

Desenvolver um sistema financeiro para o setor jurídico exige uma estrutura que suporte a incerteza dos honorários sucumbenciais (êxito) sem perder o rigor das contas fixas.

Aqui está uma proposta de **projeto lógico** refinado, focando na flexibilidade que você mencionou e adicionando camadas de inteligência para o negócio.

1. Arquitetura de Entidades (Modelo de Dados)

Para suportar previsões sem data e valores variáveis, o segredo está no status do lançamento e na vinculação a "Processos/Casos".

Estrutura de Tabelas Sugerida:

- **Participantes:** Única tabela para Clientes e Fornecedores (com uma flag de tipo), evitando duplicidade caso um fornecedor também seja cliente.
 - **Centro de Custos:** Estrutura hierárquica (ex: Operacional > Aluguel; Administrativo > Papelaria).
 - **Processos/Contratos:** Entidade central para o jurídico. É aqui que amarramos as receitas de êxito.
 - **Lançamentos Financeiros:** Onde a mágica acontece.
-

2. O Desafio do "Contas a Receber" Jurídico

Para lidar com os valores de "causa ganha", sugiro dividir os recebíveis em três tipos de **Natureza de Lançamento**:

1. **Fixos/Parcelados:** Honorários mensais (retentores). Têm data de vencimento e valor exato.
 2. **Pontuais:** Consultas avulsas ou reembolsos de custas processuais.
 3. **Honorários de Êxito (Provisórios):** * **Valor:** Pode ser um campo "Estimado".
- **Data de Vencimento:** No banco de dados, pode ser **NULL** ou uma data fictícia de longuíssimo prazo (ex: 31/12/2099), mas na interface, deve ser marcado como "**Aguardando Trânsito em Julgado**" ou "**Sem Prazo Definido**".
-

3. Ideias para Melhorar o Projeto

Para elevar o nível da sua consultoria, considere implementar os seguintes módulos:

A. Gestão de Reembolsáveis (Custas)

Muitas vezes, o advogado paga uma taxa do tribunal do próprio bolso para depois cobrar do cliente.

- **Ideia:** Criar um tipo de despesa "Reembolsável" que, ao ser lançada no Contas a Pagar, gera automaticamente um alerta ou um lançamento de "Receita de Reembolso" no Contas a Receber vinculado ao cliente.

B. Fluxo de Caixa Realista vs. Otimista

Como o êxito é incerto, o sistema deve oferecer dois filtros de visão:

- **Visão Conservadora:** Exibe apenas o que tem data e valor certo.
- **Visão Projetada:** Inclui as previsões de causas ganhas com base em uma probabilidade de vitória (ex: 25%, 50%, 75%).

C. Alocação por Advogado (Profit Sharing)

Se a consultoria divide lucros por sócio ou associado:

- **Ideia:** Vincular cada receita a um "Responsável pelo Caso". Isso facilita o cálculo automático de comissões no final do mês.

4. Dicionário de Dados Simplificado (Lançamentos)

Campo	Tipo	Descrição
ID_Lancamento	UUID	Chave primária.
Tipo	ENUM	Receita ou Despesa.
Status	ENUM	Pendente, Pago, Cancelado, Estimado (Êxito).
Valor_Previsto	DECIMAL	Valor estimado (ex: % da causa).
Valor_Real	DECIMAL	Valor que de fato entrou/saiu após liquidação.
Data_Vencimento	DATE	Pode ser nulo para lançamentos de êxito.
ID_Centro_Custo	FK	Vinculação com a árvore de custos.
ID_Processo	FK	Opcional, para rastrear a origem da receita.

Próximo Passo

Este modelo resolve a questão da incerteza dos prazos e organiza a casa para o crescimento da consultoria.

Para estruturar um banco de dados que suporte a dualidade entre o **rigor contábil** (contas fixas) e a **incerteza jurídica** (êxito), precisamos de uma modelagem que não obrigue o preenchimento de datas ou valores exatos em lançamentos de previsão.

Proposta de arquitetura relacional focada em flexibilidade e rastreabilidade:

1. Diagrama de Entidade e Relacionamento (ER)

A lógica central é separar o **Lançamento Financeiro** da sua **Liquidação**. Isso permite que você tenha um registro de "intenção de recebimento" que só se torna um dado financeiro real quando o processo é ganho.

2. Estrutura das Tabelas Principais

A. Tabela: **Participantes** (Clientes e Fornecedores)

Centralizar ajuda a gerenciar casos onde um cliente também pode ser um prestador de serviço.

- **id**: UUID (Primary Key)
- **nome_razao_social**: VARCHAR
- **cpf_cnpj**: VARCHAR (Unique)
- **tipo_perfil**: ENUM ('CLIENTE', 'FORNECEDOR', 'AMBOS')
- **contato**: TEXT

B. Tabela: **Processos** (O Coração do Recebimento)

Fundamental para vincular os honorários de êxito.

