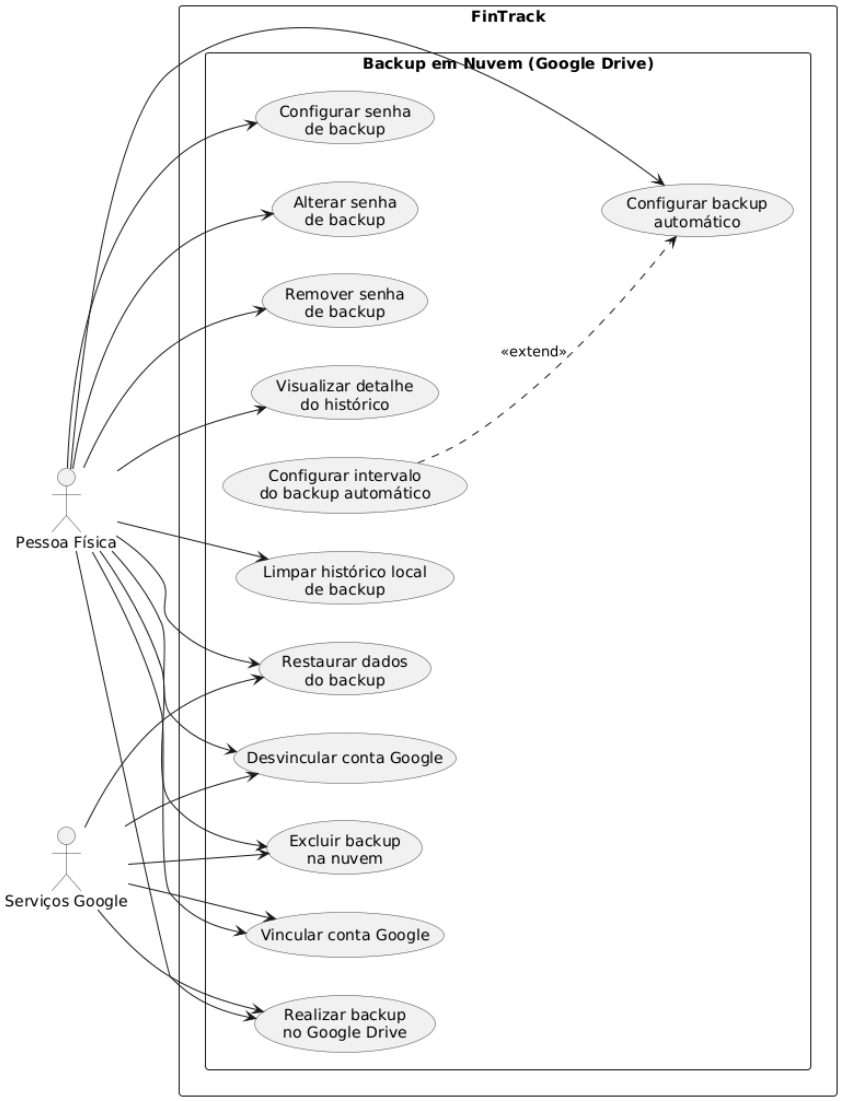


Figura 7: Diagrama de Casos de Uso do FinTrack — Backup em Nuvem

Por fim, o sexto bloco, visualizado na Figura 8, concentra os casos de uso do relatório financeiro, incluindo visualização dos relatórios, filtro por período e listagem dos comprovantes vinculados a um item do relatório.

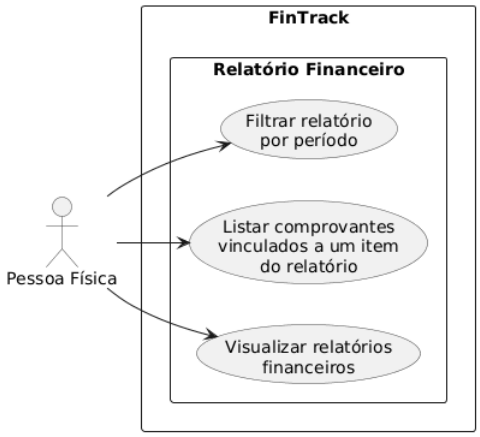


Figura 8: Diagrama de Casos de Uso do FinTrack — Relatório Financeiro

A geração de relatórios financeiros é viabilizada pelos dados estruturados extraídos automaticamente pelo OCR e persistidos na base de dados local. A partir das informações extraídas — valor, data, estabelecimento e meio de pagamento — e das categorias associadas a cada registro, o sistema agrega os dados e os apresenta ao usuário como uma visão consolidada de seus gastos, organizados por categoria, período ou tipo de comprovante.

A geração do relatório, portanto, não envolve novo OCR ou reinterpretação semântica dos documentos; ela realiza apenas processamento leve em tempo de execução, como filtragem por período, agregação de valores, separação entre receitas e despesas e cálculo de totais a partir dos dados já estruturados no momento do registro.

Diagrama de classes

O diagrama de classes, ilustrado no Apêndice A, é o modelo estrutural central do FinTrack. Ele foi projetado segundo os princípios SOLID — conjunto de diretrizes para design orientado a objetos que regem, entre outros aspectos, a separação entre contratos e implementações [51] —, expressos aqui como interfaces e serviços concretos.

Essa separação viabiliza a aplicação de três padrões de projeto principais. O Repository abstrai o acesso ao SQLite por meio das interfaces de persistência representadas no diagrama, incluindo os repositórios de registros de comprovantes, categorias, configuração, *backup*, importação em lote e tarefas de indexação semântica, desacoplando a lógica de negócio dos detalhes de armazenamento.

O Strategy está presente nos contratos de infraestrutura, que encapsulam OCR, geração de *embeddings*, manipulação de imagens, leitura de códigos visuais, criptografia e armazenamento em nuvem, permitindo a troca de implementações sem alteração nos serviços de orquestração. E o Facade se manifesta nos serviços de aplicação — especialmente `ReceiptService`, `CategoryService`, `BackupService` e `ConfigurationService` —, cada um oferecendo um ponto único de entrada para seu respectivo conjunto de casos de uso e ocultando a complexidade das cadeias de processamento detalhadas nos diagramas de sequência.

Diagramas de sequência

O diagrama de sequência detalhado no Apêndice E apresenta as interações entre os componentes do sistema durante o registro de um arquivo ou imagem de comprovante, operação que aciona a cadeia completa de processamento automático. O fluxo se inicia quando o usuário escaneia um documento físico, com uso da câmera como alternativa quando o scanner não está disponível, ou importa um arquivo, individualmente ou em lote, e a interface solicita ao `ReceiptService` o processamento de uma prévia antes da persistência definitiva.

A partir desse ponto, o `ReceiptService` atua como fachada e delega a cadeia de processamento aos serviços especializados. O `ReceiptProcessingService` conduz a validação do arquivo, a geração de variantes para OCR, a leitura de QR Code fiscal, a análise de qualidade da imagem e a extração de texto por Google ML Kit. O resultado é encaminhado ao `DataExtractorService`, que utiliza parsers especializados para documentos fiscais, recibos, pagamentos com cartão, transferências digitais e casos de fallback.

Antes de apresentar a prévia ao usuário, o fluxo ainda passa pelo `ReceiptEnrichment Service`, pelo `ReceiptCategoryService` e pelo `FinancialNatureClassifierService`, realizando enriquecimento por cache fiscal e por CNPJ, consulta de preferências de categoria por CNPJ, estabelecimento e itens, e inferência da natureza financeira da transação. Somente após a confirmação do usuário o arquivo é persistido em local definitivo, os dados são salvos no SQLite por meio de `IReceiptRepository` e as preferências e caches associados são atualizados.

Ao final do fluxo confirmado, a geração do *embedding* semântico não bloqueia a interação principal: ela é agendada em segundo plano pelo `ReceiptSemanticService`. A geração do vetor é realizada pelo `ReceiptSemanticIndexer`, que utiliza o `DistilUseMultilingualService` a partir do texto semântico composto do comprovante. Essa estratégia mantém o registro responsivo e permite que a indexação semântica seja concluída sem impor espera adicional ao usuário.

O diagrama de sequência detalhado no Apêndice F descreve o fluxo de recuperação de registros de comprovantes a partir de uma consulta digitada pelo usuário. Do ponto de vista da interface, a busca se apresenta como uma operação única e transparente: o usuário digita um termo e recebe uma lista de resultados.

Internamente, o `ReceiptService` delega a operação ao `ReceiptSearchService`, que executa duas estratégias complementares de forma coordenada. A primeira utiliza busca textual sobre o índice `receipt_fts`, baseado em FTS5, para recuperar registros por termos presentes no conteúdo extraído e nos campos estruturados.

A segunda aciona o `ReceiptSemanticService`, que gera um *embedding* da consulta por meio do `ReceiptSemanticIndexer` e do `DistilUseMultilingualService`, recuperando registros semanticamente relacionados a partir da tabela `embedding_vector` com apoio da extensão *sqlite-vector*, especialmente por meio da operação `vector_full_scan`. Os resultados textuais e semânticos são então mesclados, pontuados, deduplicados e ordenados internamente antes de serem retornados à interface.

O fluxo de recuperação também contempla mecanismos internos de manutenção da

qualidade da busca semântica. Quando necessário, o serviço agenda a reindexação de *embeddings* pendentes ou desatualizados sem exigir ação explícita do usuário. Dessa forma, a recuperação combina busca textual, busca vetorial e observação reativa por filtros, sem depender do carregamento integral dos vetores em memória pela camada Dart.

3.2 Visão arquitetural

Do ponto de vista de arquitetura, o FinTrack é uma aplicação *standalone*, isto é, toda a lógica de negócio, o processamento e a persistência residem integralmente no dispositivo do usuário, sem dependência de servidor de *backend* ou banco de dados remoto. Essa escolha implica *tradeoffs* reconhecidos — modelos locais de PLN apresentam capacidade inferior a modelos maiores executados em nuvem e o consumo de energia durante o processamento é mais elevado —, assumidos explicitamente em favor da privacidade dos dados e da operação contínua em ambientes com conectividade limitada.

No entanto, a ausência de infraestrutura distribuída não implica ausência de organização arquitetural. O FinTrack é estruturado como uma aplicação monolítica local organizada em camadas, sendo elas: Apresentação, Aplicação, Domínio e Infraestrutura.

A camada de Apresentação é composta pelos *widgets* Flutter, responsáveis pela interface visual e pela reatividade da UI. A camada de Aplicação concentra os serviços de orquestração dos casos de uso. A camada de Domínio define as entidades, *value objects* e os contratos de interface. E a camada de Infraestrutura provê as implementações concretas de persistência, OCR, PLN e criptografia. As dependências são direcionadas aos contratos das camadas internas, preservando a separação entre domínio, aplicação e infraestrutura, enquanto a composição das implementações concretas ocorre em ponto específico de inicialização da aplicação.

A arquitetura da solução foi representada por meio do C4 Model em dois níveis progressivos de detalhamento visual: contexto e contêineres.

Nível 1 — Contexto

No nível de contexto, visualizado na Figura 9, o FinTrack é situado em relação ao seu ator principal — o usuário — e às integrações externas ou locais necessárias para a execução dos seus fluxos.

A aplicação permanece *offline-first* e não depende de servidor próprio ou banco remoto, mas interage com serviços externos em situações delimitadas: autenticação OAuth 2.0 e armazenamento em nuvem para *backup* criptografado opcional, BrasilAPI para consulta cadastral automática por CNPJ e serviços fiscais das SEFAZ ou gov.br para enriquecimento automático de documentos fiscais a partir de QR Code ou chave de acesso. Também são explicitados os recursos nativos do dispositivo Android utilizados localmente, como Google ML Kit, scanner de documentos, leitura de QR Code, autenticação local e armazenamento seguro.

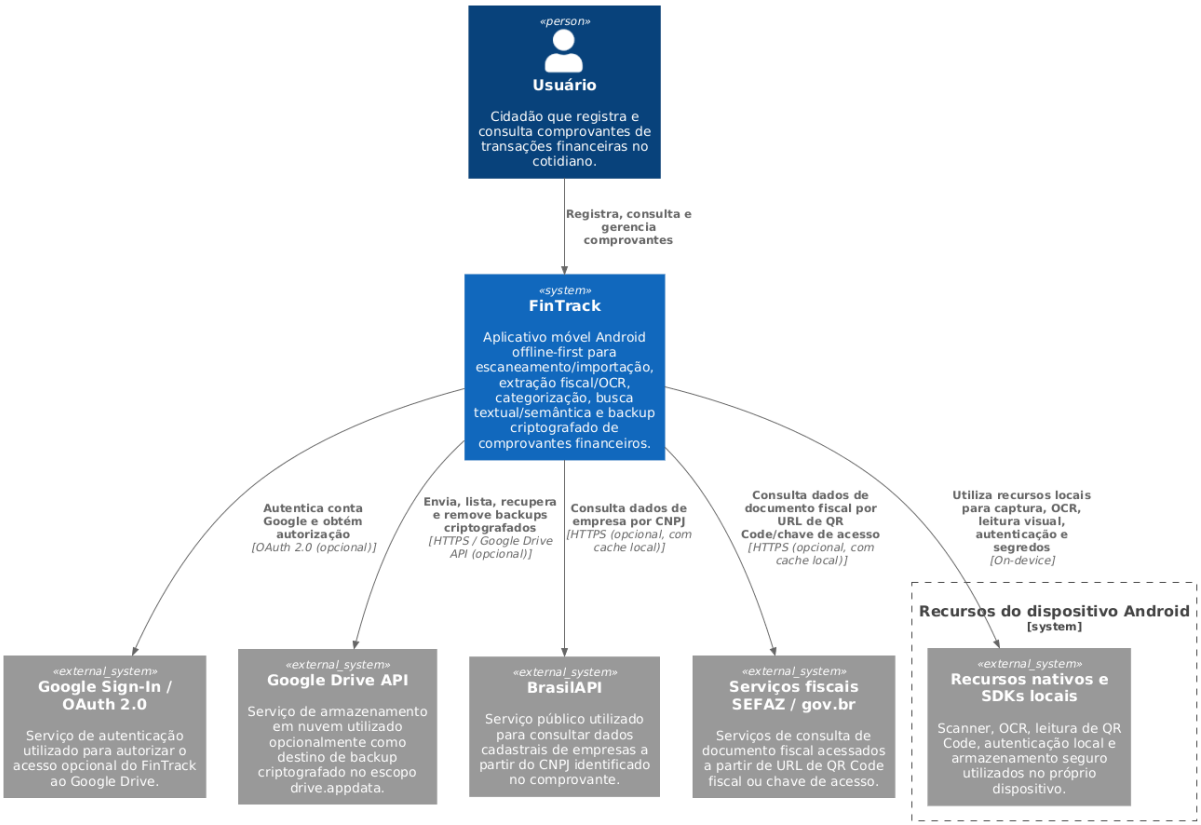


Figura 9: Diagrama de Contexto do FinTrack

Nível 2 — Contêineres

No nível de contêineres, visualizado na Figura 10, cada camada da arquitetura é representada como um contêiner independente: Apresentação, Aplicação, Domínio e Infraestrutura. O diagrama ilustra como o FinTrack está contido no dispositivo móvel Android, sem utilização de *backend* próprio ou banco de dados remoto.

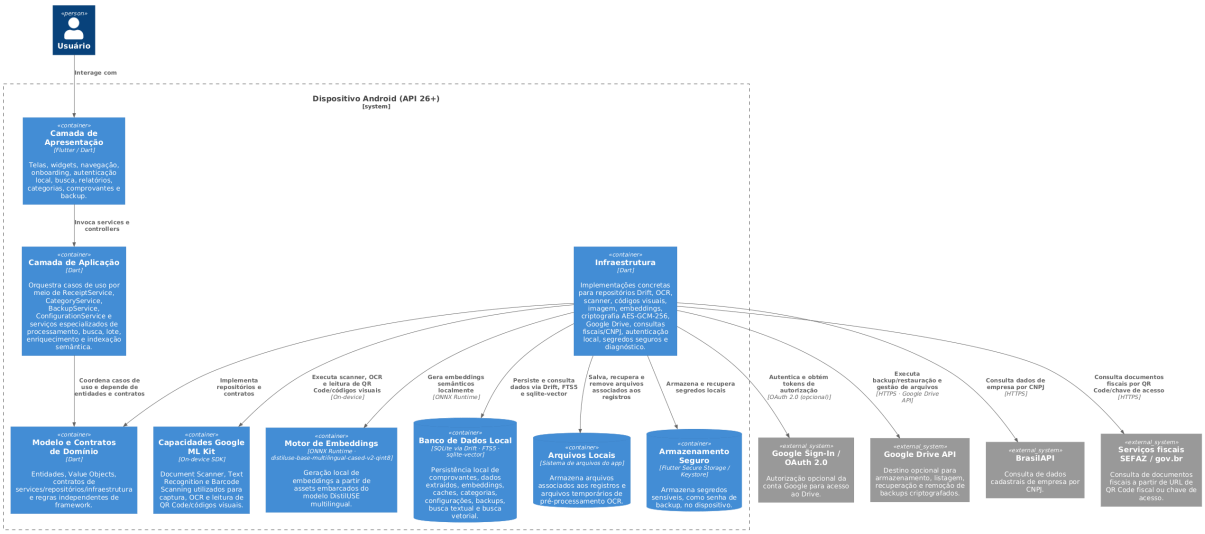


Figura 10: Diagrama de Contêiner do FinTrack

A persistência local, entretanto, não se limita ao SQLite: o banco local é acessado via Drift e complementado por FTS5 e *sqlite-vector* para busca textual e vetorial; os arquivos associados aos registros de comprovantes são mantidos no sistema de arquivos privado da aplicação; e segredos sensíveis, como a senha de *backup*, são protegidos por Flutter Secure Storage e Android Keystore. A camada de Infraestrutura agrega OCR, scanner, leitura de QR Code, PLN, geração de *embeddings*, repositórios, criptografia, consultas fiscais, consulta cadastral por CNPJ, autenticação local e comunicação opcional com serviços externos.

A interação entre as camadas ocorre por chamadas de método diretas entre objetos Dart, sem *overhead* de serialização ou protocolo de rede interno. Na persistência local, os repositórios baseados em Drift expõem *streams* que propagam alterações nos dados persistidos até a UI, permitindo que as telas sejam atualizadas de forma reativa sem acoplamento direto entre a interface e os detalhes do banco de dados.

As saídas de rede são restritas a fluxos específicos: *backup* e restauração são opcionais e utilizam HTTPS com autenticação OAuth 2.0 e arquivos criptografados localmente; consultas à BrasilAPI e a serviços fiscais são acionadas de forma oportunística, quando houver dados identificáveis e conectividade, com aproveitamento de caches locais. Falhas nessas consultas externas não bloqueiam o registro ou a recuperação local dos registros de comprovantes, preservando a experiência *offline-first*.

3.3 Modelo de banco de dados

A estrutura do banco de dados do FinTrack reflete o modelo de domínio estabelecido no diagrama de classes: as entidades centrais são persistidas em tabelas relacionais, enquanto associações, estruturas técnicas de busca e caches recebem regras de integridade compatíveis com sua semântica. Assim, algumas dependências utilizam chaves estrangeiras com remoção em cascata, outras preservam o registro principal por meio de `ON DELETE SET NULL`, e estruturas auxiliares de indexação ou cache podem ser mantidas sem chave estrangeira declarada. Os *value objects* — como `ReceiptFilter` e `OcrResult` — não são persistidos diretamente, pois existem apenas em memória durante o processamento; já os vetores representados por `EmbeddingVector` são materializados nas estruturas técnicas de busca semântica descritas a seguir.

Uma decisão de projeto relevante diz respeito ao armazenamento dos *embeddings*. Os vetores são gerados localmente pelo serviço `DistilUseMultilingualService`, que executa o modelo `distiluse-base-multilingual-cased-v2-qint8` por meio do ONNX Runtime a partir de artefatos embarcados na aplicação. Esses vetores são persistidos no SQLite em duas estruturas complementares: a tabela `embedding`, que mantém o registro lógico do vetor associado ao comprovante, e a tabela técnica `embedding_vector`, utilizada em conjunto com a extensão *sqlite-vector* para viabilizar consultas por similaridade diretamente no banco local. Desse modo, a aplicação preserva a execução *on-device* e a abordagem *offline-first*, mas evita que a recuperação semântica dependa do carregamento integral dos vetores em memória pela camada Dart.

Outra decisão de projeto relevante diz respeito ao armazenamento dos arquivos associados

aos registros de comprovantes. Persistir esses arquivos como BLOB diretamente no SQLite impactaria negativamente a latência das consultas e inflaria o arquivo do banco de forma desproporcional ao volume de metadados. Por essa razão, os arquivos são armazenados no diretório privado da aplicação no sistema de arquivos do dispositivo, inacessível a outros aplicativos e invisível na galeria do usuário.

Nesse sentido, o banco de dados armazena apenas o nome do arquivo no campo `file_name` da tabela `receipt`, a partir do qual o caminho absoluto é reconstruído em tempo de execução concatenando o diretório base atual. Essa separação também garante portabilidade do *backup*, pois ao restaurar em um novo dispositivo, os caminhos são recalculados corretamente a partir do novo diretório base, sem dependência de caminhos absolutos gravados em tempo de registro.

Modelo lógico

O modelo lógico completo, apresentado no Apêndice C, organiza os dados do FinTrack em entidades de domínio e estruturas auxiliares de busca, ordenação e cache. As composições entre `Receipt`, `ExtractedData` e `Embedding` refletem diretamente as composições do diagrama de classes — esses registros não existem sem o comprovante ao qual pertencem e são removidos em cascata. A classificação por categoria, por sua vez, é representada por uma associação opcional entre `Receipt` e `Category`, materializada pelo campo `categoryId`; assim, cada comprovante pode estar associado a, no máximo, uma categoria principal, sem necessidade de tabela associativa.

Além das entidades centrais, o modelo inclui estruturas técnicas que dão suporte ao comportamento atual da aplicação: `ReceiptFTS`, para busca textual em conteúdo extraído; `EmbeddingVector`, para busca vetorial com *sqlite-vector*; `CnpjCache`, `FiscalDocumentCache`, `EstablishmentCategoryCache` e `ItemCategoryCache`, para reaproveitamento de dados fiscais, cadastrais e preferências de categorização; e `CategoryOrder`, para persistência da ordenação definida pelo usuário.

O modelo também inclui `ReceiptBatchImportSession` e `ReceiptBatchImportItem`, que sustentam a importação em lote, além de `SemanticIndexTask`, que registra tarefas de indexação semântica pendentes, em processamento, concluídas ou com falha. No conjunto de configuração e *backup*, `BackupRecord` referencia `Configuration` por meio de `configurationId`, permitindo registrar o histórico de operações de exportação, restauração e disponibilidade do *backup* associado à configuração local vigente.

Modelo físico

O modelo físico completo, apresentado no Apêndice D, traduz o modelo lógico para o esquema de banco de dados SQLite utilizado pela aplicação. No código, os campos temporais são modelados por meio de `DateTimeColumn` do Drift e materializados no SQLite como valores inteiros, enquanto valores booleanos também são representados de forma compatível com o armazenamento inteiro adotado pelo mecanismo relacional. O campo `file_name` da tabela `receipt` recebe restrição `UNIQUE` para evitar a duplicação de registros associados ao mesmo arquivo.

As regras de integridade referencial variam conforme a semântica de cada associação: `extracted_data`, `embedding`, `embedding_vector`, `category_order`, preferências de categoria, registros de *backup*, itens de importação em lote e tarefas de indexação semântica utilizam remoção em cascata quando dependem integralmente da entidade referenciada; já a categoria opcional do comprovante e a categoria preferida associada ao cache de CNPJ utilizam `ON DELETE SET NULL`, preservando o registro principal quando a categoria é removida. Estruturas técnicas como `receipt_fts` e `fiscal_document_cache` não declaram chave estrangeira direta, pois funcionam como índices ou caches auxiliares mantidos pela própria aplicação.

A tabela `extracted_data` inclui o campo `ocr_confidence` e confianças específicas de extração, que registram metadados de qualidade do OCR e dos campos estruturados. Essas informações subsidiam a sinalização ao usuário, nas telas de confirmação e revisão em lote, quando uma revisão manual dos dados extraídos é recomendada. A tabela `configuration` inclui os campos `cloud_token_valid` e `cloud_linked_at`, que registram o estado da autenticação OAuth com o provedor de nuvem e o momento do vínculo, viabilizando o comportamento de reautenticação exigido pelo RNF19.

4 Testes de Software

A estratégia de testes adotada no desenvolvimento do FinTrack foi baseada em uma abordagem fortemente acompanhada por testes automatizados, inspirada em práticas de desenvolvimento orientado por testes, do inglês *Test-Driven Development* (TDD). Nesse modelo, comportamentos críticos são especificados e verificados por testes unitários e de *widgets* ao longo do desenvolvimento, de modo que a suíte de testes apoie tanto a definição do comportamento esperado quanto a detecção precoce de regressões [52].

Essa estratégia é particularmente relevante no contexto do FinTrack porque as funcionalidades essenciais da solução são executadas no próprio dispositivo do usuário. Como não há infraestrutura própria de *backend* ou ambiente de homologação capaz de absorver parte da validação, a confiabilidade dos serviços locais, repositórios, modelos de domínio, parsers, políticas, formatadores e componentes de interface torna-se determinante para a integridade do sistema.

Na prática, os ciclos de escrita de testes, implementação e refatoração foram aplicados de forma recorrente nos componentes exercitáveis por testes automatizados. Já os fluxos dependentes de recursos nativos do Android, como scanner de documentos, OCR, leitura de QR Code, autenticação local, sistema de arquivos e integração com armazenamento em nuvem, foram complementados por validações manuais em dispositivos Android físicos reais. Dessa forma, a estratégia combina especificação automatizada de comportamento, acompanhamento de cobertura por LCOV e verificação prática dos fluxos que dependem de recursos nativos ou de dispositivos físicos.

4.1 Projeto de testes

A organização dos testes acompanha a arquitetura em camadas adotada pelo sistema e combina testes automatizados com validações manuais em dispositivos Android físicos reais. No repositório da aplicação, a suíte automatizada é composta por testes unitários e testes de *widgets*, implementados com o Flutter Test e o Mockito, descritos na Seção 1.3. Além de *mocks*, a suíte utiliza *fakes* — implementações simplificadas e controladas de uma dependência —, serviços simulados, repositórios controlados e bancos em memória para aproximar determinados comportamentos sem acionar dependências externas reais.

Os fluxos que dependem diretamente de recursos nativos do Android, como scanner ou captura de imagem, permissões, desempenho do dispositivo, OCR, leitura de QR Code e integração com serviços externos foram validados manualmente em dispositivos Android físicos reais. Cada nível possui escopo, objetivos e ferramentas específicas, conforme detalhado a seguir.

Testes Unitários

Os testes unitários verificam o comportamento de unidades isoladas do sistema — serviços, repositórios, modelos de domínio, parsers, políticas e formatadores — de forma independente de dependências externas como scanner de documentos, sistema de arquivos real, OCR nativo, serviços fiscais, serviços cadastrais por CNPJ, armazenamento em nuvem ou serviços de PLN. Para garantir esse isolamento, as interfaces definidas na camada de Domínio são substituídas por dublês de teste, como *mocks*, *fakes* e implementações em memória, produzidos com apoio do `mockito` ou definidos diretamente na suíte de testes quando isso torna o cenário mais simples e previsível.

O principal alvo dos testes unitários é a camada de Aplicação, onde residem os serviços de orquestração dos casos de uso. Para o `ReceiptService`, os testes verificam: (i) a coordenação entre scanner de documentos, serviço de imagens, pré-processamento, OCR, leitura de QR Code, extração estruturada e repositórios durante a geração de prévia e o salvamento confirmado; (ii) o comportamento em caso de falha em etapas como validação de arquivo, processamento OCR, consulta fiscal, consulta cadastral por CNPJ, persistência ou manipulação de arquivos; (iii) a aplicação de preferências de categoria por CNPJ, estabelecimento e itens, com fallback semântico quando aplicável; (iv) a lógica de mesclagem e deduplicação dos resultados de busca textual e semântica; e (v) o agendamento e a persistência da indexação semântica sem bloqueio do fluxo principal.

Para o `CategoryService`, os testes cobrem a lógica de gestão das categorias. Para o `BackupService`, os testes cobrem a orquestração correta entre `ICryptographyService`, `ICloudStorage`, `IBackupRepository` e os demais componentes envolvidos nas operações de exportação, restauração, exclusão e histórico de *backup*. E para o `ConfigurationService`, os testes verificam a gestão do vínculo com a conta Google, a autenticação local, o limite de armazenamento e as preferências de *backup*.

No contexto de desenvolvimento acompanhado por testes, esses comportamentos foram especificados e refinados progressivamente à medida que as funcionalidades evoluíram. O

ciclo de implementação é considerado concluído quando os testes automatizados relevantes estão aprovados, as validações manuais aplicáveis foram executadas nos fluxos dependentes de dispositivo e o código foi revisado para reduzir duplicações e preservar legibilidade.

Testes de *Widgets*

Os testes de *widgets* verificam o comportamento de componentes visuais de forma isolada, sem a necessidade de executar a aplicação em um dispositivo Android físico. Cada *widget* é renderizado em um ambiente controlado, e suas interações — toques, entradas de texto, navegação entre telas — são simuladas programaticamente utilizando as APIs disponibilizadas pelo próprio Flutter Test.

No FinTrack, os testes de *widgets* cobrem fluxos críticos da interface, incluindo *onboarding*, navegação principal, autenticação local, formulário de confirmação e edição de dados extraídos após o OCR, listagem e filtragem de comprovantes, busca textual e semântica, detalhe do comprovante, revisão em lote, categorias, relatórios, configurações, *backup* em nuvem, importação compartilhada e estados de erro. Para cada componente, os testes verificam: (i) a renderização correta dos elementos visuais a partir de diferentes estados de dados; (ii) a reatividade da UI a mudanças de estado via *streams* do Drift ou controladores de tela; e (iii) o acionamento correto das ações do usuário, como confirmação, edição, exclusão, navegação, compartilhamento, salvamento externo e tratamento de falhas.

Nesse nível, os testes funcionam como contratos executáveis da interface, registrando quais elementos devem estar presentes em cada estado possível e quais ações devem ser disparadas a partir das interações do usuário. Isso reduz regressões visuais e comportamentais, além de tornar mais segura a evolução de telas com muitos estados, como confirmação de comprovante, revisão em lote, configurações, listagem e *backup*.

Validação em Dispositivos Móveis

Além dos testes automatizados, foram realizadas validações manuais de fluxos completos da aplicação em dispositivos Android físicos reais. Essas validações exercitam a interação entre múltiplas camadas do sistema em cenários próximos ao uso real, especialmente nos pontos em que o comportamento depende de recursos nativos do dispositivo ou de serviços externos. Entre os dispositivos utilizados, inclui-se um Samsung Galaxy A56 equipado com processador Exynos 1580 *octa-core* operando em até 2,9 GHz, 8 GB de RAM, 128 GB de armazenamento interno e sistema operacional Android 16 com interface Samsung One UI 8.

Os cenários cobertos por essas validações incluem o fluxo completo de registro de um comprovante — desde o escaneamento ou importação do arquivo até a revisão da prévia, confirmação do armazenamento e verificação dos dados persistidos no SQLite —, o fluxo de busca textual e semântica com verificação dos resultados retornados, e o fluxo de *backup* e restauração com o Google Drive em conta de teste. Diferentemente dos testes unitários e de *widgets*, essas validações não constituem uma suíte formal de testes de integração versionada no diretório `integration_test`; elas funcionam como verificação prática dos requisitos em dispositivos reais.

Casos de teste prioritários

A Tabela 4 apresenta os casos de teste prioritários, organizados por nível e rastreados ao requisito que verificam. Os casos de teste (CT) representam os principais comportamentos especificados e validados ao longo do desenvolvimento, combinando verificações automatizadas e validações manuais para os fluxos dependentes de recursos nativos do Android ou de dispositivos físicos.

