# Blueprint Técnico e Estratégico: Ecossistema de Gestão Stofgard Manager

## 1. Sumário Executivo e Visão Arquitetural

### 1.1 Gênese do Projeto e Objetivos de Transformação

O projeto "Stofgard Manager" representa um pivô arquitetural significativo de um framework de software jurídico legado (anteriormente denominado "JURIS-MAIDEN") para um sistema de Planejamento de Recursos Empresariais (ERP) de alta fidelidade e especializado, adaptado para o nicho de mercado de higienização e impermeabilização de estofados. O imperativo estratégico consiste em alavancar aproximadamente 80% da infraestrutura Laravel existente, expurgando rigorosamente a lógica jurídica específica do domínio (processos, tribunais, prazos processuais) para reaproveitar as capacidades centrais de CRUD (Create, Read, Update, Delete) para um modelo de negócio orientado a serviços.1

Esta transformação transcende uma mera "reestilização" da interface; trata-se de uma reimplementação fundamental da camada de lógica de negócio. O objetivo é construir um "Monólito Fechado" — um produto de software estático, robusto e acabado que não requer manutenção futura ou adições de funcionalidades externas uma vez consolidado o Produto Mínimo Viável (MVP).1 Esta filosofia dita uma abordagem de desenvolvimento "medir duas vezes, cortar uma", priorizando a estabilidade e a aderência a fluxos de trabalho específicos em detrimento da escalabilidade para uma base de usuários massiva. O sistema é projetado para um conjunto finito de quatro usuários específicos, eliminando a necessidade de módulos complexos de multi-inquilino (multi-tenancy) ou faturamento SaaS (Software as a Service).1

A análise crítica da requisição inicial, confrontada com as capacidades tecnológicas atuais e as restrições impostas pelo usuário, revela um cenário onde a precisão técnica deve encontrar a simplicidade operacional. A decisão de abandonar a tentativa anterior de "limpeza" do código e reiniciar "do zero" (mas com reaproveitamento de lógica) é validada pela necessidade de eliminar débitos técnicos ocultos que poderiam comprometer a estabilidade do novo sistema. A arquitetura proposta não é apenas um software; é a espinha dorsal digital da operação da Stofgard, desenhada para ser invisível em sua complexidade e evidente em sua utilidade.

### 1.2 A Filosofia de Experiência do Usuário "Tablet-First"

Uma característica definidora deste projeto é a diretriz estrita de "Tablet-First / Mobile-First". Diferentemente dos ERPs tradicionais projetados para interações desktop com mouse e teclado, o Stofgard Manager tem como dispositivo de interface primário tablets Android. Este requisito deriva da realidade demográfica da base de usuários, que inclui indivíduos mais velhos que necessitam de acessibilidade aprimorada e interfaces com alta legibilidade.1

A arquitetura de UI/UX deve, portanto, desviar-se das convenções padrão de painéis administrativos do FilamentPHP. A navegação lateral (sidebar), tipicamente um pilar do design de dashboards, é identificada como uma obstrução visual em telas sensíveis ao toque menores. Consequentemente, o dashboard deve funcionar como uma "Launchpad" central, utilizando blocos (tiles) grandes e de alto contraste para a navegação entre módulos, mimetizando um lançador de aplicativos nativo de sistemas móveis.1 Campos de entrada devem ser "modestos" — otimizados para alvos de toque sem exigir rolagem vertical excessiva — e a densidade de dados deve ser equilibrada com a legibilidade.

A decisão de remover a barra lateral não é apenas estética; é funcional. Em um tablet, o espaço horizontal é um recurso premium. A barra lateral consome aproximadamente 250-300 pixels de largura, o que em uma tela de 10 polegadas representa uma perda significativa de área útil para tabelas de dados e formulários. A estratégia de injetar CSS para ocultar a sidebar especificamente na rota do Dashboard, mantendo-a acessível (porém colapsada ou via menu hambúrguer) nas páginas internas, demonstra um entendimento refinado da usabilidade em dispositivos móveis.