- **id**: UUID
- **numero_processo**: VARCHAR (CNJ)
- **id_cliente**: FK (Participantes)
- **titulo_causa**: VARCHAR
- **percentual_exito**: DECIMAL(5,2) — *Ex: 30.00 para 30%.*
- **valor_pro_labore**: DECIMAL(15,2) — *Valor fixo inicial (se houver).*
- **valor_causa_atualizado**: DECIMAL(15,2) — *Base para o cálculo do êxito.*
- **status_processo**: ENUM ('ATIVO', 'SUSPENSO', 'ENCERRADO', 'TRANSITO_JULGADO')
- **valor_causa_estimado**: DECIMAL(15,2)

C. Tabela: **Lancamentos** (Contas a Pagar e Receber)

Esta tabela deve ser híbrida para aceitar dados precisos e estimativas.

- **id**: UUID
- **id_participante**: FK (Participantes)
- **id_processo**: FK (Processos) - *Opcional: Preenchido apenas em receitas de êxito ou custas.*
- **id_centro_custo**: FK (Centro_Custos)
- **tipo_operacao**: ENUM ('RECEITA', 'DESPESA')
- **natureza**: ENUM ('FIXO', 'PONTUAL', 'EXITO', 'REEMBOLSO')
- **descricao**: TEXT
- **valor_previsto**: DECIMAL(15,2) - *O percentual da causa convertido em moeda.*
- **valor_realizado**: DECIMAL(15,2) - *Preenchido apenas no momento da baixa.*
- **data_vencimento**: DATE (Allow NULL) - *Nulo para honorários de êxito sem prazo.*
- **data_pagamento**: DATE (Allow NULL)
- **status_pagamento**: ENUM ('PREVISAO', 'PENDENTE', 'PAGO', 'ATRASADO', 'CANCELADO')

D. Tabela: **Centro_Custos**

- **id**: INT
- **nome**: VARCHAR (Ex: "Operacional", "Marketing", "Custas Judiciais")
- **tipo**: ENUM ('RECEITA', 'DESPESA')

3. Lógica de Negócio Aplicada ao Banco

Para atender aos seus requisitos específicos, a aplicação deve se comportar assim no banco:

1. Lançamento de Êxito:

- `natureza` = 'EXITO'
- `data_vencimento` = `NULL`
- `status_pagamento` = 'PREVISAO'
- *Query de Fluxo de Caixa:* Quando o gestor pedir o fluxo de caixa para os próximos 30 dias, o sistema ignora esses registros. Quando pedir "Visão de Longo Prazo", o sistema soma esses valores.

2. Despesas Recorrentes (Contas a Pagar):

- Você pode criar uma tabela auxiliar `Recorrenncias` que gera automaticamente novos registros na tabela `Lancamentos` todo dia 01 de cada mês (ex: Aluguel, Software jurídico).

3. Vínculo de Custas (Reembolso):

- Uma despesa lançada como `natureza` = 'REEMBOLSO' pode disparar uma trigger ou regra de negócio para criar um lançamento de Receita espelhado para o cliente vinculado.

4. Exemplo de Consulta (SQL) para Fluxo de Caixa

Para buscar o que você tem "no radar", mas sem data definida:

```
SELECT
    p.nome_razao_social AS Cliente,
    l.descricao,
    l.valor_previsto
FROM Lancamentos l
JOIN Participantes p ON l.id_participante = p.id
WHERE l.status_pagamento = 'PREVISAO'
    AND l.natureza = 'EXITO'
    AND l.data_vencimento IS NULL;
```

Estrutura de Negócio

Incluir o **percentual de honorários** na tabela de `Processos` é uma decisão de design excelente por dois motivos principais: **automação** e **auditabilidade**.

Se o percentual estiver na tabela de processos, o sistema pode calcular automaticamente o `valor_previsto` na tabela de `Lancamentos` assim que o advogado atualizar o "Valor da Causa" ou quando a sentença for proferida.

Aqui está como a tabela de `Processos` e a lógica de automação ficariam com essa melhoria:

1. Tabela `Processos` (Versão Evoluída)

Adicionamos campos para separar os tipos de honorários (que são comuns em contratos jurídicos):

- `id`: UUID
- `numero_processo`: VARCHAR (CNJ)
- `id_cliente`: FK (Participantes)
- `titulo_causa`: VARCHAR
- `percentual_exito`: DECIMAL(5,2) — Ex: 30.00 para 30%.
- `valor_pro_labore`: DECIMAL(15,2) — Valor fixo inicial (se houver).
- `valor_causa_atualizado`: DECIMAL(15,2) — Base para o cálculo do êxito.
- `status_processo`: ENUM ('ATIVO', 'SUSPENSO', 'ENCERRADO', 'TRANSITO_JULGADO')
- `valor_causa_estimado`: DECIMAL(15,2)

2. A Lógica de Automação (Trigger ou Service Layer)

Com essa estrutura, seu aplicativo pode ter um comportamento inteligente:

1. **Cálculo Automático:** Sempre que o `valor_causa_atualizado` for alterado, o sistema recalcula:

```
Lancamentos.valor_previsto = (Processos.valor_causa_atualizado *  
Processos.percentual_exito / 100)
```

2. **Geração de Pro-labore:** No momento da abertura do processo, o sistema já pode inserir automaticamente um lançamento de "Receita Fixa" no `Contas a Receber` com o `valor_pro_labore` definido.

