



# V8.4 Manual

## MANUAL OPERACIONAL — SCHLOSSER PRO (sercarne.com)

### VERSÃO \*V8.4\* (OFICIAL • PRODUÇÃO)

Plataforma: <https://sercarne.com>

Status: ATIVA / PRODUÇÃO

Base (fonte da verdade): Google Sheets

Planilha oficial: Precificação Mix Frigo Schlosser / Zaleski 2026 (IA APP)

ID: 12wPGal\_n7PKYFGz9W\_\_bXgK4mly2NbrEEGwTrIDCzcl

Este documento é a CONSTITUIÇÃO DO SISTEMA.

O sistema executa este manual.

O manual não se adapta ao sistema.

---

### CHANGELOG (VERSÕES)

#### V8.2

Base funcional validada.

#### V8.3

- Galeria de imagens por produto
- Overlay de marca no card
- Padronização oficial das colunas AE/AF/BE/BF/BG/BH/AG/AH/AI

#### V8.4 (ATUAL)

- Integração oficial ENTRADAS\_ESTOQUE (Sheets) → entradas\_estoque (Supabase)
    - Formalização do fluxo de importação via Apps Script
    - Padronização do campo qtd\_und
    - Regra oficial de UPSERT (codigo + data\_entrada)
    - Documentação completa do cálculo de estoque por data
- 

## CAPÍTULO 0 — PRINCÍPIO FUNDAMENTAL (INALTERÁVEL)

O Schlosser PRO existe para servir o Modelo de Negócio Schlosser.

Ferramentas são descartáveis.

Regras de negócio não são.

Se houver divergência entre sistema e este manual:

 é erro de implementação

 não é melhoria

 deve ser corrigido

---

## CAPÍTULO 1 — OBJETIVO

Definir de forma inequívoca:

- Catálogo
- Visibilidade
- Precificação
- Regras de volume
- Estoque por data
- Entradas futuras
- Rotas e cutoff
- Pedidos e status
- Permissões
- Piso e ajustes
- Imagens e marca
- Governança

Permitir:

- Troca de desenvolvedor
  - Troca de IA
  - Escalabilidade
  - Zero regressão
- 

## CAPÍTULO 2 — FONTES DA VERDADE (REGRA ABSOLUTA)

### 2.1 Google Sheets

É a única fonte da verdade para:

- Produtos
- Preços
- Peso médio
- Visibilidade
- Imagens
- Marca
- Rotas

### 2.2 Supabase

É banco de dados operacional para:

- Login
- Pedidos
- Entradas futuras
- Cálculo de estoque

Supabase NÃO define regra de negócio.

### 2.3 Princípios imutáveis

- UND ≠ KG
  - Preço sempre R\$/KG
  - Peso é estimativo
  - Cliente nunca vê tabela
  - Carrinho não compromete estoque
  - Pedido é o primeiro compromisso real
- 

## CAPÍTULO 3 — ARQUITETURA OFICIAL

### 3.1 Frontend

React + Vite + Tailwind

ProductCard

Carrinho com UND / peso estimado / subtotal

### 3.2 Leitura de dados

Google Sheets via GVIZ/Query

Cache leve (performance)

### 3.3 Supabase

- Auth
  - pedidos
  - entradas\_estoque
- 

## CAPÍTULO 4 — ABA OFICIAL DO CATÁLOGO

Aba obrigatória:

 2026 Base Catalogo Precifica V2

Toda leitura deve vir dela.

---

## CAPÍTULO 5 — VISIBILIDADE

Coluna:

AX → EXIBIR NA PLATAFORMA

TRUE → aparece

FALSE → não aparece

Nenhuma lógica pode ignorar AX.

---

## CAPÍTULO 6 — ORDEM DE EXIBIÇÃO

Ordem padrão:

Maior estoque disponível → menor

Objetivo:

- Vender o que tem
  - Evitar ruptura
- 

## CAPÍTULO 7 — LÓGICA DE PREÇO (IMUTÁVEL)

Fórmula única:

VALOR ESTIMADO =

UND × PESO MÉDIO × PREÇO (R\$/KG)

Preço aplicado conforme:

- Volume no carrinho
- Perfil (logado/público)

Nunca recalcular fora dessa lógica.

---

## CAPÍTULO 8 — TABELAS E VOLUME

Exemplo modelo atual:

1 UND → TAB1

2–9 UND → TAB0

≥10 UND → TAB4

Cliente vê:

- preço final R\$/KG
- badge percentual

Nunca mostrar:

- nome da tabela

- desconto em R\$
- regra interna

Piso absoluto: TAB5

---

## CAPÍTULO 9 — ROTAS E CUTOFF

Fonte: Aba Rotas Dias De Entrega

Cidade → define rota

Rota → define dias

Rota → define cutoff

Usuário não pode forçar data inválida.