### 1.3 Análise Crítica das Avaliações Prévias de IA

Uma análise profunda da revisão preliminar do projeto conduzida pela IA Perplexity revela uma pontuação de competência de aproximadamente 9/10 para a ideia do usuário e 7/10 para a análise do Gemini, com falhas críticas na extração de dados e na lógica de conformidade.1 Especificamente, a análise anterior:

* **Falha na Extração de Dados de Precificação:** Negligenciou a lógica de precificação detalhada embutida nas tabelas HTML (TABELA-HIGI.html e TABELA-IMPER.html), que contém a lógica central de receita do negócio, incluindo a diferenciação entre preços à vista e a prazo.
* **Ignorância da Conformidade Fiscal:** Desconsiderou a exigência mandatória de emissão de Nota Fiscal de Serviço Eletrônica (NFS-e), uma obrigação legal crítica para uma entidade CNPJ no Brasil, especialmente no contexto de Ribeirão Preto.1
* **Subestimação da Complexidade de Sincronização:** Categorizou a sincronização do Google Calendar como um objetivo de Fase 2. No entanto, o usuário elevou isso a um requisito não negociável do MVP, exigindo uma implementação OAuth2 mais sofisticada imediatamente.1
* **Fragilidade Técnica no Frontend:** Sugeriu o uso de seletores CSS modernos como :has(), que, embora poderosos, possuem suporte inconsistente em versões mais antigas de navegadores móveis que podem estar presentes nos tablets da empresa.

Este relatório retifica essas omissões integrando as tabelas de preços corretas (ajustadas para Dezembro 2025), definindo um caminho concreto de integração de NF-e para Ribeirão Preto e detalhando uma arquitetura de sincronização de calendário bidirecional robusta. A seguir, detalharemos como cada um desses pontos será abordado na nova arquitetura.

## 2. Análise do Domínio de Negócio e "Modus Operandi"

### 2.1 Canais de Receita e Lógica de Parceiros

A arquitetura financeira do Stofgard é bifurcada em dois fluxos de receita distintos, cada um exigindo tratamento específico dentro dos módulos ServiceOrder e Financial. A compreensão precisa destes fluxos é vital para a modelagem correta do banco de dados e das regras de negócio automáticas.

#### 2.1.1 Canal de Parceria (70% do Volume)

Este é o fluxo crítico de receita, onde a dependência de terceiros exige um controle rigoroso de contas a pagar (comissões).

* **Mecanismo:** Grupos de lojas de móveis parceiras vendem o serviço de impermeabilização como um "add-on" (adicional) durante a venda do móvel.
* **Fluxo de Dados:** A loja fornece os dados de contato e entrega do cliente à Stofgard. O sistema deve ser capaz de registrar o "Número de Serviço do Parceiro" (um ID externo) para reconciliação futura.
* **Lógica Financeira Crítica:** O sistema deve calcular automaticamente uma **comissão de 30%** pagável ao vendedor da loja. Este cálculo deve incidir sobre o *valor acordado* (Valor Acordado), e não sobre o *valor sugerido*. Esta comissão representa uma conta a pagar no módulo financeiro e deve ter seu próprio status de pagamento (Pendente/Pago).1 A falha em gerir este passivo pode corroer a margem de lucro e prejudicar a relação com os parceiros.

#### 2.1.2 Canal Direto (30% do Volume)

* **Mecanismo:** Aquisição de clientes via marketing em redes sociais (Instagram/Facebook/Google Ads).
* **Lógica Financeira:** Retenção total da receita, sem comissão pagável a terceiros. O campo de "Vendedor Parceiro" na Ordem de Serviço deve ser nulo ou preenchido com "Interno", zerando a lógica de cálculo de comissão.

### 2.2 Tipologia de Serviço e Lógica de Garantia

O sistema é restrito a dois tipos de serviço, simplificando as relações polimórficas frequentemente encontradas em ERPs genéricos, mas exigindo profundidade na lógica de pós-venda.

1. **Higienização (Cleaning):** Um serviço padrão com datas de execução de início e fim. O ciclo de vida encerra-se com o pagamento e a confirmação da execução.
2. **Impermeabilização (Waterproofing):** Este serviço carrega uma regra de negócio crítica — uma **Garantia de 1 Ano**.
   * **Requisito Técnico:** O sistema deve instanciar uma warranty\_expiration\_date (data de expiração da garantia) no momento da conclusão de uma ordem de impermeabilização. Esta data deve ser imutável após a finalização do serviço para garantir a integridade contratual.
   * **Requisito Visual:** Um cronômetro de contagem regressiva deve ser exibido na visualização da ordem. A lógica dita um ícone Verde para "Garantia Ativa" (dias restantes > 0) e um ícone Vermelho para "Garantia Expirada" (dias restantes ≤ 0).1 Este indicador visual imediato permite que o atendente valide a cobertura de garantia em segundos durante um atendimento telefônico, sem a necessidade de cálculos manuais.