3. E se o honorário for escalonado? (Ideia Extra)

Em causas maiores, é comum o contrato dizer: "10% se ganhar na primeira instância, 20% se for para o STF".

Se você quiser que seu sistema seja **top de linha**, em vez de um único campo de percentual, você poderia ter uma tabela auxiliar: `Honorarios_Contratuais`.

Tabela `Honorarios_Contratuais`:

- `id_processo`: FK
- `fase_processual`: VARCHAR (Ex: "Liminar", "Sentença", "Recurso")
- `percentual`: DECIMAL(5,2)

Por que isso é bom? Porque o financeiro da consultoria jurídica muda conforme o processo anda. Se o processo sobe para Brasília, o seu "Contas a Receber (Previsão)" aumenta automaticamente no banco de dados porque o percentual de risco aumentou.

Resumo da sugestão:

Mantenha o campo `percentual_exito` na tabela de `Processos` para casos simples, mas considere a tabela de `Honorarios_Contratuais` se a consultoria atender clientes corporativos com contratos complexos.

Baixa de Títulos e Conciliação

A **Baixa de Títulos** e a **Conciliação** são o momento em que a "previsão" (o que você espera) encontra a "realidade" (o que caiu no banco). No setor jurídico, isso é crítico porque o valor que o juiz libera no alvará

raramente é o valor exato que você projetou inicialmente, devido a juros, correções monetárias e custas processuais.

Para que seu banco de dados suporte isso, precisamos de uma tabela intermediária de **Movimentações/Extrato**, pois um único título (receita) pode ser pago em várias vezes, ou vários títulos podem ser quitados com um único depósito.

1. Estrutura de Tabelas para Conciliação

Tabela: **Contas_Bancarias** (Caixas)

Onde o dinheiro efetivamente circula.

- **id**: UUID
- **nome**: VARCHAR (Ex: "Itaú Corrente", "Caixa Escritório", "Cofre")
- **saldo_atual**: DECIMAL(15,2)

Tabela: **Movimentacoes_Extrato** (A Realidade)

Representa cada linha do seu extrato bancário real.

- **id**: UUID
- **id_conta**: FK
- **data_movimento**: DATE
- **valor**: DECIMAL(15,2)
- **tipo**: ENUM ('ENTRADA', 'SAIDA')
- **conciliado**: BOOLEAN (Default: **false**)

Tabela: **Baixas_Vinculo** (A Ponte)

Esta tabela vincula um **Lançamento** (título) a uma **Movimentação** (dinheiro real).

- **id_lancamento**: FK
 - **id_movimentacao**: FK
 - **valor_aplicado**: DECIMAL(15,2)
-

2. A Lógica de "Baixa" (O Fluxo)

Para o jurídico, a baixa de um título de **Êxito** segue estes passos:

Passo A: Liquidação do Valor

Quando sai a sentença/alvará, o advogado atualiza o **valor_realizado** no registro que antes era apenas uma **PREVISAO**.

Regra de Negócio: O sistema pergunta: "O valor final foi diferente do previsto?". Se sim, ele ajusta o saldo para que o relatório de "Previsto vs. Realizado" faça sentido.

Passo B: A Baixa (Manual ou Automática)

1. O usuário seleciona um lançamento pendente (ex: Honorários do Cliente X).
 2. Informa a **Data do Pagamento** e a **Conta Bancária**.
 3. O sistema altera o `status_pagamento` para **PAGO**.
 4. **Ação Automática:** O sistema gera um registro na tabela `Movimentacoes_Extrato` e atualiza o `saldo_atual` da `Conta_Bancaria`.
-

3. A Lógica de Conciliação (O Check-up)

A conciliação é o processo de conferir se o que você marcou como "Pago" no sistema realmente aconteceu no banco.

