

# Visão do Produto – Cifra Organizador Financeiro

## Introdução

Este documento descreve a visão geral para o Cifra um Organizador Financeiro, uma plataforma web de finanças pessoais. O objetivo é delinear o público-alvo, os problemas que o produto se propõe a resolver, sua proposta de valor e as capacidades essenciais que o tornarão uma ferramenta indispensável para o gerenciamento financeiro acessível de qualquer navegador.

## Declaração de Visão (Elevator Pitch)

- **PARA** pessoas físicas que buscam clareza e controle sobre sua vida financeira.
- **QUE** enfrentam dificuldades em registrar e categorizar suas diversas fontes de renda e despesas diárias, correm o risco de pagar multas por esquecer vencimentos e acham o processo de controle manual repetitivo e sujeito a erros.
- **O Organizador Financeiro** é uma **plataforma web** de gerenciamento financeiro pessoal.
- **QUE** proporciona uma visão consolidada e fiel do saldo, permitindo ao usuário entender seus padrões de consumo, tomar decisões financeiras rápidas e automatizar o registro de gastos recorrentes.
- **DIFERENTE DE** planilhas manuais ou sistemas que não oferecem uma experiência integrada.
- **NOSSA PLATAFORMA** remove a complexidade inicial com uma configuração guiada, sincroniza transações automaticamente com os bancos, garante que nenhuma transação seja perdida por instabilidade de conexão e oferece alertas proativos para evitar estouros no orçamento.

## Problemas e Necessidades

O Cifra aborda as seguintes dores do usuário:

- **Falta de Clareza:** Dificuldade em ter uma visão consolidada do saldo total quando se utiliza múltiplas contas e cartões.
- **Esquecimento e Procrastinação:** O registro manual de despesas recorrentes é visto como "repetitivo e sujeito a esquecimentos".
- **Controle de Gastos:** Dificuldade em saber para onde o dinheiro está indo e em identificar categorias com gastos excessivos.
- **Medo de Atrasos:** Risco de incorrer em "multas e juros por atraso" em pagamentos recorrentes.
- **Complexidade Inicial:** A "barreira de entrada" de ter que configurar um sistema financeiro do zero, criando todas as categorias manualmente, desencoraja o uso.

## Documento de Histórias de Usuário — HU01 a HU22 (com BDD)

*Especialista: Histórias enriquecidas com cenários BDD (Dado/Quando/Então), regras, NFRs e DoD.*

### História de Usuário HU01: Registrar Receita

**Persona:** Usuário pessoa física que controla finanças pessoais no app.

**Contexto:** O usuário recebe rendas diversas (salário, bônus, freelas) e deseja registrar entradas.

**Necessidade:** Registrar receitas com valor, data, categoria e conta.

**Valor de Negócio:** Manter controle atualizado de entradas para visão fiel do saldo.

### ***Pré-condições***

- Usuário autenticado.
- Pelo menos uma conta cadastrada (ex.: Carteira, Banco X).

### ***Pós-condições***

- Receita persistida e refletida no saldo e relatórios.

### ***Regras de Negócio***

- Valor da receita deve ser  $> 0$ .
- Data não pode estar em formato inválido; aceitar datas retroativas.
- Categoria obrigatória; se ausente, sugerir 'Outras receitas'.
- Conta de destino obrigatória.

### ***Dados/Exemplos***

- R\$ 3.500,00 — 2025-09-01 — Salário — Conta: Banco XPTO
- R\$ 250,00 — 2025-09-10 — Freelance — Conta: Carteira

### ***Critérios de Aceitação (BDD)***

#### Cenário 1 — Cadastro bem-sucedido

- **Dado** que informo valor, data, categoria e conta válidos
- **Quando** confirmo o cadastro
- **Então** a receita é salva e aparece no histórico e o saldo é atualizado

#### Cenário 2 — Validação de campos obrigatórios

- **Dado** que omito a categoria ou a conta
- **Quando** tento salvar
- **Então** vejo mensagem de erro indicando os campos ausentes

#### Cenário 3 — Formato de data inválido

- **Dado** que informo uma data inválida (ex.: 2025-13-40)
- **Quando** tento salvar
- **Então** o sistema bloqueia o cadastro e orienta o formato correto

#### Cenário 4 — Edição de receita

- **Dado** que uma receita já existe
- **Quando** altero o valor/categoria/data e salvo

- **Então** o histórico e o saldo refletem a alteração imediatamente

#### Cenário 5 — Exclusão de receita

- **Dado** que desejo remover uma receita
- **Quando** confirmo a exclusão
- **Então** o registro é removido e o saldo é recalculado

#### ***Dependências/Notas Técnicas***

- Persistência em banco de dados
- Validação no backend
- Atualização reativa do saldo

#### ***Requisitos Não Funcionais***

- Operação deve concluir em < 300 ms em 95% dos casos
- Logs de auditoria para criação/edição/exclusão

#### ***Métricas/Telemetria***

- Contagem de receitas por dia
- Taxa de erro de validação
- Tempo médio de gravação

#### ***Definition of Done (DoD)***

- Testes unitários de regras
- Teste de integração de persistência
- Cobertura > 80%
- UX revisada

## História de Usuário HU02: Registrar Despesa

**Persona:** *Usuário PF que categoriza gastos mensais.*

**Contexto:** O usuário realiza despesas diárias que precisam ser classificadas.

**Necessidade:** Registrar despesas com valor, data, categoria, conta e opcionalmente notas.

**Valor de Negócio:** Entender padrões de consumo e controlar orçamento.

### ***Pré-condições***

- Usuário autenticado
- Conta cadastrada

### ***Pós-condições***

- Despesa impacta saldo, orçamento e relatórios

### ***Regras de Negócio***

- Valor da despesa deve ser  $> 0$ .
- Se houver orçamento para a categoria, comparar e sinalizar estouro.
- Permitir anexar comprovantes (imagem/PDF) opcionalmente.

