

Sistema de Compras, Estoque e Locação — Briefing Mestre

Grupo 1AMS · A Moderna Sany / ISANQUI / 1AMS LOG Destino: Claude Code · documento único e autossuficiente

Como usar este documento

Este é o escopo completo. **Construir o MVP primeiro** (ver seção MVP abaixo) — é independente e pode ser testado de imediato. O banco de dados deve nascer já contemplando todos os módulos, para que os Módulos 2 e 3 e a integração com o ADMServ entrem depois como atualizações, **sem reconstruir** o que já existe.

Visão geral — 4 frentes

1. **Módulo 1 — Compras** (requisição, aprovação, preço, desperdício, dashboard) — *construir primeiro*
2. **Módulo 2 — Estoque QR** (2A consumo · 2B ativos/banheiros químicos)
3. **Módulo 3 — Logística/Locação** (entrega via celular, vínculo por QR)
4. **Integração ADMServ** — *adiada* (o ADMServ tem API; será fornecida depois)

MVP — PRIMEIRA ENTREGA (escopo travado)

A primeira entrega é **somente** isto — construir, testar e validar antes de qualquer outra coisa:

- **Login** com perfis e permissões por função
- **Requisição** de compra
- **Aprovação por alçada**
- **Cadastro de fornecedores**
- **Cadastro de veículos**
- **Dashboard básico**
- **Exportação Excel**

Tudo o que vem depois — inteligência de preço, alerta de desperdício, checklist de aprovação, workflow de recusa, Módulo 1.5 (orçamento), Módulos 2 e 3, integração ADMServ — entra em **iterações seguintes**, sem reconstruir o que o MVP já fez.

Critério de aceite do MVP

O MVP só é considerado **entregue** quando o Bruno conseguir, sozinho e **sem intervenção do programador**:

1. Criar uma requisição;

2. Aprovar pelo celular;
3. Consultar no dashboard;
4. Exportar em Excel.

(A hospedagem inicial continua sendo do programador; “sem intervenção” refere-se à operação — uma vez no ar, o fluxo inteiro roda sem depender dele.)

Princípios de arquitetura (valem para tudo)

- O app é a **fonte de verdade**. O ADMServ recebe um espelho mais tarde.
- Banco de dados real (PostgreSQL ou similar) — nunca planilha como base.
- **Exportação limpa em planilha (CSV/Excel) desde o início** (compras, baixas, locações). É a ponte para o ADMServ e um seguro independente de qualquer integração.
- Schema desenhado de ponta a ponta: Requisição → Aprovação → Pedido → NF → Contas a Pagar → ADMServ .
- **Multiempresa por natureza**: o banco nasce com a hierarquia **Empresa → Filial → Centro de Custo** como entidades de primeira classe. Todo registro (requisição, compra, estoque, locação, orçamento, permissão) é amarrado a essa hierarquia. Adicionar empresa, filial ou centro de custo é **cadastro, nunca refatoração**. Orçamentos, permissões e relatórios respeitam e consolidam por esses níveis.
- **Responsivo / mobile-first**: aprovação, leitura de QR, foto, GPS e telas de campo precisam funcionar bem no celular (PWA).

CONTROLE DE ACESSO (USUÁRIOS E PERMISSÕES)

Camada **fundacional** — construir já no Módulo 1. Princípio: **menor privilégio**. Cada usuário só enxerga as telas e os dados da sua função; o que ele não tem permissão **nem aparece** na interface (não basta bloquear — some da tela). Permissões **editáveis por usuário**, como a matriz de alçada.

Login

- Usuário e **senha individual** por pessoa (proibido compartilhar).
- Troca de senha no primeiro acesso; reset pelo administrador.
- **Log de auditoria**: toda ação registra quem fez, o quê e quando.

Capacidades liberáveis/bloqueáveis

Ver preço de peça e alertas de preço · Ver relatórios/dashboard · Criar requisição · Aprovar (com alçada) · Operar estoque de consumo (2A) · Operar banheiros/ativos (2B) · Operar locação/entrega (M3) · Configurar (matriz, permissões, cadastros).

Funções e permissões padrão (Bruno ajusta)

	Preço		Cria		Estoque	Banheiros	Locação	
--	-------	--	------	--	---------	-----------	---------	--

Função	de peça	Relatórios	requisição	Aprova	2A	2B	M3	Config
Administrador (Bruno)	✓	✓	✓	✓ todas	✓	✓	✓	✓
Gestão (Nataly/Cristiano)	✓	✓	—	✓ sua alçada	—	—	—	—
Comprador	✓	—	✓	—	✓	—	—	—
Pátio/Operação	—	—	—	—	—	✓	✓	—
Motorista	—	—	—	—	—	—	✓ só suas entregas	—
Supervisor	—	—	—	✓ até R\$ 500	—	—	—	—

Destaques: o **Comprador** vê preço de peça (precisa pra negociar) mas **não** vê relatórios nem o lado dos banheiros. O **Pátio** opera banheiros e locação, mas **não** vê preço de peça nem relatórios. Relatórios e custos ficam restritos à Gestão e ao Administrador.