1. **Importação:** O sistema importa um arquivo (OFX ou API Bancária) para a tabela `Movimentacoes_Extrato`.
 2. **Match (Sugestão):** O sistema busca por registros onde `Movimentacao.valor == Lancamento.valor_realizado` e a data seja próxima.
 3. **Aprovação:** * Se o usuário confirma, o campo `conciliado` vira **true**.
 - Se o valor for diferente (ex: o banco cobrou uma taxa de boleto), o sistema permite criar um **Lançamento de Ajuste** (Despesa) vinculado àquela conciliação para fechar a conta em zero.
-

4. O Diferencial Jurídico: Retenção de Impostos e Taxas

No momento da baixa de um honorário de êxito, é comum haver descontos:

- **IRRF (Imposto de Renda Retido na Fonte).**
- **ISS.**

Sugestão de Lógica: No formulário de baixa, inclua campos de "Descontos/Taxas". Se o usuário preencher, o sistema deve:

1. Dar baixa no valor total do honorário.
 2. Gerar automaticamente uma **Despesa** de imposto vinculada, para que o saldo líquido da conta bancária bata exatamente com o que entrou.
-

Resumo Visual da Baixa

Ação	Tabela: Lancamentos	Tabela: Movimentacoes	Tabela: Contas_Bancarias
Antes	Status: PENDENTE	(Vazio)	Saldo: R\$ 1.000
Baixa Efetuada	Status: PAGO	Entrada de R\$ 500	Saldo: R\$ 1.500
Conciliado	(Sem alteração)	Conciliado: SIM	(Sem alteração)

O uso intensivo de **cartão de crédito** introduz uma camada de complexidade: o que você paga hoje (a compra) só sai do banco no futuro (o vencimento da fatura). Para um controle financeiro profissional, você não pode lançar a despesa apenas no dia que paga a fatura, senão perde o controle do que foi gasto em tempo real.

Aqui está a lógica para integrar o Cartão de Crédito ao seu projeto:

1. Novas Entidades no Banco de Dados

Tabela: **Cartoes_Credito**

- **id**: UUID
- **nome**: VARCHAR (Ex: "Visa Gold Escritório")
- **limite**: DECIMAL
- **dia_fechamento**: INT
- **dia_vencimento**: INT
- **id_conta_bancaria_origem**: FK (De onde sai o dinheiro para pagar a fatura)

Tabela: **Faturas_Cartao**

O cartão não é uma conta corrente, ele gera faturas mensais.

- **id**: UUID
 - **id_cartao**: FK
 - **mes_referencia**: INT
 - **ano_referencia**: INT
 - **status**: ENUM ('ABERTA', 'FECHADA', 'PAGA')
 - **valor_total**: DECIMAL
-

2. A Lógica de Lançamento (O "Pulo do Gato")

Quando o escritório assina um software (ex: Cursor, ChatGPT, JusBrasil), o sistema deve tratar isso de forma diferente de um boleto:

1. **O Registro da Despesa:** No **Contas a Pagar**, o **id_forma_pagamento** será "Cartão de Crédito".
2. **O Vínculo:** O lançamento aponta para o **id_cartao**.
3. **A Data de Saída:** O sistema calcula automaticamente em qual **id_fatura** aquela despesa cai, com base na **data_compra** e no **dia_fechamento** do cartão.

Exemplo: Se o cartão fecha dia 25 e você assina o Cursor dia 20, ele cai na fatura do mês atual. Se assinar dia 26, o sistema já joga essa previsão para a fatura do mês seguinte.

3. Conciliação de Cartão de Crédito

Diferente da conta corrente, a conciliação aqui é dupla:

- **Conciliação da Compra:** Você confere se cada software lançado no sistema aparece na fatura do cartão.
 - **Conciliação do Pagamento:** Quando o escritório paga a fatura total (ex: R\$ 5.000,00), o sistema dá uma "baixa em lote" em todas as despesas contidas naquela fatura e gera uma saída única na **Conta_Bancaria**.
-

4. Estrutura de Centro de Custo para Softwares

Como você mencionou cursos e softwares, sugiro organizar o Centro de Custo assim para facilitar a visão de ROI (Retorno sobre Investimento):

- **Operacional**
 - Softwares Jurídicos (Peticionamento, GED)
 - IA e Produtividade (Cursor, APIs)
 - **Capacitação**
 - Cursos e Treinamentos
 - Anuidades (OAB, Associações)
-

5. Visualização no Fluxo de Caixa

Para o gestor da consultoria, o sistema deve mostrar:

1. **Dívida Acumulada:** O que já foi gasto no cartão mas ainda não foi pago (Faturas abertas).
2. **Projeção de Fatura:** "Sua fatura do mês que vem já está em R\$ X devido às assinaturas recorrentes".

Próximo Passo

Essa estrutura de cartões resolve a questão das assinaturas de softwares.

Para que o sistema seja realmente útil para o dia a dia da consultoria, ele precisa parar de ser apenas um "repositório de dados" e passar a ser um **assistente proativo**.

A automação de recorrências e a geração de relatórios inteligentes transformam o caos das assinaturas (como Cursor e softwares jurídicos) em previsibilidade.