---

## CAPÍTULO 10 — ESTOQUE POR DATA (V8.4 OFICIAL)

Fórmula oficial:

Disponível(D) =

Estoque\_Base

- Entradas( $\leq D$ )

- Pedidos\_Comprometidos( $\leq D$ )

---

### 10.1 Estoque Base

Vem do Sheets

Coluna oficial do catálogo

---

### 10.2 Entradas Futuras

Vêm do Supabase

Tabela: entradas\_estoque

Origem operacional:

Sheets → ENTRADAS\_ESTOQUE

---

### 10.3 Pedidos Comprometidos

Comprometem:

- PEDIDO ENVIADO
- PEDIDO CONFIRMADO
- SEU PEDIDO SAIU PARA ENTREGA

Não comprometem:

- CANCELADO
- PEDIDO ENTREGUE

Carrinho não compromete.

---

## CAPÍTULO 11 — ENTRADAS FUTURAS (V8.4)

11.1 Aba oficial no Sheets

### ENTRADAS\_ESTOQUE

Estrutura obrigatória:

A → data\_entrada

B → codigo

C → qtd\_und

D → obs

Colunas E e F são visuais.

---

11.2 Processo oficial

Sheets é operacional.

Supabase é cálculo.

Importação via Apps Script:

Função:

```
importEntradasEstoqueToSupabase()
```

---

11.3 Script Properties obrigatórias

SUPABASE\_URL

SUPABASE\_SERVICE\_ROLE\_KEY

SUPABASE\_TABLE = entradas\_estoque

---

## 11.4 Estrutura Supabase

Tabela: entradas\_estoque

Campos mínimos:

id (uuid)  
codigo (text)  
data\_entrada (date)  
qtd\_und (numeric)  
obs (text)  
created\_at (timestamp)

Campo oficial usado:

qtd\_und

---

## 11.5 Regra anti-duplicação

UPSERT por:

(codigo, data\_entrada)

Se existir → atualiza

Se não → cria

Nunca duplicar.

---

## CAPÍTULO 12 — PEDIDOS E STATUS

Status oficiais:

PEDIDO ENVIADO  
PEDIDO CONFIRMADO  
SEU PEDIDO SAIU PARA ENTREGA  
PEDIDO ENTREGUE  
CANCELADO

Permissões:

Níveis 1–5:

- cria

- cancela apenas se ENVIADO

Níveis 6–10:

- confirma
  - avança status
  - cancela com motivo
- 

## CAPÍTULO 13 — PISO E AJUSTES

Piso absoluto: TAB5

Nunca vender abaixo.

Ajustes exigem:

- motivo
- usuário
- data/hora

Cliente nunca vê regra interna.

---

## CAPÍTULO 14 — IMAGENS E MARCA (OFICIAL)

Aba: 2026 Base Catalogo Precifica V2

Colunas oficiais:

AE → 1<sup>a</sup> foto limpa  
AF → 1<sup>a</sup> foto bruta

BE → 2<sup>a</sup> foto limpa  
BF → 2<sup>a</sup> foto bruta

BG → 3<sup>a</sup> foto limpa  
BH → 3<sup>a</sup> foto bruta

AG → marca limpa  
AH → marca bruta  
AI → código + nome marca

Regra:

Prioridade sempre link limpo.

Bolinhas aparecem apenas se 2+ fotos.

Marca é apenas visual.

---

## CAPÍTULO 15 — WHATSAPP (CIÊNCIA)

Pedido gera WhatsApp automático.

WhatsApp:

- ✗ não confirma
- ✗ não altera status
- ✗ não escreve no banco

É apenas ciência.

---

## CAPÍTULO 16 — GOVERNANÇA (ANTI-REGRESSÃO)

O sistema executa o manual.

Durante estabilidade:

- ✗ não refatorar o que funciona
- ✗ não otimizar regra validada
- ✗ não alterar estrutura de colunas

Mudança só com:

- validação
  - changelog
  - nova versão
- 

## CAPÍTULO 17 — MAPA DAS ABAS OFICIAIS

2026 Base Catalogo Precifica V2 → catálogo

Relacao Clientes Sysmo → clientes

Rotas Dias De Entrega → logística

ENTRADAS\_ESTOQUE → entradas futuras

Legenda/Objetivos → arquitetura

---

## DECLARAÇÃO FINAL

Este documento define integralmente a operação do Schlosser PRO V8.4.

Qualquer divergência entre sistema e este manual é erro de implementação.

O manual é soberano.