Segurança e acesso por localização

O sistema fica na web, mas o acesso aos dados sensíveis **não pode ser de qualquer lugar**. Regra:

- **Funções de escritório** (relatórios, compras, configuração, estoque do pátio): só de **dentro da empresa**, pela rede/IP da empresa.
- **Bruno e Nataly:** liberados de **qualquer lugar** (autorização remota).
- **Funções de campo** (entrega/locação no celular): funcionam de **qualquer lugar** por natureza — mas essas contas já não veem relatório nem preço (camada de permissão), então o acesso de fora é inofensivo.

Como implementar:

- A "trava por raio da empresa" se faz por **IP fixo** (lista de IPs permitidos da conexão da empresa) — confiável. **Pré-requisito:** contratar **IP fixo** com a operadora (adicional barato no plano empresarial). *(Trava por GPS é fraca/burlável; no máximo reforço.)*
- **HTTPS** obrigatório (conexão criptografada).
- **2FA** (código no celular) obrigatório nas contas que acessam de fora (Bruno e Nataly) — como elas furam a trava de IP, o 2º fator protege.
- **Sessão expira** por inatividade (logout automático).
- O IP permitido é campo configurável → não atrasa o Módulo 1; preenche quando a operadora liberar.

MÓDULO 1 — Compras (CONSTRUIR PRIMEIRO)

Fluxo

Requisição → verificações automáticas (histórico, preço, desperdício, estoque) → roteamento por alçada → aprovação/recusa no celular → registro.

Campos da requisição

Classificação (obrigatória):

- Empresa: Moderna Sany · ISANQUI · 1AMS LOG
- Centro de Custo: Frota · Operação · Comercial · Administrativo · Licitações · TI · Oficina
- Categoria: Peças · Pneus · Óleo · Ferramentas · Material de escritório · Uniformes · EPI · Serviços terceiros · Combustível · Outros

Dados da compra:

- **Solicitante** (quem pediu) — distinto do comprador; gera rastreabilidade (ex.: Solicitante: Danilo · Comprador: Camila)
- Comprador (quem efetua a compra)
- Peça / descrição
- Caminhão — **selecionado do Cadastro de Veículos** (pela placa); obrigatório quando Centro de Custo = Frota/Oficina ou categoria de veículo
- Fornecedor — **selecionado do Cadastro de Fornecedores** (não digitado livre)
- Preço solicitado
- Motivo da compra (obrigatório, texto livre)
- Urgência: Normal · Urgente · Emergencial
- Retirar do estoque? Sim / Não → integra com Módulo 2A
- **Anexos múltiplos** (foto da peça, orçamento 1, orçamento 2, nota técnica, garantia) — opcional, recomendado

Sistema: data/hora, status, aprovador, data da aprovação, nº da requisição.

Categoria, Centro de Custo e Caminhão são campos **distintos** (lição da planilha atual, onde a "Placa" virou categoria e só ~25% dos registros identificavam o veículo).

Aprovação por alçada (matriz CONFIGURÁVEL em tela)

Faixa	Aprovador
Até R\$ 500	Supervisor
R\$ 500 – R\$ 2.000	Cristiano
Acima de R\$ 2.000	Bruno
Acima de R\$ 10.000	Bruno + Nataly (dupla aprovação)

Valores e responsáveis editáveis sem reprogramar. Perfis: Comprador (cria) / Supervisor / Cristiano / Bruno / Nataly.

Conferência de preço (prioridade)

1. **Última compra registrada** — alerta o % de desvio. Ex.: “Última R\$ 1.280 · Atual R\$ 2.100 · 64% acima.” Principal, sempre confiável.
2. **Preço da Hora (SEFAZ-BA)** — referência fiscal regional (NF-e/NFC-e). Sem API oficial conhecida → automação de navegador (frágil) ou consulta manual; validar viabilidade. Cobertura irregular para peça pesada.
3. **Busca web** — só reserva, item sem histórico. Sem link de compra.

Alerta de desperdício

Ao registrar, cruza com o histórico: última compra da mesma peça/veículo (“comprada há 32 dias para este mesmo veículo”), frequência, garantia. **KM rodado**: campo “KM atual” no cadastro do veículo, atualizado manualmente até o ZUQ liberar dados.