Tabela 4: Casos de teste prioritários do FinTrack

Nº	Nível	Objetivo	Saída esperada	Req.
CT01	Unitário	Verificar que o `ReceiptService` processa uma imagem válida, gera uma prévia revisável e, após confirmação, persiste `Receipt` e `ExtractedData`, mantendo a categoria como associação opcional e agendando a indexação semântica quando aplicável.	Registro confirmado persistido no SQLite, dados extraídos associados e indexação semântica encaminhada sem bloquear o fluxo principal; nenhuma exceção lançada.	RF02, RF05, RF22
CT02	Unitário	Verificar que o `ReceiptService` rejeita arquivos fora do tipo ou tamanho permitidos e não realiza persistência parcial.	Exceção de validação lançada; banco de dados inalterado.	RF02, RF03
CT03	Unitário	Verificar que o `ReceiptService` sugere categoria a partir da combinação entre preferências/cache por CNPJ, estabelecimento e itens e, quando necessário, similaridade semântica entre o texto extraído e as categorias disponíveis.	Categoria esperada retornada para conjunto de entradas conhecido, respeitando preferências registradas e fallback semântico.	RF05

Nº	Nível	Objetivo	Saída esperada	Req.
CT04	Unitário	Verificar que o `ReceiptService` retorna apenas comprovantes cujo conteúdo extraído ou nome de arquivo contém o termo pesquisado, usando o método `findByTerms` de `IReceiptRepository`.	Lista com correspondência exata; sem falsos positivos.	RF08
CT05	Unitário	Verificar que o `ReceiptService` retorna comprovantes semanticamente relacionados à consulta, mesmo sem correspondência exata de termos, usando o método `findSimilar` de `IReceiptRepository`.	Comprovantes semanticamente relevantes recuperados e ordenados conforme a pontuação de relevância retornada pela busca vetorial.	RF09
CT06	Unitário	Verificar que os resultados de busca textual e semântica retornados pelo `ReceiptService` são deduplicados e ordenados por relevância.	Lista sem duplicatas; comprovante de maior pontuação na primeira posição.	RF08, RF09
CT07	Unitário	Verificar que o `BackupService` produz arquivos criptografados via `ICryptographyService`, os envia via `ICloudStorage` e registra o `BackupRecord` correspondente via `IBackupRepository`.	Arquivo cifrado enviado; entrada de `BackupRecord` persistida com status `SINCRONIZADO`.	RF16, RNF18
CT08	Unitário	Verificar que o `BackupService` recupera os arquivos via `ICloudStorage`, os descriptografa via `ICryptographyService` e repopula o banco local via `IReceiptRepository`.	Banco local restaurado; comprovantes e dados extraídos íntegros.	RF17

Nº	Nível	Objetivo	Saída esperada	Req.
CT09	*Widget*	Verificar que o formulário pós-OCR renderiza os campos de `ExtractedData` e permite edição antes do armazenamento.	Campos editáveis exibidos com os valores extraídos; ação de confirmação acionada corretamente.	RF05, RF11, RF22
CT10	*Widget*	Verificar que a tela de listagem exibe os comprovantes reativamente e aplica filtros por categoria, período, tipo ou natureza financeira sem recarregamento completo.	Lista atualizada após filtro aplicado; estado de carregamento exibido durante consulta.	RF04, RF07
CT11	Validação manual	Verificar o fluxo completo de registro por escaneamento ou importação: pré-processamento → OCR → extração → prévia → confirmação → persistência, observando o tempo médio de 8 segundos por imagem.	Comprovante persistido com dados extraídos; tela de processamento exibida durante a operação; tempo médio observado de 7 a 8 segundos por imagem, permanecendo compatível com o requisito definido.	RF02, RF05, RF22, RF24, RNF05
CT12	Validação manual	Verificar que o sistema mantém desempenho aceitável de consulta com 600 comprovantes armazenados.	Tempo de resposta da listagem inferior a 2 segundos com base de dados populada.	RNF04, RNF06
CT13	Validação manual	Verificar o fluxo de *backup* e restauração via Google Drive com conta de teste: exportação → *upload* → *download* → restauração.	Comprovantes e dados extraídos recuperados corretamente no dispositivo de destino.	RF16, RF17
CT14	*Widget*	Verificar que a tela de revisão em lote permite revisar, corrigir, salvar, reprocessar ou descartar comprovantes importados simultaneamente.	Itens válidos salvos individualmente; itens descartados ou com erro não são persistidos indevidamente.	RF03, RF23

Nº	Nível	Objetivo	Saída esperada	Req.
CT15	Unitário	Verificar que o `ConfigurationService` ativa, desativa e valida o intervalo do *backup* automático apenas quando há conta Google vinculada e senha de *backup* definida.	Configuração persistida quando os pré-requisitos são atendidos; ativação bloqueada ou normalizada quando conta ou senha não estão disponíveis.	RF21, RNF20
CT16	Unitário	Verificar que o enriquecimento fiscal e cadastral utiliza cache local e serviços externos quando há QR Code, chave de acesso ou CNPJ identificado, preservando o fluxo local em caso de falha externa.	Dados fiscais ou cadastrais reaproveitados quando disponíveis; ausência de conectividade ou falha externa não impede a geração da prévia.	RF05, RF24, RNF01, RNF10
CT17	*Widget*	Verificar que a interface de categorias permite criar, editar, excluir e reordenar categorias, respeitando restrições quando houver comprovantes associados.	Categorias atualizadas e refletidas na interface; exclusões bloqueadas ou confirmadas conforme associação existente.	RF25
CT18	*Widget*	Verificar que as telas de configurações permitem alterar tema visual, limite de armazenamento, senha de *backup*, autenticação local e reinício do tutorial.	Preferências persistidas e refletidas na interface sem exposição de segredos sensíveis.	RF01, RF26, RNF03, RNF20
CT19	Validação manual	Verificar o fluxo de remoção de *backup* em nuvem, limpeza do histórico local de operações e reporte de problema a partir da aplicação.	*Backup* removido quando a senha é validada, histórico local limpo sem apagar comprovantes e relatório de problema preparado para envio pelo usuário.	RF27, RF28

Automação e Integração Contínua

Conforme apresentado na Seção 1.3, a automação dos testes utiliza o GitHub Actions como

mecanismo de integração contínua. No projeto de testes, esse fluxo executa a suíte unitária e de *widgets* quando há alterações no diretório da aplicação Flutter, apoiando a verificação recorrente das regras de negócio e dos componentes de interface.

A automação também gera o pacote Android e as evidências de cobertura disponibilizadas como artefatos da execução. Os fluxos que dependem de recursos nativos do dispositivo ou de serviços externos são complementados por validações manuais em dispositivos Android físicos reais.

O resultado consolidado da execução no GitHub Actions é apresentado na Figura 11. A evidência demonstra a execução bem-sucedida do fluxo de integração contínua e a geração de três artefatos: o pacote Android, o relatório de cobertura em HTML e o arquivo LCOV. O detalhamento das etapas internas e da execução dos testes com cobertura é apresentado no Apêndice G.

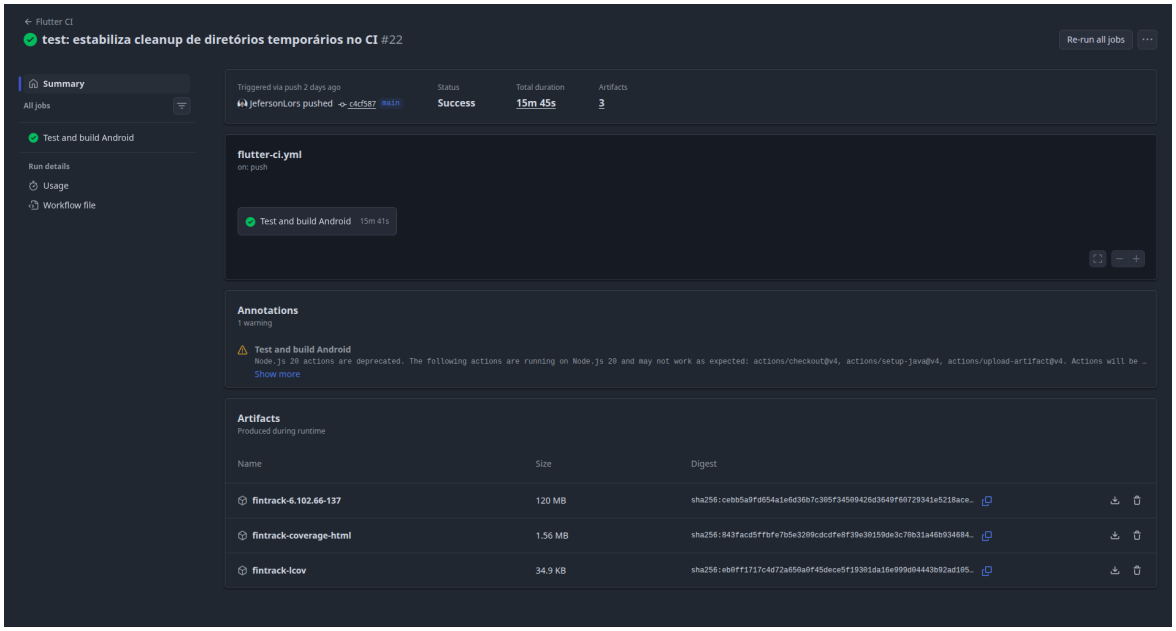


Figura 11: Resultado da execução do fluxo de integração contínua do FinTrack no GitHub Actions.

O relatório LCOV registrou cobertura de 97,1% das linhas instrumentadas, com 14.004 de 14.415 linhas cobertas, conforme apresentado na Figura 12.

LCOV - code coverage report

			Coverage	Total	Hit
Current view:	top level	Lines:	97.1 %	14415	14004
Test:	lcov.info	Functions:	-	0	0
Test Date:	2026-06-06 14:15:45				

Directory	Line Coverage			Function Coverage		
	Rate	Total	Hit	Rate	Total	Hit
lib/application/backup/lib/application/backup	95.7 %	398	381	-		
lib/application/categories/lib/application/categories	100.0 %	34	34	-		
lib/application/config/lib/application/config	100.0 %	410	410	-		
lib/application/configuration/lib/application/configuration	97.1 %	70	68	-		
lib/application/ocr/lib/application/ocr	96.4 %	991	955	-		
lib/application/policies/lib/application/policies	100.0 %	47	47	-		
lib/application/receipts/batch/lib/application/receipts/batch	99.2 %	252	250	-		
lib/application/receipts/lib/application/receipts	94.6 %	1008	954	-		
lib/application/receipts/semantic/lib/application/receipts/semantic	95.9 %	171	164	-		
lib/application/use_cases/lib/application/use_cases	100.0 %	55	55	-		
lib/bootstrap/lib/bootstrap	100.0 %	45	45	-		
lib/domain/entities/lib/domain/entities	99.5 %	191	190	-		
lib/domain/exceptions/lib/domain/exceptions	100.0 %	10	10	-		
lib/domain/infrastructure/lib/domain/infrastructure	97.4 %	38	37	-		
lib/domain/repositories/lib/domain/repositories	100.0 %	50	50	-		
lib/domain/utils/lib/domain/utils	96.1 %	282	271	-		
lib/domain/value_objects/lib/domain/value_objects	98.3 %	175	172	-		
lib/infrastructure/backup/lib/infrastructure/backup	100.0 %	7	7	-		
lib/infrastructure/category/lib/infrastructure/category	100.0 %	158	158	-		
lib/infrastructure/cloud/lib/infrastructure/cloud	98.9 %	189	187	-		
lib/infrastructure/company/lib/infrastructure/company	100.0 %	70	70	-		
lib/infrastructure/cryptography/lib/infrastructure/cryptography	96.7 %	60	58	-		
lib/infrastructure/database/lib/infrastructure/database	97.0 %	601	583	-		
lib/infrastructure/database/mappers/lib/infrastructure/database/mappers	100.0 %	86	86	-		
lib/infrastructure/database/persistence/lib/infrastructure/database/persistence	100.0 %	116	116	-		
lib/infrastructure/database/queries/lib/infrastructure/database/queries	98.5 %	204	201	-		
lib/infrastructure/database/repositories/lib/infrastructure/database/repositories	99.2 %	260	258	-		
lib/infrastructure/diagnostics/lib/infrastructure/diagnostics	97.1 %	104	101	-		
lib/infrastructure/embedding/lib/infrastructure/embedding	99.0 %	208	206	-		
lib/infrastructure/fiscal/lib/infrastructure/fiscal	94.3 %	192	181	-		
lib/infrastructure/image/lib/infrastructure/image	96.6 %	409	395	-		
lib/infrastructure/ocr/lib/infrastructure/ocr	100.0 %	117	117	-		
lib/infrastructure/receipts/lib/infrastructure/receipts	100.0 %	5	5	-		
lib/infrastructure/scanner/lib/infrastructure/scanner	100.0 %	21	21	-		
lib/infrastructure/security/lib/infrastructure/security	100.0 %	22	22	-		
lib/infrastructure/semantic/lib/infrastructure/semantic	100.0 %	5	5	-		
lib/lib	96.6 %	145	140	-		
lib/presentation/authentication/lib/presentation/authentication	97.2 %	360	350	-		
lib/presentation/backup/controllers/lib/presentation/backup/controllers	100.0 %	25	25	-		
lib/presentation/backup/pages/lib/presentation/backup/pages	94.7 %	206	195	-		
lib/presentation/backup/widgets/lib/presentation/backup/widgets	99.3 %	441	438	-		
lib/presentation/camera/lib/presentation/camera	95.1 %	102	97	-		
lib/presentation/categories/controllers/lib/presentation/categories/controllers	100.0 %	93	93	-		
lib/presentation/categories/pages/lib/presentation/categories/pages	94.9 %	256	243	-		
lib/presentation/categories/widgets/lib/presentation/categories/widgets	98.0 %	450	441	-		
lib/presentation/configuration/controllers/lib/presentation/configuration/controllers	100.0 %	62	62	-		
lib/presentation/configuration/lib/presentation/configuration	98.3 %	121	119	-		
lib/presentation/configuration/pages/lib/presentation/configuration/pages	95.4 %	216	206	-		
lib/presentation/configuration/widgets/lib/presentation/configuration/widgets	98.2 %	554	544	-		
lib/presentation/onboarding/lib/presentation/onboarding	97.6 %	82	80	-		
lib/presentation/receipts/controllers/lib/presentation/receipts/controllers	98.3 %	524	515	-		
lib/presentation/receipts/lib/presentation/receipts	96.6 %	350	338	-		
lib/presentation/receipts/pages/lib/presentation/receipts/pages	93.8 %	795	746	-		
lib/presentation/receipts/widgets/lib/presentation/receipts/widgets	96.9 %	1397	1353	-		
lib/presentation/reports/lib/presentation/reports	93.1 %	131	122	-		
lib/presentation/reports/widgets/lib/presentation/reports/widgets	99.4 %	171	170	-		
lib/presentation/shell/lib/presentation/shell	100.0 %	163	163	-		
lib/presentation/shell/widgets/lib/presentation/shell/widgets	97.0 %	132	128	-		
lib/presentation/theme/lib/presentation/theme	96.2 %	104	100	-		
lib/presentation/widgets/lib/presentation/widgets	98.3 %	474	466	-		

Figura 12: Relatório HTML de cobertura de testes gerado por LCOV.

Essa métrica não substitui a análise qualitativa dos cenários testados, mas funciona como indicador complementar de abrangência da suíte automatizada, especialmente nas camadas de Aplicação e Apresentação, onde se concentram as regras de negócio e os fluxos de interface exercitáveis sem dependência direta de recursos nativos do Android ou de dispositivos físicos.

Essa estrutura permite que a base de código evolua de forma controlada, assegurando que as regras de negócio e os componentes visuais cobertos pela suíte automatizada sejam verificados de maneira recorrente. As validações manuais complementam essa cobertura nos pontos em que o comportamento depende do ambiente físico de execução, como permissões Android, scanner de documentos, OCR, QR Code, armazenamento local, autenticação, desempenho e integração com serviços externos.

5 Implantação

Esta seção descreve como o FinTrack é distribuído, instalado e executado nos dispositivos dos usuários. Por se tratar de uma aplicação *standalone* — sem dependência de servidores próprios ou infraestrutura de *backend* —, o processo de implantação resume-se à instalação do pacote no dispositivo móvel Android, sem etapas adicionais de configuração de ambiente, provisionamento de serviços ou migração de dados.

Apesar da simplicidade do processo de instalação, a execução adequada da solução depende dos recursos disponíveis no próprio dispositivo, pois os principais fluxos de processamento e persistência são realizados localmente. Por isso, o projeto de implantação define os requisitos mínimos de compatibilidade, armazenamento, memória, processamento,

conectividade e permissões necessários ao funcionamento do FinTrack.

5.1 Projeto de implantação

O FinTrack é distribuído como um pacote Android no formato `.apk` e pode ser instalado em dispositivos com Android 8.0, correspondente ao nível de API 26, ou superior, versão mínima de compatibilidade definida no projeto. O funcionamento de recursos como biometria, câmera, scanner de documentos, OCR, leitura de QR Code, acesso a arquivos e notificações depende das capacidades do aparelho e das permissões concedidas pelo usuário.

Em termos de recursos locais, o espaço em disco necessário deve considerar o pacote Android, o modelo DistilUSE embarcado, as bibliotecas nativas e os dados criados durante o uso, como banco SQLite, arquivos associados aos registros de comprovantes, arquivos temporários de OCR e dados de *backup*. Por esse motivo, recomenda-se pelo menos 210 MB livres para instalação e uso inicial, mantendo-se 500 MB ou mais para uso recorrente.

Quanto à memória e ao processamento, como o escaneamento de documentos, o OCR, a leitura de QR Code, a geração de *embeddings* e a persistência ocorrem localmente no dispositivo, o desempenho dessas operações depende da memória de acesso aleatório (RAM) disponível e da capacidade de processamento do aparelho. Recomenda-se, portanto, no mínimo 3 GB de RAM e processador móvel de 64 bits com quatro núcleos e frequência máxima aproximada de 1,8 GHz ou superior para OCR, pré-processamento de imagens e geração local de *embeddings*, sendo preferível 4 GB de RAM ou mais e processador de oito núcleos com frequência máxima de 2,0 GHz ou superior para bases maiores.

No que se refere à necessidade de conexão com a internet, a conectividade é exigida apenas para funcionalidades externas, como vínculo com uma conta de nuvem, *backup* e restauração em armazenamento remoto, e para consultas automáticas de dados cadastrais por CNPJ ou enriquecimento fiscal a partir de portais públicos quando houver dados identificáveis. De modo semelhante, a conta de nuvem e a biometria não constituem requisitos obrigatórios de instalação: a primeira é necessária apenas para o *backup* em nuvem, enquanto a segunda é utilizada somente quando o usuário habilita autenticação local biométrica e o dispositivo oferece suporte a esse recurso.

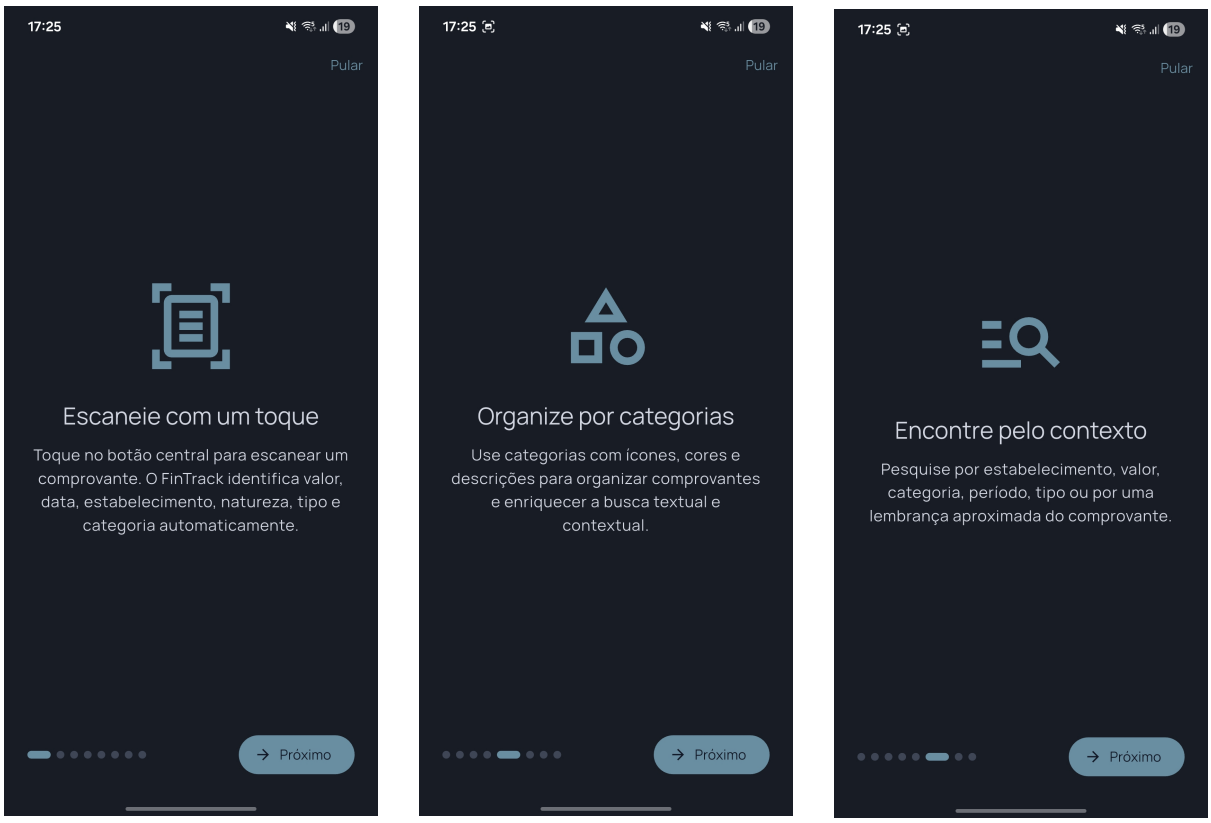
Por último, após a instalação, os fluxos que dependem de recursos do Android solicitam as permissões correspondentes, como câmera, autenticação biométrica, internet e notificações. Também é contemplada a integração com o compartilhamento nativo do sistema operacional, permitindo que o FinTrack receba imagens e documentos PDF enviados por outros aplicativos. A organização desses componentes de implantação é apresentada no Apêndice B.

6 Manual do Usuário

Esta seção apresenta as principais funcionalidades disponíveis no FinTrack sob a perspectiva do usuário final. O objetivo é orientar o uso da aplicação, desde o primeiro acesso até o cadastro, organização, recuperação, proteção e compartilhamento de registros de comprovantes financeiros.

Tutorial

Ao abrir o FinTrack pela primeira vez, o usuário é conduzido por um fluxo inicial de apresentação da aplicação. Esse fluxo apresenta os principais recursos e orienta sobre interações como captura, busca, importação, categorias e *backup*. Ao concluir essa introdução, o usuário segue para a tela principal da aplicação. A Figura 13 mostra as telas usadas nessa apresentação inicial.



Captura | Categorias | Busca contextual

Figura 13: Telas do tutorial inicial do FinTrack.

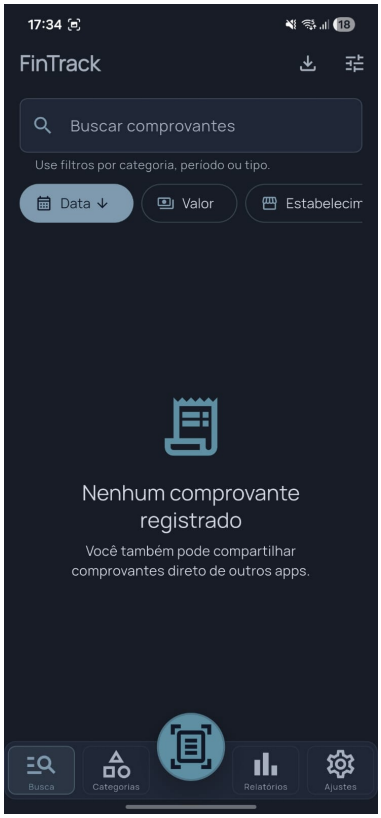
Tela de Busca e navegação principal

A Tela de Busca, acessada pelo item “Busca” da barra inferior, é a tela principal do FinTrack e funciona como ponto de partida do aplicativo. Na parte superior, a barra com o título “FinTrack” reúne as ações principais da listagem. O botão “Selecionar comprovantes”, visível apenas quando há registros salvos, inicia a seleção múltipla; a mesma seleção também pode ser acionada mantendo pressionado um cartão de registro na lista. O botão “Importar comprovante”

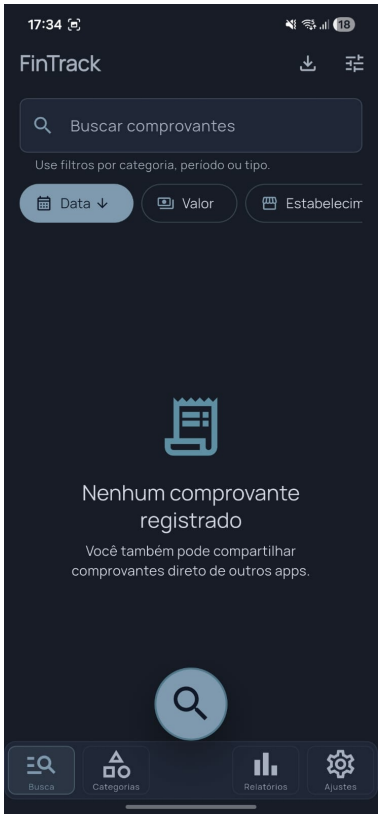
abre o fluxo de importação de arquivos de comprovante já existentes no dispositivo. O botão “Filtros” abre o painel de filtragem por categoria, período, tipo de documento e natureza financeira.

Logo abaixo da barra superior, o campo “Buscar comprovantes” permite pesquisar registros por termos extraídos do comprovante ou por contexto. Abaixo do campo de busca, a barra de ordenação apresenta os botões “Data”, “Valor” e “Estabelecimento”; ao selecionar uma dessas opções, a lista é reorganizada pelo critério escolhido, com indicação de ordem “Crescente” ou “Decrescente”. Quando filtros estão ativos, a tela também exibe um indicador do filtro aplicado antes da lista de registros.

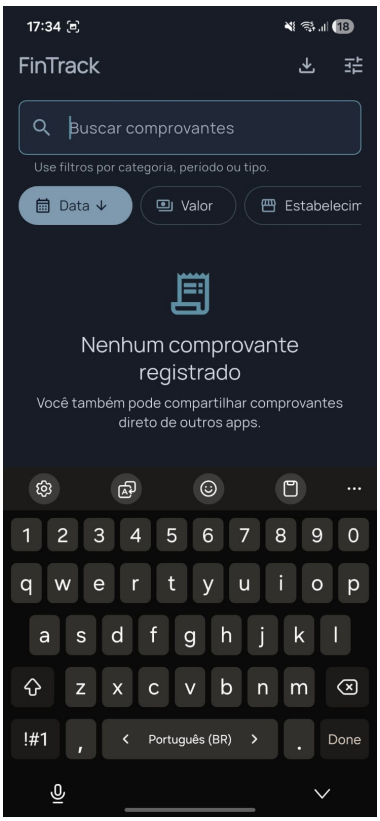
A navegação entre as áreas principais ocorre pela barra inferior persistente. Seus itens são “Busca”, “Categorias”, “Relatórios” e “Ajustes”. No centro dessa barra fica o botão circular “Escanear comprovante”. Como a barra inferior permanece disponível nas telas principais, o usuário pode pressionar esse botão e arrastá-lo para cima a partir de qualquer uma dessas áreas. Com esse gesto, o FinTrack troca o destaque do ícone de scanner para a lupa e direciona a interação para a busca de registros, mantendo a Tela de Busca como referência principal de navegação. A Figura 14 apresenta a tela principal, o gesto no botão central e o campo de busca ativo.



Tela de Busca



Gesto no botão de escaneamento



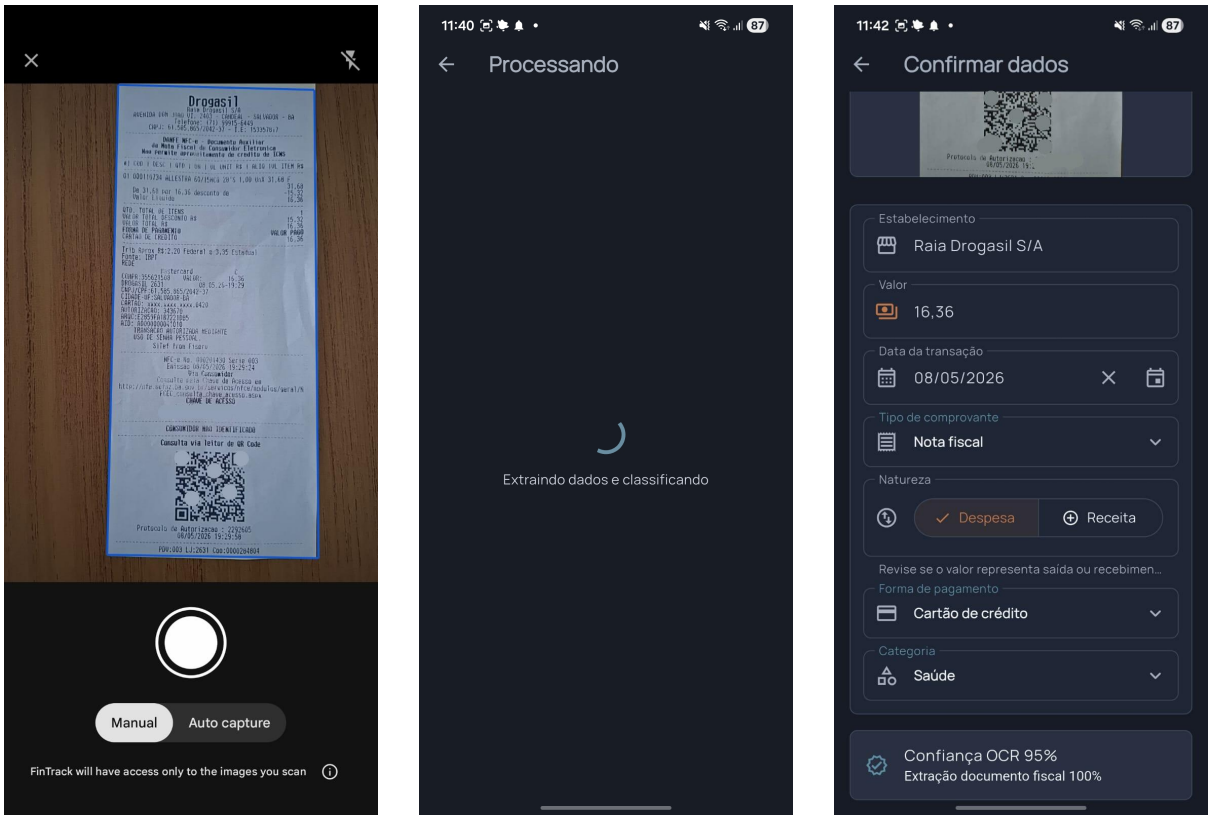
Campo “Buscar comprovantes”

Figura 14: Tela de Busca como tela principal, gesto de busca a partir do botão “Escanear comprovante” e campo de pesquisa.

Captura por scanner

Para registrar um comprovante físico, o usuário deve tocar no botão circular “Escanear comprovante”, localizado no centro da barra inferior. Como essa barra permanece disponível nas telas principais, o escaneamento pode ser iniciado a partir de “Busca”, “Categorias”, “Relatórios” ou “Ajustes”. Esse botão abre o scanner de documentos do dispositivo para capturar uma imagem digitalizada do documento físico. Caso o scanner nativo não possa ser aberto, o FinTrack apresenta o aviso “Scanner indisponível” e oferece a alternativa “Usar câmera”, que permite capturar a imagem pela câmera simples do aparelho.

Após a captura, o aplicativo exibe a tela “Processando”. Nessa etapa, o FinTrack apresenta mensagens como “Lendo comprovante” e “Extraindo dados e classificando”, enquanto executa OCR, leitura de códigos visuais quando disponíveis, extração de campos estruturados, enriquecimento automático por dados cadastrais ou fiscais quando aplicável e sugestão de categoria. Se o processamento falhar, a tela informa que não foi possível processar a imagem capturada e oferece a ação “Tentar novamente”. A Figura 15 resume a sequência entre scanner, processamento e a tela de confirmação, detalhada logo após a figura.