---

👉 V8.3  MANUAL OPERACIONAL  
— SCHLOSSER PRO sercarne



# MANUAL OPERACIONAL — SCHLOSSER PRO (sercarne.com)

## VERSÃO V8.3 (OFICIAL • PRODUÇÃO)

Plataforma: <https://sercarne.com>

Status: ATIVA / PRODUÇÃO

Base (fonte da verdade): Google Sheets

Planilha oficial: Precificação Mix Frigo Schlosser / Zaleski 2026 (IA APP)

ID: 12wPGal\_n7PKYFGz9W\_\_bXgK4mly2NbrEEGwTrIDCzcl

Formato oficial do manual: Google Docs (este texto)

Este documento é a CONSTITUIÇÃO do sistema.

---

## CHANGELOG (O QUE MUDOU DA V8.2 → V8.3)

### V8.3 — Atualizações validadas

1. Galeria de imagens por produto
    - 1ª foto: vem de **AE (limpa export)**, fallback **AF (bruta)**
    - 2ª foto: vem de **BE (limpa export)**, fallback **BF (bruta)**
    - 3ª foto: vem de **BG (limpa export)**, fallback **BH (bruta)**
    - **Bolinhas da galeria:** só aparecem quando existir 2+ fotos (regra visual)
  2. Marca do produto dentro do card (overlay)
    - Marca vem de **AG (limpa export)**, fallback **AH (bruta)**
    - Nome/código marca: **AI**
  3. Padronização oficial (memória) das colunas de imagens/marca na aba principal.
- 

## CAPÍTULO 0 — PRINCÍPIO FUNDAMENTAL (INALTERÁVEL)

O sistema existe para servir o **Modelo de Negócio Schlosser PRO**.

Ferramentas, código, IA, frameworks e plataformas são **descartáveis**.

As **regras de negócio não são**.

Qualquer divergência entre sistema e este manual:

 é erro de implementação

 não é “melhoria”  
 deve ser corrigida

---

# CAPÍTULO 1 — OBJETIVO DESTE DOCUMENTO

Definir de forma clara, técnica e inequívoca:

- Catálogo e visibilidade
- Precificação (por volume e perfil)
- Rotas, datas e cutoff
- Estoque por data
- Pedido, status e cancelamento
- Permissões
- Governança e anti-regressão
- **Imagens e marca (V8.3)**

Este documento existe para:

- Evitar interpretação
  - Permitir troca de desenvolvedor/IA
  - Proteger o que já funciona (anti-regressão)
- 

# CAPÍTULO 2 — FONTES DA VERDADE (REGRA ABSOLUTA)

1. **Google Sheets é a única fonte da verdade** (produtos, preços, imagens, rotas)
  2. **UND ≠ KG**
  3. **Preço sempre em R\$/KG**
  4. **Peso é estimativo** (peso médio vindo do Sheets)
  5. **Cliente nunca vê tabela** (TAB0/TAB4 etc)
  6. **Não existe “reserva” como compromisso** (carrinho não compromete)
  7. **Pedido é o primeiro compromisso do sistema**
  8. Supabase **não define regra**, apenas registra/consulta
  9. Código apenas executa fielmente o que está no Sheets + regras documentadas aqui
-

# CAPÍTULO 3 — ARQUITETURA OFICIAL (CAMADAS DO SISTEMA)

## 3.1 Frontend

- React + Vite + Tailwind
- Cards do catálogo (ProductCard)
- Carrinho e métricas (UND / peso / subtotal)

## 3.2 “API” do projeto (consulta)

- Leitura de Google Sheets via GVIZ/Query (schlosserApi.js)
- Cache local (localStorage) por poucos minutos para performance

## 3.3 Supabase (banco)

- Autenticação (login vendedor)
  - Tabelas principais: pedidos, entradas\_estoque (dependendo do desenho atual do estoque)
  - **Supabase não manda em regra de preço/estoque/visibilidade** — só guarda
- 

# CAPÍTULO 4 — ABA OFICIAL DO CATÁLOGO E PRECIFICAÇÃO (SHEETS)

A aba principal é **obrigatória** e é a referência padrão:

Aba principal: “**2026 Base Catalogo Precifica V2**”

Qualquer consulta de produto deve ler dessa aba.

---

# CAPÍTULO 5 — CATÁLOGO (VISIBILIDADE)

## 5.1 Regra definitiva

- ♦ AX — EXIBIR NA PLATAFORMA (visível)

- TRUE / VERDADEIRO → aparece
  - FALSE / FALSO → não aparece
- 📌 Nenhuma lógica pode ignorar AX.
- 

## CAPÍTULO 6 — ORDEM DE EXIBIÇÃO NO CATÁLOGO

Produtos são exibidos, por padrão:

➡ do maior para o menor estoque disponível (considerando data atual / lógica do estoque)

Objetivo:

- vender o que tem
  - evitar ruptura
  - facilitar a vida do vendedor
- 

## CAPÍTULO 7 — LÓGICA DE PREÇO (IMUTÁVEL)

FÓRMULA ÚNICA:

✓ VALOR ESTIMADO = UND × PESO MÉDIO × PREÇO (R\$/KG)

- UND → carrinho
- Peso médio → Sheets
- Preço → tabela aplicada automaticamente (schlosserRules)

✗ Nenhuma tela pode recalcular fora dessa lógica.