1. Automatização de Despesas Recorrentes

A melhor forma de lidar com assinaturas mensais sem intervenção humana é através de uma tabela de **Modelos de Recorrência**.

A. Tabela: Configuracao_Recorrencia

Esta tabela guarda a "regra" da assinatura, e não o gasto em si.

- id: UUID
- descricao: VARCHAR (Ex: "Assinatura Cursor AI")
- valor_fixo: DECIMAL
- dia_faturamento: INT (Ex: Todo dia 10)
- id_participante: FK (Fornecedor)
- id_metodo_pagamento: FK (Vínculo com o Cartão de Crédito ou Conta)
- id_centro_custo: FK (Software/IA)

- **proxima_geracao**: DATE (Data em que o sistema deve criar o próximo título)

B. O Processo de Automação (Worker/Cron)

O sistema deve rodar uma rotina diária (um "robô") que executa a seguinte lógica:

1. Busca todas as **Configuracao_Recorrencia** onde **proxima_geracao** <= hoje.
2. Cria um registro na tabela **Lancamentos** com os dados da configuração.
3. Se for no cartão de crédito, já vincula à **Fatura_Cartao** correspondente.
4. Atualiza a **proxima_geracao** para o mês seguinte.

Dica de Ouro: Para softwares em dólar (como o Cursor), o sistema deve lançar o valor estimado e marcar o título com um status "**Aguardando Valor Real**", permitindo o ajuste exato no momento da fatura.

2. Relatórios de Fluxo de Caixa

Com os dados de êxito (incertos), despesas fixas (recorrentes) e faturas de cartão, o Fluxo de Caixa deve ser dividido em três visões essenciais:

A. Fluxo de Caixa Realizado (Regime de Caixa)

Mostra apenas o dinheiro que **efetivamente** entrou e saiu. É o "extrato consolidado".

- **Foco:** Conciliação bancária e saldo atual.

B. Fluxo de Caixa Projetado (Curto/Médio Prazo)

Combina o saldo atual com as contas que têm data de vencimento (Boletos, Salários e Faturas de Cartão de Crédito).

- **Foco:** "Vou ter dinheiro para pagar a fatura do cartão no dia 15?".
- Aqui, as assinaturas recorrentes já aparecem como "saídas previstas" para os próximos 6 ou 12 meses.

C. Fluxo de Caixa de Oportunidade (Visão Jurídica)

Este é o diferencial do seu projeto. Ele inclui as **Previsões de Êxito** (lançamentos sem data definida).

- **Foco:** Valorização da banca.
- Apresenta o gráfico de "Cenário Otimista", onde as causas ganhas são liquidadas.

3. Exemplo de Estrutura de Relatório (Painel do Gestor)

Mês	Saldo Inicial	Receitas Fixas	Previsão de Êxito*	Despesas (Fixas + Cartão)	Saldo Final Estimado
Março	R\$ 50.000	R\$ 10.000	R\$ 200.000	(R\$ 15.000)	R\$ 45.000**
Abri	R\$ 45.000	R\$ 10.000	R\$ 0	(R\$ 12.000)	R\$ 43.000

*O Êxito não soma no Saldo Final Estimado do mês, ele fica em uma coluna separada como "Patrimônio em Trânsito" até que uma data de recebimento seja confirmada.

4. Alerta de "Burn Rate" (Taxa de Queima)

Como você usa muitos softwares e cartões, uma funcionalidade valiosa seria o **Alerta de Assinaturas**:

"Este mês seus gastos com Softwares/IA subiram 15% em relação ao mês anterior."

Isso é possível cruzando a tabela de **Lancamentos** com o **Centro de Custo** de TI.

Para que esse dashboard seja dinâmico e suporte tanto o fechamento retroativo (últimos 30 dias) quanto a exploração de períodos personalizados, a lógica de **agregação de dados** no banco precisa ser muito eficiente.

Aqui está como estruturar a inteligência por trás desse dashboard:

1. O Motor de Busca (Query Dinâmica)

O analista terá um seletor de data. Se ele não informar nada, o sistema assume **Data_Atual - 30 dias**. Para que os gráficos carreguem rápido, o ideal é criar uma **View** ou uma **Stored Procedure** que consolide os dados.

Filtros Essenciais do Dashboard:

- **Período:** De [Data Inicial] até [Data Final].
 - **Status:** Realizado (dinheiro em conta) vs. Projetado (tudo que está lançado).
 - **Centro de Custo:** Para filtrar apenas gastos com "Software/IA" ou "Operacional".
-

2. Indicadores Chave (Kpis) Sugeridos

Para uma consultoria jurídica, o dashboard deve apresentar quatro "cards" principais no topo:

1. **Saldo em Conta (Realizado):** Soma de todos os lançamentos com **status = 'PAGO'** até a data de hoje.
 2. **Burn Rate (Gasto Fixo):** Soma das despesas recorrentes e faturas de cartão do período. Útil para saber o custo de "manter as luzes acesas".
 3. **Ticket Médio de Êxito:** Média dos valores recebidos de causas ganhas no período selecionado.
 4. **Pipeline de Recebíveis (Incerteza):** Soma dos lançamentos de **natureza = 'ÊXITO'** que estão sem data, mas ativos no sistema.
-

3. Visualizações Dinâmicas (Gráficos)

A. Gráfico de Barras Empilhadas: "Entradas vs. Saídas"

Este gráfico deve mostrar o equilíbrio financeiro.