Scanner de documentos | Tela “Processando” | Tela “Confirmar dados”

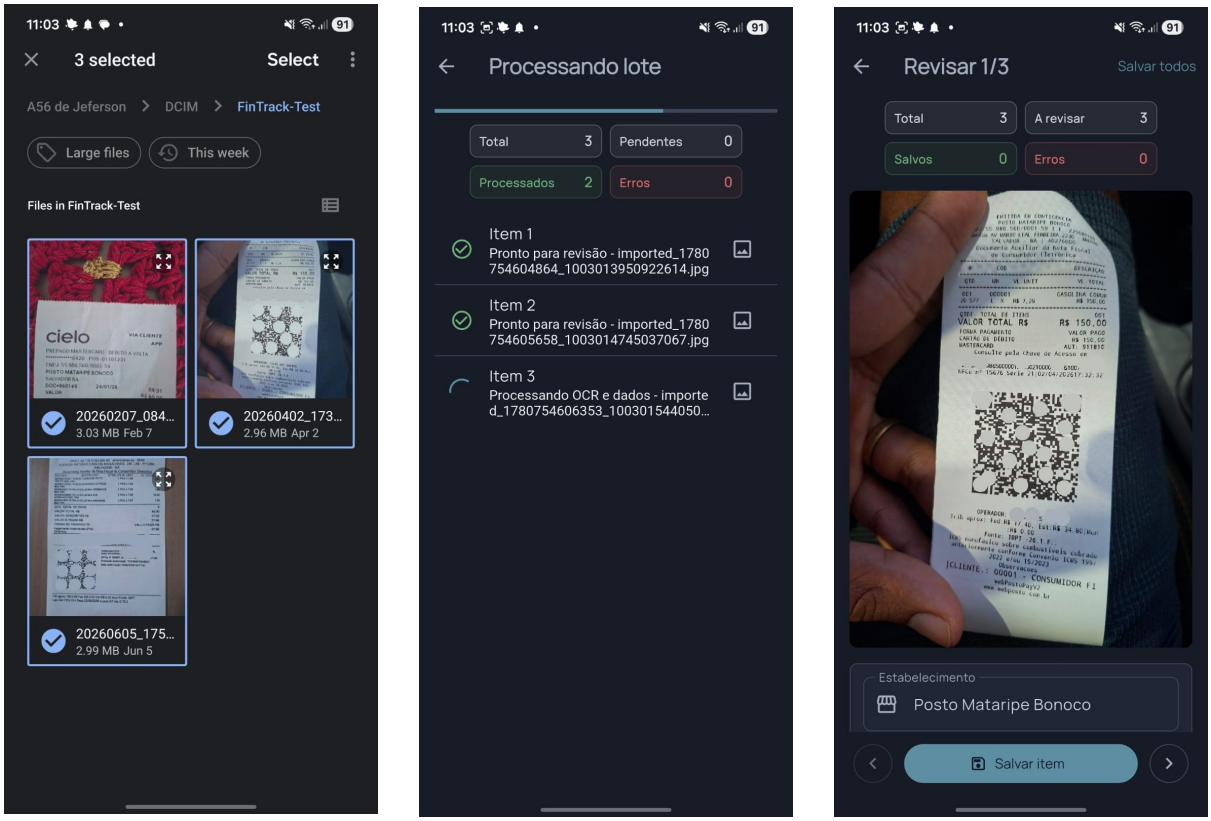
Figura 15: Fluxo de captura acionado pela barra inferior, processamento automático e revisão antes de salvar.

Quando o processamento é concluído, o usuário é encaminhado para a tela “Confirmar dados”. Nela, é possível revisar e editar os campos “Estabelecimento”, “Valor”, “Data da transação”, “Tipo de comprovante”, “Natureza”, “Forma de pagamento” e “Categoria”. A tela também pode exibir alertas quando a leitura automática ou a extração de dados ficarem incertas. Após revisar as informações, o usuário confirma o registro pelo botão “Salvar”. Ao receber um

arquivo de comprovante compartilhado a partir de outro aplicativo, o fluxo é semelhante, mas começa pelo arquivo recebido e segue diretamente para processamento e revisão, sem passar pelo scanner.

Importação de arquivos

O fluxo de importação começa na Tela de Busca, pelo botão “Importar comprovante”, localizado na barra superior ao lado das ações de seleção e filtros. Ao acionar esse botão, o FinTrack abre o seletor de arquivos do dispositivo; se nenhum arquivo for recebido, a aplicação exibe a mensagem “Nenhum arquivo foi recebido para importação”. O mesmo fluxo visual também é utilizado quando arquivos de comprovante são enviados ao FinTrack por compartilhamento a partir de outro aplicativo. A Figura 16 mostra o início da importação em lote, desde a seleção dos arquivos até a primeira tela de revisão, explicada nos parágrafos seguintes.



Seleção de arquivos | Processando lote | Revisar 1/N

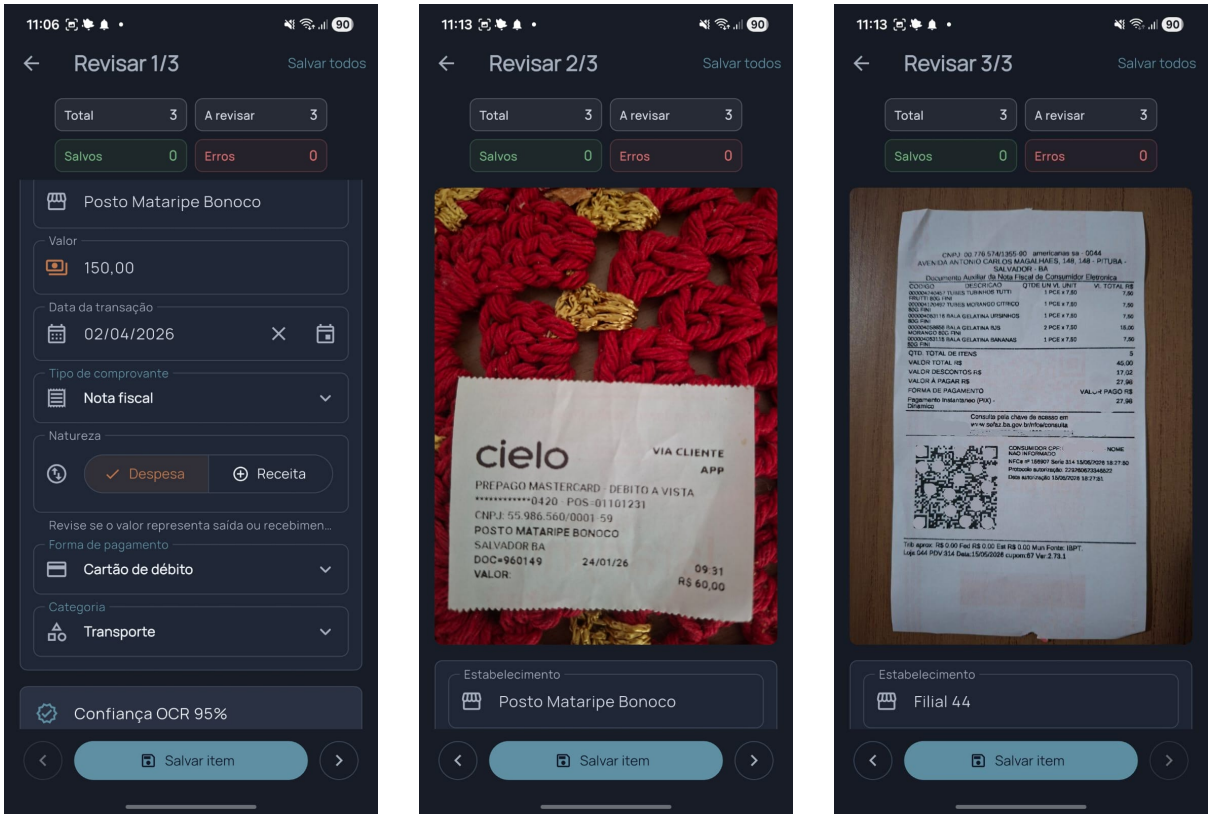
Figura 16: Seleção de arquivos, processamento em lote e abertura da revisão dos registros importados.

Quando apenas um arquivo é selecionado ou recebido por compartilhamento, as telas seguintes são as mesmas já apresentadas no fluxo de captura: primeiro a tela de processamento individual e, em seguida, a tela “Confirmar dados”, na qual o usuário revisa as informações extraídas antes de salvar o registro. Quando mais de um arquivo é selecionado ou compartilhado com o FinTrack, a aplicação abre a tela “Processando lote”, cria uma sessão de importação e acompanha cada item separadamente.

Essa tela apresenta o progresso geral por meio dos indicadores “Total”, “Pendentes”, “Processados” e “Erros”, além de listar cada item com estados como “Pendente”, “Processando OCR e dados”, “Pronto para revisão”, “Falha no processamento” ou “Salvo”. Durante essa etapa, a aplicação pode executar OCR, leitura de QR Code, extração de campos estruturados, enriquecimento cadastral ou fiscal quando aplicável e sugestão de categoria.

Ao final do processamento em lote, os itens prontos são abertos na tela de revisão, identificada pelo título “Revisar 1/N”, conforme a posição do item no conjunto importado. Nessa tela, o usuário revisa a pré-visualização do arquivo e edita os campos “Estabelecimento”, “Valor”, “Data da transação”, “Tipo de comprovante”, “Natureza”, “Forma de pagamento” e “Categoria”. A navegação entre os arquivos é feita pelos botões “Anterior” e “Próximo”, e o registro atual pode ser confirmado pelo botão “Salvar item”.

Quando desejar confirmar todos os itens pendentes de uma vez, o usuário pode utilizar a ação “Salvar todos”. Itens com falha permanecem disponíveis para análise e oferecem as ações “Excluir item” e “Reprocessar”. A Figura 17 detalha a navegação entre itens e os campos editáveis da revisão em lote.



Dados editáveis | Revisão do item 2 | Revisão do item 3

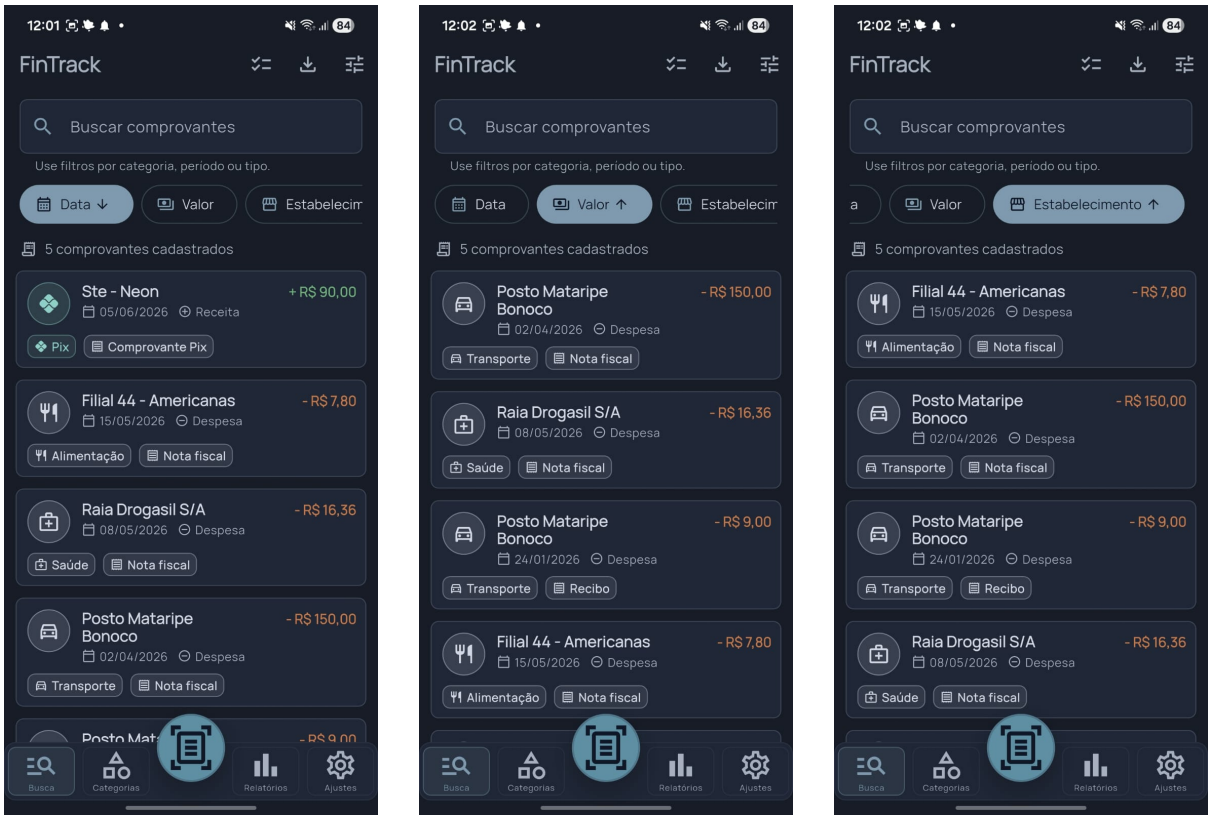
Figura 17: Edição dos dados extraídos e navegação entre itens na revisão da importação em lote.

Lista e detalhes de registros

Os registros salvos ficam disponíveis na lista da Tela de Busca. Cada cartão apresenta o estabelecimento identificado, a data da transação, a natureza financeira como “Despesa” ou

“Receita”, o valor com sinal negativo ou positivo, a categoria associada ou o marcador “Sem categoria” e o tipo do registro, como “Nota fiscal”, “Recibo”, “Comprovante Pix” ou “Outros”. Acima da lista, o FinTrack informa a quantidade de itens carregados, encontrados, filtrados ou cadastrados, conforme o estado atual da busca e dos filtros. Para manter a navegação fluida em bases maiores, os resultados são carregados em páginas de até 30 registros; ao rolar a lista para baixo, o aplicativo busca automaticamente a próxima página por meio de carregamento contínuo.

A ordenação é controlada pelos botões “Data”, “Valor” e “Estabelecimento”, exibidos logo abaixo do campo “Buscar comprovantes”. Ao tocar no critério ativo, a direção alterna entre “Crescente” e “Decrescente”; ao tocar em outro critério, a lista passa a ser ordenada por esse campo. A ordenação por valor considera a natureza do registro, diferenciando despesas e receitas. A Figura 18 apresenta a mesma lista organizada pelos três critérios disponíveis.



Ordenação: Data | Ordenação: Valor | Ordenação: Estabelecimento

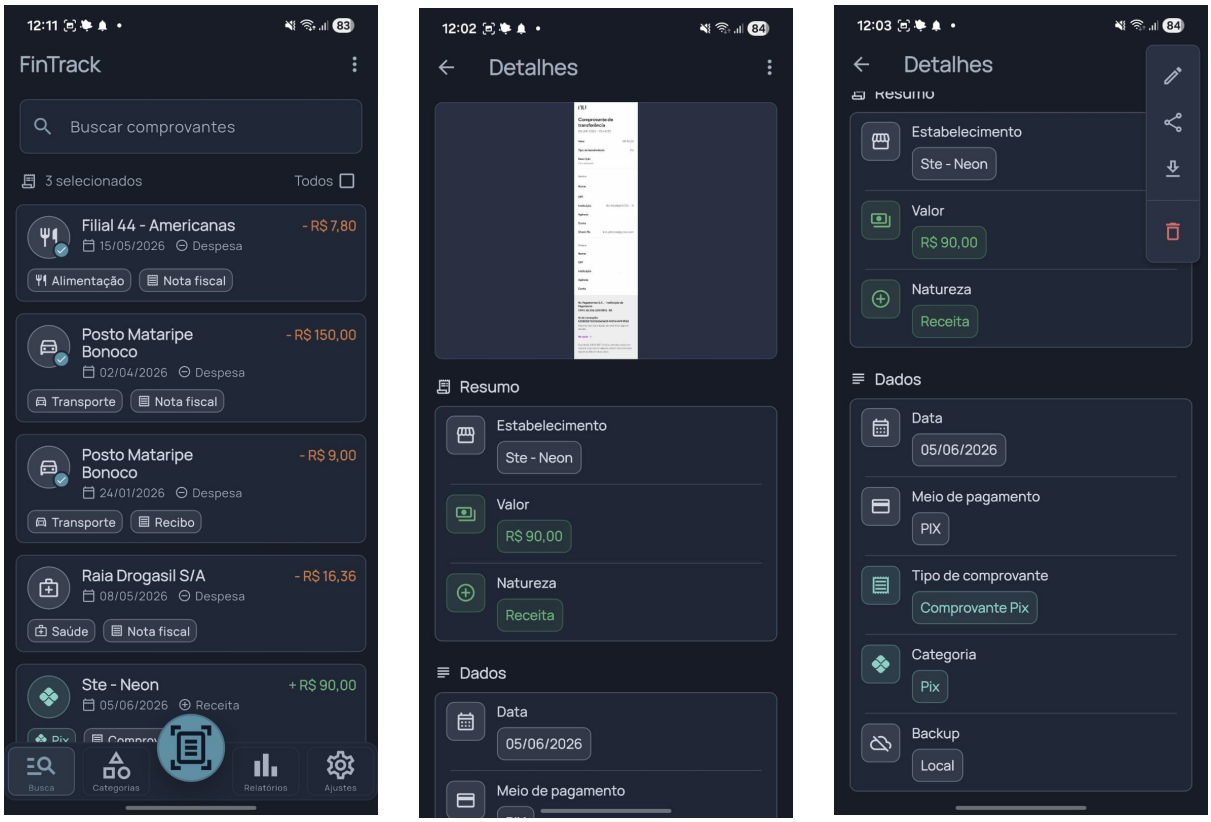
Figura 18: Critérios de ordenação disponíveis na Tela de Busca: data, valor e estabelecimento.

Para abrir um registro, o usuário toca no cartão correspondente. A tela “Detalhes” apresenta a pré-visualização do arquivo associado ao registro e organiza as informações nas seções “Resumo” e “Dados”. Em “Resumo”, são exibidos “Estabelecimento”, “Valor” e “Natureza”. Em “Dados”, são exibidos “Data”, “Meio de pagamento”, “Tipo de comprovante”, “Categoria” e o estado de “Backup”, como “Local” ou “Sincronizado”. Quando a pré-visualização corresponde a uma imagem, o toque sobre ela abre a visualização ampliada “Imagem do comprovante”.

Quando for necessário agir sobre mais de um registro, o usuário pode usar o botão

“Selecionar comprovantes”, localizado na barra superior, ou manter pressionado um cartão da lista. No modo de seleção, o cabeçalho passa a indicar a quantidade de itens “selecionado” ou “selecionados” e oferece o controle “Todos” para marcar ou desmarcar os registros visíveis. O menu “Mais opções” reúne as ações em lote “Compartilhar selecionados”, “Salvar no dispositivo” e “Excluir selecionados”. As duas primeiras atuam sobre os arquivos associados aos registros; a exclusão remove os registros, seus dados e os dados de busca inteligente após confirmação.

Na tela “Detalhes”, o menu “Mais opções” oferece as ações “Editar”, “Compartilhar”, “Salvar no dispositivo” e “Excluir”. A ação “Editar” reabre a tela “Confirmar dados” para ajuste das informações extraídas. As ações “Compartilhar” e “Salvar no dispositivo” operam sobre o arquivo associado ao registro, enquanto “Excluir” abre o diálogo “Excluir comprovante?” antes de remover o registro da base local. A Figura 19 mostra o modo de seleção, a tela de detalhes e o menu de ações de um registro.



Modo de seleção | Tela “Detalhes” | Menu “Mais opções”

Figura 19: Seleção múltipla, detalhamento e ações sobre registros salvos no FinTrack.

Categorias

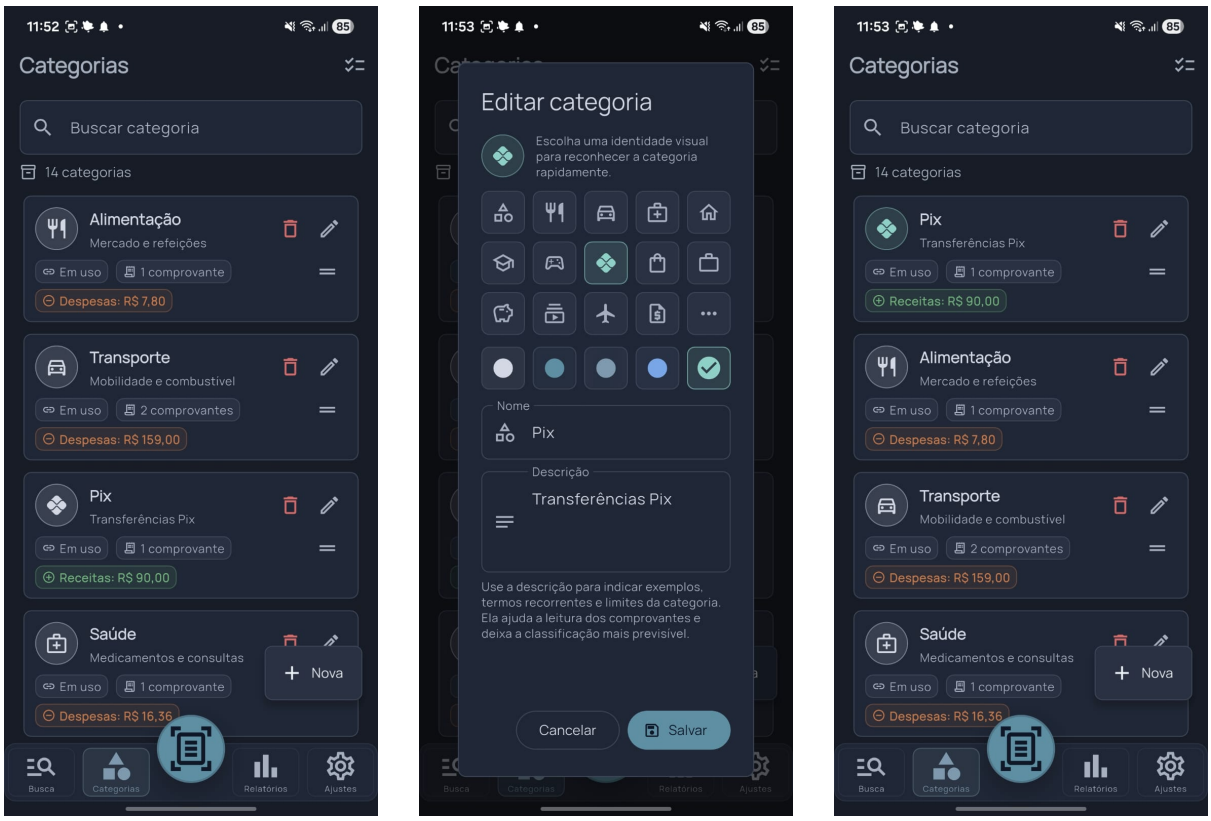
O item “Categorias” da barra inferior abre a tela de gerenciamento das categorias usadas para organizar os registros. Cada categoria possui nome, descrição, ícone e cor associados. O nome e a descrição dão significado à categoria; o ícone e a cor funcionam como identificação visual, aparecendo nos cartões de categoria, nos registros classificados, nos filtros e nos relatórios. No topo, a tela apresenta o título “Categorias” e, quando há categorias cadastradas, o botão “Selecionar categorias”, usado para iniciar a seleção múltipla. A mesma seleção pode

ser acionada mantendo pressionado um cartão de categoria.

Abaixo do cabeçalho, o campo “Buscar categoria” permite localizar categorias pelo nome ou pela descrição. Cada cartão de categoria exibe nome, descrição, ícone, cor, estado de uso e estatísticas associadas, como quantidade de registros, despesas e receitas. Ao tocar em uma categoria fora do modo de seleção, o FinTrack abre a lista de registros filtrada por ela, com o indicador “Categoria: nome da categoria”. No próprio cartão, o botão “Renomear” abre a edição da categoria e o botão “Excluir” inicia a tentativa de remoção. Quando a categoria está associada a registros, a aplicação exibe o aviso “Categoria em uso” e impede a exclusão até que os registros sejam desassociados.

Para criar uma categoria, o usuário aciona o botão flutuante “Nova”. O FinTrack abre o diálogo “Nova categoria”, no qual é possível escolher ícone e cor, informar “Nome” e preencher “Descrição”. A descrição deve ser preenchida com cuidado, pois ela não serve apenas como observação: esse texto ajuda a explicar o escopo da categoria para a busca contextual e para as sugestões automáticas.

Uma boa descrição pode incluir exemplos, termos recorrentes e limites de uso, como os tipos de estabelecimento, produtos, serviços ou situações que devem ser associados à categoria. Ao editar uma categoria existente, o diálogo passa a ser “Editar categoria” e mantém os mesmos campos. A confirmação é feita pelo botão “Salvar”. A Figura 20 reúne a tela de categorias, o diálogo de cadastro ou edição e a ação de reordenação.



Tela “Categorias” Diálogo de categoria Arrastar para reordenar

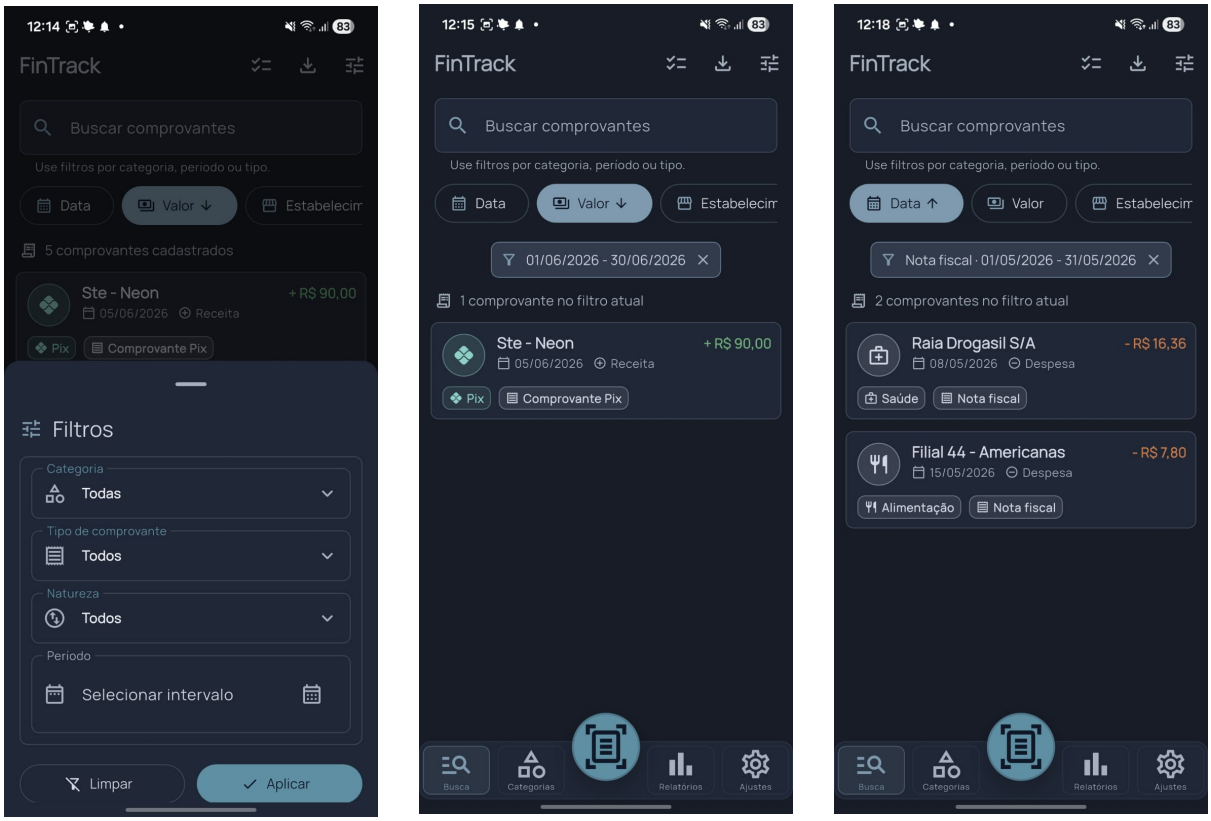
Figura 20: Busca, criação, edição, exclusão e reordenação de categorias no FinTrack.

As categorias também podem ser reorganizadas diretamente na lista. Quando a busca não está ativa e a tela não está em modo de seleção, o usuário pode usar o controle “Arrastar para reordenar” para alterar a posição das categorias. Em modo de seleção, iniciado pelo botão “Selecionar categorias” ou pelo toque prolongado em um cartão, é possível marcar uma ou mais categorias e usar a ação “Excluir selecionadas”; categorias sem registros associados são removidas após confirmação, enquanto categorias em uso geram o aviso “Categorias em uso”.

Durante o cadastro ou edição de um registro, a aplicação pode sugerir uma categoria com base em preferências já aprendidas, como CNPJ, estabelecimento e itens recorrentes. Quando essas informações não são suficientes, a classificação pode utilizar similaridade semântica entre o conteúdo extraído e as categorias existentes. Nesse caso, nomes e descrições mais claros aumentam a chance de a categoria representar corretamente o contexto do registro, além de enriquecer a busca textual e contextual. A categoria sugerida é opcional e pode ser alterada pelo usuário na tela de confirmação ou revisão do registro.

Filtros de registros

Os filtros são acessados na Tela de Busca pelo botão “Filtros”, localizado na barra superior. Ao acionar essa opção, o FinTrack abre um painel inferior com quatro critérios: “Categoria”, “Tipo de comprovante”, “Natureza” e “Período”. Esses critérios podem ser usados separadamente ou combinados para reduzir a lista aos registros que interessam naquele momento. A Figura 21 mostra o painel de filtros e exemplos iniciais de aplicação por período e tipo de comprovante.



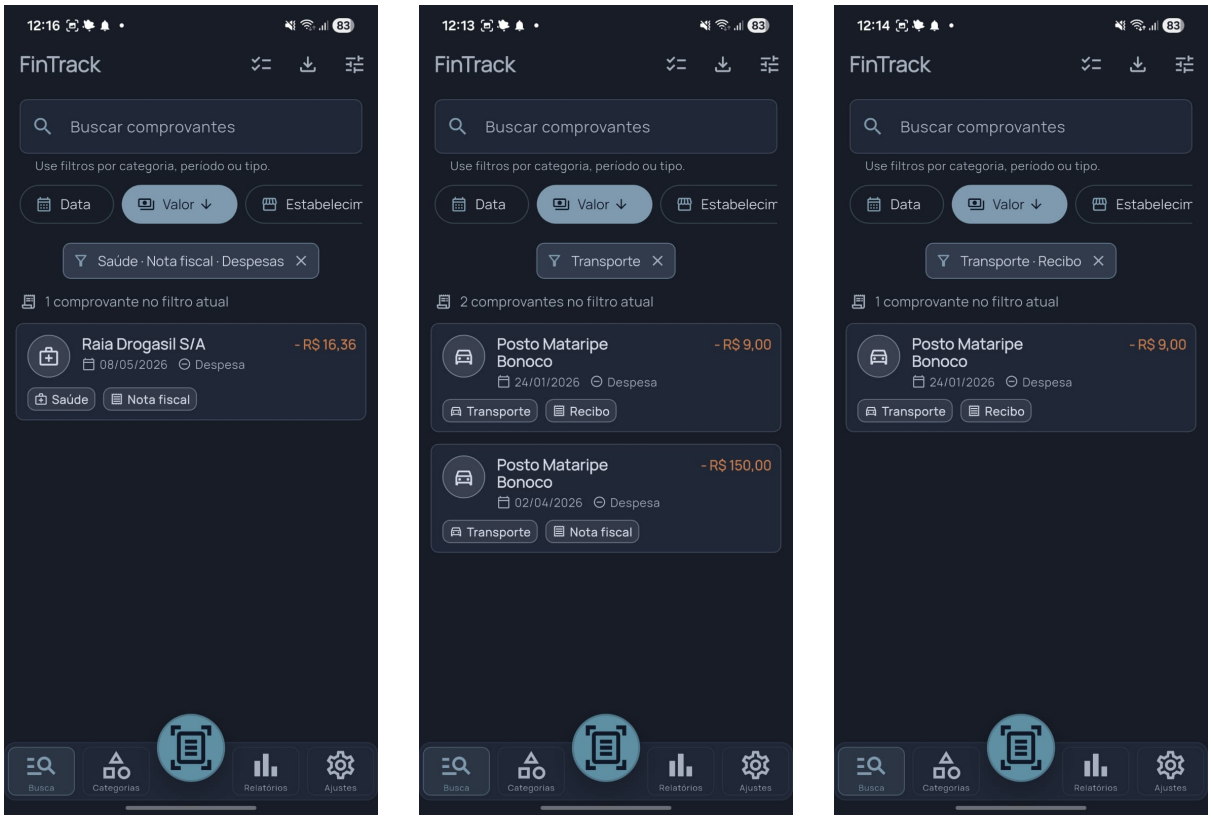
Painel “Filtros” | Filtro “Período” aplicado | Filtros “Tipo” e “Período”

Figura 21: Painel de filtros e aplicação inicial de período e tipo de comprovante na Tela de Busca.