### 2.3 Estrutura de Precificação e Ajuste Inflacionário

A base de dados de precificação não é estática; ela baseia-se em um valor base indexado pela inflação (IPCA). A análise das tabelas HTML fornecidas (TABELA-HIGI e TABELA-IMPER) estabelece a data de referência de Dezembro de 2025. O sistema deve ser capaz de armazenar estes valores históricos para garantir que orçamentos antigos não sejam alterados retroativamente por atualizações de preço futuras.

* **Lógica de Ajuste:** Os valores atuais refletem um ajuste de **+22,50%** sobre os preços base de Novembro de 2021. O sistema não deve calcular isso em tempo real para evitar erros de arredondamento em queries de banco de dados; os valores já ajustados devem ser persistidos na tabela de preços.
* **Níveis de Pagamento:**
  + **À Vista:** O preço base com desconto implícito.
  + **À Prazo:** Inclui um *markup* de aproximadamente 10-15% para cobrir custos financeiros (taxas de cartão, risco de inadimplência). O sistema deve apresentar ambos os valores ao operador durante a confecção do orçamento.

Tabela 1: Análise Amostral de Precificação (Impermeabilização) 1

| **Item** | **Preço Base (Nov 2021)** | **Preço Ajustado (Dez 2025) (+22,5%)** | **Preço a Prazo (Dez 2025)** |
| --- | --- | --- | --- |
| **Sofá 2 Lugares (Retrátil)** | R$ 350,00 | **R$ 428,75** | R$ 453,25 |
| **Sofá 3 Lugares (Retrátil)** | R$ 420,00 | **R$ 514,50** | R$ 551,25 |
| **Colchão Casal** | R$ 380,00 | **R$ 465,50** | R$ 490,00 |
| **Cadeira de Jantar** | R$ 50,00 | **R$ 61,25** | R$ 73,50 |

O usuário declarou explicitamente que os valores atuais da tabela são válidos até Dezembro de 2026, simplificando a lógica imediata para usar as colunas de "Preço Ajustado" diretamente, sem a necessidade de uma *engine* de cálculo de IPCA em tempo real para o MVP.1

## 3. Arquitetura Técnica e Estratégia de Stack

### 3.1 Definição do Stack Central

A escolha tecnológica deve refletir a necessidade de robustez, rapidez de desenvolvimento e longevidade do código ("Closed Monolith").

* **Framework: Laravel 12.x (PHP 8.2+):** Esta escolha fornece a robustez necessária para a estrutura de monólito e garante suporte de segurança a longo prazo sem atualizações frequentes.1 O ecossistema Laravel oferece ferramentas nativas para filas, agendamento de tarefas e autenticação que são essenciais para as regras de negócio da Stofgard.
* **Painel Administrativo: FilamentPHP v3:** Escolhido por suas capacidades de desenvolvimento rápido (stack TALL: Tailwind, Alpine.js, Laravel, Livewire). O Filament é a escolha ideal para criar a interface "Tablet-First" devido à sua natureza responsiva, embora exija personalizações específicas para ocultar a barra lateral em desktops e tablets em rotas específicas.2 A modularidade do Filament permite criar "Resources" isolados para Clientes e OS, facilitando a manutenção.
* **Banco de Dados: MySQL 8.0 (ou MariaDB):** O esquema deve utilizar estritamente **UUIDs** (Identificadores Únicos Universais) para todas as chaves primárias. Isso garante a portabilidade dos dados e a segurança, prevenindo ataques de enumeração de ID (onde um concorrente poderia estimar o volume de vendas apenas incrementando IDs na URL).1
* **Motor de Relatórios: DomPDF:** Selecionado para gerar orçamentos e ordens de serviço em PDF formais, aderindo a requisitos rigorosos de layout (Logo, CNPJ, remoção de endereço físico).1 A facilidade de estilizar PDFs com CSS/HTML torna o DomPDF superior a soluções binárias para este caso de uso.

### 3.2 Infraestrutura e Implantação

* **Provedor: VPS Contabo (Alemanha):** Esta escolha oferece uma alta relação desempenho-custo. A localização na Alemanha não introduz latência perceptível para uma aplicação administrativa interna no Brasil.
* **Servidor Web:** Nginx configurado com PHP-FPM.
* **Gerenciador de Processos:** Supervisor (essencial para *queue workers*, necessários para a Sincronização de Calendário e envio de E-mails). Sem o Supervisor, a sincronização do calendário falharia silenciosamente ou causaria timeouts na interface do usuário.
* **CI/CD:** Recomenda-se a implantação automatizada via GitHub Actions para agilizar o fluxo de trabalho de codificação baseado em Agentes.1 Isso permite que o código gerado pela IA seja testado e implantado com um único comando git push.