Checklist na aprovação (painel de decisão)

Antes de aprovar (principalmente Cristiano), a tela mostra um checklist **pré-preenchido** pelo sistema com os dados que ele já calcula; o aprovador precisa confirmar — evita aprovação automática/carimbada:

- Existe em estoque? (do Módulo 2A)
- Está na garantia? (da última compra)
- Última compra analisada (valor e % de desvio)
- Valor compatível?
- Dentro do orçamento do centro de custo? (do Módulo 1.5)

Workflow de recusa

Recusar exige **motivo obrigatório** (em estoque · valor acima do mercado · sem justificativa · compra duplicada · fora do orçamento · outro). A requisição recusada **volta ao solicitante com o motivo** — não morre. Os motivos viram diagnóstico no dashboard (ex.: “38% das recusas foram por já haver material em estoque” → sinaliza problema de visibilidade de estoque).

Dashboard gerencial

Mês atual: Total comprado · Aprovado · Recusado · **% de compras emergenciais** (indicador de falha de planejamento/estoque) · Compras por veículo. Inclui o **Ranking de fornecedores** e atalho pro **Histórico por veículo**. Filtros por Empresa, Centro de Custo, Categoria.

MÓDULO 1.5 — CONTROLE ORÇAMENTÁRIO

Fecha o ciclo: deixa de ser só “onde gastei” e passa a “quanto eu podia gastar e quanto estourei”.

- **Orçamento mensal por centro de custo** (ex.: Frota = R\$ 50.000/mês).

- **Orçamento anual por empresa.**
- **Consumo medido na aprovação** (compromisso), não só no pagamento — assim o alerta dispara *antes* de estourar, não depois.
- **Alertas:** 80% → amarelo · 100% → vermelho · 120% → **aprovação especial** (trava e exige liberação do Bruno).
- **Dashboard de desvio orçamentário:** orçado × consumido × saldo, por centro de custo e por empresa.

A “aprovação especial” por estouro é um **segundo gatilho** de aprovação, somado à alçada por valor: uma compra pode caber na alçada do Cristiano, mas se estourar o orçamento da Frota, sobe pro Bruno.

CADASTROS DE APOIO

Cadastro de veículos

A frota vira tabela própria; a requisição referencia o veículo (não digita placa livre):

- Placa · Modelo · Ano · KM atual · Centro de custo · Status (Operacional · Oficina · Vendido · Inativo)
- Habilita: histórico por veículo, **custo por km**, garantias e controle de manutenção — sem reestruturar depois.
- O **KM atual** alimenta o custo/km e o alerta de desperdício; atualizado manualmente até o ZUQ liberar o hodômetro.

Cadastro de fornecedores

Fornecedor deixa de ser texto livre e vira cadastro: Razão social · CNPJ · Telefone · Contato · Categoria. Padroniza o dado e habilita o ranking.

Ranking de fornecedores

Dashboard: Fornecedor | Nº de compras | Valor total | Último pedido. Revela concentração (ex.: “42% das compras de peças num só fornecedor”).

Histórico por veículo

Visão por placa, montada sobre os dados que o sistema **já captura** (placa + categoria + valor) — sem digitação extra:

- Histórico de compras e de manutenção
- Custo acumulado, com quebra por categoria
- Ex.: **Ford Cargo 2429 — QMA6F40 · 2026:** Peças R\$ 18.000 · Pneus R\$ 12.000 · Óleo R\$ 3.000 · **Total R\$ 33.000**

Mostra **quais veículos consomem caixa** — insumo direto pra decisão de manter, reformar ou vender.

MÓDULO 2 — Estoque QR

2A — Estoque de consumo (peças, óleo, EPI, ferramentas)

- Controla **quantidade / saldo**. Entrada soma; saída (baixa por leitura) subtrai.
- Alimenta o alerta “tem em estoque” e o roteamento do “Retirar do estoque?” do Módulo 1.

2B — Ativos: banheiros químicos (modelo de composição)

Princípio: o modelo (Simples/Luxo/Super Luxo) **nunca é digitado** — é calculado pelas peças instaladas na cabine, registradas por leitura de QR. Nunca diverge da realidade.

- **Cabine** — ativo com QR único (banheiro nº 1, 2, 3...).
- **Componentes serializados** (QR + número próprio): Pia · Caixa de dejetos (tipo: `visualiza dejetos | com descarga`) · Porta papel toalha · Porta sabonete.
- **Montagem:** bipa a cabine + bipa as peças instaladas → grava a composição e calcula o modelo.
Conversão: troca física + re-bip → recalcula sozinho.