---

## CAPÍTULO 8 — TABELAS E REGRA DE VOLUME

A tabela aplicada depende do **total UND no carrinho + perfil** (logado/público).

Regra padrão de volume (exemplo do modelo atual):

- 1 UND → TAB1
- 2–9 UND → TAB0
- $\geq 10$  UND → TAB4

Exibição ao cliente:

- **Preço “público referência”**: TAB3
- Cliente vê apenas:
  - preço R\$/KG
  - badge de benefício (% abaixo do público), quando logado
  - ✗ Nunca mostrar nome da tabela ao cliente
  - ✗ Nunca mostrar desconto em R\$

---

## CAPÍTULO 9 — ROTAS, DIAS E CUTOFF

Fonte da verdade: Sheets

Aba: **Rotas Dias De Entrega**

Regras:

- Cidade define rota/grupo
- Rota define dias e cutoff
- Cutoff é o horário limite (normalmente no dia anterior)
- Usuário não pode forçar data inválida

---

## CAPÍTULO 10 — ESTOQUE POR DATA (REGRA V8+)

Fórmula oficial (cumulativa):

✓  $\text{Disponível}(D) = \text{Estoque\_Base\_Hoje} + \text{Entradas}(\leq D) - \text{Pedidos\_Comprometidos}(\leq D)$

Comprometem estoque:

- PEDIDO ENVIADO
- PEDIDO CONFIRMADO
- SEU PEDIDO SAIU PARA ENTREGA

NÃO comprometem:

- CANCELADO
- PEDIDO ENTREGUE (histórico)

📌 Carrinho não compromete estoque.  
📌 Validação ocorre ao adicionar/enviar conforme implementação (mas **nunca** vira “reserva”).

“Na V8.3 em produção, o cálculo de estoque por data utiliza as tabelas entradas\_estoque e pedidos no Supabase. O Sheets segue como fonte da verdade do catálogo/preços/imagens/rotas. A sincronização de entradas do Sheets para o Supabase pode ser manual ou automatizada (fase 2).”

---

## CAPÍTULO 11 — PEDIDOS E STATUS (FLUXO OFICIAL)

Status oficiais:

1. PEDIDO ENVIADO
2. PEDIDO CONFIRMADO
3. SEU PEDIDO SAIU PARA ENTREGA
4. PEDIDO ENTREGUE
5. CANCELADO

Permissões (resumo):

- Níveis 1–5: cria pedido; edita/cancela só se estiver **PEDIDO ENVIADO**
  - Níveis 6–10: confirma; avança status; cancela em qualquer status (com motivo)
- 

## CAPÍTULO 12 — PISO E AJUSTES (REGRA DE PROTEÇÃO)

Piso absoluto: **TAB5**

Nunca vender abaixo do piso.

Ajustes exigem:

- motivo
- usuário

- data/hora

Cliente nunca vê:

- nome de tabela
  - piso
  - regra de ajuste
  - log interno
- 

## CAPÍTULO 13 — IMAGENS E MARCA (NOVO V8.3 • OFICIAL)

Este capítulo define **exatamente** de onde vem cada imagem no catálogo e como o site exibe.

### 13.1 Fonte das imagens (Google Sheets)

Aba: 2026 Base Catalogo Precifica V2

#### PADRÃO OFICIAL (MEMÓRIA / COLUNAS)

- **AE (Coluna 31)** = fórmula limpa 1ª foto principal (export)
- **AF (Coluna 32)** = URL bruto 1ª foto principal
- **BE (Coluna 57)** = fórmula limpa 2ª foto (export)
- **BF (Coluna 58)** = URL bruto 2ª foto
- **BG (Coluna 59)** = fórmula limpa 3ª foto (export)
- **BH (Coluna 60)** = URL bruto 3ª foto
- **AG (Coluna 33)** = fórmula limpa marca (export)
- **AH (Coluna 34)** = URL bruto marca
- **AI (Coluna 35)** = Código + Nome da marca (ex.: “5010 BAH BEEF”)

 Regra operacional:

- Você cola o link bruto no (AF/BF/BH/AH)
- As fórmulas geram os links “limpos export” (AE/BE/BG/AG)
- O site **prioriza sempre os links limpos** (export)

### 13.2 Como o site lê e monta a galeria

O site busca:

- Foto 1: **AE** (fallback AF se necessário)
- Foto 2: **BE**

- Foto 3: **BG**

E cria `product.images = [foto1, foto2, foto3]` (somente as existentes).