- **Eixo X:** Dias ou Semanas.
- **Eixo Y:** Valor em R\$.
- **Diferencial:** As barras de entrada devem ser divididas em "Honorários Fixos" e "Honorários de Êxito", para o analista ver o quanto a empresa depende de vitórias judiciais.

B. Gráfico de Pizza/Donut: "Composição de Despesas"

Aqui entram os dados dos **Cartões de Crédito** e **Assinaturas**.

- Fatiadas baseadas no **Centro de Custo**.
- O analista consegue ver rapidamente se o gasto com "Cursor/IA/Software" está abocanhando uma fatia muito grande do faturamento.

4. Lógica de "Drill-down" (Aprofundamento)

Um dashboard dinâmico de verdade permite clicar no dado.

- **Exemplo:** O analista vê que gastou R\$ 2.000 em "Softwares" nos últimos 30 dias. Ao clicar na fatia do gráfico, o sistema abre uma tabela abaixo filtrada com todos os lançamentos que compõem aquele valor (Cursor, ChatGPT, JusBrasil, etc.).

5. Exemplo de Estrutura SQL para o Dashboard

Para alimentar o gráfico de linha de fluxo de caixa, o sistema usaria algo assim:

```
SELECT
    DATE(data_vencimento) as dia,
    SUM(CASE WHEN tipo_operacao = 'RECEITA' THEN valor_previsto ELSE 0 END) as entradas,
    SUM(CASE WHEN tipo_operacao = 'DESPESA' THEN valor_previsto ELSE 0 END) as saidas
FROM Lancamentos
WHERE data_vencimento BETWEEN :data_inicio AND :data_fim
GROUP BY DATE(data_vencimento)
ORDER BY dia;
```

6. O Alerta de "Previsão de Caixa Vazio"

Como você tem a data de fechamento da fatura do cartão de crédito, o dashboard pode ter um widget de **Alerta de Liquidez**:

"Atenção: A fatura do Cartão Visa vence em 5 dias (R\$ 4.500) e o saldo projetado em conta para esta data é de apenas R\$ 3.800."

Isso dá tempo ao analista para cobrar um cliente ou antecipar uma taxa.

Próximo Passo

O projeto lógico está robusto: banco de dados flexível, automação de assinaturas e agora a inteligência do dashboard.

Para dar continuidade e fechar o projeto lógico com chave de ouro, vamos detalhar como o sistema deve lidar com o **Multimoeda (Dólar vs. Real)** — essencial para as assinaturas de softwares como Cursor e OpenAI — e como consolidar a **Interface do Dashboard** para que o analista tenha controle total.

1. Tratamento de Moeda Estrangeira (Assinaturas Internacionais)

Como a consultoria utiliza softwares pagos em dólar via cartão de crédito, o sistema não pode simplesmente salvar um valor fixo.

A. Estrutura de Tabela: `Moedas_Cotacoes`

- `moeda_origem`: CHAR(3) (USD, EUR, BRL)
- `data_cotacao`: DATE
- `valor_cambio`: DECIMAL(10,4)

B. Lógica de Lançamento de Software (Ex: Cursor)

1. **No lançamento recorrente:** O usuário define a assinatura como \$20.00 USD.
 2. **Na projeção (Dashboard):** O sistema busca a última cotação salva e aplica um "markup" de segurança (ex: PTAX + 5% de IOF + Spread bancário) para mostrar o valor estimado em Reais.
 3. **Na baixa da fatura:** Quando o analista recebe a fatura do cartão, ele edita o valor real pago em BRL. O sistema calcula a diferença e ajusta o saldo.
-

2. Protótipo Lógico do Dashboard Dinâmico

Imagine a tela dividida em três camadas de profundidade:

Camada 1: Os Widgets de Topo (Snapshots)

- **Saldo Consolidado:** (Soma de Contas Bancárias).
 - **Faturas em Aberto:** (Soma do que já foi gasto no Cartão, mas não pago).
 - **Contas a Receber (Próximos 30 dias):** Somente o que tem data certa (pro-labore/mensalidades).
 - **Receita em Potencial (Êxito):** A soma de todos os percentuais sobre causas que estão "em trânsito".
-

Camada 2: O Gráfico Central (Fluxo de Caixa Dinâmico)

Um gráfico de linhas onde o usuário pode ligar e desligar "camadas":

1. **Linha Realizada:** O que já aconteceu (Passado).
2. **Linha Projetada:** Contas fixas e faturas que vencerão (Futuro certo).
3. **Linha de Oportunidade (Ghost Line):** Uma linha pontilhada que soma o **Êxito** caso ele ocorra nas datas estimadas.