### ***Dados/Exemplos***

- R\$ 75,90 — 2025-09-25 — Alimentação — Cartão X
- R\$ 120,00 — 2025-09-26 — Transporte — Débito Banco Y

### ***Critérios de Aceitação (BDD)***

#### Cenário 1 — Cadastro com sucesso

- **Dado** que informo campos válidos
- **Quando** salvo a despesa
- **Então** o histórico é atualizado e o saldo diminui

#### Cenário 2 — Estouro de orçamento

- **Dado** que o orçamento da categoria já foi excedido
- **Quando** registro nova despesa
- **Então** sou notificado sobre estouro e vejo o percentual excedido

#### Cenário 3 — Anexo de comprovante

- **Dado** que anexo um arquivo permitido
- **Quando** salvo
- **Então** o comprovante fica vinculado à despesa

#### Cenário 4 — Validação de valor

- **Dado** que informo valor zero ou negativo
- **Quando** tento salvar
- **Então** recebo mensagem de erro e o registro é bloqueado

### ***Dependências/Notas Técnicas***

- Regras de orçamento
- Armazenamento de arquivos
- Conversão/thumbnail de anexos

### ***Requisitos Não Funcionais***

- Upload até 10MB por arquivo
- Tempo de resposta < 500 ms

***Métricas/Telemetria***

- % de despesas com anexo
- Incidência de estouro de orçamento

***Definition of Done (DoD)***

- Testes de validação
- Fluxos de erro cobertos

## História de Usuário HU03: Visualizar Saldo Atual

*Persona: Usuário PF*

**Contexto:** Necessita visão consolidada de saldo entre contas.

**Necessidade:** Exibir saldo total e por conta em tempo real.

**Valor de Negócio:** Tomada de decisão rápida de gastos e transferências.

### *Pré-condições*

- Receitas/despesas existentes ou saldo zero inicial

### *Pós-condições*

- Saldo sincronizado após operações

### *Regras de Negócio*

- Saldo =  $\square$  receitas -  $\square$  despesas por conta
- Destacar saldo negativo em vermelho

### *Dados/Exemplos*

- Conta Banco X: R\$ 1.250,00; Carteira: R\$ 100,00; Total: R\$ 1.350,00

### *Critérios de Aceitação (BDD)*

#### Cenário 1 — Cálculo em tempo real

- **Dado** que registro uma transação
- **Quando** a operação é concluída
- **Então** o saldo é recalculado e exibido sem recarregar a tela

#### Cenário 2 — Destaque negativo

- **Dado** que o saldo da conta fica abaixo de zero
- **Quando** abro o dashboard
- **Então** o saldo é destacado com cor de alerta e ícone

### *Dependências/Notas Técnicas*

- Canal de eventos (WebSocket) ou polling
- Cache local para atualização instantânea

### *Requisitos Não Funcionais*

- Atualização sob 200 ms após confirmação da transação

### *Métricas/Telemetria*

- Latência média de atualização
- Erros de cálculo

### *Definition of Done (DoD)*

- Testes de precisão de cálculo
- Testes visuais de estados (positivo/negativo)

# História de Usuário HU04: Consultar Histórico Financeiro

*Persona: Usuário PF*

**Contexto:** Analisar transações em período selecionado.

**Necessidade:** Filtro por intervalo, tipo e categoria com paginação e busca textual.

**Valor de Negócio:** Auditar gastos e localizar itens rapidamente.

## ***Pré-condições***

- Transações existentes ou estado vazio

## ***Pós-condições***

- Resultados conforme filtros; mensagem clara se vazio

## ***Regras de Negócio***

- Paginar 50 itens por página
- Ordenação por data desc (padrão)

## ***Dados/Exemplos***

- Filtro: 2025-09-01 a 2025-09-30; Tipo: Despesa; Categoria: Alimentação

## ***Critérios de Aceitação (BDD)***

### *Cenário 1 — Filtro por período*

- **Dado** que seleciono um intervalo válido
- **Quando** aplico o filtro
- **Então** vejo apenas as transações no período

### *Cenário 2 — Busca textual*

- **Dado** que informo um termo (ex.: 'uber')
- **Quando** pesquiso
- **Então** o histórico retorna transações que contenham o termo

### *Cenário 3 — Sem resultados*

- **Dado** que o filtro não encontra transações
- **Quando** vejo a lista
- **Então** apresenta mensagem 'Nenhuma transação encontrada' com opção de limpar filtros

## ***Dependências/Notas Técnicas***

- Indexação de textos
- API de listagem com filtros

## ***Requisitos Não Funcionais***

- Resposta < 700 ms em 95%

## ***Métricas/Telemetria***

- Taxa de uso de filtros
- Tempo para primeiro resultado

## ***Definition of Done (DoD)***

- Testes de API de filtros
- Teste UX estado vazio

# História de Usuário HU05: Visualizar Gráficos Financeiros

*Persona: Usuário PF visual*

**Contexto:** Entender distribuição por categoria e tendências.

**Necessidade:** Gráficos por categoria/tempo (pizza, barras, linhas).

**Valor de Negócio:** Facilita identificação de excessos e sazonalidade.

## ***Regras de Negócio***

- Permitir alternância de tipos de gráfico
- Mostrar legenda e totais

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Geração de gráfico por categoria

- **Dado** que existe ao menos uma despesa no período
- **Quando** seleciono 'por categoria'
- **Então** um gráfico é exibido com percentuais e valores

### Cenário 2 — Sem dados suficientes

- **Dado** que não existem transações no período
- **Quando** tento visualizar
- **Então** o sistema informa ausência de dados e sugere alterar filtros

## ***Dependências/Notas Técnicas***

- Biblioteca de gráficos
- Agregações no backend

## ***Requisitos Não Funcionais***

- Renderização < 500 ms
- Acessível com leitores de tela (descrição textual)

## ***Métricas/Telemetria***

- Tempo de renderização
- Engajamento por tipo de gráfico

## ***Definition of Done (DoD)***

- Testes de acessibilidade ARIA
- Teste de performance de renderização



# História de Usuário HU06: Criar Orçamento por Categoria

*Persona: Usuário PF*

**Contexto:** Planejar gastos mensais por categoria.

**Necessidade:** Definir limite mensal por categoria.

**Valor de Negócio:** Evitar ultrapassagens com alertas proativos.

## ***Regras de Negócio***

- Um orçamento por categoria por mês
- Permitir editar/pausar

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Criação de orçamento

- **Dado** que informo categoria, mês e limite válidos
- **Quando** salvo
- **Então** o orçamento aparece na lista e passa a monitorar despesas