### 3.3 Modelo de Segurança de "Sistema Fechado"

O controle de acesso é codificado para uma lista branca específica de quatro usuários, aplicando uma política estrita de "Zero Trust" para qualquer registro externo. Não haverá página de registro público.

Tabela 2: Matriz de Usuários Autorizados 1

| **Nome** | **Função** | **Email** | **Senha Inicial (Hash)** |
| --- | --- | --- | --- |
| **Allisson Gonçalves** | **Admin** | allisson@stofgard.com.br | Swordfish |
| **Maria de Jesus** | Operador | maria@stofgard.com.br | Stofgard2026 |
| **Jaelsa Maria** | Operador | jaelsa@stofgard.com.br | Stofgard2026 |
| **Raelcia Maria** | Operador | raelcia@stofgard.com.br | Stofgard2026 |

A base de código deve remover explicitamente as rotas do RegistrationController e desativar os fluxos de trabalho genéricos de Illuminate\Auth\MustVerifyEmail para impedir qualquer vetor de acesso não autorizado. As senhas devem ser alteradas no primeiro login, forçando uma atualização de segurança.

## 4. Aprofundamento: Implementação de Funcionalidades Críticas

### 4.1 O Dashboard "Tablet-First" & Personalização de UI

O layout padrão do FilamentPHP inclui uma barra lateral colapsável que persiste em visualizações de desktop/tablet. Para atender ao requisito de "Landing Page" do usuário para o Dashboard, devemos implementar um renderHook específico ou uma sobreposição de CSS que suprima a barra lateral *apenas* na rota do dashboard, criando uma experiência focada, semelhante a um aplicativo.1

**Estratégia de Implementação:**

1. **Injeção de CSS:** Utilizar Filament::renderHook('panels::body.end') dentro do AdminPanelProvider para injetar um bloco <style> que alveja a classe da barra lateral (.fi-sidebar) e define display: none!important especificamente quando a classe do corpo indica a página do dashboard (ex: .fi-page-dashboard). Isso garante que a navegação lateral não consuma pixels valiosos na tela inicial do tablet.
2. **Grid de Navegação:** Substituir o grid de widgets padrão por um componente Blade personalizado que renderiza "Tiles" (Cartões) grandes e amigáveis ao toque para Clientes, OS, Agenda, Financeiro, Orçamento e Inventário. Estes cartões devem ocupar 100% da largura em celulares e 50% ou 25% em tablets, utilizando as classes de grid responsivo do Tailwind (grid-cols-1 md:grid-cols-2 lg:grid-cols-4).
3. **Widget de Clima:** Integrar a API OpenWeatherMap (ou similar gratuita) visando as coordenadas de **Ribeirão Preto (-21.1775, -47.8103)**. Este widget serve como uma âncora visual no dashboard branco, fornecendo utilidade imediata e personalização local.1
4. **Otimização de Toque:** Aumentar o padding de todos os botões e inputs no arquivo de configuração do Tailwind ou através de um tema personalizado do Filament. Botões devem ter altura mínima de 44px (padrão de toque da Apple/Google) para evitar erros de clique por usuários com menor precisão motora.

### 4.2 A Sincronização "Não Negociável" do Google Calendar

Este é o requisito técnico mais complexo e arriscado. O usuário exige **sincronização bidirecional em tempo real**. Uma "Conta de Serviço" genérica (que compartilha um único calendário) é insuficiente porque cada um dos 4 usuários precisa que sua própria agenda de trabalho seja refletida em seu Google Calendar pessoal.1

**Abordagem Arquitetural:**

1. **Multi-Inquilino OAuth2:** Não podemos usar um arquivo JSON de credencial simples de conta de serviço. Devemos implementar a **Biblioteca PHP Google\_Client** combinada com o **Laravel Socialite** para gerenciar o *handshake* OAuth2 individualmente para cada usuário.
2. **Armazenamento de Tokens:** A tabela users requer colunas para google\_access\_token, google\_refresh\_token e google\_calendar\_id. O token de atualização é crítico para manter a conexão viva sem exigir que o usuário faça login novamente a cada hora.
3. **Lógica de Sincronização:**
   * *Stofgard para Google:* Quando um Event ou ServiceOrder é salvo (via Eloquent Observer), um Job é despachado (SyncToGoogleCalendar). Este Job utiliza o token armazenado do usuário para enviar o evento para a API do Google. O uso de filas (Queues) é obrigatório aqui para não travar a interface do usuário enquanto a API do Google responde.
   * *Google para Stofgard:* Implementar webhooks (push notifications) do Google é tecnicamente exigente para uma configuração VPS (requer domínio HTTPS verificado e endpoints públicos sempre ativos). Uma alternativa robusta para este MVP é um **Comando Agendado** (executado a cada 5-10 minutos) que consulta a API do Google Calendar buscando alterações (events.list com syncToken) e atualiza o banco de dados local.4 Isso evita a complexidade de gerenciar webhooks e lida melhor com falhas de rede.