Camada 3: Tabela de Análise por Centro de Custo

Abaixo dos gráficos, uma tabela dinâmica que responde aos filtros de data:

Centro de Custo	Previsto (BRL)	Realizado (BRL)	Desvio (%)
Softwares & IA	R\$ 1.200,00	R\$ 1.350,00	+ 12,5% (Câmbio)
Custas Processuais	R\$ 5.000,00	R\$ 4.800,00	- 4%
Marketing/Ads	R\$ 2.000,00	R\$ 2.000,00	0%

3. Regras de Negócio para o Analista

Para garantir que o dashboard seja confiável, o sistema deve impor algumas validações:

- **Validação de Data:** Se o analista filtrar um período futuro, os campos de "Realizado" devem ser automaticamente zerados, mostrando apenas o "Projetado".
 - **Alerta de Pendência:** Se houver um lançamento cuja data de vencimento passou e não houve baixa, ele aparece em **vermelho piscante** no dashboard como "Atraso Crítico".
 - **Log de Alterações:** Como honorários de êxito mudam conforme o processo avança, o sistema deve guardar um histórico de quem alterou o **valor_previsto** da causa.
-

4. Próximos Passos Recomendados

Com este projeto lógico, você já temos estrutura para iniciar o desenvolvimento (Backend e Banco de Dados).

Essa é uma excelente decisão. No setor jurídico, onde o tempo dos advogados é faturado por hora, qualquer minuto gasto digitando "Pagamento Cursor US\$ 20.00" é desperdício de dinheiro. A automação de entrada de dados transforma o sistema de um "diário" em uma ferramenta de **auditoria automática**.

Aqui está como estruturar essa automação de forma profissional e segura:

1. Métodos de Captura de Dados

Para extratos bancários e cartões, você tem três caminhos, do mais manual ao mais automatizado:

A. Importação de Arquivos OFX (O Padrão Ouro)

O OFX é o formato universal de extratos bancários.

- **Como funciona:** O analista baixa o arquivo no Internet Banking e faz o upload no seu app.
- **Vantagem:** Seguro e gratuito. Não precisa de integração direta com o banco.
- **Lógica de Banco:** O sistema lê o arquivo, ignora transações já importadas (baseado no ID da transação bancária) e apresenta as novas para o analista "categorizar".

B. Integração via Open Finance / APIs (Automatização Total)

Utilizar agregadores (como Pluggy, Belvo ou Linker) que conectam o seu app diretamente à conta bancária e ao cartão do escritório.

- **Como funciona:** O sistema puxa os lançamentos de hora em hora.
- **Vantagem:** O dashboard está sempre atualizado sem que ninguém move um dedo.
- **Desafio:** Custo por conexão e complexidade de segurança (consentimento do usuário).

C. Leitura de PDFs e Web Scraping (Último Recurso)

Usar bibliotecas de IA (como OCR) para ler faturas de cartão em PDF.

- **Vantagem:** Útil para bancos que não exportam OFX.
- **Desafio:** Propenso a erros se o banco mudar o layout da fatura.

2. A Inteligência de Categorização (O "Cérebro" do App)

Não basta importar os dados; o sistema precisa saber que "[Apple.com/Bill](#)" é uma despesa de software. Para isso, você precisa de uma tabela de **Regras de De-Para**:

Tabela: **Regras_Categorizacao**

- **termo_busca:** VARCHAR (Ex: "CURSOR", "OPENAI", "OAB/SP")
- **id_centro_custo_sugerido:** FK (Software)
- **id_participante_sugerido:** FK (Cursor AI)

Lógica: Ao ler o extrato, o sistema faz um "match" por palavra-chave. Se encontrar "CURSOR" no histórico bancário, ele já preenche automaticamente o lançamento no sistema, deixando apenas para o analista confirmar.

3. O Fluxo de Conciliação Inteligente

Com a automação, o trabalho do analista muda. Em vez de **cadastrar**, ele passará a **conferir**. O sistema deve apresentar uma tela de "Conciliação Pendente":

1. **Lado Esquerdo (Extrato):** O que veio do banco.
2. **Lado Direito (Sistema):** O que já estava previsto (as assinaturas recorrentes que criamos antes).
3. **Ação:** Se os valores e datas batem, o sistema exibe um botão de "Conciliar". Se o sistema não encontrar um par, ele sugere: "*Identificamos um gasto novo de R\$ 150,00 no Cartão. Deseja criar um lançamento agora?*"

4. Vantagens Estratégicas para o Jurídico

- **Identificação de Fraudes/Erros:** Como o cartão de crédito é muito usado, assinaturas esquecidas ou cobranças em duplicidade saltam aos olhos no dashboard.
- **Câmbio Real:** No caso do Cursor, a automação lê o valor exato em Reais que o banco cobrou (já com IOF), substituindo a sua "estimativa" de forma automática.
- **Velocidade no Êxito:** Se o alvará de uma causa ganha cair na conta, o dashboard avisa na hora, sem esperar o advogado avisar o financeiro.