No campo “Categoria”, o usuário pode manter a opção “Todas” ou escolher uma categoria cadastrada, como saúde, transporte ou outra organização definida no aplicativo. No campo “Tipo de comprovante”, é possível manter “Todos” ou selecionar tipos como “Nota fiscal”, “Recibo”, “Comprovante Pix” ou “Outros”. Em “Natureza”, o painel permite filtrar por “Todos”, “Despesas” ou “Receitas”. Em “Período”, o controle “Selecionar intervalo” abre o seletor de datas, com os campos “Início” e “Fim”, e permite aplicar um recorte temporal aos registros.

Após configurar os critérios, o usuário confirma a filtragem pelo botão “Aplicar”. O botão “Limpar” remove todos os critérios do painel. Quando há filtros ativos, a Tela de Busca exibe um marcador acima da lista com o resumo dos critérios aplicados, por exemplo com o nome da categoria, o tipo de comprovante, a natureza e o intervalo de datas. Esse marcador também pode ser removido para voltar à listagem sem filtros.

Os filtros atuam sobre os registros de comprovantes salvos na base local, não sobre os arquivos externos originais. Assim, um arquivo importado ou uma imagem digitalizada passa a aparecer nos filtros de acordo com os dados revisados e salvos no registro, como categoria, tipo, natureza e data da transação. A Figura 22 exemplifica combinações de categoria e tipo de comprovante já aplicadas à listagem.



Filtros “Categoria” e “Tipo” | Filtro “Categoria” aplicado | Filtros “Categoria” e “Recibo”

Figura 22: Exemplos de filtros combinados por categoria e tipo de comprovante.

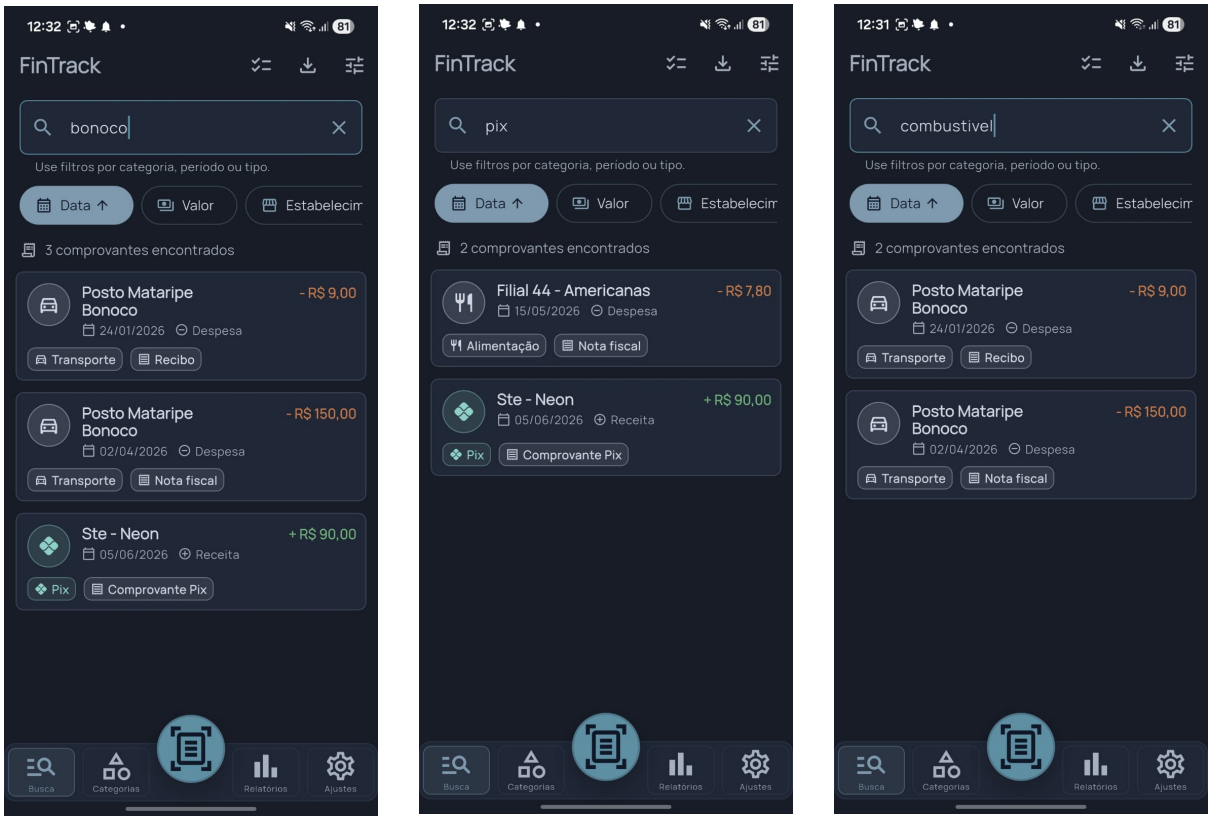
Busca textual e contextual

Na Tela de Busca, o campo “Buscar comprovantes” permite localizar registros por texto ou por contexto. O campo fica abaixo da barra superior e pode ser acionado diretamente pelo

toque ou pelo gesto de pressionar o botão central “Escanear comprovante” e arrastá-lo para cima a partir de qualquer uma das telas principais. Quando há texto digitado, o botão “Limpar busca” aparece no próprio campo para remover a consulta e atualizar novamente a lista.

A busca textual considera os dados estruturados do registro, como estabelecimento, categoria, meio de pagamento, tipo de comprovante, natureza financeira, valor e data da transação. Também considera o texto identificado por OCR, isto é, o conteúdo lido automaticamente a partir da imagem digitalizada ou do arquivo importado. Para tornar a consulta mais tolerante, a comparação textual normaliza letras maiúsculas, minúsculas e acentos.

A busca contextual utiliza similaridade semântica para encontrar registros pelo significado da consulta, mesmo quando as palavras digitadas não aparecem literalmente no documento. Para isso, o FinTrack considera informações como estabelecimento, itens identificados, categoria, descrição da categoria, contexto extraído e meio de pagamento. Por esse motivo, descrições claras nas categorias ajudam não apenas na organização visual, mas também na recuperação contextual dos registros. A Figura 23 apresenta exemplos de resultados localizados por texto e por termos recorrentes.



Resultado textual | Busca por “Pix” | Busca por “Combustível”

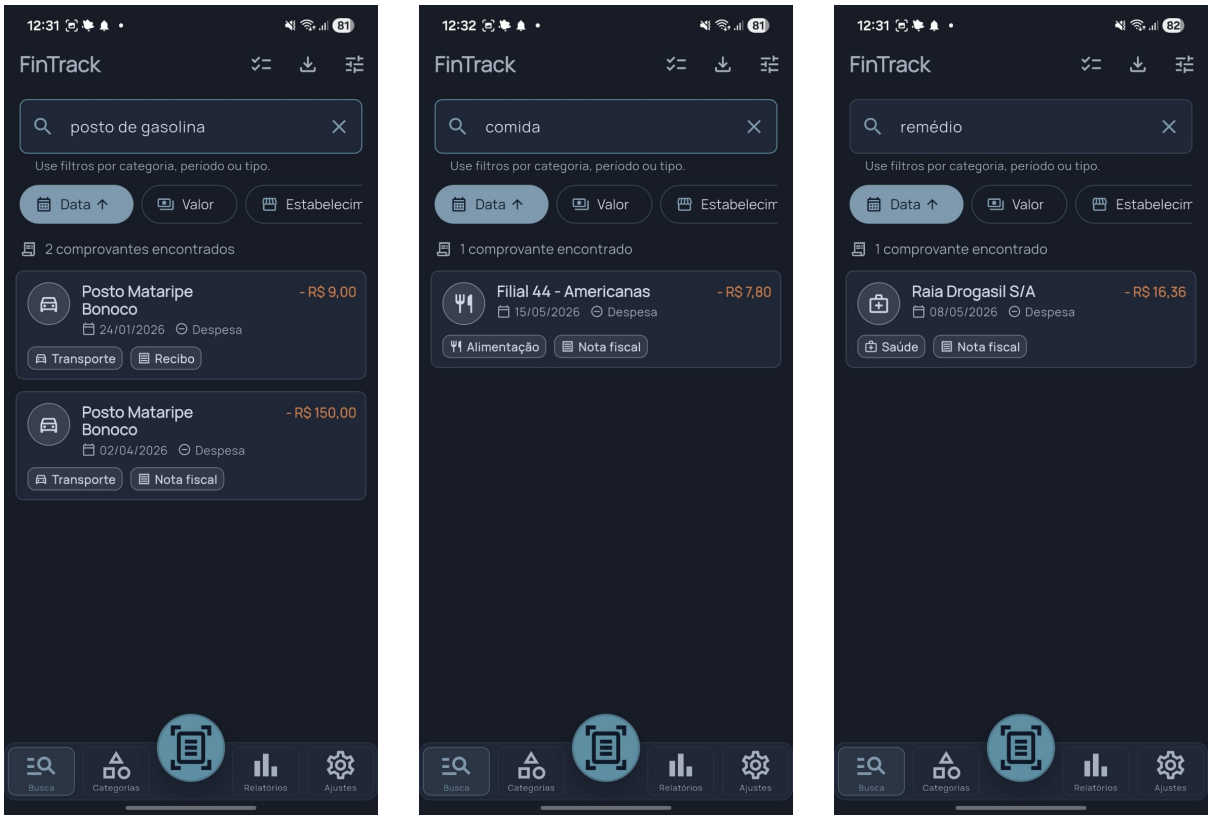
Figura 23: Resultados de busca textual e contextual por estabelecimento, meio de pagamento e conteúdo financeiro.

Ao pesquisar, a aplicação combina os resultados textuais e contextuais, remove duplicidades e ordena os registros por relevância. Quando há empate, registros com data extraída mais recente tendem a aparecer antes. Os resultados da busca ainda podem ser refinados pelos fil-

tros de “Categoria”, “Tipo de comprovante”, “Natureza” e “Período”, e podem ser reorganizados pelos botões de ordenação quando uma ordenação personalizada estiver ativa. Assim como na listagem geral, os resultados são exibidos em páginas de até 30 registros e novas páginas são carregadas conforme o usuário rola a tela.

Durante a execução da busca, a interface pode apresentar o estado “Buscando comprovantes”. Se nenhum item corresponder à consulta, o FinTrack exibe “Nenhum resultado encontrado” e orienta o usuário a tentar outros termos ou ajustar os filtros aplicados. A indexação contextual pode ocorrer em segundo plano após o salvamento ou edição de um registro; por isso, registros recém-criados podem levar alguns instantes para aparecer nos resultados contextuais.

Por exemplo, o usuário pode buscar por termos literais, como “Pix” ou o nome de um estabelecimento, e também por descrições aproximadas, como “combustível”, “posto de gasolina”, “comida” ou “remédio”. Nos dois casos, os resultados apresentados continuam sendo registros salvos no FinTrack, associados aos seus dados extraídos e aos arquivos armazenados pela aplicação. A Figura 24 mostra consultas contextuais com termos que representam situações de consumo.



Busca por “Posto de gasolina” Busca por “Comida” Busca por “Remédio”

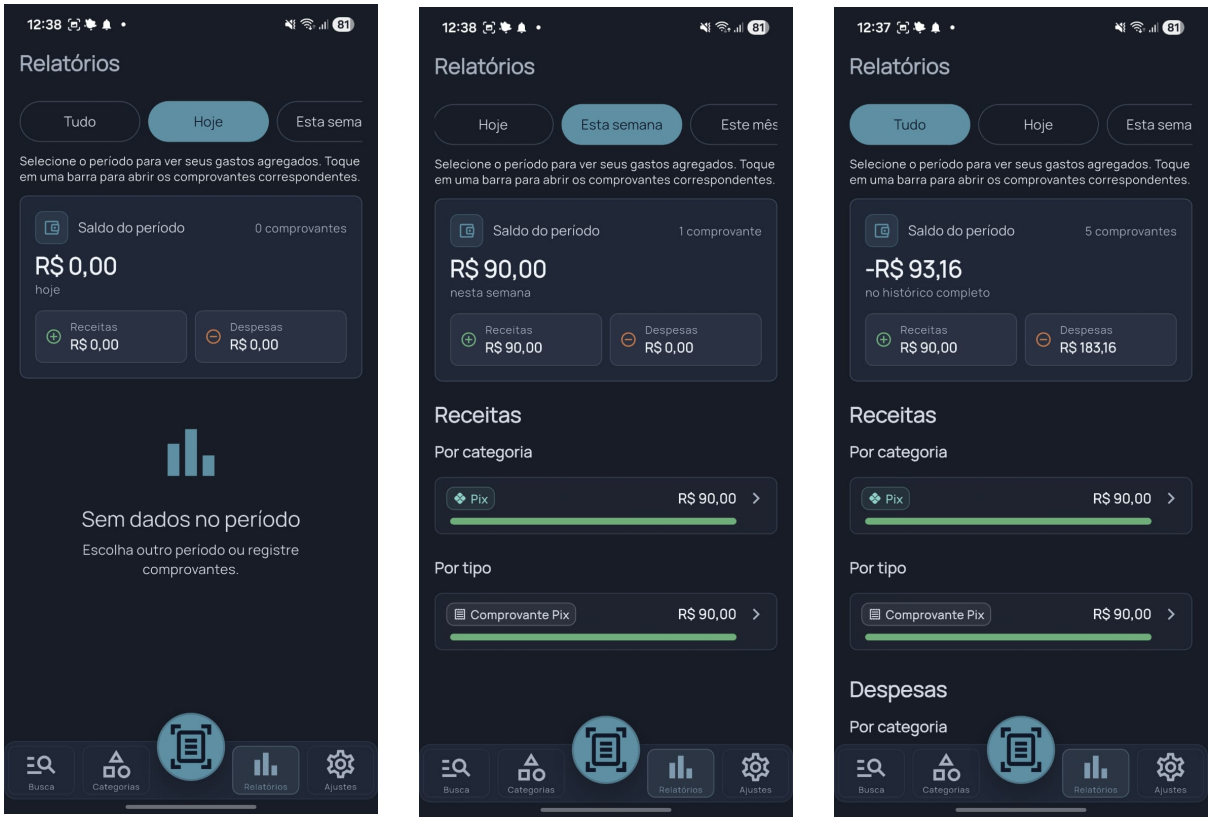
Figura 24: Exemplos de busca contextual por situações de consumo e significado aproximado.

Relatórios financeiros

A tela “Relatórios”, acessada pelo item “Relatórios” da barra de navegação inferior, apresenta uma visão consolidada dos dados extraídos dos registros de comprovantes salvos no FinTrack. No topo da tela, o usuário escolhe o período por meio dos botões “Tudo”, “Hoje”,

“Esta semana”, “Este mês” e “Este ano”. Abaixo do seletor, a interface orienta o usuário a selecionar o período para ver gastos agregados e tocar em uma barra para abrir os registros correspondentes.

O cartão “Saldo do período” mostra o saldo calculado pela diferença entre receitas e despesas, a quantidade de registros considerados e o resumo do período selecionado, como “hoje”, “nesta semana” ou “no histórico completo”. No mesmo cartão, os blocos “Receitas” e “Despesas” exibem os totais de cada natureza financeira e podem ser tocados para abrir a Tela de Busca já filtrada pelos registros daquele grupo. A Figura 25 mostra a mudança de visualização entre os filtros de período.



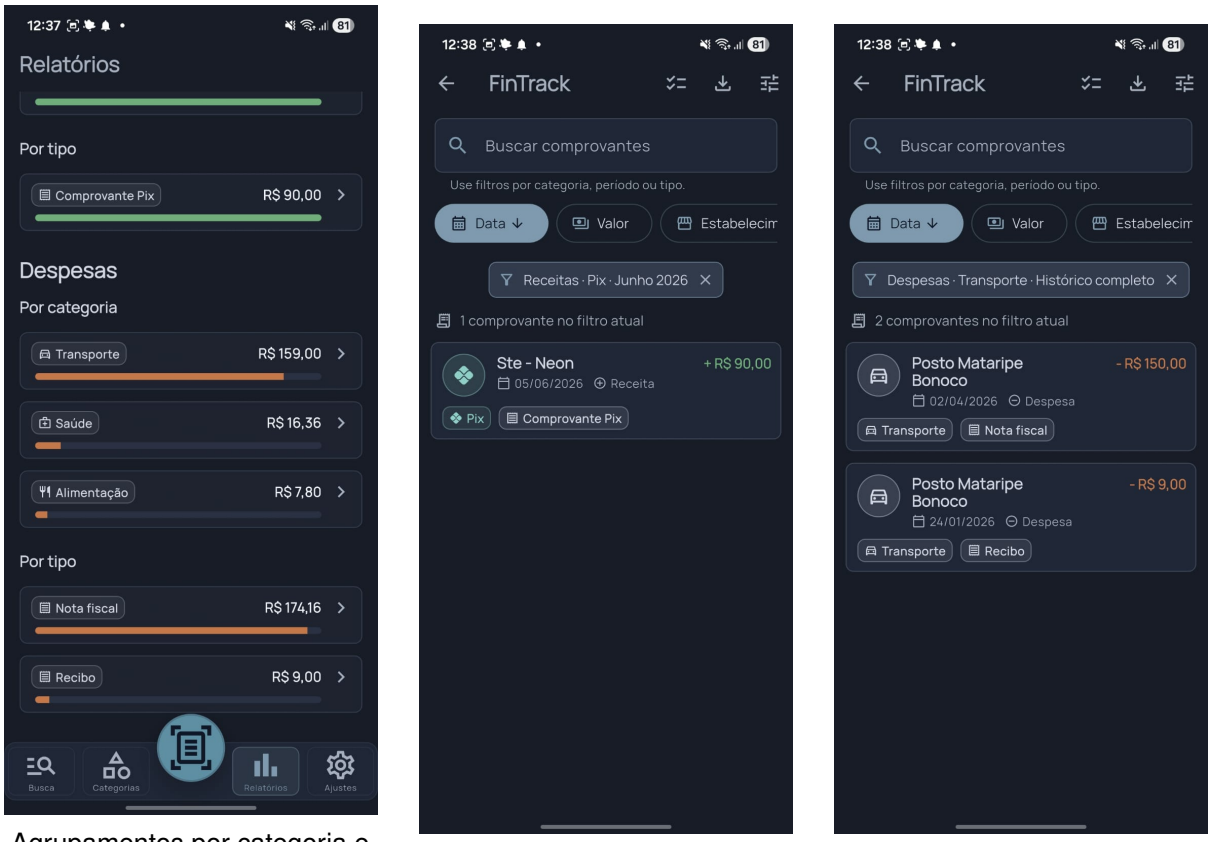
Filtro “Hoje” Filtro “Esta semana” Filtro “Tudo”

Figura 25: Relatórios financeiros filtrados por hoje, semana atual e histórico completo.

Quando há dados no período, o FinTrack separa os agrupamentos em duas seções: “Receitas” e “Despesas”. Em cada seção, os valores aparecem em “Por categoria” e “Por tipo”. As barras são ordenadas do maior para o menor valor, utilizam o ícone e a cor da categoria quando disponíveis e mostram o valor acumulado de cada grupo. Ao tocar em uma barra, como uma categoria de transporte ou o tipo “Comprovante Pix”, a aplicação abre a lista de registros com os filtros correspondentes já aplicados.

Os relatórios são calculados localmente a partir dos registros armazenados e dependem dos campos extraídos ou revisados no momento do cadastro, especialmente valor, data da transação, natureza financeira, tipo de comprovante e categoria. Registros sem data de transação extraída entram no histórico completo, mas não aparecem nos períodos que

dependem de uma data específica. Caso não existam registros no período selecionado, a tela exibe “Sem dados no período” e orienta o usuário a escolher outro período ou cadastrar novos registros. A Figura 26 apresenta os agrupamentos e exemplos de detalhamento aberto a partir das barras do relatório.



Agrupamentos por categoria e tipo

Detalhe de “Comprovante Pix”

Detalhe da categoria transporte

Figura 26: Agrupamentos do relatório financeiro e abertura dos registros correspondentes.

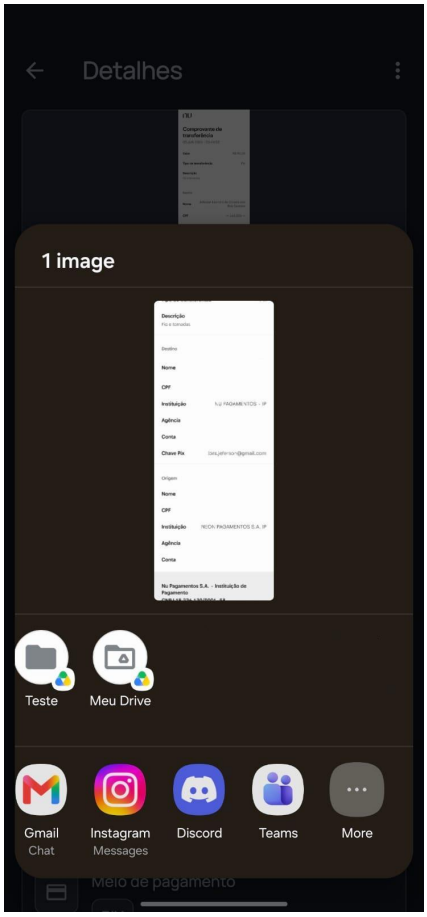
Compartilhamento e salvamento local

O FinTrack permite compartilhar ou salvar no dispositivo o arquivo associado a um registro de comprovante. Esse arquivo pode ter origem em um comprovante digital importado, em um arquivo recebido de outro aplicativo ou em uma imagem gerada a partir da digitalização de um comprovante físico. Por isso, essas ações atuam sobre o arquivo vinculado ao registro salvo, e não necessariamente sobre um comprovante originalmente digital.

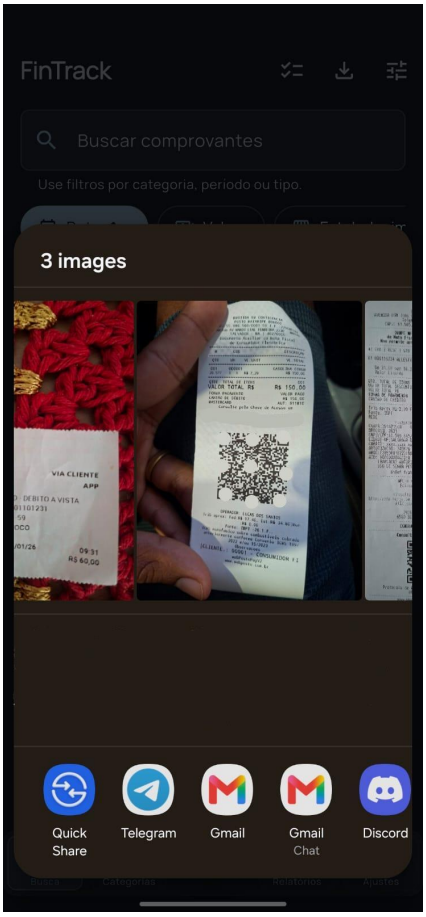
O compartilhamento pode ocorrer de forma individual ou em lote. No fluxo individual, o usuário abre a tela “Detalhes” de um registro, aciona o menu “Mais opções”, localizado na barra superior, e escolhe “Compartilhar”. O FinTrack envia o arquivo associado ao registro para o seletor nativo de compartilhamento do Android; quando o seletor é aberto com sucesso, a aplicação exibe a mensagem “Compartilhamento aberto”.

No fluxo em lote, o usuário entra no modo de seleção múltipla pela Tela de Busca, usando o botão “Selecionar comprovantes” da barra superior ou mantendo pressionado um cartão de registro, seleciona os registros desejados, abre o menu “Mais opções” e escolhe “Compartilhar

selecionados”. A Figura 27 mostra as duas formas de acionamento do compartilhamento.



Ação “Compartilhar”

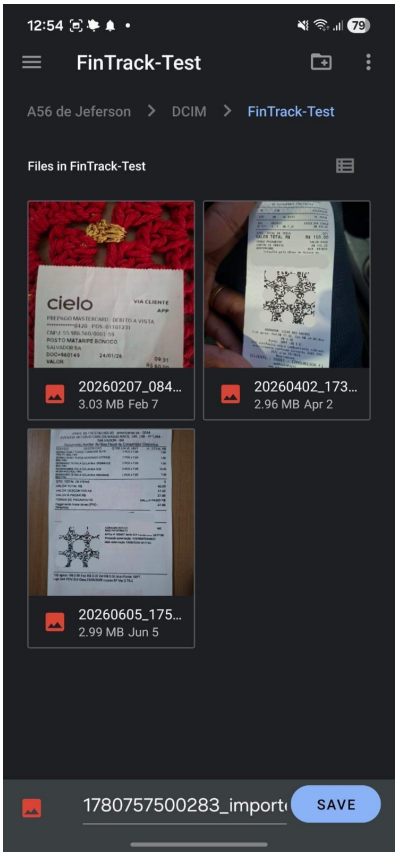


“Compartilhar selecionados”

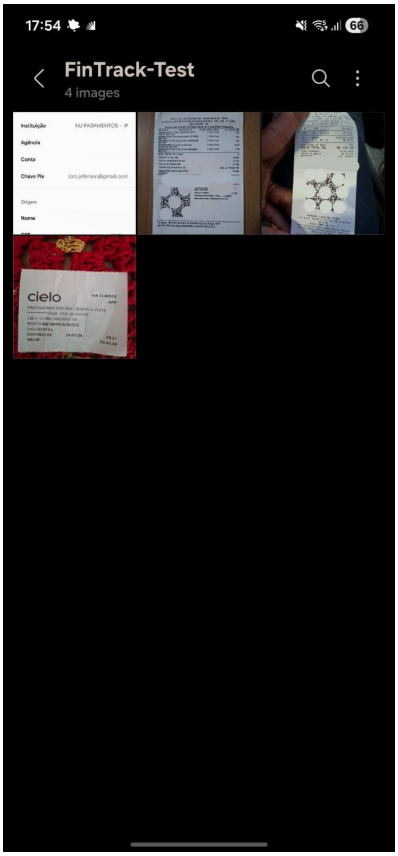
Figura 27: Compartilhamento individual e em lote dos arquivos associados aos registros salvos.

O salvamento no dispositivo segue a mesma separação entre ação individual e ação em lote. No fluxo individual, a ação “Salvar no dispositivo”, disponível no menu “Mais opções” da tela “Detalhes”, envia o arquivo associado ao registro para um diretório acessível ao usuário fora da área privada do aplicativo; em caso de sucesso, a mensagem exibida é “Arquivo salvo no dispositivo”.

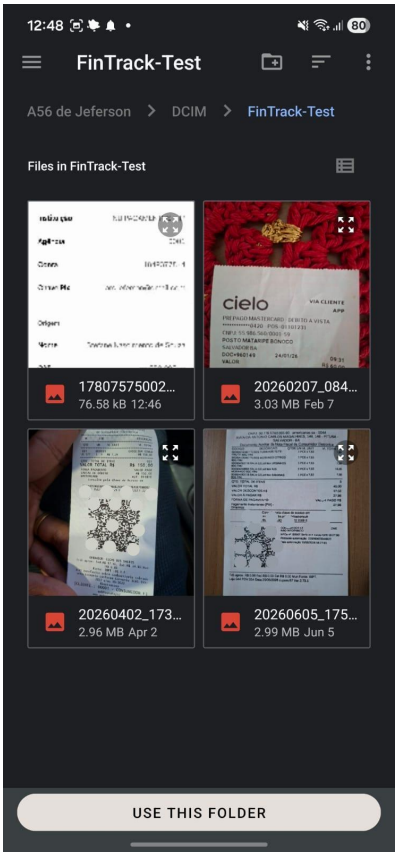
No fluxo em lote, a opção “Salvar no dispositivo” fica disponível no menu “Mais opções” do modo de seleção múltipla e grava os arquivos associados aos registros selecionados; quando mais de um arquivo é salvo, a aplicação confirma com “Arquivos salvos no dispositivo”. A Figura 28 agrupa os dois fluxos, mostrando a escolha do diretório e o diretório atualizado após o salvamento individual e em lote.



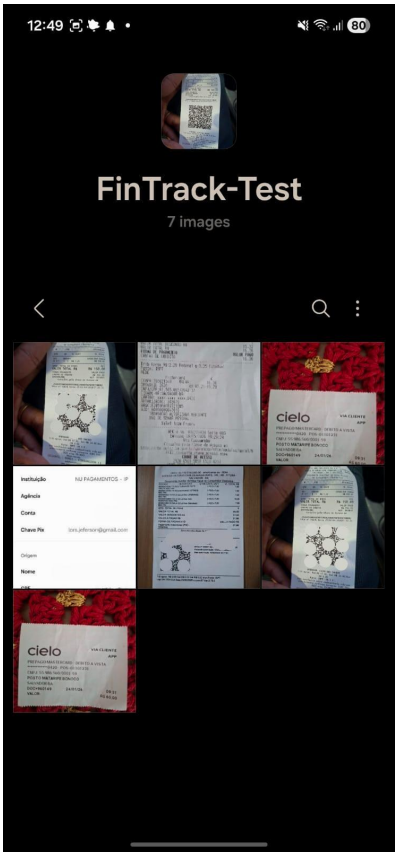
Seleção do diretório individual



Arquivo salvo no diretório



Seleção do diretório em lote



Diretório atualizado em lote

Figura 28: Salvamento local individual e em lote dos arquivos associados aos registros.

Esses recursos são úteis quando o usuário precisa enviar o arquivo associado a um registro, anexar uma nota fiscal, compartilhar arquivos por aplicativos externos ou preservar uma cópia local independente do armazenamento interno do FinTrack. Caso o Android não consiga abrir o compartilhamento ou salvar os arquivos, o FinTrack apresenta mensagens como “Não foi possível abrir o compartilhamento”, “Não foi possível salvar o arquivo” ou “Não foi possível salvar os arquivos”.

Exclusão de registros e categorias

A exclusão de um registro de comprovante pode ser feita a partir da tela “Detalhes”. Para isso, o usuário abre o menu “Mais opções”, localizado na barra superior, escolhe “Excluir” e confirma a operação no diálogo “Excluir comprovante?”. A mensagem de confirmação informa que “A imagem, os dados e os dados de busca inteligente serão removidos”, indicando que a remoção inclui o registro salvo, o arquivo associado e os dados usados pela busca contextual. Ao concluir, o FinTrack exibe “Comprovante excluído” e retorna para a tela anterior. A Figura 29 reúne essa confirmação individual e a confirmação em lote descrita a seguir.

“Excluir comprovante?”