**Por que o pacote da Spatie pode ser insuficiente:** O pacote padrão spatie/laravel-google-calendar é otimizado para uma única Conta de Serviço (servidor-para-servidor). Para sincronização de calendário pessoal multiusuário, uma implementação personalizada encapsulando o google/apiclient é preferível para lidar corretamente com as atualizações de token individuais e escopos de permissão.6

### 4.3 Emissão Automatizada de NF-e (Contexto Ribeirão Preto)

O usuário é um MEI em Ribeirão Preto. A cidade utiliza um provedor de webservice específico (frequentemente **ISSDigital**, **Ginfes** ou **Coderp**, dependendo do contrato municipal vigente). A integração direta com APIs SOAP municipais é notoriamente difícil devido à documentação pobre e instabilidade frequente.8

Recomendação Estratégica:

Para evitar "reinventar a roda" e arriscar a não conformidade fiscal, o sistema deve integrar-se com uma API de Middleware como NFe.io, Focus NFe ou Webmania. Estes serviços abstraem a complexidade do SOAP municipal em uma API JSON REST limpa.

* **Custo vs. Benefício:** O custo (~R$ 0,30 - R$ 1,00 por nota) é negligenciável comparado às horas de desenvolvimento necessárias para manter uma conexão SOAP direta, que frequentemente quebra quando a prefeitura atualiza seus sistemas.
* **Implementação:**
  1. **Fase 1 (MVP):** Emissão manual via Gov.br (Portal Nacional).9 O sistema gera um relatório com os dados necessários para o usuário copiar e colar.
  2. **Fase 2 (Automatizada):** Job Laravel (EmitirNFS-e) acionado quando uma Transação Financeira é marcada como "Paga". O Job envia um payload JSON para a API de Middleware contendo os detalhes da ServiceOrder.
* **Especificidades de Ribeirão Preto:** A integração deve respeitar os códigos de tributação municipal para serviços de "Limpeza/Conservação". O middleware lida com a geração do XML ABRASF 2.04 exigido pela cidade, garantindo que a nota não seja rejeitada por erros de sintaxe.8

### 4.4 Estratégia de Importação de Contatos Móveis

O usuário solicitou a importação de contatos diretamente do telefone. Navegadores restringem o acesso direto à agenda de endereços do dispositivo por razões de segurança, tornando uma "Web API" tradicional inviável para leitura em massa sem interação do usuário.

A Solução: Contact Picker API:

O Chrome moderno para Android suporta a API navigator.contacts.select(['name', 'tel'], {multiple: true}). Isso permite que um aplicativo web (PWA) acione um seletor nativo do sistema operacional onde o usuário seleciona os contatos que deseja compartilhar com o aplicativo.11 Esta é a solução mais elegante e nativa.

* **Fallback (Plano B):** Se a API não for suportada (ex: em Desktop ou iOS antigo), o sistema deve ter um recurso de **Upload de vCard (.vcf)**. O usuário exporta seus contatos para um arquivo no celular e faz o upload para o sistema, que utiliza uma biblioteca PHP (como jeroendesloovere/vcard) para parsear e importar os dados em massa.12

## 5. Esquema de Banco de Dados e Modelagem de Dados

O banco de dados deve utilizar **UUIDs** como chaves primárias. Abaixo está a definição refinada do esquema para a lógica de negócio Stofgard.

### 5.1 Entidades Centrais

**Tabela: users**

* id (UUID, PK)
* name, email, password
* google\_id, google\_token, google\_refresh\_token (Para sincronização do Calendar)
* role (enum: admin, operator)
* initials (String, 2-3 chars, para logs de auditoria)

**Tabela: clients**

* id (UUID, PK)
* name (String, Obrigatório)
* phone, mobile (String, com lógica auxiliar para WhatsApp)
* address\_line, district, city (Padrão: Ribeirão Preto)
* google\_maps\_link (Coluna virtual/computada ou helper)
* notes (Text)
* source (enum: partner\_store, instagram, google, referral)
* created\_by (User Initials)