### Cenário 2 — Ultrapassagem de limite

- **Dado** que as despesas atingem 100% do limite
- **Quando** registro nova despesa
- **Então** recebo alerta de estouro com recomendação de ação

## ***Dependências/Notas Técnicas***

- Regras de comparação a cada nova despesa
- Notificações in-app/push

## ***Requisitos Não Funcionais***

- Cálculo incremental sob 150 ms

## ***Métricas/Telemetria***

- % de categorias com orçamento
- Taxa de estouro

## ***Definition of Done (DoD)***

- Testes de limite e bordas (0, valores altos)

# História de Usuário HU07: Exportar Relatório Financeiro (PDF/Excel)

*Persona: Usuário PF*

**Contexto:** Necessita enviar/guardar cópia estruturada.

**Necessidade:** Exportar período, com totais e categorização.

**Valor de Negócio:** Compartilhar com contabilidade e manter histórico.

## ***Regras de Negócio***

- Cabeçalho com período e moeda
- Pagar PDF automaticamente

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Exportação em PDF

- **Dado** que seleciono período e formato PDF
- **Quando** confirmo exportar
- **Então** o download inicia com arquivo legível e sumário por categoria

### Cenário 2 — Exportação em Excel

- **Dado** que escolho Excel
- **Quando** abro o arquivo
- **Então** os lançamentos aparecem em colunas normalizadas (data, tipo, categoria, conta, valor)

## ***Dependências/Notas Técnicas***

- Módulo de exportação PDF/XLSX
- Fila assíncrona se volume > 10k linhas

## ***Requisitos Não Funcionais***

- Geração < 5s para até 5k lançamentos

## ***Métricas/Telemetria***

- Tempo de geração
- Taxa de erro de exportação

## ***Definition of Done (DoD)***

- Testes de conteúdo e layout
- Validação de codificação CSV/XLSX

# História de Usuário HU08: Configurar Lembrete de Vencimento de Contas

*Persona: Usuário PF*

**Contexto:** Evitar atrasos em pagamentos recorrentes.

**Necessidade:** Agendar lembretes por data e frequência.

**Valor de Negócio:** Reduzir multas e juros por atraso.

## ***Regras de Negócio***

- Permitir lembrete único e recorrente (mensal)
- Horas de envio configuráveis

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Lembrete criado

- **Dado** que informo nome, data e frequência
- **Quando** salvo o lembrete
- **Então** recebo notificação na véspera e no dia de vencimento

### Cenário 2 — Edição de lembrete

- **Dado** que altero a data
- **Quando** salvo
- **Então** o agendamento é reprogramado sem duplicidades

## ***Dependências/Notas Técnicas***

- Serviço de notificações push/e-mail
- Agendador de jobs

## ***Requisitos Não Funcionais***

- Entrega de notificação em até 2 min do horário

## ***Métricas/Telemetria***

- Taxa de abertura de notificação
- Cliques em 'pagar agora'

## ***Definition of Done (DoD)***

- Teste de fuso horário e DST
- Teste de recorrência

## História de Usuário HU09: Visualizar Desempenho Financeiro por Período

*Persona: Usuário PF analítico*

**Contexto:** Comparar meses/trimestres/anos.

**Necessidade:** Visões comparativas e drill-down por categoria.

**Valor de Negócio:** Ajustar gastos com base em tendências.

### **Regras de Negócio**

- Comparar período N vs N-1
- Indicadores: variação %, ticket médio

### **Crítérios de Aceitação (BDD)**

#### Cenário 1 — Comparação mensal

- **Dado** que escolho 'mensal' e um mês com histórico
- **Quando** abro a análise
- **Então** vejo variação vs mês anterior e categorias que mais cresceram

#### Cenário 2 — Drill-down

- **Dado** que cliço em uma categoria
- **Quando** navego
- **Então** vejo transações detalhadas daquela categoria no período

### **Dependências/Notas Técnicas**

- Consultas agregadas
- Gráficos interativos

### **Requisitos Não Funcionais**

- Resposta < 800 ms

### **Métricas/Telemetria**

- Tempo de consulta
- Uso de drill-down

### **Definition of Done (DoD)**

- Teste de consistência dos agregados

## História de Usuário HU10: Criação de Categorias Padrão

- **Persona:** Novo usuário que acabou de criar sua conta.
- **Contexto:** Para registrar receitas e despesas de forma organizada, o usuário precisa de categorias. Criar uma lista inicial do zero é uma barreira que pode desencorajar o uso.
- **Necessidade:** Ter um conjunto de categorias padrão de receitas e despesas (ex: Salário, Alimentação, Transporte, Moradia) criado automaticamente no momento do meu cadastro.
- **Valor de Negócio:** Remover a fricção inicial de configuração, acelerar o tempo até o usuário perceber o valor do produto (Time-to-Value) e fornecer uma base de organização financeira sólida desde o início.
- **Pré-condições:**
  - Um novo usuário acaba de ser criado no sistema.
- **Pós-condições:**

- Uma lista de categorias padrão está associada ao perfil do novo usuário e pronta para ser utilizada.
- **Regras de Negócio:**
  - A criação das categorias padrão deve ser um processo automático e transparente, executado logo após o cadastro bem-sucedido.
  - O usuário deve ter a liberdade de editar ou excluir essas categorias padrão posteriormente.
- **Critérios de Aceitação (BDD):**
  - **Cenário 1 — Verificação da existência das categorias**
    - **Dado** que minha conta de usuário acabou de ser criada
    - **Quando** acesso a tela para registrar uma nova despesa pela primeira vez
    - **Então** a lista de seleção de "Categoria" já contém opções como "Alimentação", "Transporte", "Lazer", "Moradia" e "Saúde"
- **Dependências/Notas Técnicas:**
  - Um template/seeder no backend que é executado para popular as categorias na criação de um novo usuário.
- **Requisitos Não Funcionais:**
  - A criação das categorias padrão não deve adicionar um tempo de espera perceptível ao processo de cadastro (idealmente < 100ms).
- **Métricas/Telemetria:**
  - Taxa de utilização das categorias padrão.
  - Número de categorias personalizadas criadas em adição às padrão.
- **Definition of Done (DoD):**
  - A lista de categorias padrão foi revisada e validada como um bom ponto de partida para a maioria dos usuários.
  - O processo de "seeding" de categorias foi testado e é confiável.

