

Svelto SaaS - Roteiro de Execução & Contexto (Status Report)

Este documento contextualiza o estado atual do projeto Svelto para novos desenvolvedores ou agentes de IA.

1. Onde Estamos (Milestone 0 - Concluído)

Fase de Fundação e Arquitetura.

Nós completamos a estrutura base do sistema, definindo as regras inquebráveis de engenharia financeira e montando o ambiente de desenvolvimento.

O que já está pronto:

1. **Ambiente de Desenvolvimento (Monorepo):**
 - Estrutura criada com NestJS (API) e Next.js (Web).
 - Versões travadas para estabilidade: Node LTS e **Prisma v5.22.0**.
 - Ambiente configurado para rodar localmente no Windows sem conflitos de npx.
2. **Infraestrutura Cloud (Railway):**
 - Banco de Dados **PostgreSQL** provisionado e conectado.
 - Sistema de Filas **Redis** provisionado (para processamento futuro).
3. **Arquitetura de Dados (Schema v2.0 - Blindado):**
 - Banco de dados desenhado com suporte a **Multi-tenancy (RLS)**.
 - Tipagem financeira correta: DECIMAL(19,4) para valores monetários.
 - **Engenharia da Verdade:** Separação clara entre tabelas Transaction (Venda), Settlement (Liquidação) e Payout (Saque).
 - **Auditoria:** Tabela AuditLog com suporte a JSON Diff.
4. **Deep Research (Regras de Negócio):**
 - Mapeamento completo da API do Mercado Pago (Diferença entre /payments e /settlement/report).
 - Regra de Baixa do Omie: Taxas (MDR) devem ir no campo nValDesc.
 - Lógica de Conciliação: Vínculo reverso (Relatório -> Venda) e não o contrário.

2. Próximos Passos (Milestone 1 - Execução / MVP)

Fase de Conectividade e Ingestão.

Agora que a casa está arrumada, vamos começar a trazer dados reais.

Passo 2.1: Segurança & Criptografia (PRIORIDADE IMEDIATA)

Antes de conectar qualquer conta, precisamos de um cofre seguro.

- **Tarefa:** Implementar o SecurityService no NestJS.
- **Requisito:** Usar AES-256 (Envelope Encryption) para cifrar os tokens de acesso do Mercado Pago antes de salvar no banco Integration.
- **Motivo:** Compliance de segurança. Não podemos ter chaves de API expostas no banco.

Passo 2.2: O Conector Mercado Pago

- **Tarefa:** Criar o MercadoPagoService.
- **Requisito:** Implementar método auth (OAuth ou validação de token) e método fetchPayments (Ingestão Operacional).
- **Detalhe:** Mapear o JSON da API para a tabela Transaction, explodindo as taxas (fee_details) nos campos corretos (amountMdrFee, amountFinancingFee).

Passo 2.3: API de Ingestão (Backend)

- **Tarefa:** Criar o Controller POST /integrations/mercadopago-sync.
- **Funcionalidade:** Receber o comando do usuário para "Sincronizar Agora" e disparar o Job de busca.

Passo 2.4: Dashboard de Conexão (Frontend)

- **Tarefa:** Criar a tela no Next.js onde o usuário cola o Access Token do Mercado Pago.
- **Fluxo:** Frontend -> API (Criptografa e Salva) -> Retorna Sucesso.

3. Visão de Futuro (Milestone 2 - Conciliação)

Fase de Inteligência e ERP.

- **Motor de Auditoria:** Script que compara amountNetGateway vs amountNetCalculated.
- **Conector Omie:** Implementar a lógica de "Check-then-Act" (Consultar antes de Baixar).
- **Conciliador Automático:** O "Worker" que roda toda madrugada, lê o Relatório de Liquidação e executa as baixas no Omie.

Resumo para o Agente IA (Svelto Architect)

Você está assumindo o projeto no início da Fase de Execução.

A arquitetura já foi decidida e não deve ser alterada sem forte justificativa.

Seu foco agora é escrever código (Services, Controllers) que respeite o Schema v2.0 definido no Blueprint.