**Tabela: partners (Lojas de Móveis)**

* id (UUID, PK)
* name (ex: "Magazine Luiza", "Loja X")
* default\_commission\_rate (Decimal, Padrão: 30.00)
* contact\_person (String)

**Tabela: service\_orders (OS)**

* id (UUID, PK)
* client\_id (Chave Estrangeira)
* partner\_id (Chave Estrangeira, Nulável - para 70% dos casos)
* salesperson\_name (String, nome do vendedor da loja)
* partner\_os\_number (String, ID Externo da loja)
* service\_type (enum: higienizacao, impermeabilizacao)
* status (enum: pending, scheduled, in\_progress, completed, cancelled)
* execution\_date (Date)
* warranty\_end\_date (Date, Nulável. Lógica: execution\_date + 1 ano se o tipo for Impermeabilização)
* value\_agreed (Decimal, 10,2)
* commission\_amount (Decimal, 10,2. Lógica: value\_agreed \* 0.30 se parceiro existir)
* payment\_status\_client (enum: pending, paid)
* payment\_status\_commission (enum: pending, paid)
* created\_by (User Initials)
* created\_at, updated\_at

**Tabela: budgets (Orçamentos)**

* id (UUID, PK)
* client\_name (String, permite orçamento rápido sem criação completa de cliente)
* items (JSON: Array de itens de serviço e preços)
* total\_value (Decimal)
* discount\_type (enum: none, pix\_10, round\_down, manual)
* expires\_at (DateTime, Padrão: +7 dias)
* status (enum: draft, sent, approved, expired)

**Tabela: inventory\_items (Almoxarifado)**

* id (UUID, PK)
* name (ex: "Impermeabilizante Apex", "Extratora")
* type (enum: equipment, chemical)
* quantity (Decimal - Litros para químicos, Unidades para equipamentos)
* min\_threshold (Decimal, Padrão: 20.0 para químicos - o gatilho de "um galão")

**Tabela: financial\_transactions**

* id (UUID, PK)
* type (enum: income, expense)
* category (enum: service\_revenue, commission\_payment, supply\_purchase, operational)
* amount (Decimal)
* due\_date (Date)
* payment\_date (Date, Nullable)
* related\_os\_id (FK, Nullable)

## 6. Especificações Funcionais dos Módulos

### 6.1 Dashboard (A "Launchpad")

* **Visuais:** Fundo branco puro (#FFFFFF) para maximizar o contraste em ambientes iluminados.
* **Canto Superior Esquerdo:** Logo Stofgard (SVG) - Link fixo (Hardcoded) para /admin.
* **Canto Superior Direito:** Sino de Notificação (Filament Database Notifications) para alertas de estoque e lembretes.
* **Grid Central:**
  + **Cartão de Clima:** Busca dados em https://api.open-meteo.com para Ribeirão Preto. Exibe Temperatura, Ícone e "Bom trabalho, [Usuário]".
  + **Ações Rápidas:** Azulejos Grandes para "Novo Cliente", "Nova OS", "Novo Orçamento". Estes devem ser os elementos mais proeminentes da UI.
  + **Alertas:**
    - "Estoque Crítico" (Vermelho) se qualquer químico < 20L.
    - "Lembretes Pendentes" se Agenda::where('status', 'pending')->count() > 0.

### 6.2 Lógica de Ordens de Serviço e Garantia

* **Contagem Regressiva da Garantia:** Uma entrada personalizada na Infolist do Filament.
  + *Lógica:* Carbon::parse($warranty\_end\_date)->diffInDays(now()).
  + *Visual:*
    - Se > 30 dias: Badge Verde "Garantia Ativa".
    - Se < 30 dias: Badge Amarelo "Vencendo".
    - Se < 0: Badge Vermelho "Expirada".
* **Comissões:** O modelo ServiceOrder deve ter um método booted para calcular automaticamente a comissão ao salvar, garantindo que o valor financeiro nunca desincronize do valor do serviço:  
  PHP  
  static::saving(function ($os) {  
   if ($os->partner\_id && $os->value\_agreed) {  
   // Assume-se que a taxa padrão é 30%, a menos que o parceiro tenha uma taxa específica  
   $rate = $os->partner->default\_commission\_rate?? 0.30;  
   $os->commission\_amount = $os->value\_agreed \* $rate;  
   } else {  
   $os->commission\_amount = 0;  
   }  
  });