# História de Usuário HU11: Adicionar Categoria de Receita/Despesa

*Persona: Usuário PF*

**Contexto:** Necessita granularidade além das categorias padrão.

**Necessidade:** Criar/editar/excluir categorias com ícones e cor.

**Valor de Negócio:** Relatórios mais significativos.

## ***Regras de Negócio***

- Nome único por tipo
- Bloquear exclusão se houver transações (ou exigir reclassificação)

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Criação de categoria

- **Dado** que informo nome e tipo válidos
- **Quando** salvo
- **Então** a categoria aparece disponível no cadastro de transações

### Cenário 2 — Exclusão com dependências

- **Dado** que existem transações associadas
- **Quando** tento excluir
- **Então** sou instruído a reclassificar ou a manter a categoria

## ***Dependências/Notas Técnicas***

- Tabela de categorias
- Migração/reclassificação assistida

## ***Requisitos Não Funcionais***

- Resposta < 300 ms

## ***Métricas/Telemetria***

- Qtde de categorias ativas por usuário

## ***Definition of Done (DoD)***

- Testes de unicidade e reclassificação

# História de Usuário HU12: Configurar Metas de Economia

*Persona: Usuário PF*

**Contexto:** Acompanhar progresso de metas (viagem, emergência).

**Necessidade:** Definir valor alvo e prazo, com barra de progresso.

**Valor de Negócio:** Estimular poupança e disciplina financeira.

## ***Regras de Negócio***

- Permitir múltiplas metas
- Contribuições vinculadas a receitas/transferências

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Criação de meta

- **Dado** que informo objetivo, valor e prazo
- **Quando** salvo
- **Então** a meta aparece com 0% e passa a somar contribuições

### Cenário 2 — Conclusão da meta

- **Dado** que o total acumulado atinge o alvo
- **Quando** abro a tela de metas
- **Então** a meta é marcada como concluída e notificada

## ***Dependências/Notas Técnicas***

- Cálculo de progresso
- Associação de transações à meta

## ***Requisitos Não Funcionais***

- Atualização de progresso em < 200 ms

## ***Métricas/Telemetria***

- Taxa de metas concluídas

## ***Definition of Done (DoD)***

- Testes de arredondamento e prazos

# História de Usuário HU13: Configurar Notificação de Despesas Elevadas

*Persona: Usuário PF*

**Contexto:** Evitar gastos surpresas acima de um limite.

**Necessidade:** Definir limite que dispare alerta ao registrar despesa acima.

**Valor de Negócio:** Controle e prevenção de estouros.

## ***Regras de Negócio***

- Limite por usuário; opcional por categoria
- Silenciar por período (soneca)

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Alerta disparado

- **Dado** que defino limite de R\$ 500
- **Quando** registro despesa de R\$ 650
- **Então** recebo notificação com valor, categoria e atalho para revisar

### Cenário 2 — Silenciar alertas

- **Dado** que ativo 'silenciar por 24h'
- **Quando** registro despesa acima do limite
- **Então** o alerta não é enviado durante o período de soneca

## ***Dependências/Notas Técnicas***

- Motor de regras e notificações
- Preferências por usuário

## ***Requisitos Não Funcionais***

- Entrega de notificação < 60 s após registro

## ***Métricas/Telemetria***

- % de despesas acima do limite
- Taxa de interação com alerta

## ***Definition of Done (DoD)***

- Testes de limites, silenciar e reativar



# História de Usuário HU14: Integração Automática com Banco

*Persona: Usuário PF com contas em múltiplas instituições*

**Contexto:** Deseja sincronizar transações sem esforço manual.

**Necessidade:** Conectar bancos e importar lançamentos recorrentes.

**Valor de Negócio:** Reduz digitação e erros; visão unificada.

## ***Pré-condições***

- Banco parceiro compatível
- Consentimento (LGPD) do usuário

## ***Regras de Negócio***

- Sincronização incremental (últimos 90 dias no onboarding)
- Evitar duplicidades via hash de transação

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Conexão inicial

- **Dado** que autorizo acesso via OAuth2 ao banco
- **Quando** concluo a autenticação
- **Então** as transações dos últimos 90 dias são importadas e categorizadas automaticamente

### Cenário 2 — Revogação de consentimento

- **Dado** que desativo a conexão
- **Quando** confirmo a revogação
- **Então** novas importações são bloqueadas e um aviso é exibido

## ***Dependências/Notas Técnicas***

- APIs bancárias/OFX
- Serviço de categorização automática

## ***Requisitos Não Funcionais***

- Importação inicial < 60 s
- Conectores com retry e backoff

## ***Métricas/Telemetria***

- % de transações categorizadas automaticamente
- Taxa de duplicidade

## ***Definition of Done (DoD)***

- Testes com sandbox bancário
- Registro de consentimento LGPD

## História de Usuário HU15: Agendar Despesa Recorrente

- **Persona:** Usuário PF (Pessoa Física) que deseja automatizar o registro de seus gastos fixos mensais.
- **Contexto:** O usuário possui despesas recorrentes com valores e datas previsíveis (aluguel, assinaturas de streaming, academia) e considera o lançamento manual repetitivo e sujeito a esquecimentos.
- **Necessidade:** Agendar o lançamento automático de despesas com uma frequência definida (ex: mensal, semanal) para economizar tempo e garantir a precisão do fluxo de caixa.
- **Valor de Negócio:** Aumentar a conveniência e o engajamento do usuário, reduzir o esforço manual e melhorar a previsibilidade dos relatórios financeiros.
- **Pré-condições:**
  - Usuário autenticado.
  - Pelo menos uma conta e uma categoria de despesa cadastradas.
- **Pós-condições:**
  - Um agendamento de despesa é criado e armazenado.
  - Nas datas programadas, uma nova transação de despesa é criada automaticamente, impactando o saldo e os relatórios.
- **Regras de Negócio:**
  - O agendamento deve conter valor, categoria, conta, data de início e frequência (ex: diária, semanal, mensal, anual).
  - O agendamento pode ter uma data de término (opcional).
  - O usuário deve poder editar, pausar ou excluir um agendamento. A alteração afeta apenas as futuras ocorrências.