Exclusão em lote

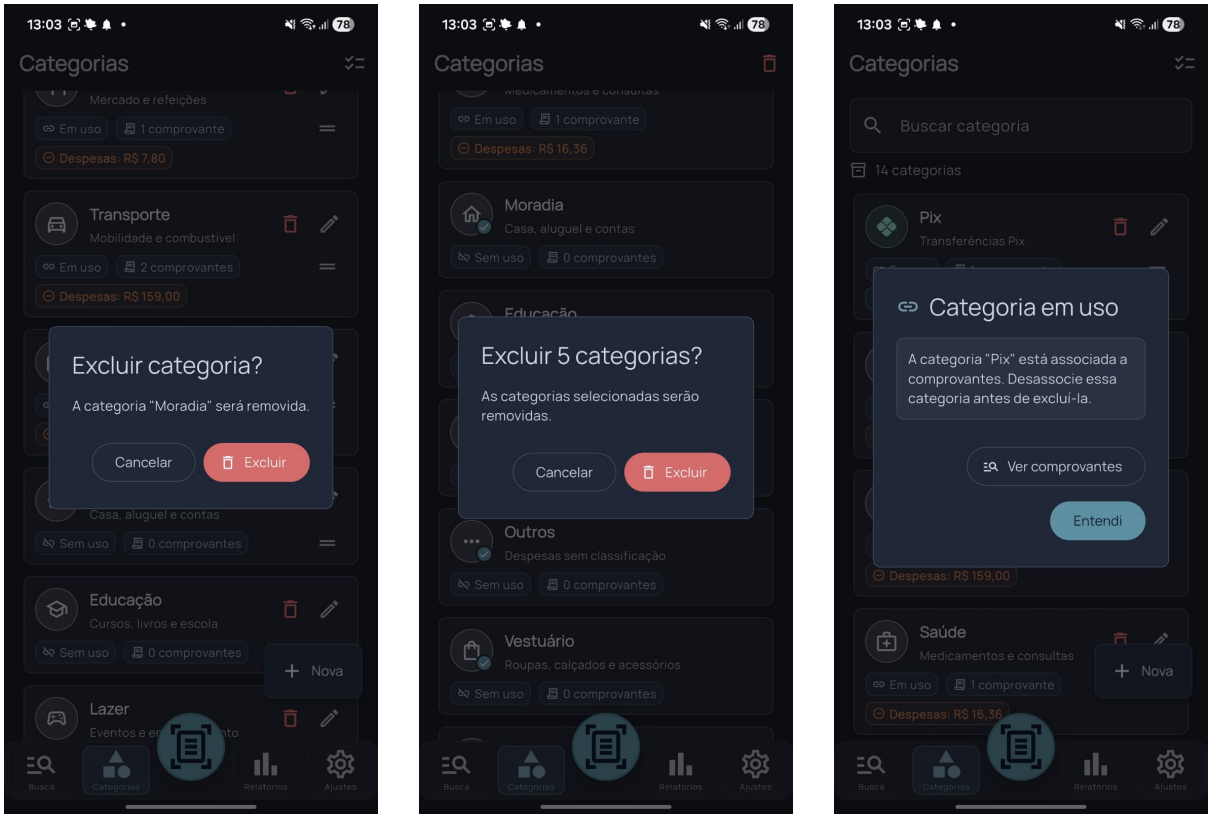
Figura 29: Confirmações de exclusão individual e em lote de registros de comprovantes.

Na Tela de Busca, também é possível excluir registros em lote. O usuário aciona “Selecionar comprovantes” na barra superior ou mantém pressionado um cartão de registro, marca os registros desejados, abre o menu “Mais opções” e escolhe “Excluir selecionados”. A confirma-

ção usa o título “Excluir comprovante?” quando apenas um item está selecionado e “Excluir X comprovantes?” quando há vários itens. Durante a execução, a tela informa “Excluindo comprovantes” e “Aguarde enquanto os comprovantes selecionados são removidos”. Ao final, a aplicação confirma com “Comprovante excluído” ou “X comprovantes excluídos”.

Na tela “Categorias”, cada cartão informa se a categoria está “Em uso” ou “Sem uso” e mostra a quantidade de registros associados. Categorias sem registros associados podem ser removidas pelo botão “Excluir” do próprio cartão ou, em modo de seleção, pelo botão “Selecionar categorias” da barra superior seguido da ação “Excluir selecionadas”. Esse modo de seleção também pode ser iniciado mantendo pressionado um cartão de categoria. A confirmação individual usa “Excluir categoria?” e informa que a categoria selecionada será removida; para múltiplas categorias, o título passa a ser “Excluir X categorias?”.

Quando uma categoria está associada a registros, o FinTrack bloqueia a exclusão para preservar a consistência dos dados. Nessa situação, a aplicação exibe “Categoria em uso” ou “Categorias em uso” e informa que a categoria está associada a comprovantes. Na tentativa individual, o diálogo oferece “Ver comprovantes”, abrindo a Tela de Busca filtrada por aquela categoria, e “Entendi”, para fechar o aviso. Para excluir a categoria posteriormente, o usuário deve primeiro desassociá-la dos registros correspondentes. A Figura 30 apresenta a exclusão individual e em lote de categorias sem uso, além do bloqueio aplicado quando há registros associados; o mesmo bloqueio também vale para tentativas de exclusão múltipla com categorias em uso.



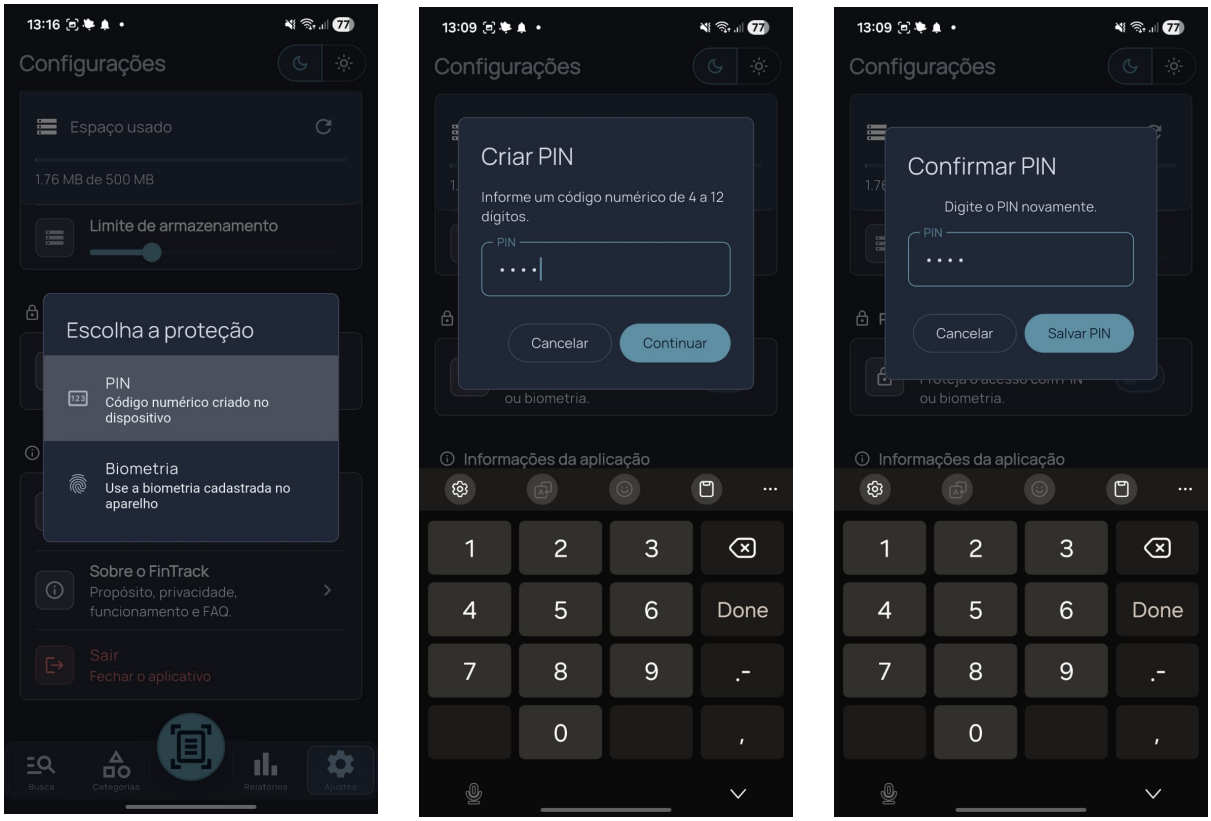
Categoria “Sem uso” | Categorias selecionadas | Aviso “Categoria em uso”

Figura 30: Exclusão individual e em lote de categorias sem uso e bloqueio de exclusão para categoria vinculada a registros salvos.

Bloqueio por PIN

O bloqueio por PIN é configurado em Ajustes, na seção “Proteção do aplicativo”. Ao ativar “Autenticação local”, o FinTrack exibe o diálogo “Escolha a proteção”, no qual a opção “PIN” aparece com a descrição “Código numérico criado no dispositivo”. Depois da escolha, o aplicativo abre o fluxo de criação do código local.

Na tela “Criar PIN”, o usuário informa um código numérico de 4 a 12 dígitos no campo “PIN” e aciona “Continuar”. Em seguida, a tela “Confirmar PIN” solicita que o código seja digitado novamente e confirmado por meio da ação “Salvar PIN”. Se o código tiver menos de quatro dígitos, a interface apresenta a mensagem “Informe ao menos 4 dígitos”; se a confirmação não coincidir com o primeiro código, apresenta “Os PINs não conferem. Comece novamente”. Quando a configuração não é concluída, a autenticação local não é ativada ou o método anterior é mantido. A Figura 31 mostra a escolha do método e a criação do código.

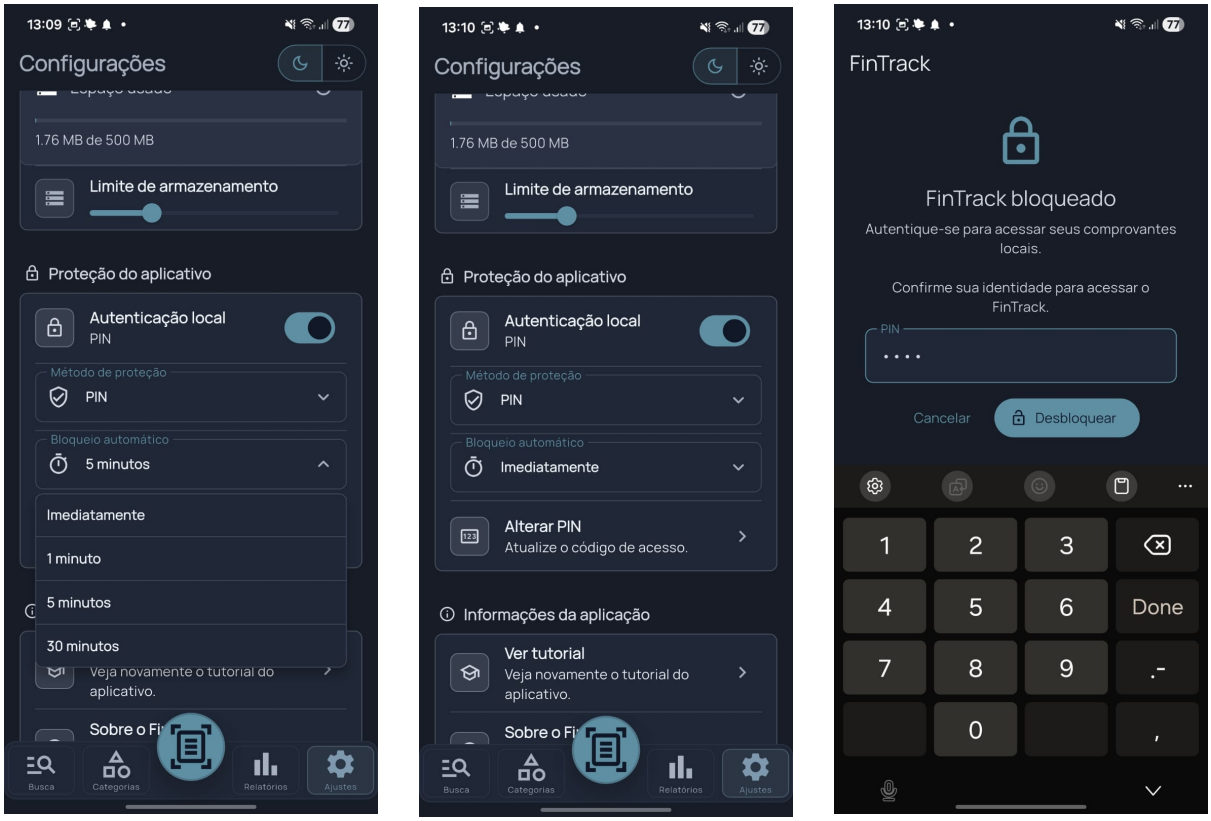


Opção “PIN” Tela “Criar PIN” Tela “Confirmar PIN”

Figura 31: Escolha do método PIN e definição do código local de acesso ao FinTrack.

Após a criação do PIN, o usuário define o campo “Bloqueio automático”, com as opções “Imediatamente”, “1 minuto”, “5 minutos” ou “30 minutos”. Esse intervalo determina quando o FinTrack deve exigir nova autenticação ao retornar do segundo plano ou ao ser reaberto. Quando o bloqueio está ativo, a tela “FinTrack bloqueado” é exibida; nela, o usuário informa o PIN cadastrado e aciona “Desbloquear”. Caso o campo seja enviado incompleto, a interface orienta “Informe o PIN cadastrada”; caso o código esteja incorreto, exibe “PIN incorreto. Acesso não autorizado”.

Quando o PIN já está configurado, a seção de proteção também apresenta a ação “Alterar PIN”, usada para substituir o código de acesso. O PIN é uma proteção local do dispositivo e não substitui a senha de *backup*: ele restringe o acesso à aplicação e aos registros salvos naquele aparelho, enquanto a senha de *backup* protege os arquivos criptografados enviados à nuvem. A Figura 32 apresenta a definição do intervalo e a tela bloqueada.



Campo “Bloqueio automático” Intervalo selecionado Tela “FinTrack bloqueado”

Figura 32: Intervalo de bloqueio automático e desbloqueio do FinTrack por PIN.

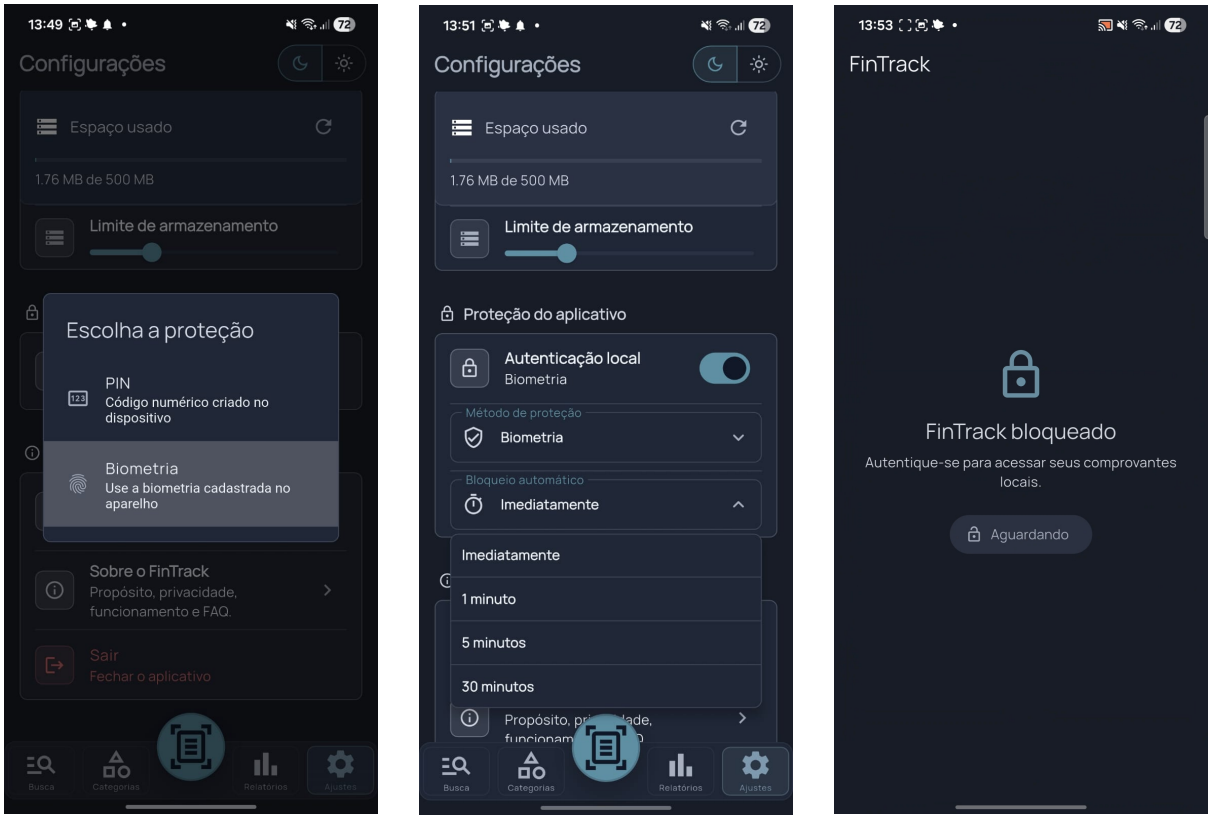
Bloqueio por biometria

O bloqueio por biometria também é configurado em Ajustes, na seção “Proteção do aplicativo”. Ao ativar “Autenticação local”, o usuário escolhe a opção “Biometria” no diálogo “Escolha a proteção”. Essa opção aparece com a descrição “Use a biometria cadastrada no aparelho”, indicando que o FinTrack utiliza a autenticação biométrica já configurada no Android, como impressão digital, reconhecimento facial ou outro método disponível no dispositivo.

Antes de ativar o método, o FinTrack verifica se a biometria está disponível. Quando o dispositivo não oferece suporte, não possui biometria cadastrada ou o recurso está temporariamente indisponível, a aplicação apresenta mensagens como “Configure a biometria nas configurações do sistema operacional”, “Este dispositivo não possui autenticação biométrica disponível” ou “A biometria está temporariamente indisponível neste dispositivo”. Se a verificação for bem-sucedida, o Android exibe o prompt nativo com o título “Ativar biometria” e a orientação “Confirme sua identidade para ativar a proteção”. Caso a confirmação não seja concluída, o FinTrack informa “A biometria não foi confirmada”.

Depois de ativada, a biometria usa o mesmo campo “Bloqueio automático” do PIN, com as opções “Imediatamente”, “1 minuto”, “5 minutos” ou “30 minutos”. Quando o intervalo definido é atingido e o aplicativo volta do segundo plano ou é reaberto, a tela “FinTrack bloqueado” é exibida. Ao tocar em “Desbloquear”, o Android abre o prompt biométrico com o título “Desbloquear FinTrack” e a mensagem “Confirme sua identidade para acessar o FinTrack”. Se a autenticação falhar ou for cancelada, o FinTrack apresenta “Não foi possível confirmar a biometria. Acesso não autorizado”.

A tela nativa de autenticação biométrica não aparece nas imagens porque o Android não permite capturar tela desse diálogo por razões de segurança. Por isso, o manual apresenta a escolha do método, a configuração do bloqueio automático e a tela bloqueada, mas não o prompt biométrico exibido pelo sistema operacional. A Figura 33 mostra as etapas que podem ser documentadas por captura de tela.



Opção “Biometria” Campo “Bloqueio automático” Tela “FinTrack bloqueado”

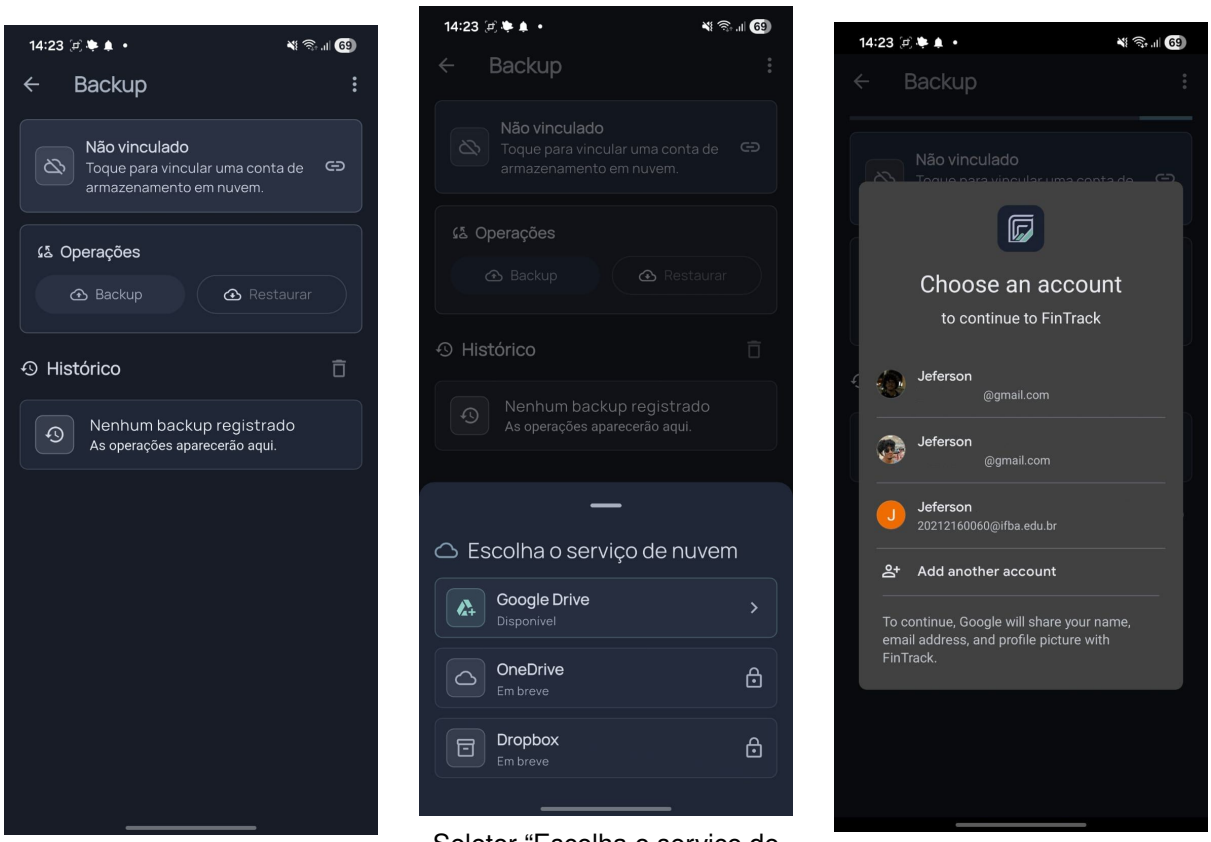
Figura 33: Escolha do método biométrico, definição do intervalo de bloqueio automático e tela bloqueada antes da autenticação nativa do Android.

Conta de nuvem para *backup*

O *backup* em nuvem é opcional e fica concentrado na tela “Backup”, acessada em Ajustes, na seção “Configurações de backup”, pelo item “Backup”. Essa área é descrita na própria interface como “Nuvem e histórico de sincronização”. Ao abrir a tela, o primeiro cartão indica o estado da conta de armazenamento: quando nenhuma conta está vinculada, o cartão exibe “Não vinculado” e a orientação “Toque para vincular uma conta de armazenamento em nuvem”.

Ao tocar nesse cartão, o FinTrack abre o seletor “Escolha o serviço de nuvem”. A opção disponível é “Google Drive”, apresentada como “Disponível”; “OneDrive” e “Dropbox” aparecem como “Em breve”, indicando que ainda não podem ser selecionados. Após escolher “Google Drive”, o usuário é encaminhado ao seletor de conta do Google no Android. Quando o vínculo é concluído, a aplicação registra a conta, marca o token de nuvem como válido e apresenta a mensagem “Conta de nuvem vinculada”. A partir desse momento, o cartão passa a exibir “Google Drive” e o e-mail vinculado.

Se o usuário tocar novamente no cartão de conta vinculada, o FinTrack solicita confirmação para “Desvincular conta de nuvem”. A mensagem informa que, ao desvincular a conta, o *backup* automático será desativado. Ao confirmar em “Desvincular”, a conta é removida da configuração local e a aplicação exibe “Conta de nuvem desvinculada”. A desvinculação não apaga automaticamente os dados já enviados à nuvem; a remoção desses dados é feita separadamente pela ação “Limpar nuvem”, descrita na subseção de operações e histórico. A Figura 34 apresenta o vínculo com o Google Drive, único serviço de nuvem disponível na solução.



Cartão “Não vinculado” | Seletor “Escolha o serviço de nuvem” | Seleção de conta Google

Figura 34: Vínculo do Google Drive como serviço de nuvem disponível para *backup* no FinTrack.

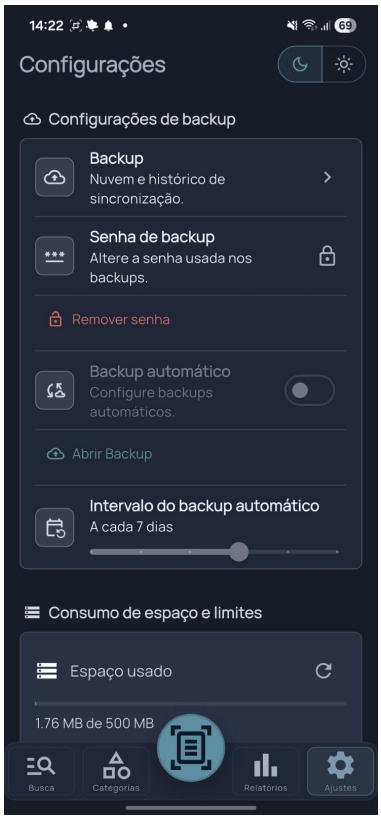
Senha de *backup*

A senha de *backup* é configurada em Ajustes, na seção “Configurações de backup”, pelo item “Senha de backup”. Quando nenhuma senha foi definida, esse item exibe “Crie uma senha

para seus backups”; quando já existe uma senha, passa a exibir “Altere a senha usada nos backups”. Essa senha protege os arquivos enviados para a nuvem e não deve ser confundida com o PIN local: o PIN desbloqueia o aplicativo no aparelho, enquanto a senha de *backup* é necessária para restaurar dados protegidos em outro dispositivo.

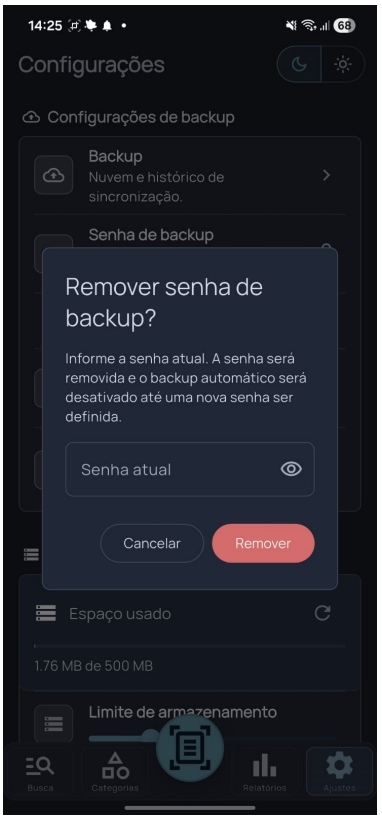
Ao tocar em “Senha de backup”, o FinTrack abre o diálogo “Definir senha de backup” ou “Alterar senha de backup”. O próprio diálogo orienta: “Essa senha protege seus backups. Se você esquecê-la, não será possível restaurar arquivos protegidos por ela”. Para definir uma nova senha, o usuário preenche “Nova senha” e “Confirmar nova senha”, podendo usar os botões “Mostrar senha” e “Ocultar senha” para alternar a visibilidade dos campos. A senha deve ter pelo menos 8 caracteres; caso contrário, a interface informa “Use pelo menos 8 caracteres”. Se os campos não coincidirem, informa “As senhas não conferem”.

Quando uma senha já existe, a alteração exige também o preenchimento de “Senha atual”. Se ela não corresponder à senha armazenada, a interface apresenta “Senha atual incorreta”. Se a nova senha for igual à atual, informa “Use uma senha diferente da atual”. Para remover a proteção, o usuário aciona “Remover senha” e confirma no diálogo “Remover senha de backup?”, informando a senha atual. A mensagem deixa claro que a senha será removida e que o *backup* automático será desativado até que uma nova senha seja definida. A Figura 35 mostra o item de configuração, o diálogo de definição e a confirmação de remoção da senha.



Item “Senha de backup”

Diálogo “Definir senha de backup”



Diálogo “Remover senha de backup?”

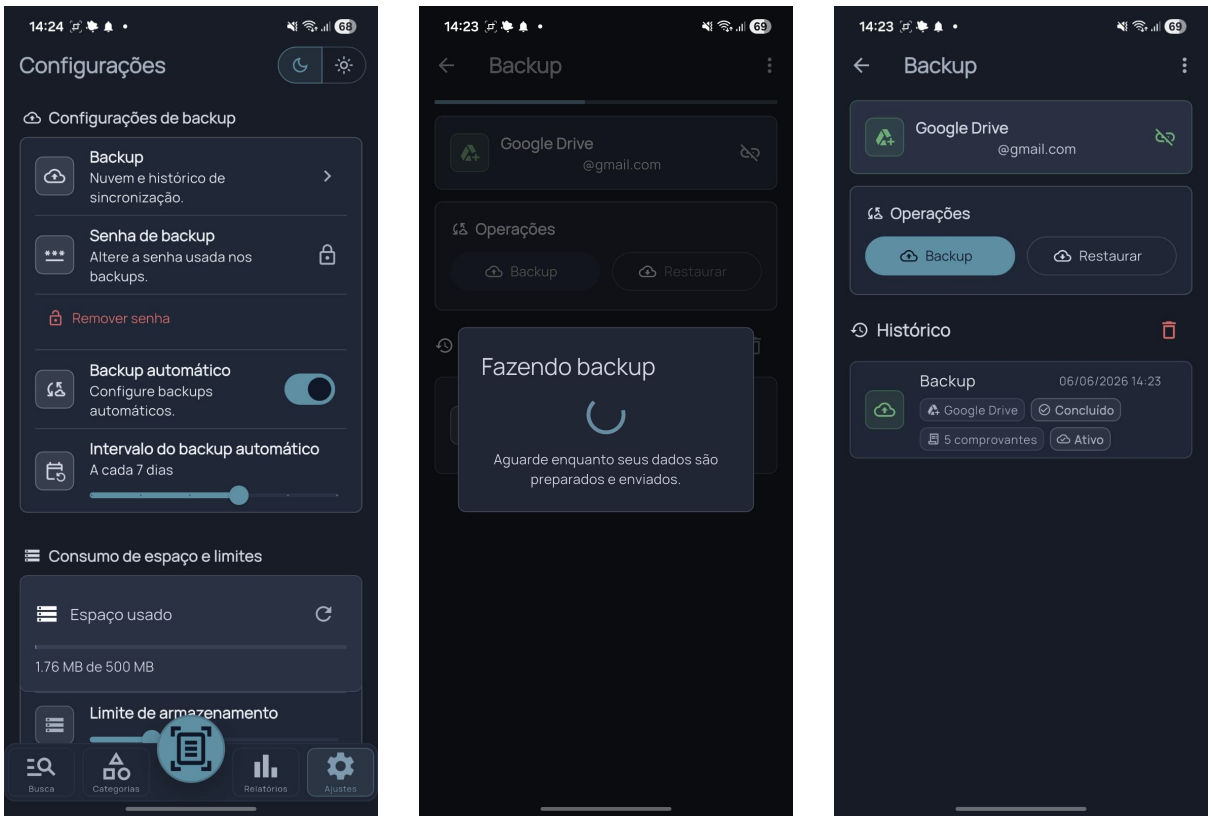
Figura 35: Definição, alteração e remoção da senha usada para proteger os arquivos de *backup*.

Backup, restauração e histórico

Com a conta de nuvem vinculada e a senha configurada, a tela “Backup” habilita o painel “Operações”. Nele, o botão “Backup” executa o envio manual dos dados para a nuvem. Se a senha ainda não tiver sido definida, a tela informa “Defina uma senha de backup nas configurações para proteger seus arquivos” e a tentativa de execução apresenta “Defina uma senha de backup nas configurações antes de fazer backup”. Durante o envio, o FinTrack mostra o diálogo “Fazendo backup” com a mensagem “Aguarde enquanto seus dados são preparados e enviados”; ao concluir, apresenta “Backup concluído”.

O *backup* automático é configurado em Ajustes, na seção “Configurações de backup”, pelo controle “Backup automático”. Ele só pode ser ativado quando há conta de nuvem vinculada, token válido e senha de *backup* definida. Se a senha ainda não existir, a aplicação abre o aviso “Senha necessária”, orientando: “Para ativar o backup automático, defina uma senha de backup primeiro”. Se a senha existir, mas a conta de nuvem não estiver válida, a seção apresenta o botão “Abrir Backup” para conduzir o usuário à tela de vínculo.