### 6.3 Financeiro e Orçamentação

* **Orçamentação:** A lógica de "Arredondamento para Baixo" é específica e deve ser implementada no método de cálculo do orçamento.
  + *Algoritmo:* $total = floor($total / 50) \* 50; (Arredonda 280 -> 250, 145 -> 100). Isso força "centenas ou metades" conforme solicitado.
* **Geração de PDF:**
  + **Cabeçalho:** Logo à Esquerda, Texto à Direita.
  + **Conteúdo:** "CNPJ: 58.794.846/0001-20", "Tel: (16) 99104-0195".
  + **Exclusão:** O modelo de visualização (Blade template) deve filtrar agressivamente qualquer linha de endereço para cumprir a regra de "Sem Endereço Físico".1

### 6.4 Inventário e "O Galão"

* **Lógica Visual:** Um componente Blade personalizado no recurso do Filament.
  + Exibição: floor($quantity / 20) ícones de um Galão (Jerrycan SVG).
  + Se quantity < 20, exibir um ícone de alerta vermelho piscante e disparar uma notificação de banco de dados para o Admin.

## 7. Roteiro de Implementação (8 Semanas)

### Fase 1: Fundação (Semana 1)

* **Objetivo:** Limpar código legado e configurar ambiente.
* **Tarefas:**
  + Clonar repositório atual.
  + Excluir modelos/controladores de Processo, Audiencia, Tribunal.
  + Configurar VPS Contabo (Nginx, PHP 8.2, MySQL).
  + Configurar Google Cloud Console (Tela de Consentimento OAuth, Ativar API Calendar).5

### Fase 2: Banco de Dados e Modelos Centrais (Semanas 2-3)

* **Objetivo:** Estrutura de dados estável.
* **Tarefas:**
  + Criar Migrations (apenas UUIDs).
  + Implementar Seeders para os 4 usuários (garantindo acesso imediato).
  + Semear a tabela de Preços com os valores de +22,5% extraídos dos arquivos HTML.
  + Implementar lógica de ContactPicker no frontend para importação de contatos.

### Fase 3: O Dashboard e Calendário (Semanas 4-5)

* **Objetivo:** A Experiência "Tablet".
* **Tarefas:**
  + CSS Personalizado para ocultar Sidebar na rota do Dashboard.
  + Implementar widget OpenWeather.
  + **Crítico:** Implementar Sincronização do Google Calendar.
    - Passo 1: Fluxo de login Socialite para vincular Conta Google.
    - Passo 2: Configuração Spatie\GoogleCalendar usando o token do usuário (Configuração Dinâmica).
    - Passo 3: Criar Job de Sincronização Bidirecional.

### Fase 4: Lógica de Negócio e Relatórios (Semanas 6-7)

* **Objetivo:** Automação e Documentação.
* **Tarefas:**
  + Ação de conversão Orçamento -> OS (Replicar dados, criar registros).
  + Lógica de Cálculo de Comissão.
  + Templates DomPDF (Formatação estrita).
  + Lógica de Contagem Regressiva da Garantia.

### Fase 5: Testes e Implantação (Semana 8)

* **Objetivo:** Prontidão para Produção.
* **Tarefas:**
  + Testes Unitários (PestPHP) para cálculos de comissão e arredondamento (crítico para não perder dinheiro).
  + Testes de Navegador (Dusk) para a responsividade da UI em Tablet.
  + Deploy Final para sistema.stofgard.com.br.

## 8. Instruções para o Agente de IA (A Estratégia ".cursorrules")

Para garantir que o agente de IA (Cursor/Windsurf) execute este plano sem alucinações, o seguinte conjunto de regras deve ser aplicado no arquivo .cursorrules da raiz do projeto 14:

#.cursorrules para Stofgard Manager

## Persona

Você é um Arquiteto Laravel Sênior. Você favorece soluções robustas e estáticas em detrimento de complexas e escaláveis. Este é um "Monólito Fechado".

## Restrições Críticas

1. **Sem Sidebar no Dashboard:** Você deve injetar CSS para ocultar .fi-sidebar APENAS na rota do dashboard.
2. **Apenas UUIDs:** Todas as migrações devem usar $table->uuid('id')->primary();.
3. **Lista de Usuários Estrita:** Permitir login apenas para os 4 emails hardcoded no Seeder.
4. **Tablet First:** Todos os formulários Filament devem usar ->columns(1) ou ->columns(2) no máximo. Sem grids complexos. Inputs devem ter ->size('lg') para alvos de toque.

## Imposições de Lógica de Negócio

1. **Comissão:** Sempre calcular 30% de comissão se partner\_id estiver presente na OS.
2. **Precificação:** Usar os valores cacheados do PriceTableSeeder (Valores de Dez 2025). Não aplicar matemática de IPCA dinâmica; usar os valores fixos.
3. **Relatórios:** NUNCA renderizar o endereço da empresa em resources/views/pdf. APENAS CNPJ e Telefone.