- **Critérios de Aceitação (BDD):**

### Cenário 1 — Criação bem-sucedida do agendamento

- **Dado** que informo os detalhes de uma despesa e seleciono a frequência 'Mensal, todo dia 10'
- **Quando** confirmo o agendamento
- **Então** o agendamento é salvo e exibido na lista de "Despesas Agendadas"

### Cenário 2 — Lançamento automático da despesa

- **Dado** que hoje é o dia programado para o lançamento de uma despesa recorrente
- **Quando** o sistema executa o agendador
- **Então** uma nova despesa é registrada no meu histórico, o saldo da conta é atualizado e eu recebo uma notificação (opcional)

### Cenário 3 — Exclusão de um agendamento

- **Dado** que um agendamento de despesa recorrente existe
- **Quando** eu o excluo

- **Então** ele é removido da lista de agendamentos e não gerará mais novas despesas, mas as despesas já criadas permanecem no histórico

- **Dependências/Notas Técnicas:**

- Necessidade de um serviço de agendamento de tarefas no backend (ex: Cron Job, Hangfire, Quartz).
- Serviço de notificações (push/e-mail) para avisar sobre o lançamento.

- **Requisitos Não Funcionais:**

- A execução do agendamento deve ocorrer com uma margem de no máximo 10 minutos do horário previsto.
- O sistema deve ser resiliente a falhas, garantindo que um lançamento não seja duplicado em caso de reinicialização do serviço.

- **Métricas/Telemetria:**

- Número de agendamentos ativos por usuário.
- Percentual de despesas criadas automaticamente vs. manualmente.

- **Definition of Done (DoD):**

- Testes unitários para a lógica de cálculo das próximas datas de execução.
- Teste de integração do serviço de agendamento.
- Fluxo de edição e exclusão de agendamentos testado.

# História de Usuário HU16: Importar Arquivo OFX/CSV

*Persona: Usuário PF*

**Contexto:** Possui extratos em arquivos do banco.

**Necessidade:** Importar OFX/CSV e mapear colunas.

**Valor de Negócio:** Acelerar carga histórica e migração.

## ***Regras de Negócio***

- Detectar encoding e delimitador
- Pré-visualizar linhas antes de confirmar

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Importação bem-sucedida

- **Dado** que escolho um arquivo OFX/CSV válido
- **Quando** confirmo a importação
- **Então** as transações são criadas e o saldo é atualizado

### Cenário 2 — Arquivo inconsistente

- **Dado** que colunas obrigatórias estão ausentes
- **Quando** tento importar
- **Então** vejo erros por linha e posso baixar relatório de inconsistências

## ***Dependências/Notas Técnicas***

- Parser robusto
- Mecanismo de de-duplicação

## ***Requisitos Não Funcionais***

- Até 50k linhas em < 30 s

## ***Métricas/Telemetria***

- Erros por mil linhas
- Tempo de importação

## ***Definition of Done (DoD)***

- Testes com amostras reais de bancos

# História de Usuário HU17: Compartilhar Relatórios

*Persona: Usuário PF*

**Contexto:** Precisa enviar relatórios a terceiros.

**Necessidade:** Compartilhar por e-mail/app de mensagem.

**Valor de Negócio:** Colaboração e prestação de contas.

## ***Regras de Negócio***

- Proteger links com expiração
- Anexar PDF/XLSX

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Compartilhamento por e-mail

- **Dado** que um relatório foi gerado
- **Quando** seleciono 'Compartilhar' e informo e-mail
- **Então** o destinatário recebe o arquivo ou link seguro com expiração

### Cenário 2 — Compartilhamento via app

- **Dado** que escolho WhatsApp/Telegram
- **Quando** confirmo
- **Então** o arquivo é anexado e enviado pelo aplicativo selecionado

## ***Dependências/Notas Técnicas***

- APIs de compartilhamento do sistema
- Serviço de e-mail com domain keys

## ***Requisitos Não Funcionais***

- Entrega de e-mail em < 2 min
- Links válidos por 7 dias

## ***Métricas/Telemetria***

- Taxa de abertura de e-mail
- Cliques em links

## ***Definition of Done (DoD)***

- Teste de expiração e revogação de links

# História de Usuário HU18: Múltiplos Perfis de Usuário

*Persona: Usuário PF/MEI*

**Contexto:** Deseja separar finanças pessoais e do negócio.

**Necessidade:** Criar e alternar perfis independentes.

**Valor de Negócio:** Evitar misturar lançamentos e relatórios.

## ***Regras de Negócio***

- Cada perfil tem contas, categorias e metas próprias
- Controle de permissões por perfil (compartilháveis)

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Troca de perfil

- **Dado** que possuo dois perfis criados
- **Quando** alterno para 'Empresa'
- **Então** vejo apenas transações, categorias e saldos daquele perfil

### Cenário 2 — Exclusão de perfil

- **Dado** que seleciono excluir um perfil
- **Quando** confirmo após aviso de impacto
- **Então** os dados do perfil são removidos de forma irreversível

## ***Dependências/Notas Técnicas***

- Modelagem multi-tenant por usuário
- Guia visual de perfil ativo

## ***Requisitos Não Funcionais***

- Troca de perfil < 300 ms

## ***Métricas/Telemetria***

- # de usuários com múltiplos perfis
- Erros de contexto de perfil

## ***Definition of Done (DoD)***

- Testes de isolamento de dados

# História de Usuário HU19: Modo Offline

*Persona: Usuário móvel com conectividade instável*

**Contexto:** Usuário precisa registrar transações sem internet.

**Necessidade:** Operar offline e sincronizar depois.

**Valor de Negócio:** Evitar perda de dados e atritos.

## ***Regras de Negócio***

- Fila local de eventos
- Resolução de conflitos na sincronização (last-write-wins com merge assistido)