✓ Regra visual:

- Se `product.images.length < 2` → NÃO mostrar bolinhas
- Se `product.images.length >= 2` → mostrar bolinhas e permitir trocar

### 13.3 Marca no canto do card

O site busca:

- Marca: **AG** (fallback AH)
- Exibe como overlay (canto superior esquerdo) no card do produto

📌 A marca é apenas visual:

- não altera preço
- não altera estoque
- não altera regra

---

## CAPÍTULO 14 — WHATSAPP (CIÊNCIA) — REGRA OBRIGATÓRIA

Regra absoluta:

Todo pedido gera WhatsApp automático de ciência.

WhatsApp:

- ✗ não confirma
- ✗ não altera pedido
- ✗ não muda status
- ✗ não escreve no banco

📌 É ciência, não validação.

---

## CAPÍTULO 15 — GOVERNANÇA (ANTI-REGRESSÃO)

## Regra de ouro

O SISTEMA EXECUTA O MANUAL.

O MANUAL NÃO SE ADAPTA AO SISTEMA.

Durante modo estabilidade:

-  não refatorar o que funciona
  -  não “otimizar” comportamento validado
  -  apenas correção de bug/diferença comprovada
- 

## ANEXO A — ABAS OFICIAIS DA PLANILHA (MAPA RÁPIDO)

- **2026 Base Catalogo Precifica V2** → produtos, preços, peso, visibilidade, imagens, marca
  - **Relacao Clientes Sysmo** → clientes (consulta)
  - **Rotas Dias De Entrega** → cidades / rotas / cutoff
  - **Cidades** → fallback de cidades (se necessário)
  - **Legenda / Objetivos sercarne.com** → arquitetura lógica (faz parte do sistema)
- 

## DECLARAÇÃO FINAL

Este documento define integralmente a operação da plataforma **Schlosser PRO V8.3**. Qualquer divergência entre sistema e este manual é **erro de implementação**.

**---Ideias melhoria---**

Ideias:

1. Pensando nas parcerias como Vandal/Ray, deveria ter um campo consulta CNPJ para ver se o cliente já é atendido por outro vendedor ou não.
2. Na tela catálogo ter um filtro se o cliente quiser ordenar os itens por ordem de descrição, por preço, por estoque (de menor para maior ou do contra).
- 3.



# V8.2 MANUAL OPERACIONAL



# MANUAL OPERACIONAL

## SCHLOSSER PRO — VERSÃO V8.2 (OFICIAL • PRODUÇÃO)

**Plataforma:** <https://sercarne.com>

**Versão:** V8

**Status:** ATIVA / PRODUÇÃO

**Formato oficial:** Google Docs / GitHub

**Este documento é a CONSTITUIÇÃO do sistema**

---

## CAPÍTULO 0 — PRINCÍPIO FUNDAMENTAL (INALTERÁVEL)

O sistema existe para servir o **Modelo de Negócio Schlosser PRO**.

Ferramentas, código, IA, frameworks e plataformas são descartáveis.

**As regras de negócio não são.**

Qualquer divergência entre sistema e este manual:

- é erro de implementação
  - não é melhoria
  - deve ser corrigida
- 

## CAPÍTULO 1 — OBJETIVO DESTE DOCUMENTO

Definir de forma clara, técnica e inequívoca:

- Catálogo e visibilidade
- Precificação
- Rotas, datas e cutoff
- Estoque por data
- Pedido, status e cancelamento
- Permissões
- Governança e anti-regressão

Este documento existe para:

- Evitar interpretação
  - Permitir troca de desenvolvedor/IA
  - Proteger o que já funciona
- 

## CAPÍTULO 2 — FONTES DA VERDADE (REGRA ABSOLUTA)

1. **Google Sheets é a única fonte da verdade**
  2. UND ≠ KG
  3. Preço sempre em R\$/KG
  4. Cliente nunca vê tabela
  5. Peso é sempre estimativo
  6. **NÃO EXISTE RESERVA**
  7. **Pedido é o primeiro compromisso**
  8. Supabase não define regra
  9. Código apenas executa o Sheets
- 

## CAPÍTULO 3 — ARQUITETURA OFICIAL

A arquitetura lógica está documentada na planilha:

### Legenda / Objetivos sercarne.com

Essa aba define oficialmente:

- Onde o dado nasce
  - Onde é consumido
  - Responsabilidades
-  **Ela faz parte do sistema.**
- 

## CAPÍTULO 4 — CATÁLOGO (VISIBILIDADE)

### REGRA DEFINITIVA

#### ♦ COLUNA AX — EXIBIR NA PLATAFORMA

- TRUE → aparece
- FALSE → não aparece

 Nenhuma lógica pode ignorar essa coluna.