O intervalo é definido em “Intervalo do backup automático”, com opções como “A cada 1 minuto”, “A cada 1 dia”, “A cada 3 dias”, “A cada 7 dias”, “A cada 15 dias” e “A cada 30 dias”. Ao salvar a configuração, o FinTrack informa “Backup automático definido” ou “Backup automático desativado”. A Figura 36 apresenta o controle de *backup* automático, a execução manual e o registro gerado no histórico.

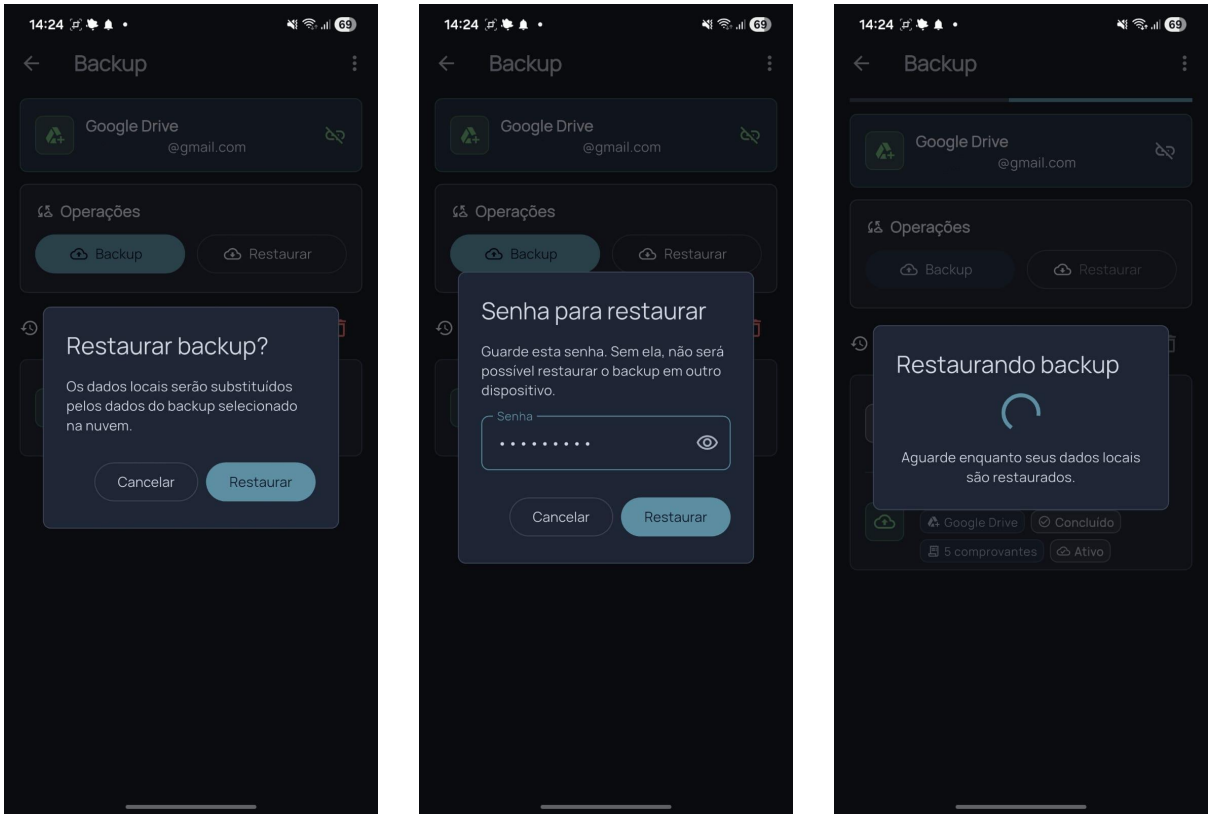


Controle “Backup automático” | Diálogo “Fazendo backup” | Registro no “Histórico”

Figura 36: Ativação do *backup* automático, execução manual de *backup* e registro da operação.

O arquivo de *backup* contém os registros de comprovantes, os arquivos associados a esses registros e configurações necessárias para a recuperação, como preferências relacionadas ao próprio *backup*. Esses dados são protegidos pela senha definida pelo usuário antes de serem enviados ao serviço de nuvem. Por isso, a restauração depende da mesma senha: sem ela, os dados protegidos não podem ser recuperados.

O botão “Restaurar” inicia a recuperação dos dados salvos na nuvem. Antes de prosseguir, o FinTrack exibe o diálogo “Restaurar backup?”, informando que “Os dados locais serão substituídos pelos dados do backup selecionado na nuvem”. Após a confirmação em “Restaurar”, o usuário informa a senha no diálogo “Senha para restaurar”. Durante o processo, a aplicação mostra “Restaurando backup” e “Aguarde enquanto seus dados locais são restaurados”; ao final, exibe “Restauração concluída”. Caso o pacote seja incompatível, esteja corrompido, use senha incorreta ou ultrapasse o limite local de armazenamento, a operação é interrompida e registrada como falha. A Figura 37 mostra as três etapas visuais da restauração.

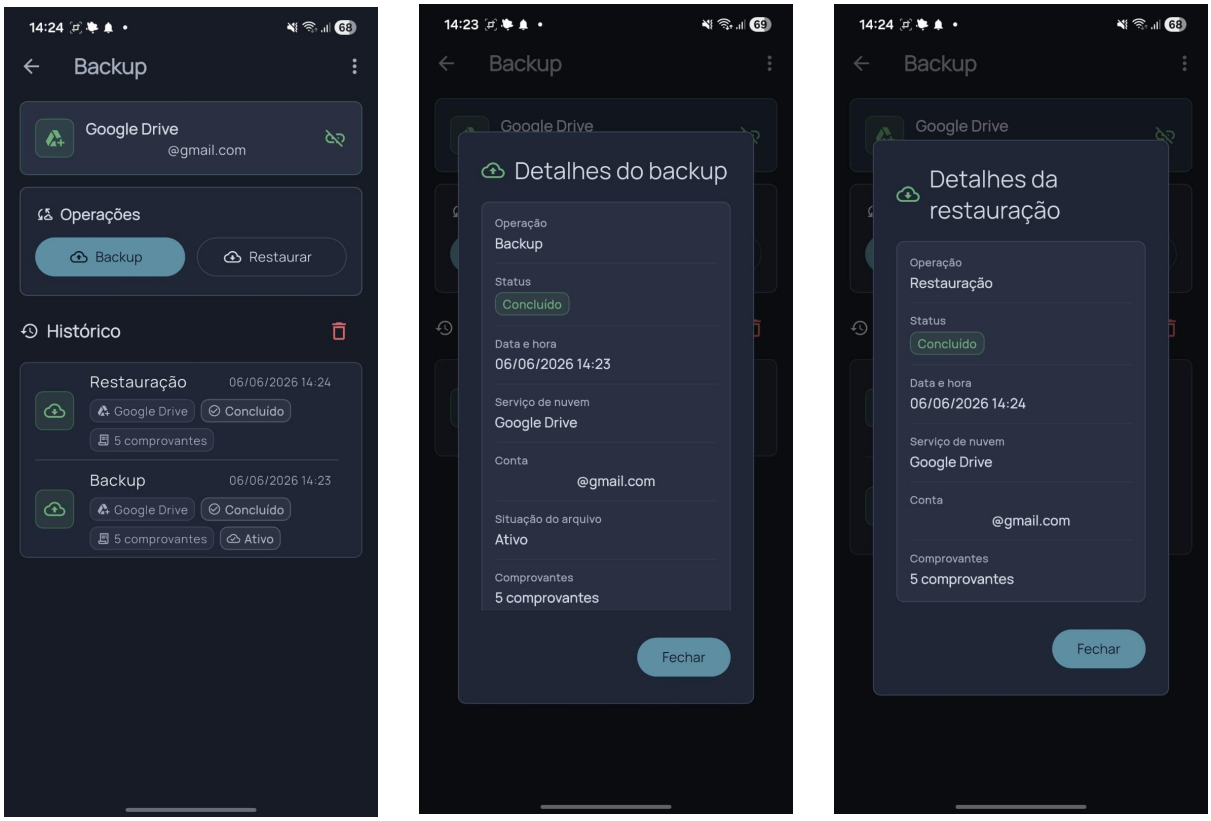


Diálogo “Restaurar backup?” Diálogo “Senha para restaurar” Diálogo “Restaurando backup”

Figura 37: Confirmação, senha e execução da restauração de dados a partir da nuvem.

As operações ficam registradas na seção “Histórico”. Quando não há registros, a tela exibe “Nenhum backup registrado”. Cada item do histórico indica se a operação foi “Backup” ou “Restauração”, a data e hora, o serviço de nuvem, o status, a quantidade de comprovantes envolvidos e, no caso de *backup*, a situação do arquivo em nuvem, como “Ativo”, “Inativo” ou “Excluído”. Ao tocar em um item, o FinTrack abre “Detalhes do backup” ou “Detalhes da restauração”, com campos como “Operação”, “Status”, “Data e hora”, “Serviço de nuvem”, “Conta”, “Situação do arquivo” e “Comprovantes”. A Figura 38 mostra a lista de histórico e os

detalhes de cada tipo de operação.

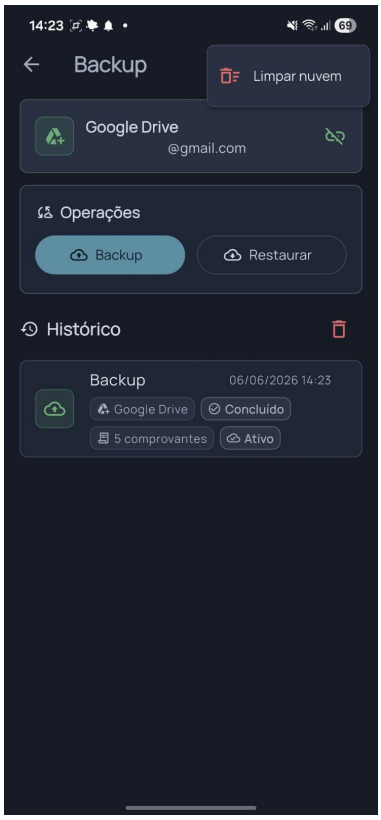


Lista “Histórico” “Detalhes do backup” “Detalhes da restauração”

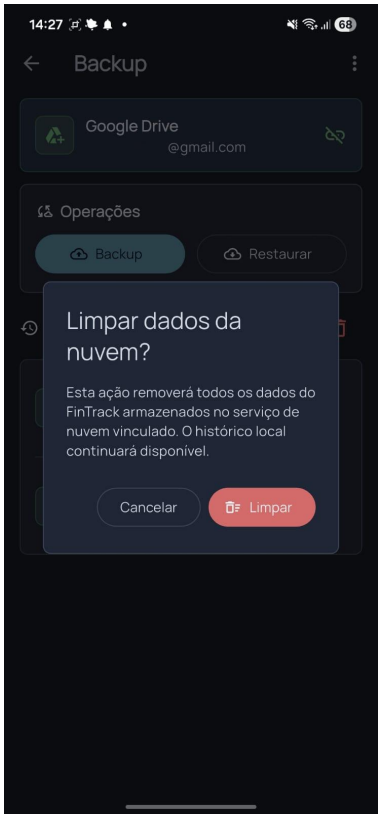
Figura 38: Histórico de operações de *backup* e restauração e suas telas de detalhamento.

No menu “Mais opções” da tela “Backup”, a ação “Limpar nuvem” permite remover os dados do FinTrack armazenados no serviço de nuvem vinculado. A confirmação ocorre no diálogo “Limpar dados da nuvem?”, que informa que o histórico local continuará disponível. Em seguida, o FinTrack solicita a “Senha do backup” para autorizar a remoção e executa a operação com o diálogo “Removendo dados da nuvem”. Ao concluir, apresenta “Dados da nuvem removidos”.

O botão “Apagar histórico”, na seção “Histórico”, remove apenas os registros exibidos nessa tela; a própria confirmação informa que comprovantes, arquivos de *backup* e configurações não são apagados por essa ação. A Figura 39 apresenta o diálogo de limpeza da nuvem e o diálogo de confirmação para apagar apenas o histórico local.



Ação “Limpar nuvem”



Diálogo “Limpar dados da nuvem?”

Confirmação de histórico

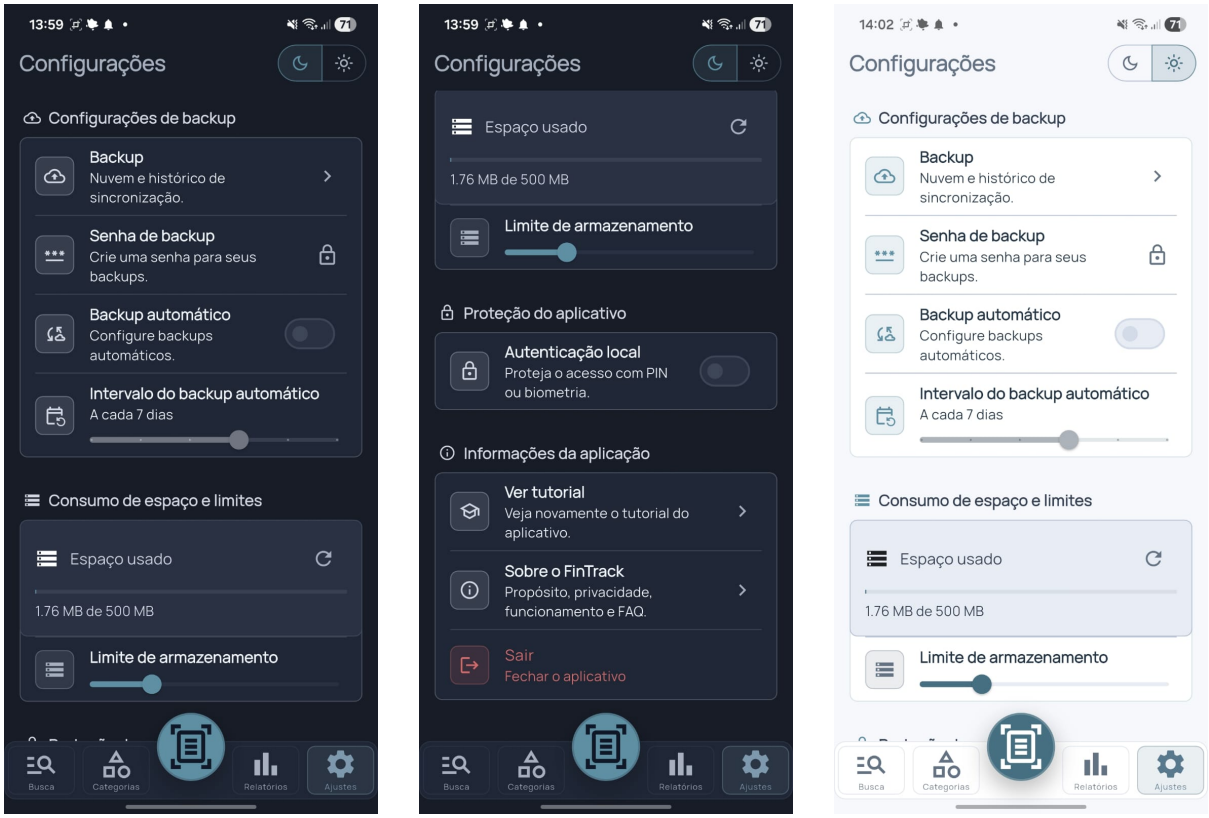
Figura 39: Remoção dos dados de *backup* da nuvem e limpeza do histórico local.

Ajustes e configurações gerais

O item “Ajustes” da barra inferior abre a tela “Configurações”. Essa tela reúne preferências de funcionamento e atalhos administrativos do FinTrack. Na barra superior, ao lado do título “Configurações”, há um controle de alternância de tema com os botões “Tema escuro” e “Tema claro”. A Figura 40 mostra a tela de configurações, as primeiras seções exibidas e o mesmo ambiente já com o tema claro selecionado.

A primeira seção é “Configurações de backup”. Nela, o item “Backup” abre a tela de nuvem e histórico, identificada pela descrição “Nuvem e histórico de sincronização”. O item “Senha de backup” permite definir, alterar ou remover a senha usada para proteger os arquivos enviados à nuvem. O controle “Backup automático” habilita ou desabilita agendamentos quando há conta de nuvem vinculada e senha configurada, e o campo “Intervalo do backup automático” permite ajustar a periodicidade.

A seção “Consumo de espaço e limites” mostra o cartão de armazenamento. Esse cartão pode indicar “Espaço usado”, “Espaço alto” ou “Limite atingido”, exibe a relação de uso em MB, oferece o botão “Atualizar espaço” e permite ajustar o “Limite de armazenamento” por controle deslizante. Quando o limite é atingido, a interface orienta o usuário a liberar espaço ou aumentar o limite.



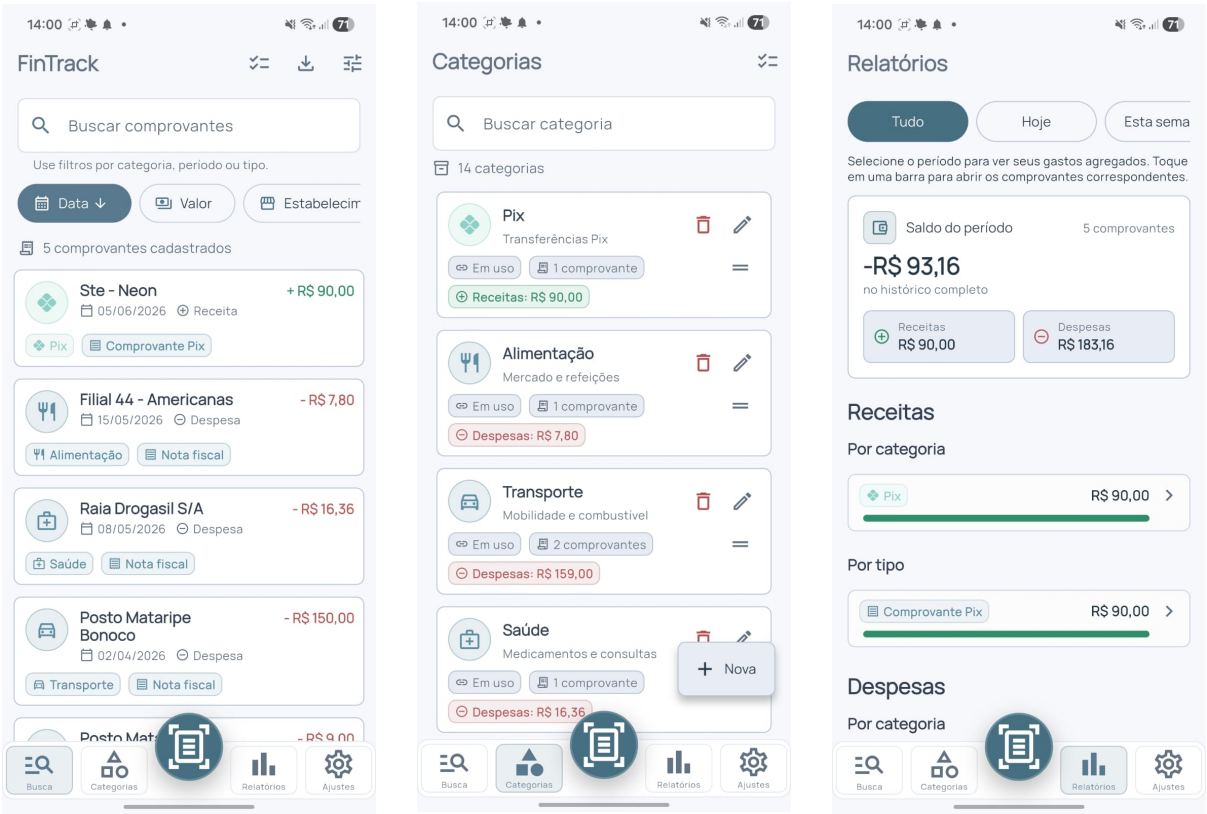
Tela “Configurações” Seções de preferências “Configurações” em tema claro

Figura 40: Tela “Configurações”, seções de preferências e alternância para o tema claro.

Na sequência da mesma tela, a seção “Proteção do aplicativo” concentra a autenticação local. O controle “Autenticação local” ativa a proteção por “PIN” ou “Biometria”. Quando a proteção está ativa, o campo “Método de proteção” permite escolher o método, o campo “Bloqueio automático” define o intervalo para novo bloqueio e, quando o método atual é PIN, a ação “Alterar PIN” permite atualizar o código de acesso.

Por fim, a seção “Informações da aplicação” reúne “Ver tutorial”, para reiniciar o tutorial inicial, “Sobre o FinTrack”, que abre a tela institucional e de suporte do aplicativo, e “Sair”, descrito como “Fechar o aplicativo”. Ao acionar “Sair”, o FinTrack apresenta o diálogo “Sair do FinTrack?” antes de fechar a aplicação.

Ao selecionar o modo claro, a aplicação mantém as mesmas telas e funcionalidades, mas adapta cores, fundos, cartões e textos para a paleta clara. A Figura 41 ilustra a aplicação desse tema na Tela de Busca, em “Categorias” e em “Relatórios”.



Tela de Busca em tema claro | “Categorias” em tema claro | “Relatórios” em tema claro

Figura 41: Aplicação do tema claro nas três principais áreas de consulta e organização do FinTrack.

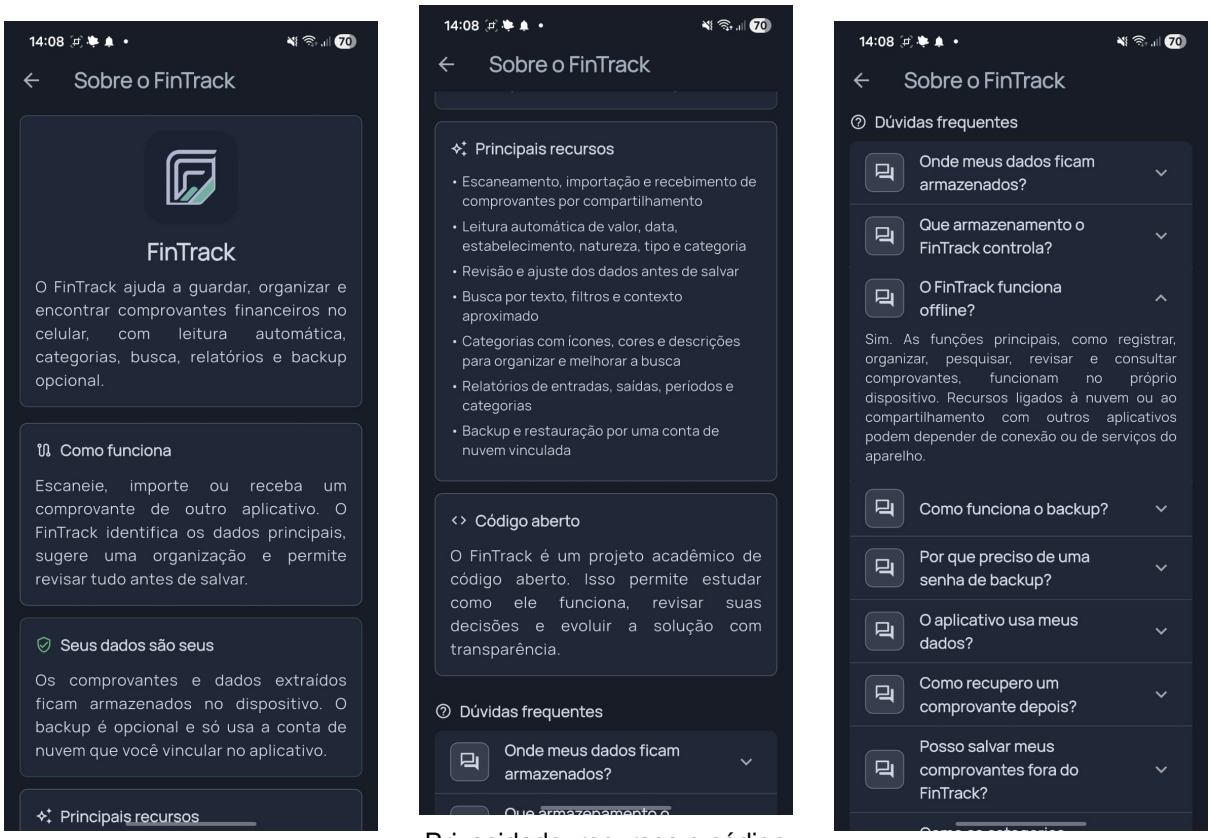
Sobre o FinTrack

A tela “Sobre o FinTrack” é acessada por “Ajustes” > “Sobre o FinTrack”, dentro da seção “Informações da aplicação”. Ela apresenta a identificação do aplicativo, com logotipo, nome “FinTrack” e uma descrição resumida do objetivo da solução: guardar, organizar e encontrar registros financeiros no celular com leitura automática, categorias, busca, relatórios e *backup* opcional.

Na sequência, a tela organiza as informações em blocos. O bloco “Como funciona” explica que o usuário pode escanear, importar ou receber um arquivo de comprovante de outro aplicativo, revisar os dados identificados e salvar o registro. O bloco “Seus dados são seus” resume a abordagem local da aplicação, informando que registros e dados extraídos ficam no dispositivo e que o *backup* em nuvem é opcional. O bloco “Principais recursos” lista funções como escaneamento, importação, recebimento por compartilhamento, revisão dos dados, busca textual e contextual, filtros, categorias, relatórios, *backup* e restauração. O bloco “Código aberto” informa que o FinTrack é um projeto acadêmico de código aberto.

A área “Dúvidas frequentes” usa itens expansíveis para responder perguntas sobre armazenamento local, funcionamento offline, *backup*, senha de *backup*, uso dos dados, recuperação posterior de registros, salvamento fora do FinTrack e contribuição das categorias para organização e busca. Ao final da tela, o botão “Reportar problema” inicia o fluxo de suporte descrito na

seção seguinte, e o rodapé exibe informações de autoria, instituição e versão do aplicativo. A Figura 42 apresenta esses blocos da tela institucional.



Identificação e funcionamento

Privacidade, recursos e código aberto

“Dúvidas frequentes”

Figura 42: Tela “Sobre o FinTrack”, com identificação, funcionamento, privacidade, recursos, FAQ e acesso ao reporte de problemas.

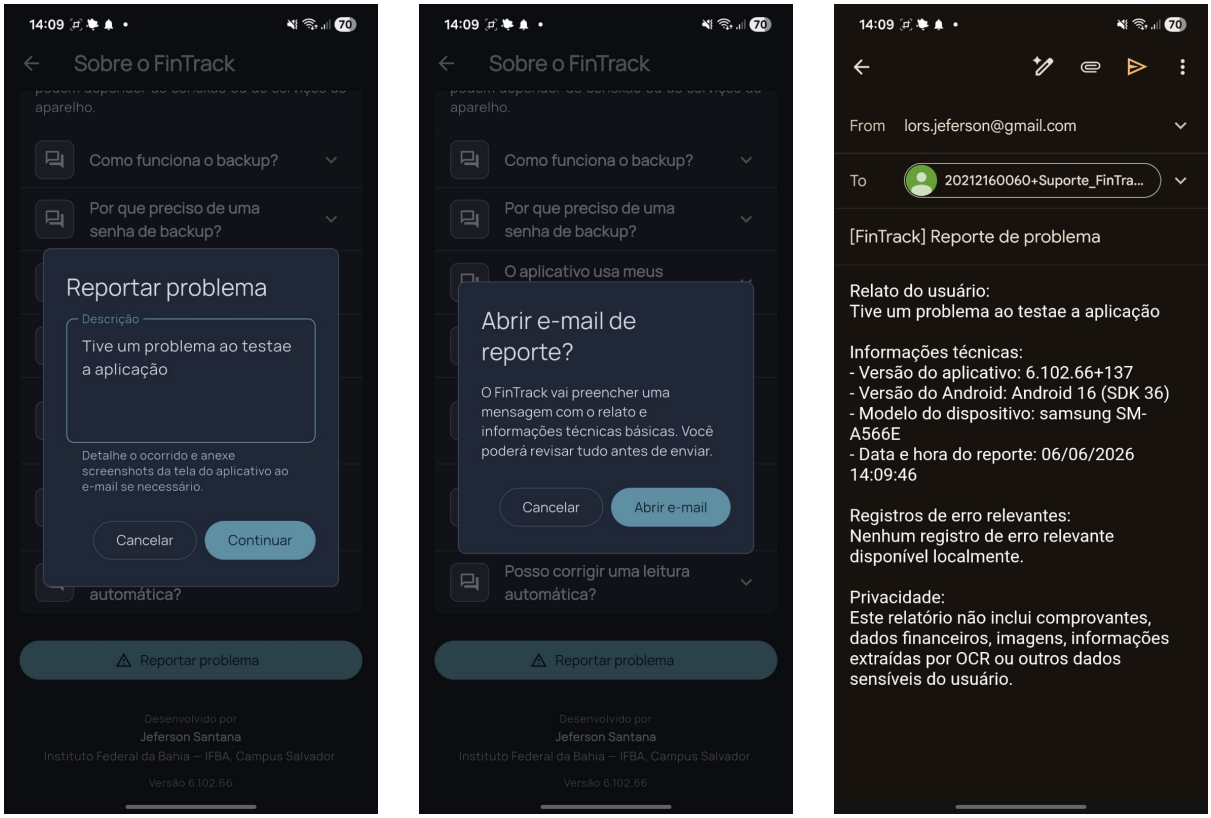
Reporte de problemas

A função de reporte é acessada a partir da tela “Sobre o FinTrack”, pelo caminho “Ajustes” > “Sobre o FinTrack” > “Reportar problema”. Ao tocar no botão, o FinTrack abre o diálogo “Reportar problema”, com o campo “Descrição”, a orientação “Descreva o que aconteceu” e o texto auxiliar que recomenda detalhar o ocorrido e anexar capturas de tela ao e-mail quando necessário. O botão “Continuar” só fica disponível depois que o usuário preenche uma descrição.

Depois da descrição, a aplicação apresenta a confirmação “Abrir e-mail de reporte?”. Essa etapa informa que o FinTrack preencherá uma mensagem com o relato e informações técnicas básicas, mas que o usuário poderá revisar tudo antes de enviar. Ao acionar “Abrir e-mail”, o sistema abre o aplicativo de e-mail do dispositivo com destinatário de suporte, assunto “[FinTrack] Reporte de problema” e corpo pré-preenchido.

O corpo do e-mail contém o “Relato do usuário”, a seção “Informações técnicas”, com versão do aplicativo, versão do Android, modelo do dispositivo e data/hora do reporte, a seção “Registros de erro relevantes”, com registros locais recentes quando existirem, e uma

declaração de privacidade. Essa declaração informa que o relatório não inclui comprovantes, dados financeiros, imagens, informações extraídas por OCR ou outros dados sensíveis. Caso o aplicativo de e-mail não possa ser aberto, o FinTrack exibe a mensagem “Não foi possível abrir o aplicativo de e-mail”. A Figura 43 mostra o preenchimento da descrição, a confirmação e o e-mail gerado.



Diálogo “Reportar problema” | Confirmação para abrir e-mail | E-mail pré-preenchido

Figura 43: Fluxo de reporte de problemas a partir da tela “Sobre o FinTrack”, com descrição, confirmação e e-mail revisável pelo usuário.

Funcionamento sem internet

O FinTrack mantém as funções principais disponíveis sem internet, incluindo captura de comprovantes físicos, importação de arquivos, revisão de dados extraídos, salvamento de registros, consulta à lista e aos detalhes, edição, filtros, ordenação, busca em registros já indexados, categorias e autenticação local. Recursos ligados à conta Google, *backup*, restauração e enriquecimento cadastral ou fiscal dependem de conexão quando forem executados.

Cuidados recomendados

Recomenda-se revisar campos sinalizados com baixa confiança antes de salvar registros, especialmente valor, data e estabelecimento. Também é importante manter a senha de *backup* em local seguro, executar *backups* periodicamente quando a proteção em nuvem estiver configurada e evitar remover manualmente arquivos internos da aplicação, pois isso pode comprometer a associação entre registros e imagens.