## ***Critérios de Aceitação (BDD)***

### Cenário 1 — Registro offline

- **Dado** que estou sem conexão
- **Quando** lanço uma despesa
- **Então** o app armazena localmente e marca para sincronizar depois

### Cenário 2 — Sincronização posterior

- **Dado** que a conexão foi restabelecida
- **Quando** o app sincroniza
- **Então** as transações locais são enviadas e o saldo é atualizado

## ***Dependências/Notas Técnicas***

- Banco local (ex.: SQLite)
- Agente de sync em background

## ***Requisitos Não Funcionais***

- Capacidade offline de 30 dias
- Sincronizar em < 10 s após reconexão

## ***Métricas/Telemetria***

- Falhas de sync
- Conflitos resolvidos automaticamente

## ***Definition of Done (DoD)***

- Testes em modo avião
- Teste de conflitos simultâneos

## História de Usuário HU20: Central de Ajuda e FAQ

**Persona:** Usuário PF

**Contexto:** Precisa de autoatendimento para dúvidas comuns.

**Necessidade:** Consultar FAQ, tutoriais e abrir chamado.

**Valor de Negócio:** Reduz esforço de suporte e acelera resolução pelo usuário.

### Regras de Negócio

- Busca por palavras-chave
- Artigos versionados e avaliáveis

### Critérios de aceitação(BDD)

- Cenário 1 — Consulta de artigo
  - **Dado** que acesso a central e busco 'orçamento'
  - **Quando** abro um artigo relevante
  - **Então** visualizo instruções passo a passo e links relacionados
- Cenário 2 — Escalonamento ao suporte
  - **Dado** que não encontro solução
  - **Quando** clico em 'falar com suporte'
  - **Então** posso abrir um ticket com dados do contexto (app/versão/dispositivo)

### Dependências/Notas Técnicas

- Base de conhecimento
- Sistema de chamados/integr. e-mail

### Requisitos Não Funcionais

- Tempo de busca < 300 ms
- Disponibilidade 99,9%

### Métricas/Telemetria

- Taxa de resolução via FAQ
- Satisfação (CSAT) por artigo

### Definition of Done (DoD)

- Revisão editorial
- Medição de utilidade dos artigos

## História de Usuário HU21: Cadastro de Novo Usuário

- **Persona:** Visitante que baixou o aplicativo e deseja começar a usá-lo.
- **Contexto:** O visitante não possui uma conta e precisa criar uma para acessar as funcionalidades do gerenciador financeiro.
- **Necessidade:** Criar uma conta de forma simples e segura, utilizando um e-mail e senha ou, opcionalmente, através de um provedor social (Google, Apple).
- **Valor de Negócio:** Constitui a porta de entrada para a aquisição de novos usuários, sendo um passo fundamental para o crescimento da base de clientes.
- **Pré-condições:** Nenhuma.
- **Pós-condições:**



- A conta do usuário é criada e persistida no banco de dados.
- O usuário é autenticado automaticamente e direcionado para a próxima etapa (configuração inicial ou dashboard).
- **Regras de Negócio:**
  - O endereço de e-mail deve ser único na base de dados.
  - A senha deve seguir requisitos mínimos de segurança (ex: 8 caracteres, letras e números).
  - O usuário deve concordar com os Termos de Serviço e a Política de Privacidade.
- **Critérios de Aceitação (BDD):**
  - **Cenário 1 — Cadastro com e-mail e senha bem-sucedido**
    - **Dado** que preencho um e-mail válido e único, uma senha forte e aceito os termos
    - **Quando** clico em "Cadastrar"
    - **Então** minha conta é criada, sou logado no sistema e direcionado para a tela de boas-vindas
  - **Cenário 2 — Tentativa de cadastro com e-mail duplicado**
    - **Dado** que insiro um endereço de e-mail que já existe no sistema
    - **Quando** tento me cadastrar
    - **Então** vejo uma mensagem de erro informando "Este e-mail já está em uso" e o cadastro é bloqueado
  - **Cenário 3 — Cadastro com provedor social (Google)**
    - **Dado** que escolho a opção "Entrar com Google" e autorizo o acesso
    - **Quando** o Google confirma minha identidade
    - **Então** uma conta é criada para mim com meu e-mail do Google e sou logado no sistema
- **Dependências/Notas Técnicas:**
  - Integração com provedores OAuth 2.0 (Google/Apple).
  - Armazenamento seguro de senhas com hash (ex: Argon2, Bcrypt).
- **Requisitos Não Funcionais:**
  - O processo de cadastro deve ser concluído em menos de 3 segundos após o envio dos dados.
  - A página de cadastro deve ter proteção contra ataques comuns (CSRF, XSS).
- **Métricas/Telemetria:**
  - Taxa de conversão de visitantes em usuários cadastrados.
  - Distribuição de métodos de cadastro (e-mail vs. social).
- **Definition of Done (DoD):**
  - Testes de validação de campos (e-mail, senha).
  - Testes de integração com os provedores sociais.

- Revisão de segurança (pentest) do fluxo de autenticação.