## Tech Stack

* Laravel 12
* Filament v3
* DomPDF
* Spatie/Google-Calendar (Customizado para OAuth multiusuário)

## 9. Conclusão

A transformação do software jurídico legado no Stofgard Manager é um projeto viável e de alto valor. A complexidade não reside nas operações CRUD, mas nas integrações específicas (Google Calendar, NF-e) e nos requisitos rigorosos de UI (Tablet-first, amigável para idosos). Ao aderir à arquitetura de "Monólito Fechado" e alavancar o robusto ecossistema Filament com substituições direcionadas, o sistema fornecerá a estabilidade e a funcionalidade específica exigida pelo negócio familiar da Stofgard. A inclusão da sincronização "Não Negociável" do Calendário no MVP adiciona peso técnico, mas é solucionável via a abordagem Híbrida Socialite/Spatie delineada acima. O roteiro é realista em ~140-170 horas, garantindo um deploy pronto para produção em 8 semanas. A aderência estrita a este plano garantirá que o software não seja apenas uma ferramenta, mas um ativo estratégico que respeita o fluxo de trabalho e a demografia de seus usuários.

#### Referências citadas

1. Gemini.txt
2. Filament: Collapse Sidebar Menu (with or without icons), acessado em dezembro 30, 2025, <https://filamentexamples.com/tutorial/collapsible-navigation-sidebar>
3. Navigation - Panels - Filament, acessado em dezembro 30, 2025, <https://filamentphp.com/docs/3.x/panels/navigation>
4. Compact Guide: How to sync Google calendar with Laravel - DEV Community, acessado em dezembro 30, 2025, <https://dev.to/dnsinyukov/compact-guide-how-to-sync-google-calendar-with-laravel-145g>
5. How To Integrate Google Calendar API and friendship with Laravel. Part 2 - DEV Community, acessado em dezembro 30, 2025, <https://dev.to/dnsinyukov/how-to-integrate-google-calendar-api-and-friendship-with-laravel-part-2-48ba>
6. spatie/laravel-google-calendar: Manage events on a Google Calendar - GitHub, acessado em dezembro 30, 2025, <https://github.com/spatie/laravel-google-calendar>
7. How to connect users' calendars to my app? - Laracasts, acessado em dezembro 30, 2025, <https://laracasts.com/discuss/channels/general-discussion/how-to-connect-users-calendars-to-saas-app>
8. Guia técnico de emissão de NFS-e em Ribeirão Preto/SP pela API da Focus NFe, acessado em dezembro 30, 2025, <https://focusnfe.com.br/guides/nfse/municipios-integrados/ribeirao-preto-sp/>
9. Nota Fiscal de Serviço Eletrônica (NFS-e) — Empresas & Negócios - Portal Gov.br, acessado em dezembro 30, 2025, <https://www.gov.br/empresas-e-negocios/pt-br/empreendedor/servicos-para-mei/nota-fiscal/nota-fiscal-de-servico-eletronica-nfs-e>
10. Liberação para emissão de NFS-e - Ribeirão Preto - SP - Webmania® - Central de Ajuda, acessado em dezembro 30, 2025, <https://ajuda.webmania.com.br/pt-BR/articles/12680519-liberacao-para-emissao-de-nfs-e-ribeirao-preto-sp>
11. A contact picker for the web | Capabilities - Chrome for Developers, acessado em dezembro 30, 2025, <https://developer.chrome.com/docs/capabilities/web-apis/contact-picker>
12. jeroendesloovere/vcard: This vCard PHP library can easily parse or generate/export vCards as .vcf - GitHub, acessado em dezembro 30, 2025, <https://github.com/jeroendesloovere/vcard>
13. A Straightforward Guide to Code Documentation & Handovers | Praxent, acessado em dezembro 30, 2025, <https://praxent.com/blog/software-handover-documentation-checklist>
14. Comprehensive Laravel PHP Cursor Rules: Best Practices and Key Principles., acessado em dezembro 30, 2025, <https://cursor.directory/laravel-cursor-rules>
15. awesome-cursorrules/rules/laravel-php-83-cursorrules-prompt-file/.cursorrules at main · PatrickJS/awesome-cursorrules - GitHub, acessado em dezembro 30, 2025, <https://github.com/PatrickJS/awesome-cursorrules/blob/main/rules/laravel-php-83-cursorrules-prompt-file/.cursorrules>