---

## CAPÍTULO 5 — ORDEM DE EXIBIÇÃO

Produtos são exibidos:

➡ do maior para o menor estoque disponível (data atual)

Objetivo:

- Vender o que tem
  - Evitar ruptura
  - Ajudar o vendedor
- 

## CAPÍTULO 6 — LÓGICA DE PREÇO (IMUTÁVEL)

### FÓRMULA ÚNICA

$$\text{VALOR ESTIMADO} = \text{UND} \times \text{PESO MÉDIO} \times \text{PREÇO R$/KG}$$

- UND → carrinho
- Peso médio → Sheets
- Preço → tabela aplicada automaticamente

✗ Nenhuma tela pode recalcular fora dessa lógica.

---

## CAPÍTULO 7 — TABELAS E VOLUME

UND total no carrinho → tabela interna

### UND Tabela

1 TAB1

2–9 TAB0

≥10 TAB4

Exibição ao cliente:

- Preço base: TAB3
  - Cliente vê apenas:
    - R\$/KG
    - Benefício (%)
- ✗ Nunca vê tabela ou desconto em R\$

---

## CAPÍTULO 8 — ROTAS, DIAS E CUTOFF

Fonte da verdade:

- Sheets → Aba **Rotas Dias De Entrega**

Regras:

- Cidade define rota
  - Rota define dias e cutoff
  - Cutoff sempre no **dia anterior**
  - Usuário não pode forçar data inválida
- 

## CAPÍTULO 9 — ESTOQUE POR DATA (REGRA V8)

### FÓRMULA OFICIAL

Disponível(D) =  
Estoque\_Base\_Hoje  
+ Entradas( $\leq D$ )  
- Pedidos\_Comprometidos( $\leq D$ )

### Comprometem estoque:

- PEDIDO ENVIADO
- PEDIDO CONFIRMADO
- SEU PEDIDO SAIU PARA ENTREGA

### NÃO comprometem:

- CANCELADO
- PEDIDO ENTREGUE (histórico)

💡 Carrinho **não compromete** estoque.  
💡 Validação ocorre **no envio do pedido**.

---

## CAPÍTULO 10 — PEDIDOS E STATUS

### Status oficiais:

1. PEDIDO ENVIADO
2. PEDIDO CONFIRMADO
3. SEU PEDIDO SAIU PARA ENTREGA
4. PEDIDO ENTREGUE
5. CANCELADO

## Permissões

### Níveis 1–5

- Criar pedido
- Editar/cancelar apenas em PEDIDO ENVIADO

### Níveis 6–10

- Confirmar
  - Avançar status
  - Cancelar em qualquer status (com motivo)
- 

## CAPÍTULO 11 — AJUSTE DE PREÇO E PISO

- Piso absoluto: **TAB5**
  - Nunca vender abaixo do piso
  - Ajustes exigem:
    - motivo
    - usuário
    - data/horaCliente nunca vê ajuste.
- 

## CAPÍTULO 12 — MODO CONGELAMENTO (GOVERNANÇA)

### REGRA DE OURO

**O SISTEMA EXECUTA O MANUAL.  
O MANUAL NÃO SE ADAPTA AO SISTEMA.**

Durante o congelamento:

- Não refatorar
- Não “otimizar”
- Não reinterpretar regras
- Apenas correção de divergência

---

## CAPÍTULO 13 — WHATSAPP OBRIGATÓRIO (V8.3)

### Regra absoluta

Todo pedido gera WhatsApp automático de ciência.

WhatsApp:

-  não confirma
-  não altera pedido
-  não muda status
-  não escreve no banco

 É ciência, não validação.

---

## DECLARAÇÃO FINAL

Este documento define integralmente a operação do  
**Schlosser PRO V8.2.**

Qualquer divergência entre sistema e este manual  
**é erro de implementação.**



V8



MANUAL OPERACIONAL



# MANUAL OPERACIONAL

## SCHLOSSER PRO — VERSÃO V8 (OFICIAL • FINAL)

**Plataforma:** <https://sercarne.com>

**Versão:** V8 — Primeira versão operacional completa

**Status:** ATIVA / PRODUÇÃO

**Formato oficial:** Google Docs

**Este documento é a CONSTITUIÇÃO do sistema**

---

## CAPÍTULO 0 — PRINCÍPIO FUNDAMENTAL (INALTERÁVEL)

**O sistema existe para servir o Modelo de Negócio Schlosser PRO.**

Ferramentas, código, IAs e plataformas são descartáveis.

As regras de negócio **não são**.