## História de Usuário HU22: Configuração Inicial Guiada (Wizard)

- **Persona:** Usuário recém-cadastrado que está acessando o aplicativo pela primeira vez.
- **Contexto:** Após o primeiro login, o usuário se depara com uma interface vazia e pode não saber por onde começar para tirar proveito do app.
- **Necessidade:** Ser conduzido por um guia passo a passo (wizard) para realizar as configurações essenciais, como criar a primeira conta (ex: "Carteira") e entender onde registrar uma despesa.
- **Valor de Negócio:** Aumentar a taxa de ativação de novos usuários, reduzindo a curva de aprendizado e garantindo que eles executem ações de valor logo no início, o que diminui a probabilidade de abandono.
- **Pré-condições:**
  - Usuário está logado pela primeira vez.
- **Pós-condições:**
  - O usuário possui ao menos uma conta financeira criada (ex: Carteira, com saldo inicial).
  - O usuário é direcionado ao dashboard principal, agora pronto para uso.
- **Regras de Negócio:**
  - O guia deve permitir que o usuário cadastre uma conta e defina um saldo inicial para ela.
  - O guia de configuração deve poder ser ignorado ("Pular por enquanto").
- **Critérios de Aceitação (BDD):**
  - **Cenário 1 — Conclusão do guia com sucesso**
    - **Dado** que sou um novo usuário na tela de configuração inicial
    - **Quando** defino o nome "Carteira" e o saldo inicial de "R\$ 50,00" e finalizo o guia
    - **Então** a conta "Carteira" é criada com o saldo informado e sou levado ao dashboard principal
  - **Cenário 2 — Ignorar o guia**
    - **Dado** que estou na tela de configuração inicial
    - **Quando** clico na opção "Pular"
    - **Então** sou levado diretamente ao dashboard principal, que exibirá um lembrete para eu adicionar minha primeira conta
- **Dependências/Notas Técnicas:**
  - Controle de estado no perfil do usuário para não exibir o guia novamente após concluído ou ignorado.
- **Requisitos Não Funcionais:**
  - A interface do guia deve ser simples, rápida e carregar em menos de 1 segundo.
- **Métricas/Telemetria:**

- Taxa de conclusão do guia de configuração.
- Ponto de abandono (em qual etapa o usuário desiste).
- **Definition of Done (DoD):**
  - Revisão de UX/UI do fluxo do guia.
  - Teste do fluxo completo e da opção de "Pular".

## **Regras de Negócio – Organizador Financeiro**

### **RN-001 – Registro de Transação**

Descrição: O sistema deve permitir ao usuário registrar transações financeiras, associando uma categoria (ex.: despesa ou receita), valor, data e uma descrição. As transações devem ser armazenadas no banco de dados de forma estruturada.

Fonte/autoridade: Requisito funcional do Organizador Financeiro.

Impacto: UC-001 Registrar Transação; UC-003 Consultar Transações.

### **RN-002 – Categorias de Transação**

Descrição: O usuário pode definir categorias personalizadas para suas transações (ex.: Alimentação, Transporte, Lazer), e cada transação deve ser associada a uma dessas categorias. O sistema deve permitir a modificação de categorias existentes.

Fonte/autoridade: Requisito funcional do Organizador Financeiro.

Impacto: UC-002 Cadastrar Categoria; UC-001 Registrar Transação.

### **RN-003 – Limite de Gastos**

Descrição: O usuário pode definir um limite de gastos mensais por categoria. O sistema deve alertar o usuário quando o limite de gastos para uma categoria for atingido ou ultrapassado.

Fonte/autoridade: Política de Controle Financeiro.

Impacto: UC-004 Gerenciar Orçamento; UC-001 Registrar Transação.

### **RN-004 – Metas de Economia**

Descrição: O sistema deve permitir que o usuário defina metas de economia mensais ou anuais, com valor e prazo. O progresso em relação a essas metas será atualizado automaticamente com base nas receitas e despesas registradas.

Fonte/autoridade: Requisito funcional do Organizador Financeiro.

Impacto: UC-005 Definir Meta de Economia; UC-001 Registrar Transação.

### **RN-005 – Alerta de Transações Pendentes**

Descrição: O sistema deve enviar um alerta ao usuário quando houver transações registradas mas não confirmadas, por exemplo, transações de débito ou pagamento parcelado.

Fonte/autoridade: Requisito funcional do Organizador Financeiro.

Impacto: UC-006 Monitorar Transações; UC-007 Enviar Notificação.

### **RN-006 – Histórico de Transações**

Descrição: O usuário deve ser capaz de visualizar um histórico completo de suas transações financeiras com a capacidade de filtrá-las por categoria, data e valor.

Fonte/autoridade: Requisito funcional do Organizador Financeiro.

Impacto: UC-008 Consultar Histórico de Transações.

### **RN-007 – Exportação de Relatórios**

Descrição: O sistema deve permitir que o usuário exporte relatórios financeiros em formatos PDF ou CSV, com a possibilidade de escolher o período, categorias e tipo de transação.

Fonte/autoridade: Requisito funcional do Organizador Financeiro.

Impacto: UC-009 Exportar Relatórios; UC-010 Gerar Relatórios.

### **RN-008 – Definição de Limite de Gastos Mensal**

Descrição: O usuário poderá definir um limite máximo de gastos para cada categoria (ex.: alimentação, transporte). Caso as despesas se aproximem de 90% do limite, o sistema deve emitir um alerta preventivo.

Fonte/autoridade: Configuração definida pelo usuário.

Impacto: UC-011 Definir Orçamento; UC-012 Monitorar Categorias de Gastos.

### **RN-009 – Proibição de Saldo Negativo em Conta Corrente Virtual**

Descrição: O sistema não deve permitir que uma conta cadastrada apresente saldo negativo caso a opção

'não permitir saldo devedor' esteja habilitada pelo usuário.  
Fonte/autoridade: Política de uso definida pelo próprio usuário.  
Impacto: UC-013 Atualizar Saldo; UC-014 Controlar Contas.

#### **RN-010 – Backup e Recuperação de Dados**

Descrição: O sistema deve realizar backups automáticos periódicos das transações e informações do usuário, permitindo restaurar dados em caso de falha ou perda de informações.  
Fonte/autoridade: Boa prática de segurança da informação e continuidade de negócios. Impacto: UC-015 Realizar Backup; UC-016 Restaurar Dados.

#### **RN-011 – Autenticação e Controle de Acesso**

Descrição: O sistema deve exigir autenticação do usuário para acesso às funcionalidades, garantindo que apenas o proprietário da conta possa visualizar ou alterar suas informações financeiras.  
Fonte/autoridade: Requisito de segurança e proteção de dados pessoais.  
Impacto: UC-017 Login de Usuário; UC-018 Gerenciar Permissões.