Receitas (regras):

Modelo	Caixa de dejetos	Pia	Papel toalha	Sabonete
Simples	Visualiza dejetos	—	—	—
Luxo	Visualiza dejetos	✓	✓	✓
Super Luxo	Com descarga	✓	✓	✓

Configuração que não casa com nenhuma receita = “**indefinida/incompleta**” + alerta. Nunca atribui modelo por aproximação.

Rastreabilidade por item: histórico de cada peça (em qual cabine, desde quando, manutenções); consultas tipo “quantos Super Luxo montados agora?”, “onde está a pia nº X?”.

MÓDULO 3 — Logística / Locação

Entrega de banheiros via **celular do motorista** (lê QR pela câmera).

QR da ordem de locação (estampado pelo programador no documento) carrega: cliente, número da ordem, quantidade e tipo de banheiro. Uma ordem cobre **várias unidades**.

Fluxo de entrega:

1. Bipa o QR da ordem → app preenche cliente / nº / quantidade / tipo.
2. Bipa cada cabine → vincula as unidades específicas à ordem **e confere tipo e quantidade**. Se o motorista bipar a quantidade errada ou um tipo diferente do pedido, o sistema **barra e alerta** (sabe o tipo de cada cabine pela composição do 2B).
3. Foto + GPS → comprovante de entrega anexado à ordem.
4. (com integração) espelha no ADMServ.

Devolução/retirada: bipa a cabine (e/ou a ordem) → registra devolução → unidade volta a “disponível”.

Status da unidade (disponível / locado / manutenção) é derivado desses eventos → mostra a **ocupação real da frota de banheiros**.

Offline: como o QR carrega o payload completo, a leitura funciona sem sinal no cliente; o número da ordem é a chave para sincronizar depois.

A **captura funciona no app desde já**, sem ADMServ. A sincronização com o ADMServ é a camada de integração (adiada).

INTEGRAÇÃO ADMServ (ADIADA — entra como atualização)

- O ADMServ **tem API**; será fornecida pelo dono do sistema depois.
 - Até lá: o app é a fonte de verdade e **exporta planilha limpa** dos lançamentos (compras aprovadas, baixas, locações).
 - **A confirmar com o fornecedor:** formato/endpoints da API e se há import/export por planilha.
 - Quando a API chegar, a integração pluga sem reconstruir nada (schema e exportação já preparados).
-

HARDWARE

- **Motoristas / campo:** celular comum (lê QR pela câmera; registra foto + GPS).
- **Base — banheiros:** um celular dedicado.
- **Base — peças/estoque/inventário: scanner sem fio (Bluetooth)** pareado ao celular/tablet da base (sem cabo, sem USB).
- O app roda como **PWA** no próprio celular. Geração e impressão de etiquetas QR no sistema.

IDENTIDADE VISUAL

Padrão visual: identidade da **A Moderna Sany**.

Paleta:

- Verde institucional #2E7D32 — botões de ação e destaques
- Verde apoio #4CAF50 — realces secundários / estados
- Grafite #263238 — menu lateral e textos fortes
- Branco #FFFFFF — fundo

Diretrizes:

- Interface moderna e corporativa, no estilo dos SaaS atuais (Monday, ClickUp, Omie).
- **Menu lateral** em grafite; **botões de ação** em verde institucional.

- Dashboard com **cartões limpos** e indicadores visuais.
- Excelente leitura em **celular e desktop**.
- **Logomarca oficial** da A Moderna Sany na tela de login e no cabeçalho.
- Os alertas (amarelo 80% / vermelho 100% do orçamento) devem harmonizar com a paleta.

Nota prática: fornecer o arquivo da logomarca oficial dentro do projeto, para o Claude Code usá-la no login e no cabeçalho.

STACK (recomendação — ajustável)

React (front) + backend leve (Node) + PostgreSQL. Login com perfis por alçada. Alerta de aprovação no celular via WhatsApp (API oficial/serviço) ou push (PWA). **Backup automático** do PostgreSQL: diário, com retenção mínima de 30 dias.

SEMENTES DE DADOS

- Planilha “Controle de Compras” (jan–abr/2026, 399 registros) → histórico de compra/pagamento dos alertas.
- ZUQ (despesa por carro) → vínculo peça↔caminhão e KM. *Verificar export CSV/API.*

ORDEM DE CONSTRUÇÃO

1. **MVP** (login, requisição, aprovação por alçada, fornecedores, veículos, dashboard básico, exportação Excel) — construir, **testar** e validar primeiro.
2. **Resto do Módulo 1** (inteligência de preço, alerta de desperdício, checklist de aprovação, workflow de recusa) + **Módulo 1.5** (controle orçamentário).
3. **Módulo 2** (2A consumo + 2B banheiros).
4. **Módulo 3** (logística/locação — captura).
5. **Integração ADMServ** — quando a API for fornecida.