Qualquer divergência entre sistema e este manual:

- **X** é erro de implementação
  - **X** não é melhoria
  - **X** deve ser corrigida
- 

## CAPÍTULO 1 — OBJETIVO DESTE DOCUMENTO

Definir **de forma clara, técnica e inequívoca**:

- Catálogo de produtos
- Regras de visibilidade e ordenação
- Precificação
- Rotas, dias de entrega e cutoff
- Estoque disponível por data
- Pedido, status, cancelamento
- Autonomia do vendedor e do supervisor

Este documento existe para:

- Evitar interpretação errada
- Evitar perguntas repetidas
- Permitir troca de desenvolvedor, IA ou tecnologia **sem perda de regra**

---

## CAPÍTULO 2 — FONTES DA VERDADE (REGRA INEGOCIÁVEL)

1. Google Sheets é a única fonte da verdade
  2. UND ≠ KG
  3. Preço sempre é R\$/KG
  4. Cliente nunca vê tabela
  5. Peso é sempre estimativo
  6. Não existe reserva
  7. Pedido é o primeiro compromisso do sistema
  8. Supabase não define regras
  9. Código apenas executa o que está no Sheets
- 

## CAPÍTULO 3 — ARQUITETURA OFICIAL (INDEPENDENTE DE TECNOLOGIA)

A arquitetura lógica do projeto está documentada na planilha:

🔗 Legenda / Objetivos sercarne.com

[https://docs.google.com/spreadsheets/d/12wPGal\\_n7PKYFGz9W\\_bXgK4mly2NbrEEGwTrIDCzcl/edit](https://docs.google.com/spreadsheets/d/12wPGal_n7PKYFGz9W_bXgK4mly2NbrEEGwTrIDCzcl/edit)

A aba “Legenda/Objetivos sercarne.com” define oficialmente:

- Stakeholders
- Responsabilidades
- Onde cada dado nasce
- Onde cada dado é consumido

👉 Esta aba faz parte do sistema.

---

## CAPÍTULO 4 — CATÁLOGO DE PRODUTOS (VISIBILIDADE)

### REGRA DEFINITIVA V8

Quem decide se o produto aparece na plataforma é a planilha.

- ♦ COLUNA AX (COLUNA 50) — EXIBIR NA PLATAFORMA

- **TRUE / VERDADEIRO** → produto aparece
- **FALSE / FALSO** → produto não aparece

✖ Regras:

- **Somente produtos com AX = TRUE aparecem**
  - Código do produto **não é critério**
  - Front-end e back-end **não podem ignorar essa coluna**
- 

## CAPÍTULO 5 — ORDEM DE EXIBIÇÃO NO CATÁLOGO

### REGRA AUTOMÁTICA (ANTES DE QUALQUER FILTRO)

Produtos são exibidos do **MAIOR** para o **MENOR** estoque disponível.

Objetivos:

- Vender o que tem mais disponibilidade
- Evitar ruptura
- Ajudar o vendedor sem ele decidir

Filtros futuros (não implementados ainda):

- Marca
- Espécie
- Tipo de embalagem

⚠ Até existirem, **somente a ordenação por estoque é aplicada**.

---

## CAPÍTULO 6 — LÓGICA DE PREÇO (IMUTÁVEL)

**Fórmula única:**

**VALOR ESTIMADO = UND × PESO MÉDIO × PREÇO POR KG**

- UND → carrinho
- Peso médio → planilha
- Preço → tabela aplicada automaticamente

✖ Nenhuma tela recalcula preço fora dessa lógica.

---

## CAPÍTULO 7 — TABELAS E REGRA DE VOLUME

UND total no carrinho    Tabela interna

1	TAB1
2–9	TAB0
≥10	TAB4

**Exibição ao cliente:**

- Preço base sempre TAB3
- Cliente vê apenas:
  - Preço R\$/KG
  - Benefício (%)

✗ Nunca mostrar tabela

✗ Nunca mostrar desconto em R\$

---

## CAPÍTULO 8 — ROTAS, DIAS DE ENTREGA E CUTOFF

**Fonte da verdade**

- Planilha: Google Sheets
- Aba: Rotas Dias De Entrega

✗ Supabase não define rotas

✗ Código não inventa regra

**Regras centrais**

- Rota é definida pela **CIDADE**
- Cada cidade pertence a um grupo logístico
- Cada grupo define:
  - Dias de entrega
  - Horário de corte (cutoff)

**Cutoff (regra crítica)**

- Sempre considerado **no dia anterior à entrega**
  - Após o horário de corte:
    - a próxima entrega válida é automaticamente pulada
  - Usuário **não pode forçar** data inválida
-

# CAPÍTULO 9 — ESTOQUE DISPONÍVEL POR DATA (REGRA OFICIAL)

## Fórmula cumulativa

Para cada data D:

$$\text{Disponível}(D) = \text{Estoque\_Base\_Hoje} + \text{Entradas}(\leq D) - \text{Pedidos\_Comprometidos}(\leq D)$$

Onde:

- **Estoque\_Base\_Hoje** = valor do Sheets (coluna H)
  - **Entradas( $\leq D$ )** = entradas até a data D
  - **Pedidos\_Comprometidos( $\leq D$ )** = pedidos com entrega até D e status comprometedor
- 

## Status que COMPROMETEM estoque:

- **PEDIDO ENVIADO**
- **PEDIDO CONFIRMADO**
- **SEU PEDIDO SAIU PARA ENTREGA**

## Status que NÃO comprometem:

- **CANCELADO**
- **PEDIDO ENTREGUE** (histórico)

💡 Ao cancelar um pedido, o estoque **volta automaticamente** (deixa de ser comprometido).