Falhas e suporte

Quando houver falhas de processamento, conexão, autenticação, *backup*, restauração ou enriquecimento automático, o usuário deve seguir a orientação exibida pela interface. Se o problema persistir, a função “Reportar problema”, disponível em “Sobre o FinTrack”, pode ser usada para enviar um relato com informações técnicas básicas, sem incluir comprovantes, imagens ou dados financeiros sensíveis no relatório gerado automaticamente.

7 Limitações

Embora o FinTrack atenda ao objetivo de centralizar, estruturar e recuperar registros de comprovantes de transações financeiras em uma aplicação móvel *offline-first*, algumas limitações devem ser reconhecidas. Essas limitações decorrem tanto de escolhas arquiteturais assumidas em favor da privacidade e da execução local quanto das restrições práticas de escopo, tempo e recursos disponíveis para o desenvolvimento do trabalho.

Cobertura de testes

A aplicação possui testes automatizados unitários e de *widgets*, voltados principalmente às regras de negócio, serviços de aplicação, parsers, formatações e componentes de interface. Entretanto, não conta com uma suíte formal automatizada de testes de integração cobrindo fluxos completos em dispositivo, como escaneamento, OCR, persistência, busca, *backup* e restauração em conjunto. Esses cenários foram validados manualmente em dispositivos Android físicos reais, o que mitiga parte do risco, mas não substitui a rastreabilidade, a repetibilidade e a execução recorrente oferecidas por uma suíte automatizada de integração.

A cobertura instrumentada também se concentra nos trechos exercitáveis por testes unitários e de *widgets*. Pontos de entrada da aplicação e integrações com APIs nativas ou serviços externos foram excluídos pontualmente da métrica de cobertura para evitar distorções, pois sua validação depende de permissões Android, credenciais, canais nativos ou execução em dispositivo físico.

Qualidade do OCR e da extração

O reconhecimento óptico de caracteres está sujeito à qualidade da imagem capturada ou importada. Comprovantes borrados, cortados, amassados, com baixa iluminação, fontes degradadas ou leiautes incomuns podem reduzir a precisão do OCR e, consequentemente, prejudicar a extração automática de valor, data, estabelecimento, meio de pagamento e demais campos estruturados. O FinTrack reduz esse impacto por meio de pré-processamento, indicadores de confiança e revisão manual antes da confirmação, mas não elimina integralmente a necessidade de conferência pelo usuário em documentos de baixa qualidade.

Controle da interface de escaneamento

O fluxo de escaneamento utiliza um componente externo de scanner de documentos, integrado à aplicação por meio de biblioteca nativa. Por esse motivo, o FinTrack não controla integralmente os textos, botões e etapas exibidos durante a captura. Como consequência, parte

da interface do scanner pode não seguir a mesma padronização textual adotada no restante da aplicação em português, além de apresentar ações adicionais, redundantes ou pouco ajustáveis ao fluxo específico do sistema. Essa limitação não compromete o processamento posterior do registro, mas reduz a capacidade de refinamento da experiência durante a etapa de escaneamento.

Variação dos formatos de comprovantes

Os parsers implementados contemplam padrões relevantes de documentos fiscais, recibos, pagamentos com cartão, transferências digitais e casos genéricos de fallback. Ainda assim, comprovantes financeiros apresentam grande diversidade de formatos, abreviações, disposições visuais e nomenclaturas. Por esse motivo, documentos fora dos padrões previstos podem ter extração incompleta ou exigir correção manual, especialmente quando os dados relevantes aparecem com rótulos pouco usuais ou em posições não contempladas pelas regras atuais.

Dependência de serviços externos

Apesar de a aplicação permanecer funcional sem internet para registro, organização, consulta e recuperação local dos registros de comprovantes, algumas etapas automáticas de enriquecimento dependem de serviços externos. Consultas à BrasilAPI e a serviços fiscais SEFAZ ou gov.br podem falhar por indisponibilidade, mudança de formato das respostas, alteração de regras de acesso, instabilidade de rede ou ausência de dados públicos para determinado documento. Nesses casos, o FinTrack preserva o fluxo local e utiliza informações obtidas por OCR e caches existentes, mas o enriquecimento fiscal ou cadastral pode ficar incompleto.

Busca semântica local

A busca semântica do FinTrack utiliza um modelo DistilUSE multilíngue quantizado, executado localmente por meio do ONNX Runtime. Essa escolha preserva a privacidade dos dados e evita o envio de textos financeiros a serviços de PLN em nuvem, mas impõe limitações de precisão quando comparada a modelos maiores executados em infraestrutura remota. Além disso, a geração e a indexação dos *embeddings* podem ocorrer em segundo plano após o salvamento do registro, o que significa que resultados contextuais de registros recém-salvos podem não estar imediatamente disponíveis.

Modelos maiores de PLN ou LLMs poderiam ampliar a capacidade de interpretação contextual e melhorar determinados casos de extração ou classificação, mas tenderiam a elevar o custo computacional, o tempo de processamento e o consumo de memória em execução local. Alternativamente, seu uso em nuvem introduziria dependência de conectividade e maior exposição de dados sensíveis, contrariando parcialmente a diretriz de privacidade adotada pela solução.

Desempenho no dispositivo

O desempenho de etapas como escaneamento, OCR, pré-processamento de imagem, geração de *embeddings* e busca vetorial depende das características do dispositivo utilizado. Embora o sistema seja compatível com Android 8.0 (API 26) ou superior, dispositivos mais

antigos, com menor capacidade de processamento ou pouca memória disponível, podem apresentar maior latência em operações computacionalmente mais custosas. Essa limitação decorre diretamente da opção por processamento *on-device*, adotada para preservar privacidade e funcionamento local.

***Backup* e recuperação**

O *backup* em nuvem é opcional e depende de conta de nuvem vinculada, autorização de acesso, conexão com a internet e senha de *backup* definida pelo usuário. Como os arquivos são criptografados localmente antes do envio ao armazenamento remoto, a perda da senha impede a descriptografia e, consequentemente, a recuperação dos dados.

Além disso, tokens de acesso podem expirar ou ser revogados, exigindo nova autenticação, e a restauração depende da disponibilidade do serviço de armazenamento em nuvem adotado, bem como da permanência do arquivo de *backup* no espaço remoto associado à aplicação. Essas restrições são compatíveis com o modelo de segurança adotado, mas tornam a recuperação dependente da guarda correta da senha e da manutenção de *backups* recentes pelo usuário.

Armazenamento local

Os arquivos associados aos registros de comprovantes são armazenados no sistema de arquivos privado da aplicação, enquanto os metadados e dados extraídos são persistidos no SQLite. Essa separação melhora a organização e evita inflar o banco com BLOBs, mas o espaço ocupado no dispositivo cresce conforme o volume de registros e arquivos armazenados. O FinTrack informa o uso de armazenamento e adota estratégias de controle, porém a gestão de espaço continua dependente do comportamento do usuário, especialmente em bases com grande quantidade de imagens digitalizadas ou arquivos importados.

Escopo de plataforma

O FinTrack foi desenvolvido para dispositivos Android e não contempla execução em iOS, web ou desktop. Essa restrição decorre tanto do escopo definido para o trabalho quanto das dependências técnicas adotadas na implementação, como Google ML Kit para scanner, OCR e leitura de QR Code, ONNX Runtime para geração local de *embeddings*, *sqlite-vector* para busca vetorial, Flutter Secure Storage/Android Keystore para proteção de segredos e permissões específicas do ambiente Android.

A adaptação para outro ecossistema exigiria validação de compatibilidade das bibliotecas, adaptação dos fluxos nativos, revisão das permissões, ajustes de empacotamento e testes específicos. Também não há integração com sistemas contábeis, plataformas corporativas de reembolso ou ERPs, pois o escopo do trabalho está centrado em pessoas físicas e na gestão pessoal de registros de comprovantes de transações financeiras.

Segurança condicionada ao ambiente do dispositivo

Recursos como autenticação local, armazenamento seguro de segredos e proteção de acesso dependem dos mecanismos oferecidos pelo sistema operacional e pelo dispositivo

utilizado. O aplicativo pode empregar PIN, biometria e armazenamento seguro quando disponíveis, mas não substitui boas práticas do usuário, como manter o dispositivo bloqueado, não compartilhar a senha de *backup* e preservar a conta de nuvem utilizada para recuperação. Além disso, dispositivos comprometidos, com configurações de segurança enfraquecidas ou sem mecanismos nativos adequados podem reduzir a efetividade dessas proteções.

Validação empírica

A validação do FinTrack foi realizada por meio de testes automatizados e validações manuais em dispositivos Android físicos reais, com comprovantes representativos utilizados durante o desenvolvimento. Entretanto, não foi construído um grande conjunto público e anotado de comprovantes para medir formalmente métricas como acurácia de OCR, precisão da extração, qualidade da categorização e relevância da busca semântica.

Nesse sentido, a avaliação demonstra a viabilidade técnica e funcional da solução, mas não quantifica de forma estatística o desempenho dos componentes de OCR, extração e recuperação semântica em uma base ampla e padronizada. Os testes automatizados verificam o comportamento do sistema a partir dos resultados fornecidos pelo OCR e pelos *embeddings*, mas não avaliam isoladamente a qualidade intrínseca desses componentes, tarefa que exigiria base de referência e métricas específicas.

8 Conclusão

Este trabalho apresentou o FinTrack, uma aplicação móvel Android voltada ao registro, à organização e à recuperação de comprovantes de transações financeiras no uso pessoal cotidiano. A solução foi concebida a partir da constatação de que a quantidade de comprovantes físicos e digitais gerados no cotidiano do brasileiro tem aumentado, enquanto a gestão desses documentos tende a permanecer dispersa e pouco estruturada.

A proposta do FinTrack fundamenta-se na centralidade do comprovante como objeto de gestão, na redução do esforço de interação do usuário e na preservação da autonomia sobre os dados. Para isso, a aplicação integra captura e importação de arquivos, extração automatizada de dados por OCR, enriquecimento fiscal ou cadastral quando aplicável, categorização, filtros, relatórios e mecanismos de busca textual e contextual, mantendo as funcionalidades essenciais em uma abordagem *offline-first*.

Durante o desenvolvimento e a validação da solução, observou-se que o FinTrack atende aos principais requisitos definidos para o escopo do trabalho. Os testes automatizados e as validações manuais realizadas indicaram a viabilidade técnica da aplicação, especialmente nos fluxos de registro, revisão, persistência, recuperação, organização, proteção local e *backup* criptografado opcional.

Entretanto, a solução não elimina todas as limitações inerentes ao domínio tratado. A qualidade do OCR e da extração depende das condições dos documentos processados; os formatos de comprovantes apresentam grande diversidade; e a busca semântica local possui restrições associadas ao uso de modelos compactos executados no próprio dispositivo. Além disso, a avaliação realizada não substitui uma validação empírica ampla com usuários finais em situações reais de uso.

Como impacto esperado, a solução pode contribuir para tornar a gestão pessoal de comprovantes menos onerosa e mais sistemática, reduzindo a dependência de buscas manuais em múltiplos ambientes físicos e digitais. Ao centralizar os registros no próprio dispositivo, com apoio de extração automatizada, organização estruturada e busca contextual, o FinTrack tende a favorecer a preservação e a recuperação desses documentos em situações cotidianas, como controle de gastos, comprovação de despesas, reembolsos ou resolução de divergências comerciais. Além disso, sua abordagem local e o uso opcional de *backup* criptografado reforçam a autonomia do usuário sobre informações financeiras sensíveis.

Dessa forma, conclui-se que o FinTrack representa uma contribuição válida no contexto de soluções voltadas à gestão pessoal de comprovantes de transações financeiras, ao oferecer uma alternativa especializada, local, revisável e orientada à recuperação eficiente desses documentos. Como desdobramentos futuros, destacam-se a ampliação dos testes de integração, a criação de uma base anotada de comprovantes, o refinamento dos parsers e dos modelos locais de PLN e a realização de estudos de uso com usuários finais.

9 Agradecimentos

Eu tenho muita sorte, pois, ao longo da minha vida, pude contar com o amor e o apoio de diversas pessoas. Algumas delas, infelizmente, já não estão mais aqui, mas, sem cada uma delas, eu jamais teria conseguido sequer iniciar uma graduação.

Agradeço, primeiramente, à minha avó e à minha mãe, que sempre se preocuparam em me proporcionar a melhor educação possível, dentro das nossas condições financeiras, que nunca foram fáceis. Desde muito cedo, elas me ensinaram que o estudo era a única forma de transformar a nossa realidade.

Agradeço também ao meu pai de criação e à minha bisavó, ambos já falecidos, que sempre fizeram tudo o que podiam para pavimentar o meu caminho até aqui. Meu pai, sempre que tinha oportunidade, dizia que "ia fazer a minha festa de formatura". Com seu jeito simples, essa era a forma que encontrava de demonstrar que sabia o quanto essa fase era importante.

À minha tia de coração, meu muito obrigado por ter "comprado a minha briga" desde o dia em que me conheceu. Ela sempre me apoiou de inúmeras formas, inclusive nos estudos, e se tornou um dos meus maiores exemplos.

Ao meu grande amigo Daniel, agradeço por todo o apoio e incentivo ao longo dessa jornada. Em um daqueles momentos em que pensamos em desistir da graduação, eu realmente desisti. Foi ele quem desempenhou um papel fundamental para que eu encontrasse forças para retornar e seguir em frente.

Meu agradecimento especial a todos os amigos que contribuíram para este trabalho, oferecendo sugestões, opiniões e participando dos testes da solução. Em especial, à Ana, que me deu um apoio extraordinário durante a fase de testes funcionais do FinTrack. Quando, despretensiosamente, pedi que realizasse alguns testes, jamais imaginei o tamanho do comprometimento que demonstraria. Sem sua colaboração, a solução certamente seria entregue, mas não com o nível de qualidade e corretude que conseguimos alcançar.

Agradeço também a todos os colegas e amigos que fiz no IFBA, com quem compartilhei momentos inesquecíveis, dividi as dificuldades naturais dessa caminhada e encontrei motivação quando ela parecia faltar. Muitas vezes, quando nem a própria realidade era suficiente para me impulsionar, era a companhia deles que me fazia continuar.

Minha gratidão se estende a todos os profissionais comprometidos em fazer o curso de Análise e Desenvolvimento de Sistemas do IFBA acontecer. Em especial, aos professores Grinaldo e Flávia, com quem tive o privilégio não apenas de assistir às aulas, mas também de compartilhar conversas e reflexões importantes sobre a vida, a profissão e o futuro.

Gostaria de registrar também minha gratidão aos membros da banca examinadora, professor Manoel Neto e professor Luciano Pacífico, pela disponibilidade em avaliar este trabalho e pelas inúmeras contribuições apresentadas durante a defesa. Suas observações foram fundamentais para o aprimoramento deste trabalho e servirão de base para seus desdobramentos futuros.

Agradeço, de forma especial, ao professor Dihego, meu orientador. Embora eu tenha tido pouco tempo de convivência com ele em sala de aula, desde o início admirei sua forma leve e inspiradora de ensinar. Durante a orientação, acolheu minhas ideias desde o primeiro momento, ajudou-me a lapidá-las e sempre me incentivou a seguir em frente, respeitando meu tempo e meu processo até chegarmos à conclusão deste trabalho.

Ao meu amor, deixo um agradecimento impossível de resumir em palavras. Foi ela quem sempre “comprou” minhas ideias de projeto e embarcou comigo em cada uma delas desde o começo, discutindo possibilidades, ouvindo pacientemente minhas reflexões e dividindo comigo tanto as inquietações quanto as conquistas. Foi quem acolheu minhas preocupações, me tranquilizou nos momentos mais difíceis, me tirou da frente do computador quando percebeu que eu precisava descansar, mas que também aceitou remarcar passeios ou voltar para casa mais cedo para que eu pudesse concluir mais uma etapa.

Por fim, espero que um dia eu consiga retribuir todo o apoio que recebi de cada uma das pessoas mencionadas aqui e de tantas outras que, mesmo sem serem citadas, também contribuíram, direta ou indiretamente, para que eu pudesse concluir esta etapa tão importante da minha vida.

Referências

1 Banco Central do Brasil. **Pix**. 2026. Acesso em: 14 abr. 2026. Disponível em: <https://www.bcb.gov.br/estabilidadefinanceira/pix>.

2 ____. **Relatório Anual do Sistema de Pagamentos Instantâneos – SPI 2024**. 2025. P. 9. Acesso em: 12 mar. 2026. Disponível em: <https://www.bcb.gov.br/content/estabilidadefinanceira/relatorios_SPI/relatorio_anual_spi_2024.pdf>.

3 ____. **O brasileiro e sua relação com o dinheiro – 2024**. 2024. Acesso em: 21 jul. 2026. Disponível em: <https://www.bcb.gov.br/content/cedulasemoedas/pesquisabrasileirodinheiro/Apresentacao_brasileiro_relacao_dinheiro_2024.pdf>.

4 Fundação PROCON-SP. **Guarda de documentos**. 2016. Acesso em: 25 fev. 2026. Disponível em: <https://www.procon.sp.gov.br/guarda-de-documentos/>.

5 IBGE. **No Brasil, 88,9% da população de 10 anos ou mais tinha celular em 2024**. 2025. Acesso em: 5 mar. 2026. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/noticias/44032-no-brasil-88-9-da-populacao-de-10-anos-ou-mais-tinha-celular-em-2024>.

6 Google LLC. **Flutter – Build apps for any screen**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://docs.flutter.dev/learn/pathway/how-flutter-works>.

7 ____. **Develop Android apps with Kotlin**. 2026. Acesso em: 7 jun. 2026. Disponível em: <https://developer.android.com/kotlin>.

8 ____. **Writing custom platform-specific code**. 2026. Acesso em: 7 jun. 2026. Disponível em: <https://docs.flutter.dev/platform-integration/platform-channels>.

9 SQLite Consortium. **SQLite**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://www.sqlite.org/about.html>.

10 Simon Binder. **Drift – Reactive persistence library for Flutter and Dart**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://drift.simonbinder.eu>.

11 SQLite Consortium. **SQLite FTS5 Extension**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://www.sqlite.org/fts5.html>.

12 SQLite AI. **sqlite-vector**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://github.com/sqliteai/sqlite-vector>.

13 Google LLC. **Google ML Kit**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://developers.google.com/ml-kit?hl=pt-br>.

14 Hugging Face. **sentence-transformers/distiluse-base-multilingual-cased-v2**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2>.

15 Microsoft. **ONNX Runtime**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://onnxruntime.ai/docs/>.

16 Dart. **image package**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://pub.dev/packages/image>.

17 OpenAI. **Codex**. 2026. Acesso em: 7 jun. 2026. Disponível em: <https://developers.openai.com/codex/overview>.

18 Flutter Secure Storage. **flutter_secure_storage**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://pub.dev/packages/flutter_secure_storage>.

19 Google LLC. **Android Keystore system**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://developer.android.com/privacy-and-security/keystore>.

20 Dart. **cryptography package**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://pub.dev/packages/cryptography>.

21 TURAN, M. S. et al. **Recommendation for Password-Based Key Derivation: Part 1: Storage Applications**. [S.l.], 2010. Acesso em: 8 jun. 2026. Disponível em: <https://csrc.nist.gov/pubs/sp/800/132/final>.

22 Wikipédia. **Google Drive**. 2026. Página da enciclopédia Wikipédia. Acesso em: 14 abr. 2026. Disponível em: <https://pt.wikipedia.org/wiki/Google_Drive>.

23 Flutter. **google_sign_in package**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://pub.dev/packages/google_sign_in>.

24 Google LLC. **Google Drive API**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://developers.google.com/workspace/drive/api/guides/about-sdk?hl=pt-br>.

25 Software Freedom Conservancy. **Git – Fast Version Control**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://git-scm.com>.

26 Stack Overflow. **Stack Overflow Developer Survey 2022**. 2022. Acesso em: 29 mar. 2026. Disponível em: <https://survey.stackoverflow.co/2022/#version-control-version-control-system-prof>.

27 Git Large File Storage. **Git Large File Storage**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://git-lfs.com/>.

28 GitHub Inc. **GitHub**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git#sobre-o-github>.

29 ____. **GitHub Actions – Automate your workflow**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://docs.github.com/pt/actions/get-started/understand-github-actions>.

30 Arnaud Roques. **PlantUML – Open-source tool for UML diagrams**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://plantuml.com>.

31 Google LLC. **Flutter Test – Testing Flutter Apps**. 2024. Acesso em: 29 mar. 2026. Disponível em: <https://docs.flutter.dev/testing/overview>.

32 Linux Test Project. **LCOV**. 2026. Acesso em: 4 jun. 2026. Disponível em: <https://github.com/linux-test-project/lcov>.

33 Flutter Team. **Mock dependencies using Mockito**. 2025. <https://docs.flutter.dev/cookbook/testing/unit/mocking>. Acesso em: 15 de abril de 2026.

34 INTSIG Information Co., Ltd. **CamScanner - PDF Scanner App**. 2026. Aplicativo disponível na App Store. Acesso em: 4 jun. 2026. Disponível em: <https://apps.apple.com/us/app/camscanner-pdf-scanner-app/id388627783>.

35 ____. **CamScanner: Scan, Edit, and Manage Files All in One Place**. 2026. Site oficial. Acesso em: 22 maio 2026. Disponível em: <https://www.camscanner.com/>.

36 Google LLC. **Google Drive**. 2026. Aplicativo disponível na App Store. Acesso em: 4 jun. 2026. Disponível em: <https://apps.apple.com/us/app/google-drive/id507874739>.

37 ____. **How your Google storage works**. 2026. Ajuda oficial da Conta Google. Acesso em: 4 jun. 2026. Disponível em: <https://support.google.com/accounts/answer/9312312?hl=en-EN>.

38 ____. **Scan documents with Google Drive**. 2026. Ajuda oficial do Google Drive. Acesso em: 22 maio 2026. Disponível em: <https://support.google.com/drive/answer/3145835?hl=en-GB>.

39 Expensify, Inc. **Expensify**. 2026. Site oficial do serviço de gestão de despesas. Acesso em: 14 abr. 2026. Disponível em: <https://www.expensify.com/>.

40 ____. **Expensify Mobile App: Expense management on the go**. 2026. Página oficial do aplicativo móvel. Acesso em: 4 jun. 2026. Disponível em: <https://use.expensify.com/expensify-mobile-app>.

41 ____. **Free Features in Expensify**. 2026. Central de ajuda oficial. Acesso em: 22 maio 2026. Disponível em: <https://help.expensify.com/articles/new-expensify/getting-started/Free-Features-in-Expensify>.

42 Mobills. **Mobills: Controle Financeiro Pessoal**. 2026. Aplicativo de controle financeiro pessoal disponível na Google Play Store. Acesso em: 14 abr. 2026. Disponível em: <https://play.google.com/store/apps/details?id=br.com.gerenciadorfinanceiro.controller&hl=pt_BR>.

43 ____. **Mobills - Controle de gastos**. 2026. Aplicativo de controle financeiro pessoal disponível na App Store. Acesso em: 4 jun. 2026. Disponível em: <https://apps.apple.com/br/app/mobills-controle-de-gastos/id921838244>.

44 ____. **Attachments**. 2026. Central de ajuda oficial. Acesso em: 22 maio 2026. Disponível em: <https://mobills.zendesk.com/hc/en-us/sections/360009149653-Attachments>.

45 StatCounter Global Stats. **Mobile & Tablet Operating System Market Share Brazil**. 2026. Dados de janeiro de 2026. Acesso em: 5 jun. 2026. Disponível em: <https://gs.statcounter.com/os-market-share/mobile-ta/brazil>.

46 ICIU Studio. **FinTrack: Track, Plan, Prosper**. 2026. Aplicativo disponível na Google Play Store. Acesso em: 4 jun. 2026. Disponível em: <https://play.google.com/store/apps/details?id=com.sonu.fintrack>.

47 Proper Solutions. **FinTrack - Personal Finance**. 2026. Aplicativo disponível na Google Play Store. Acesso em: 4 jun. 2026. Disponível em: <https://play.google.com/store/apps/details?id=app.web.moneymanager>.

48 Abzer Mobility Team. **FinTrack**. 2026. Aplicativo disponível na Google Play Store. Acesso em: 4 jun. 2026. Disponível em: <https://play.google.com/store/apps/details?id=com.abzer.fintrack>.

49 Valor Technologies. **Fintrack: Expense Tracker**. 2026. Aplicativo disponível na Google Play Store. Acesso em: 4 jun. 2026. Disponível em: <https://play.google.com/store/apps/details?id=com.harshvardhan.fintrack>.

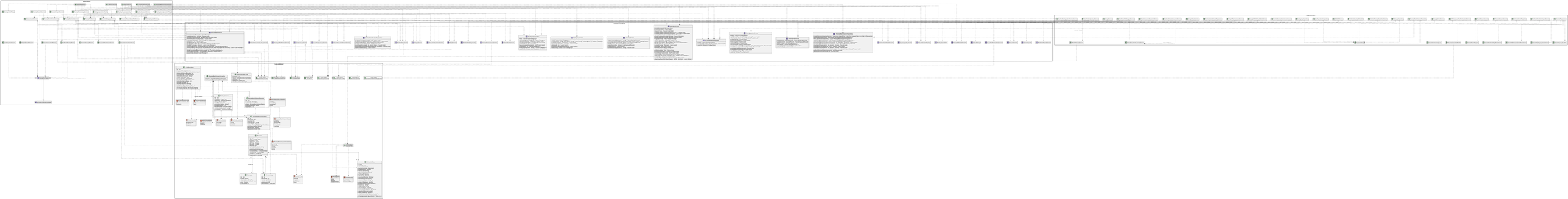
50 cajudevs. **Entendendo o C4 Model: Uma Abordagem para Arquitetura de Software**. Disponível em: <https://medium.com/cajudevs/entendendo-o-c4-model-uma-abordagem-para-arquitetura-de-software-3ed0f007ae66>.

51 ARAúJO, I.; vinelouzada. **SOLID: o que é e quais os 5 princípios da Programação Orientada a Objetos (POO)**. 2024. <https://www.alura.com.br/artigos/solid>. Acesso em: 15 de abril de 2026.

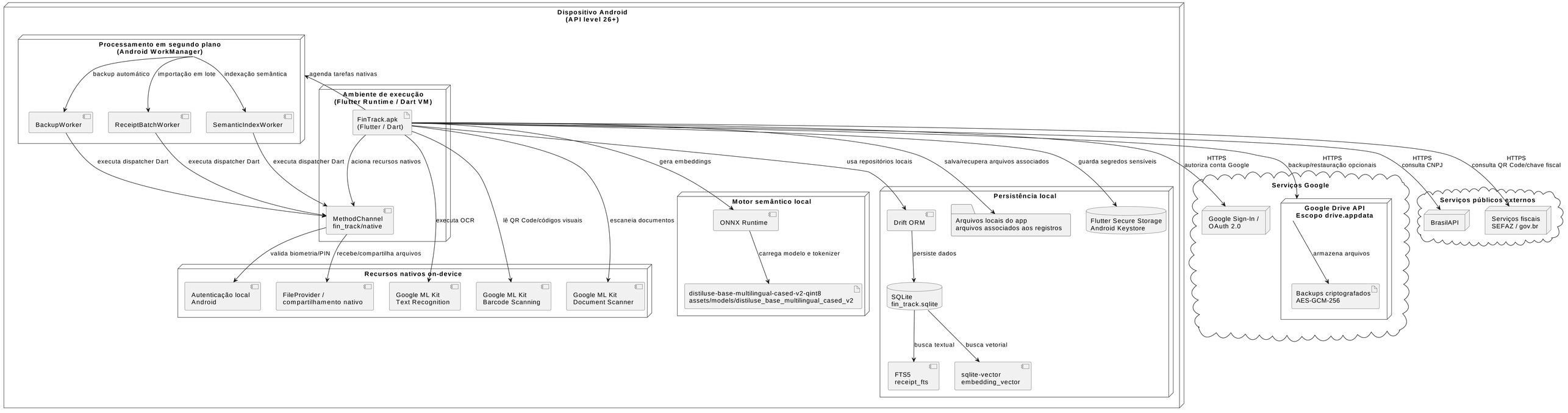
52 SCHNEIDER, J.; SMALLEY, I. O que é desenvolvimento orientado por testes (tdd)? **IBM Think**, jul 2025. Acesso em: 15 de abril de 2026. Disponível em: <https://www.ibm.com/br-pt/think/topics/test-driven-development>.