### **Requisitos Não Funcionais**

#### **RNF-001 – Desempenho: Tempo de Resposta**

Descrição: O sistema deve responder às operações principais (registro de transação, consulta de saldo e geração de relatório) em  $\leq 1$  s para 95 % das requisições, considerando apenas um usuário ativo por vez.  
Classificação: Eficiência de desempenho.  
Método de verificação: Teste de carga com simulação de 1 usuário realizando 100 operações consecutivas.  
Critério de aceitação: Relatório de teste demonstra que  $\geq 95$  operações foram concluídas em  $\leq 1$  s.

#### **RNF-002 – Confiabilidade: Backup Periódico**

Descrição: O sistema deve realizar cópias de segurança completas do banco de dados a cada 24 horas, armazenadas em local seguro por no mínimo 30 dias.  
Classificação: Confiabilidade.  
Método de verificação: Auditoria de logs de backup e restauração em ambiente de teste.  
Critério de aceitação: Evidência de pelo menos 30 backups diários armazenados e restauração bem-sucedida de um backup aleatório.

#### **RNF-003 – Usabilidade: Onboarding Simples**

Descrição: Usuários novatos devem conseguir cadastrar sua primeira transação em  $\leq 3$  min, sem auxílio externo, durante teste de usabilidade moderado.  
Classificação: Usabilidade.  
Método de verificação: Teste de usabilidade com 10 participantes representativos do público-alvo.  
Critério de aceitação:  $\geq 8$  participantes concluem o fluxo de primeira transação em  $\leq 3$  min.

#### **RNF-004 – Segurança: Armazenamento Seguro Local**

Descrição: Dados sensíveis (valores, categorias, notas) devem ser armazenados usando keystore/keychain do SO (Android/iOS), com chaves gerenciadas pelo sistema e sem segredos hardcoded no app.  
Classificação: Segurança.  
Método de verificação: Auditoria de código + teste dinâmico (instrumentação) para verificar uso de Keystore/Keychain.  
Critério de aceitação: Auditoria não encontra segredos hardcoded; chamadas de leitura/escrita passam por APIs seguras do SO em 100% dos pontos.

#### **RNF-005 – Estabilidade: Taxa de Crash**

Descrição: Manter  $\geq 99,5\%$  de sessões livres de crash por mês (crash-free sessions).  
Classificação: Confiabilidade (Estabilidade).  
Método de verificação: Ferramenta de monitoramento (Crashlytics, Sentry ou equivalente).  
Critério de aceitação: Relatório mensal  $\geq 99,5\%$  crash-free.

#### **RNF-006 – Segurança: Bloqueio por Biometria/PIN (opcional)**

Descrição: Se habilitado pelo usuário, o app deve exigir biometria ou PIN para reabrir após 2 minutos de inatividade em foreground.  
Classificação: Segurança / Usabilidade.  
Método de verificação: Teste funcional em dispositivo real com temporizador de inatividade.  
Critério de aceitação: Em 10/10 tentativas após 2 min de inatividade, o app exige biometria/PIN antes de exibir dados.

#### **RNF-007 – Confiabilidade: Integridade de Transações**

Descrição: O sistema deve garantir que as transações sejam processadas sem duplicação, mesmo em caso de falha no processo de salvamento ou comunicação.

Classificação: Confiabilidade.

Método de verificação: Teste de simulação de falha durante transações para verificar ausência de duplicação de dados.

Critério de aceitação: Nenhuma transação duplicada após falhas simuladas.

#### **RNF-008 – Manutenibilidade: Código Documentado**

Descrição: O código do aplicativo deve ter documentação de código de 90% das funções e classes principais, com uso de comentários significativos.

Classificação: Manutenibilidade.

Método de verificação: Análise do código fonte por ferramenta de inspeção de qualidade.

Critério de aceitação: Relatório de qualidade do código mostra que 90% das funções e classes estão documentadas.

#### **RNF-009 – Compatibilidade: Dispositivos e OS**

Descrição: O aplicativo deve ser compatível com as versões mais recentes de Android ( $\geq 12$ ) e iOS ( $\geq 15$ ).

Classificação: Compatibilidade.

Método de verificação: Teste de instalação e uso em dispositivos físicos e emuladores.

Critério de aceitação: O aplicativo é testado e funciona corretamente em dispositivos Android e iOS especificados.

#### **RNF-010 – Estabilidade: Tratamento de Erros de Rede**

Descrição: Em falha de rede, o app deve exibir mensagem única e clara, não repetir requisições mais de 3x com backoff exponencial, e retornar ao estado estável sem travar a UI.

Classificação: Confiabilidade / Usabilidade.

Método de verificação: Testes instrumentados com simulação de perda/latência.

Critério de aceitação: 100% dos cenários: 1 mensagem de erro, máx. 3 tentativas com backoff, UI responsiva após a falha.

### **Declaração de Uso Ético e Responsável de Inteligência Artificial**

Declaramos que, durante a elaboração deste Documento de Requisitos do projeto Organizador Financeiro, utilizamos ferramentas de inteligência artificial generativa como ChatGPT (modelo GPT-5, OpenAI), no período de setembro de 2025, com o objetivo de auxiliar na formulação, estruturação e revisão de partes específicas do documento.

Atribuição:

- Ferramenta utilizada: ChatGPT (OpenAI, modelo GPT-5).
- Período de uso: setembro/2025.

Supervisão Humana:

Todas as saídas sugeridas pela ferramenta foram revisadas criticamente. Nenhuma seção foi incluída sem passar por avaliação humana.

Checagem de Confiabilidade:

As informações produzidas pela IA foram confrontadas com boas práticas de engenharia de software, modelos de requisitos e referências teóricas adequadas. Eventuais ajustes, correções e adequações às necessidades do projeto foram realizados pela equipe antes da versão final do documento.

Integridade Acadêmica:

O uso de IA não substituiu a autoria humana, servindo como ferramenta de apoio e inspiração. Todo o conteúdo gerado foi adaptado, reestruturado e revisado, garantindo que não haja plágio. Onde partes significativas foram mantidas, a contribuição da IA está explicitamente reconhecida nesta declaração.