O que você acha de priorizar a importação via OFX como primeira entrega?

É a forma mais rápida de validar a lógica de automação antes de partir para APIs complexas de bancos.

Excelente escolha. O algoritmo de **Matching** (conciliação) é o que separa um sistema financeiro "burro" (que apenas acumula dados) de um sistema inteligente que economiza horas de trabalho.

Abaixo, detalho a lógica do algoritmo e como estruturar o banco para evitar duplicidade.

1. O Identificador Único de Transação (**FitID**)

Para evitar duplicar o mesmo lançamento toda vez que você fizer o upload do mesmo arquivo OFX, precisamos de um campo chamado **banco_transacao_id** (conhecido no padrão OFX como **FITID**).

Regra de Ouro:

Antes de inserir qualquer linha do extrato no banco, o sistema faz um check:

```
IF NOT EXISTS (SELECT 1 FROM Movimentacoes_Extrato WHERE banco_transacao_id = :fitid_importado)
```

2. A Lógica do Algoritmo de Matching

Quando o analista importa o extrato, o algoritmo deve processar cada linha e atribuir um **Status de Confiança**.

Passo A: O Match Perfeito (100% de Confiança)

O sistema marca como conciliado automaticamente se encontrar um lançamento no sistema onde:

- **Valor** é idêntico.
- **Data** é igual ou tem uma variação de até 3 dias (D+3).
- **Tipo** (Receita/Despesa) é o mesmo.
- **Conta Bancária** coincide.

Passo B: O Match Sugerido (70-90% de Confiança)

Se o sistema encontrar o valor exato, mas a data está distante ou o nome do favorecido no extrato é ligeiramente diferente do cadastro (ex: "CURS* AI" vs "Cursor AI"), ele apresenta ao analista:

"Encontramos este lançamento que parece ser o pagamento do Cursor. Deseja vincular?"

Passo C: O Novo Lançamento (0% de Confiança)

Se não houver nada parecido no sistema, o "robô" aplica as **Regras de De-Para** que discutimos antes.

- Se o extrato diz "UBER *TRIP", o sistema já sugere: **Centro de Custo: Transportes / Tipo: Despesa**.

3. Fluxo de Processamento de Dados

4. Estrutura Técnica do Algoritmo (Pseudocódigo)

```
# Exemplo de lógica para processar uma linha do extrato
for transacao in arquivo_ofx:
    # 1. Checa duplicidade de importação
    if existe_no_banco(transacao.fitid):
        continue

    # 2. Tenta encontrar um lançamento equivalente já cadastrado
    match = buscar_lancamento_pendente(
        valor = transacao.valor,
        intervalo_data = (transacao.date - 3, transacao.date + 3)
    )

    if match:
        vincular_e_baixar(match.id, transacao)
    else:
        # 3. Se não achou, aplica regra de IA/Palavra-chave
        sugestao = aplicar_regras_de_para(transacao.memo)
        criar_pre_lancamento(transacao, sugestao)
```

5. Como lidar com o "Custo de Cartão" (IOF e Taxas)

Especialmente para o **Cursor** e softwares estrangeiros, o valor que você previu (ex: R\$ 100,00) nunca será o valor que virá no OFX (ex: R\$ 106,38 devido ao IOF).

A Solução Lógica: No momento do Matching, se o valor for diferente por uma margem pequena (ex: até 10%), o sistema pergunta:

"O valor no extrato é R\$ 6,38 maior. Deseja lançar a diferença como 'Impostos/Taxas Bancárias'?" Se o analista clicar em "Sim", o sistema:

1. Baixa o título principal (R\$ 100,00).
2. Gera um lançamento automático de despesa (R\$ 6,38) no centro de custo "Impostos".
3. Fecha a conciliação com perfeição.

6. Próximo Passo: A Stack Tecnológica

Agora que temos o projeto lógico completo (Banco, Dashboard, Automação de Cartões, Regras de Êxito e Algoritmo de Conciliação), precisamos decidir **como construir isso**.

Stack do Projeto

- **Backend:** Node.js ou Python (Django/FastAPI).

- **Frontend:** React ou Vue.js (com bibliotecas de gráficos como Recharts).
- **Banco:** PostgreSQL (pela robustez com dados financeiros).