10 Apêndices

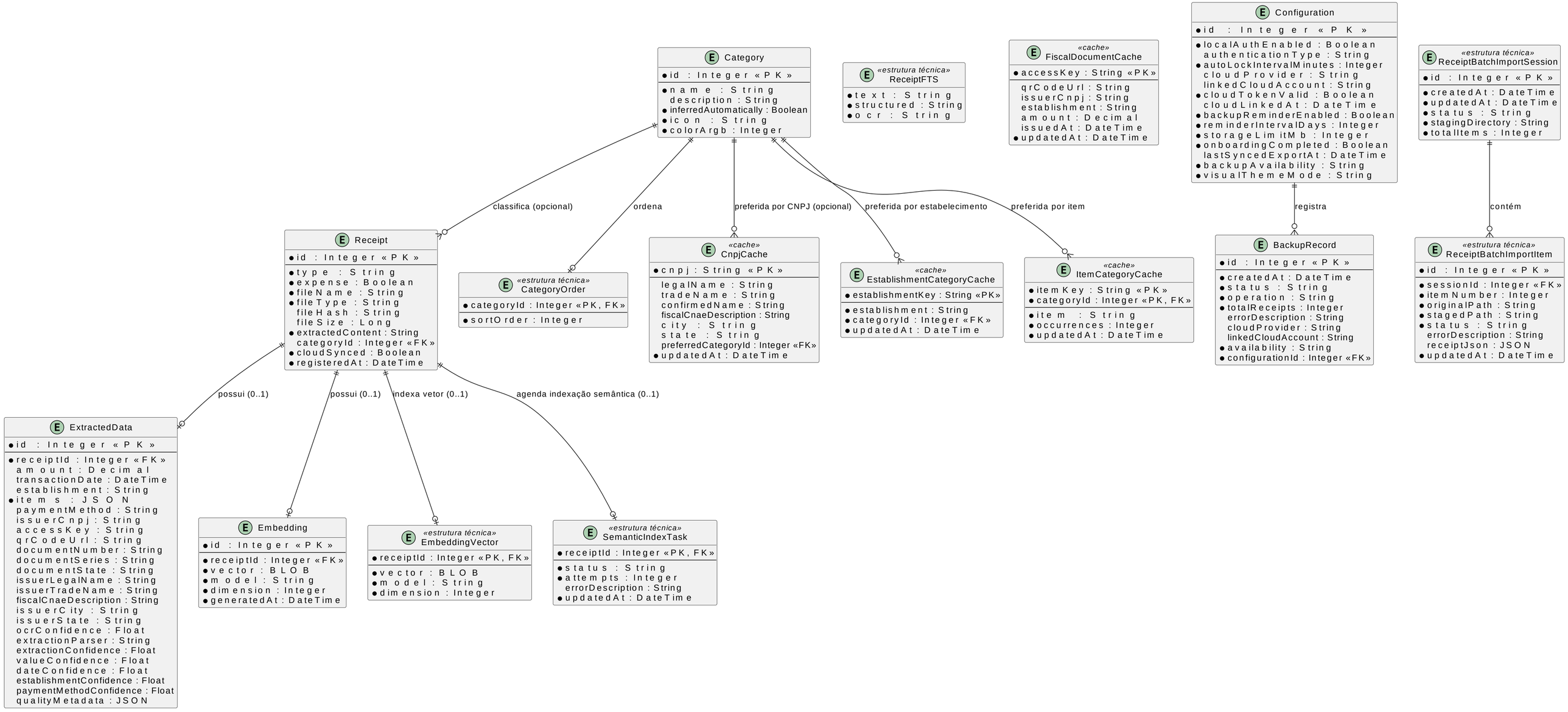
A Diagrama de Classes


Controller
Service Interface
Service Impl
Mapper
Entity
Vo
Dto
Enum
Exception
Utils
Config
Aspect
Interceptor
Filter
Listener
Task
Event
Cache
Lock
Client
Feign
Mq
Redis
Es
Mapper.xml
Service Mapper
Service Entity
Service Vo
Service Dto
Service Enum
Service Exception
Service Utils
Service Config
Service Aspect
Service Interceptor
Service Filter
Service Listener
Service Task
Service Event
Service Cache
Service Lock
Service Client
Service Feign
Service Mq
Service Redis
Service Es
Service Mapper.xml
Service Controller
Service Interface
Service Impl

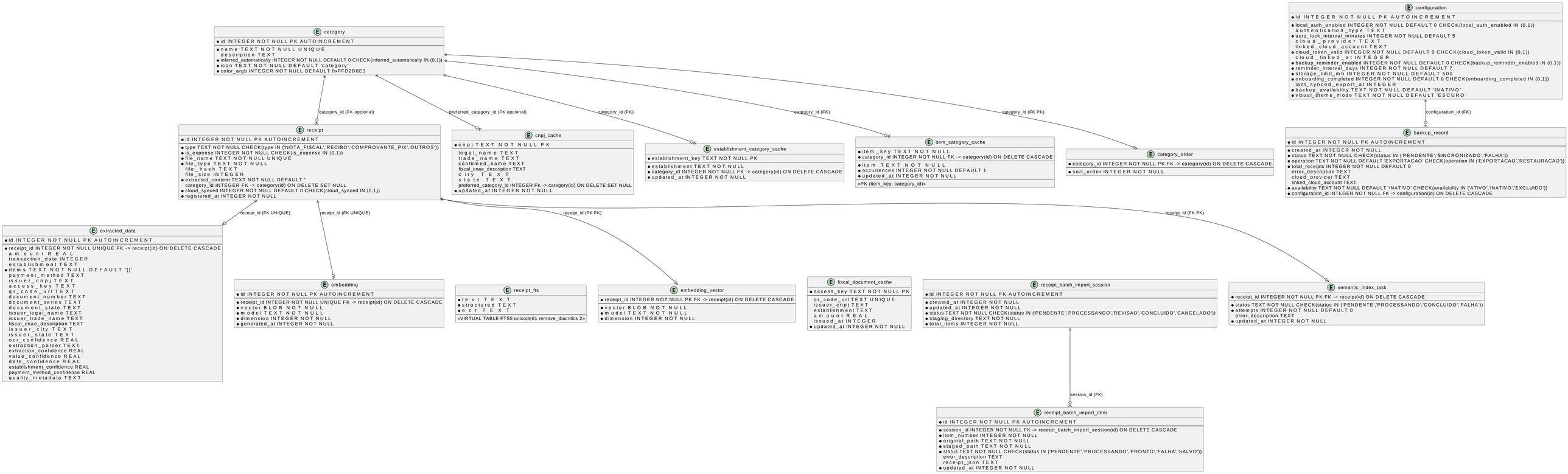
B Diagrama de Implantação


Dispositivo Android
(API level 26+)
Processamento em segundo plano
(Android WorkManager)
backup automático
importação em lote
indexação semântica
BackupWorker
ReceiptBatchWorker
SemanticIndexWorker
agenda tarefas nativas
Ambiente de execução
(Flutter Runtime / Dart VM)
FinTrack.apk
(Flutter / Dart)
executa dispatcher Dart
executa dispatcher Dart
executa dispatcher Dart
aciona recursos nativos
MethodChannel
fin_track/native
executa OCR
lê QR Code/códigos visuais
escaneia documentos
gera embeddings
usa repositórios locais
salva/recupera arquivos associados
guarda segredos sensíveis
HTTPS
autoriza conta Google
HTTPS
backup/restauração opcionais
HTTPS
consulta CNPJ
HTTPS
consulta QR Code/chave fiscal
valida biometria/PIN
recebe/compartilha arquivos
Recursos nativos on-device
Autenticação local
Android
FileProvider /
compartilhamento nativo
Google ML Kit
Text Recognition
Google ML Kit
Barcode Scanning
Google ML Kit
Document Scanner
Motor semântico local
ONNX Runtime
carrega modelo e tokenizer
distiluse-base-multilingual-cased-v2-qint8
assets/models/distiluse_base_multilingual_cased_v2
Persistência local
Drift ORM
Arquivos locais do app
arquivos associados aos registros
Flutter Secure Storage
Android Keystore
persiste dados
SQLite
fin_track.sqlite
busca textual
busca vetorial
FTS5
receipt_fts
sqlite-vector
embedding_vector
Serviços Google
Google Sign-In /
OAuth 2.0
Google Drive API
Escopo drive.appdata
armazena arquivos
Backups criptografados
AES-GCM-256
Serviços públicos externos
BrasilAPI
Serviços fiscais
SEFAZ / gov.br

C Modelo Lógico do Banco de Dados


Category
id : Integer «PK»
name : String
description : String
inferredAutomatically : Boolean
icon : String
colorArgb : Integer
«estrutura técnica»
ReceiptFTS
text : String
structured : String
ocr : String
«cache»
FiscalDocumentCache
accessKey : String «PK»
qrCodeUrl : String
issuerCnpj : String
establishment : String
amount : Decimal
issuedAt : DateTime
updatedAt : DateTime
Configuration
id : Integer «PK»
localAuthEnabled : Boolean
authenticationType : String
autoLockIntervalMinutes : Integer
cloudProvider : String
linkedCloudAccount : String
cloudTokenValid : Boolean
cloudLinkedAt : DateTime
backupReminderEnabled : Boolean
reminderIntervalDays : Integer
storageLimitMb : Integer
onboardingCompleted : Boolean
lastSyncedExportAt : DateTime
backupAvailability : String
visualThemeMode : String
«estrutura técnica»
ReceiptBatchImportSession
id : Integer «PK»
createdAt : DateTime
updatedAt : DateTime
status : String
stagingDirectory : String
totalItems : Integer
classifica (opcional)
ordena
preferida por CNPJ (opcional)
preferida por estabelecimento
preferida por item
registra
contém
Receipt
id : Integer «PK»
type : String
expense : Boolean
fileName : String
fileType : String
fileHash : String
fileSize : Long
extractedContent : String
categoryId : Integer «FK»
cloudSynced : Boolean
registeredAt : DateTime
«estrutura técnica»
CategoryOrder
categoryId : Integer «PK, FK»
sortOrder : Integer
«cache»
CnpjCache
cnpj : String «PK»
legalName : String
tradeName : String
confirmedName : String
fiscalCnaeDescription : String
city : String
state : String
preferredCategoryId : Integer «FK»
updatedAt : DateTime
«cache»
EstablishmentCategoryCache
establishmentKey : String «PK»
establishment : String
categoryId : Integer «FK»
updatedAt : DateTime
«cache»
ItemCategoryCache
itemKey : String «PK»
categoryId : Integer «PK, FK»
item : String
occurrences : Integer
updatedAt : DateTime
BackupRecord
id : Integer «PK»
createdAt : DateTime
status : String
operation : String
totalReceipts : Integer
errorDescription : String
cloudProvider : String
linkedCloudAccount : String
availability : String
configurationId : Integer «FK»
«estrutura técnica»
ReceiptBatchImportItem
id : Integer «PK»
sessionId : Integer «FK»
itemNumber : Integer
originalPath : String
stagedPath : String
status : String
errorDescription : String
receiptJson : JSON
updatedAt : DateTime
possui (0..1)
possui (0..1)
indexa vetor (0..1)
agenda indexação semântica (0..1)
ExtractedData
id : Integer «PK»
receiptId : Integer «FK»
amount : Decimal
transactionDate : DateTime
establishment : String
items : JSON
paymentMethod : String
issuerCnpj : String
accessKey : String
qrCodeUrl : String
documentNumber : String
documentSeries : String
documentState : String
issuerLegalName : String
issuerTradeName : String
fiscalCnaeDescription : String
issuerCity : String
issuerState : String
ocrConfidence : Float
extractionParser : String
extractionConfidence : Float
valueConfidence : Float
dateConfidence : Float
establishmentConfidence : Float
paymentMethodConfidence : Float
qualityMetadata : JSON
Embedding
id : Integer «PK»
receiptId : Integer «FK»
vector : BLOB
model : String
dimension : Integer
generatedAt : DateTime
«estrutura técnica»
EmbeddingVector
receiptId : Integer «PK, FK»
vector : BLOB
model : String
dimension : Integer
«estrutura técnica»
SemanticIndexTask
receiptId : Integer «PK, FK»
status : String
attempts : Integer
errorDescription : String
updatedAt : DateTime

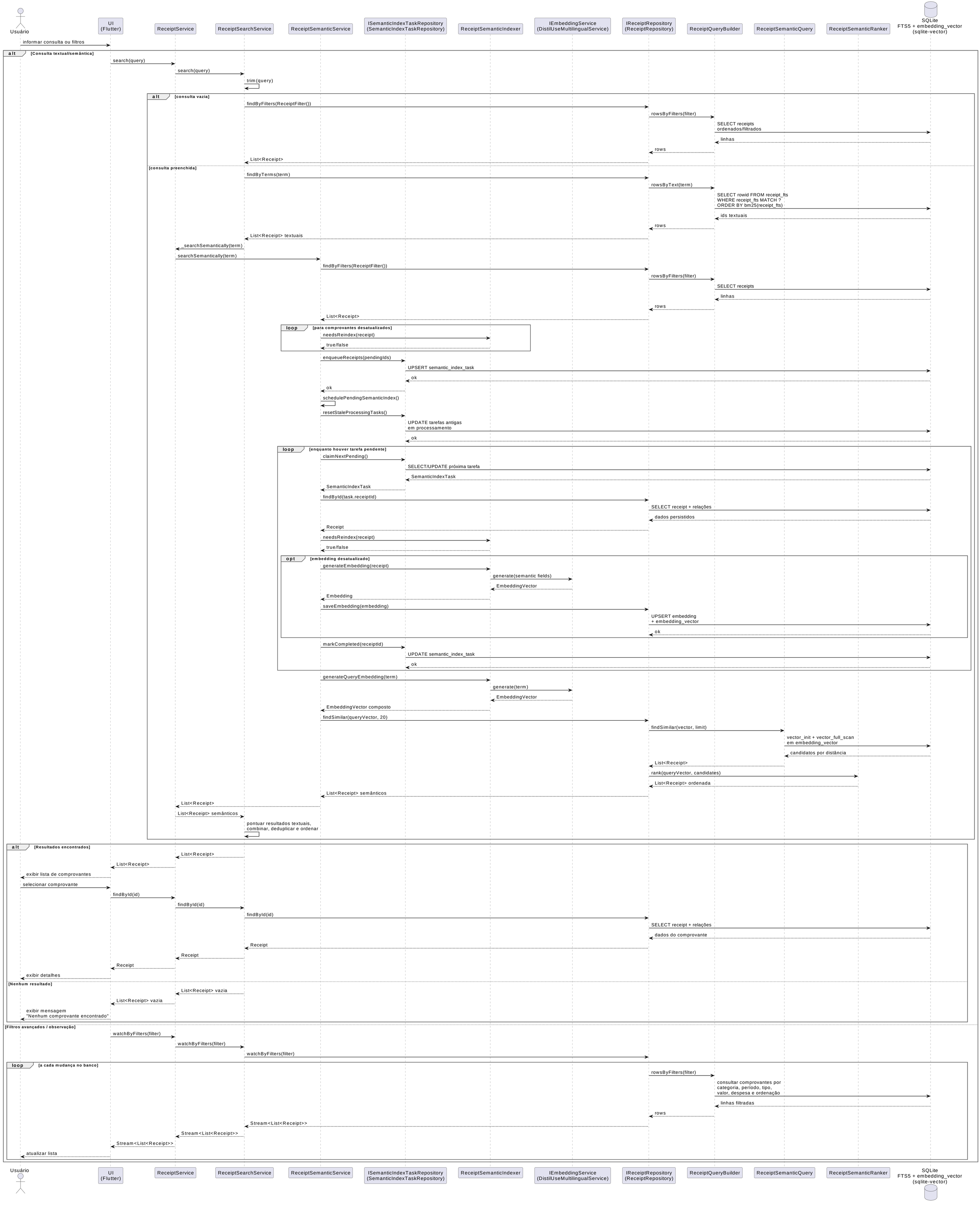
D Modelo Físico do Banco de Dados


category
id INTEGER NOT NULL PK AUTOINCREMENT
name TEXT NOT NULL UNIQUE
description TEXT
inferred_automatically INTEGER NOT NULL DEFAULT 0 CHECK(inferred_automatically IN (0,1))
icon TEXT NOT NULL DEFAULT 'category'
color_argb INTEGER NOT NULL DEFAULT 0xFFD2D8E3
configuration
id INTEGER NOT NULL PK AUTOINCREMENT
local_auth_enabled INTEGER NOT NULL DEFAULT 0 CHECK(local_auth_enabled IN (0,1))
authentication_type TEXT
auto_lock_interval_minutes INTEGER NOT NULL DEFAULT 5
cloud_provider TEXT
linked_cloud_account TEXT
cloud_token_valid INTEGER NOT NULL DEFAULT 0 CHECK(cloud_token_valid IN (0,1))
cloud_linked_at INTEGER
backup_reminder_enabled INTEGER NOT NULL DEFAULT 0 CHECK(backup_reminder_enabled IN (0,1))
reminder_interval_days INTEGER NOT NULL DEFAULT 7
storage_limit_mb INTEGER NOT NULL DEFAULT 500
onboarding_completed INTEGER NOT NULL DEFAULT 0 CHECK(onboarding_completed IN (0,1))
last_synced_export_at INTEGER
backup_availability TEXT NOT NULL DEFAULT 'INATIVO'
visual_theme_mode TEXT NOT NULL DEFAULT 'ESCURO'
receipt
id INTEGER NOT NULL PK AUTOINCREMENT
type TEXT NOT NULL CHECK(type IN ('NOTA_FISCAL','RECIBO','COMPROVANTE_PIX','OUTROS'))
is_expense INTEGER NOT NULL CHECK(is_expense IN (0,1))
file_name TEXT NOT NULL UNIQUE
file_type TEXT NOT NULL
file_hash TEXT
file_size INTEGER
extracted_content TEXT NOT NULL DEFAULT ''
category_id INTEGER FK -> category(id) ON DELETE SET NULL
cloud_synced INTEGER NOT NULL DEFAULT 0 CHECK(cloud_synced IN (0,1))
registered_at INTEGER NOT NULL
cnpj_cache
cnpj TEXT NOT NULL PK
legal_name TEXT
trade_name TEXT
confirmed_name TEXT
fiscal_cnae_description TEXT
city TEXT
state TEXT
preferred_category_id INTEGER FK -> category(id) ON DELETE SET NULL
updated_at INTEGER NOT NULL
establishment_category_cache
establishment_key TEXT NOT NULL PK
establishment TEXT NOT NULL
category_id INTEGER NOT NULL FK -> category(id) ON DELETE CASCADE
updated_at INTEGER NOT NULL
item_category_cache
item_key TEXT NOT NULL
category_id INTEGER NOT NULL FK -> category(id) ON DELETE CASCADE
item TEXT NOT NULL
occurrences INTEGER NOT NULL DEFAULT 1
updated_at INTEGER NOT NULL
«PK (item_key, category_id)»
category_order
category_id INTEGER NOT NULL PK FK -> category(id) ON DELETE CASCADE
sort_order INTEGER NOT NULL
backup_record
id INTEGER NOT NULL PK AUTOINCREMENT
created_at INTEGER NOT NULL
status TEXT NOT NULL CHECK(status IN ('PENDENTE','SINCRONIZADO','FALHA'))
operation TEXT NOT NULL DEFAULT 'EXPORTACAO' CHECK(operation IN ('EXPORTACAO','RESTAURACAO'))
total_receipts INTEGER NOT NULL DEFAULT 0
error_description TEXT
cloud_provider TEXT
linked_cloud_account TEXT
availability TEXT NOT NULL DEFAULT 'INATIVO' CHECK(availability IN ('ATIVO','INATIVO','EXCLUIDO'))
configuration_id INTEGER NOT NULL FK -> configuration(id) ON DELETE CASCADE
extracted_data
id INTEGER NOT NULL PK AUTOINCREMENT
receipt_id INTEGER NOT NULL UNIQUE FK -> receipt(id) ON DELETE CASCADE
amount REAL
transaction_date INTEGER
establishment TEXT
items TEXT NOT NULL DEFAULT '[]'
payment_method TEXT
issuer_cnpj TEXT
access_key TEXT
qr_code_url TEXT
document_number TEXT
document_series TEXT
document_state TEXT
issuer_legal_name TEXT
issuer_trade_name TEXT
fiscal_cnae_description TEXT
issuer_city TEXT
issuer_state TEXT
ocr_confidence REAL
extraction_parser TEXT
extraction_confidence REAL
value_confidence REAL
date_confidence REAL
establishment_confidence REAL
payment_method_confidence REAL
quality_metadata TEXT
embedding
id INTEGER NOT NULL PK AUTOINCREMENT
receipt_id INTEGER NOT NULL UNIQUE FK -> receipt(id) ON DELETE CASCADE
vector BLOB NOT NULL
model TEXT NOT NULL
dimension INTEGER NOT NULL
generated_at INTEGER NOT NULL
receipt_fts
text TEXT
structured TEXT
ocr TEXT
«VIRTUAL TABLE FTS5 unicode61 remove_diacritics 2»
embedding_vector
receipt_id INTEGER NOT NULL PK FK -> receipt(id) ON DELETE CASCADE
vector BLOB NOT NULL
model TEXT NOT NULL
dimension INTEGER NOT NULL
fiscal_document_cache
access_key TEXT NOT NULL PK
qr_code_url TEXT UNIQUE
issuer_cnpj TEXT
establishment TEXT
amount REAL
issued_at INTEGER
updated_at INTEGER NOT NULL
receipt_batch_import_session
id INTEGER NOT NULL PK AUTOINCREMENT
created_at INTEGER NOT NULL
updated_at INTEGER NOT NULL
status TEXT NOT NULL CHECK(status IN ('PENDENTE','PROCESSANDO','REVISAO','CONCLUIDO','CANCELADO'))
staging_directory TEXT NOT NULL
total_items INTEGER NOT NULL
semantic_index_task
receipt_id INTEGER NOT NULL PK FK -> receipt(id) ON DELETE CASCADE
status TEXT NOT NULL CHECK(status IN ('PENDENTE','PROCESSANDO','CONCLUIDO','FALHA'))
attempts INTEGER NOT NULL DEFAULT 0
error_description TEXT
updated_at INTEGER NOT NULL
receipt_batch_import_item
id INTEGER NOT NULL PK AUTOINCREMENT
session_id INTEGER NOT NULL FK -> receipt_batch_import_session(id) ON DELETE CASCADE
item_number INTEGER NOT NULL
original_path TEXT NOT NULL
staged_path TEXT NOT NULL
status TEXT NOT NULL CHECK(status IN ('PENDENTE','PROCESSANDO','PRONTO','FALHA','SALVO'))
error_description TEXT
receipt_json TEXT
updated_at INTEGER NOT NULL
category_id (FK opcional)
preferred_category_id (FK opcional)
category_id (FK)
category_id (FK)
category_id (FK PK)
configuration_id (FK)
receipt_id (FK UNIQUE)
receipt_id (FK UNIQUE)
receipt_id (FK PK)
receipt_id (FK PK)
session_id (FK)

E Diagrama de Sequência — Registro no FinTrack

Usuário
UI (Flutter)
ReceiptBatchImportService
ReceiptService
ReceiptImportService
ReceiptFileService
IDocumentScannerService (MLKitDocumentScannerService)
IImageService (ImageService)
StorageLimitPolicy
ReceiptProcessingService
ImagePreprocessorService
GoogleMLKitVisualCodeService
GoogleMLKitService
DataExtractorService + ReceiptOcrParser
ReceiptEnrichmentService
CachedFiscalDocumentService
CachedCnpjLookupService
ReceiptCategoryService
CachedCategoryPreferenceService
FinancialNatureClassifierService
ReceiptSemanticService
ISemanticIndexTaskRepository (SemanticIndexTaskRepository)
ReceiptSemanticIndexer
IEmbeddingService (DistilUseMultilingualService)
IReceiptBatchImportRepository (ReceiptBatchImportRepository)
ICategoryRepository (CategoryRepository)
IReceiptRepository (ReceiptRepository)
SQLite Drift + FTS5 + embedding_vector
escanear ou importar comprovante
alt [Escanear documento]
scanDocument()
scanDocument()
scanDocument()
File
File
File
File
[Scanner indisponível / usar câmera]
captureImage()
captureImage()
captureImage()
capture()
File
File
File
File
[Importar arquivo ou lote]
importFiles()
importFiles()
importFiles()
importMany()
List<File>
List<File>
List<File>
List<File>
opt [Importação em lote]
createSession(files)
stage(files)
createSession(stagedFiles)
inserir sessão e itens
sessionId
sessionId
sessionId
scheduePendingBatchImports()
sessionId
processSession(sessionId)
resetStaleProcessingItems()
markSessionStatus(processing)
atualizar sessão e itens
ok
ok
loop [até não haver item pendente]
claimNextPendingItem(sessionId)
reservar próximo item pendente
ReceiptBatchImportItem
ReceiptBatchImportItem
validateSpaceForNewReceipt(file)
validateSpaceForNewReceipt(file)
ok
acionar processamento da prévia do item
refreshSessionStatus(sessionId)
recalcular status da sessão
ok
ok
itens prontos para revisão
alt [Arquivo individual]
processPreview(file)
[Item de lote]
processPreview(item.file)
processPreview(file, callbacks de domínio)
preparePreviewFile(file)
validateSpaceForNewReceipt(file)
validateSpaceForNewReceipt(file)
ok
ok
cleanOldTemporaryFiles()
generateVariants(file)
List<OcrImageVariant>
par [leitura de códigos visuais]
readCodes(variant.file)
códigos visuais
[análise de qualidade]
analyzeQuality(file, hasQrCode, hasBarcode)
OcrImageQuality
loop [para cada variante OCR]
process(variant.file)
OcrResult
processBestOcrVariant()
processResult(ocrResult, codes: visualCodes)
OcrProcessingResult
extractEstablishment(ocrResult.text)
estabelecimento inferido
enrichByLocalFiscalCache(data)
findAccessKey, urlQrCode)
dados fiscais ou vazio
ExtractedData
enrichByLocalCnpj(data)
lookupLocal(cnpj)
CompanyData ou vazio
ExtractedData
categoryText(normalizedText, processing, data)
texto de categoria
suggest(text, data)
list()
SELECT categorias ordenadas
List<Category>
List<Category>
findPreferredCategory() itemBoosts()
preferências ou vazio
generate(category/source text)
EmbeddingVector
Category?
infer(originalText, normalizedText, type, data, categories)
natureza financeira
scheduleRemoteBackgroundEnrichment(data)
montar Receipt preview registrar diagnóstico
discardTemporaryVariants(variants)
Receipt preview
alt [Arquivo individual]
Receipt preview
exibir confirmação
[Item de lote]
Receipt preview
markItemReady(itemId, receipt)
atualizar item do lote
ok
ok
item pronto para revisão
exibir revisão do lote
alt [Usuário descarta prévia]
descartar
discartPreview(receipt)
discartPreview(receipt)
deleteIfManaged(receipt.fileName)
ok
ok
ok
voltar ao fluxo anterior
[Usuário confirma prévia]
confirmar
saveConfirmed(receipt)
previewFile(receipt)
File
saveToFileSystem(sourceFile)
finalName
save(Receipt)
INSERT receipt, extracted_data e FTS
id gerado
Receipt salvo
findById(id)
SELECT receipt + relações
dados persistidos
Receipt persistido
registerPreference(persisted)
registerCnpjConfirmation() registerPreferredCategory() registerCategoryItems()
UPSERT preferências
ok
ok
ok
registerFiscalCache(extractedData)
save(extractedData)
UPSERT documento_fiscal_cache
ok
ok
ok
scheduleBackgroundSemanticEmbedding(persisted)
Receipt persistido
exibir comprovante salvo
opt [Embedding semântico em segundo plano]
enqueueReceipt(receiptId)
UPSERT semantic_index_task
ok
ok
schedulePendingSemanticIndex()
claimNextPending()
SELECT/UPDATE próxima tarefa
SemanticIndexTask
SemanticIndexTask
findById(receiptId)
SELECT receipt + relações
dados persistidos
Receipt
generateEmbedding(receipt)
generate(semantic fields)
EmbeddingVector
Embedding
saveEmbedding(embedding)
UPSERT embedding + embedding_vector
ok
ok
markCompleted(receiptId)
UPDATE semantic_index_task
ok

F Diagrama de Sequência — Recuperação no FinTrack


Usuário
UI (Flutter)
ReceiptService
ReceiptSearchService
ReceiptSemanticService
ISemanticIndexTaskRepository (SemanticIndexTaskRepository)
ReceiptSemanticIndexer
IEmbeddingService (DistilUseMultilingualService)
IReceiptRepository (ReceiptRepository)
ReceiptQueryBuilder
ReceiptSemanticQuery
ReceiptSemanticRanker
SQLite FTS5 + embedding_vector (sqlite-vector)
informar consulta ou filtros
alt [Consulta textual/semântica]
search(query)
search(query)
trim(query)
alt [consulta vazia]
findByFilters(ReceiptFilter())
rowsByFilters(filter)
SELECT receipts ordenados/filtrados
linhas
rows
List<Receipt>
[consulta preenchida]
findByTerms(term)
rowsByText(term)
SELECT rowid FROM receipt_fts WHERE receipt_fts MATCH ? ORDER BY bm25(receipt_fts)
ids textuais
rows
List<Receipt> textuais
searchSemantically(term)
searchSemantically(term)
findByFilters(ReceiptFilter())
rowsByFilters(filter)
SELECT receipts
linhas
rows
List<Receipt>
loop [para comprovantes desatualizados]
needsReindex(receipt)
true/false
enqueueReceipts(pendingIds)
UPSERT semantic_index_task
ok
ok
schedulePendingSemanticIndex()
resetStaleProcessingTasks()
UPDATE tarefas antigas em processamento
ok
loop [enquanto houver tarefa pendente]
claimNextPending()
SELECT/UPDATE próxima tarefa
SemanticIndexTask
SemanticIndexTask
findById(task.receiptId)
SELECT receipt + relações
dados persistidos
Receipt
needsReindex(receipt)
true/false
opt [embedding desatualizado]
generateEmbedding(receipt)
generate(semantic fields)
EmbeddingVector
Embedding
saveEmbedding(embedding)
UPSERT embedding + embedding_vector
ok
markCompleted(receiptId)
UPDATE semantic_index_task
ok
generateQueryEmbedding(term)
generate(term)
EmbeddingVector
EmbeddingVector composto
findSimilar(queryVector, 20)
findSimilar(vector, limit)
vector_init + vector_full_scan em embedding_vector
candidatos por distância
List<Receipt>
rank(queryVector, candidates)
List<Receipt> ordenada
List<Receipt> semânticos
List<Receipt>
List<Receipt> semânticos
pontuar resultados textuais, combinar, deduplicar e ordenar
alt [Resultados encontrados]
List<Receipt>
List<Receipt>
exibir lista de comprovantes
selecionar comprovante
findById(id)
findById(id)
findById(id)
SELECT receipt + relações
dados do comprovante
Receipt
Receipt
Receipt
exibir detalhes
[Nenhum resultado]
List<Receipt> vazia
List<Receipt> vazia
exibir mensagem "Nenhum comprovante encontrado"
[Filtros avançados / observação]
watchByFilters(filter)
watchByFilters(filter)
watchByFilters(filter)
loop [a cada mudança no banco]
rowsByFilters(filter)
consultar comprovantes por categoria, período, tipo, valor, despesa e ordenação
linhas filtradas
rows
Stream<List<Receipt>>
Stream<List<Receipt>>
Stream<List<Receipt>>
atualizar lista

G Evidências de Integração Contínua

Este apêndice apresenta registros complementares do fluxo de integração contínua utilizado no projeto, detalhando as etapas executadas pelo GitHub Actions e a execução específica da suíte de testes com cobertura.

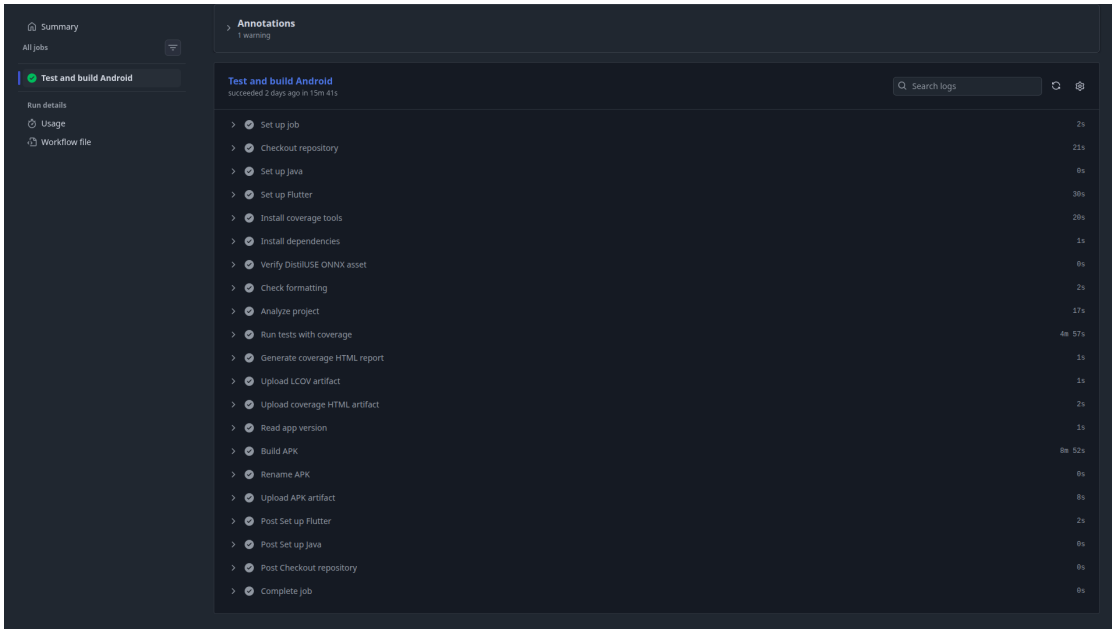


Figura 44: Etapas do fluxo de integração contínua do FinTrack no GitHub Actions.

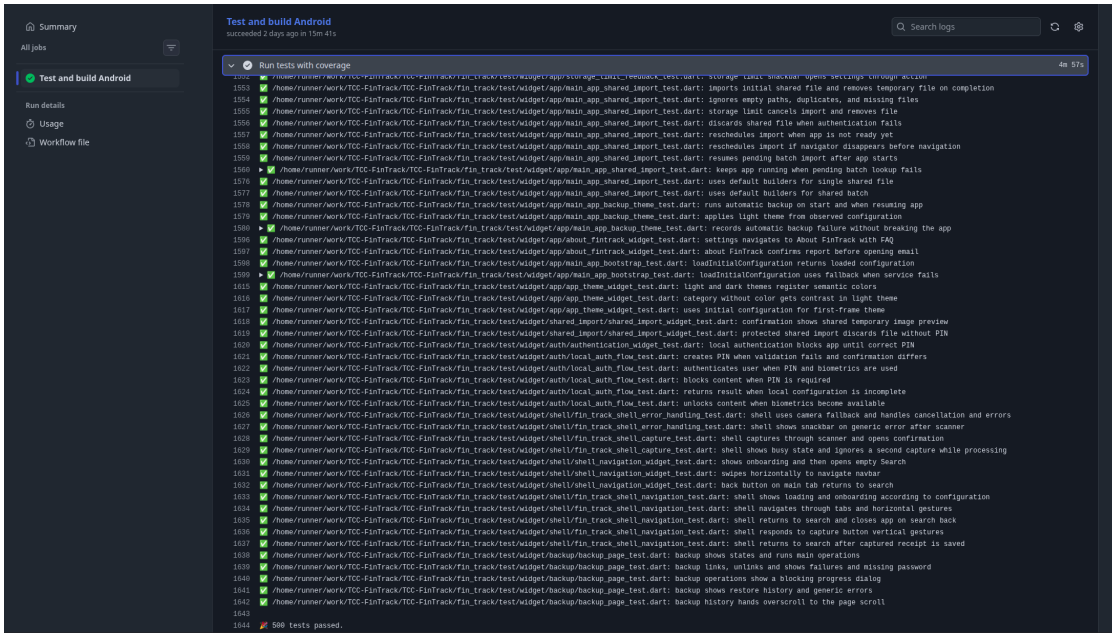


Figura 45: Detalhamento da etapa de testes com cobertura executada no GitHub Actions.