---

## Exemplo de validação (TESTE DE ACEITE)

Produto **400010**

Estoque base hoje (Sheets): **2 UND**

Entradas:

- +8 UND em 02/02/2026
- +100 UND em 05/02/2026

Pedidos:

- -2 UND com entrega 04/02/2026

- -5 UND com entrega 06/02/2026

Resultado:

Data	Disponível
------	------------

30/01	2
-------	---

31/01	2
-------	---

02/02	10
-------	----

03/02	10
-------	----

04/02	8
-------	---

05/02	108
-------	-----

06/02	103
-------	-----

07/02	103
-------	-----

---

## CAPÍTULO 10 — PEDIDOS E STATUS (FLUXO OFICIAL)

### Status oficiais:

1. **PEDIDO ENVIADO**
  2. **PEDIDO CONFIRMADO**
  3. **SEU PEDIDO SAIU PARA ENTREGA**
  4. **PEDIDO ENTREGUE**
  5. **CANCELADO**
- 

### Permissões

#### Níveis 1 a 5 (Cliente / Vendedor)

- Criar pedido (se houver estoque)
- Editar pedido **somente enquanto = PEDIDO ENVIADO**
- Cancelar pedido enquanto = PEDIDO ENVIADO

✗ Não podem:

- Confirmar pedido
- Alterar após confirmado

#### Níveis 6 a 10 (Supervisor / Admin)

- Confirmar pedido
  - Avançar status até entregue
  - Cancelar pedido em qualquer status (com motivo obrigatório)
  - Ajustar preço após confirmado (ver capítulo 11)
- 

## CAPÍTULO 11 — VOUCHER E AJUSTE DE PREÇO (COM PISO)

### Piso inegociável

**Preço final por KG nunca pode ser menor que TAB5 (coluna W).**

---

### Voucher do vendedor

- Usado apenas em **PEDIDO ENVIADO**
  - Aplicado por níveis 1–5
  - Sempre respeitando o piso TAB5
- 

### Ajuste de supervisor

- Pode ser feito **após PEDIDO CONFIRMADO**
- Apenas níveis 6–10
- Sempre com:
  - motivo
  - usuário
  - data/hora
- Respeitando o piso TAB5

Cliente **nunca vê tabela**, voucher ou ajuste — apenas o preço final.

---

## CAPÍTULO 12 — CANCELAMENTO

- Pode ocorrer em qualquer status (conforme permissão)
- Sempre exige:
  - motivo
  - usuário
  - data/hora
- Pedido CANCELADO:
  - **não compromete estoque**

- devolve disponibilidade automaticamente
- 

## CAPÍTULO 13 — REGRA DE ALTERAÇÕES (ANTI-REGRESSÃO)

 Proibido:

- Ajustar regra validada
- “Aproveitar” mudanças
- Criar exceções locais

 Qualquer evolução:

- Nova versão (V9+)
  - Documentada
  - Aprovada
- 

## ANEXO — CAMADAS INTOCÁVEIS

1. Modelo de Negócio
  2. Este Manual V8
  3. Google Sheets (abas oficiais)
  4. Lógica UND × KG × R\$/KG
  5. Estoque cumulativo por data
  6. Coluna AX (visibilidade)
  7. Piso TAB5
  8. Regra de status e permissões
- 

## DECLARAÇÃO FINAL

Este documento define integralmente a operação da plataforma Schlosser PRO V8.

Qualquer divergência entre sistema e este manual é erro de implementação.



# MANUAL SCHLOSSER PRO — V8

## CAPÍTULO 12 — MODO CONGELAMENTO E GOVERNANÇA DO SISTEMA

**Este capítulo é parte integrante do Manual Schlosser PRO V8 e tem força de regra absoluta.**

Ele se sobrepõe a qualquer interpretação técnica, sugestão de IA ou decisão pontual de desenvolvimento.

---

### 12.1 OBJETIVO DESTE CAPÍTULO

Definir **como o sistema deve ser mantido, alterado e evoluído** quando:

- o projeto entra em fase de lançamento
- existe prazo crítico de operação
- já há regras de negócio validadas
- a plataforma passa a ser usada no dia-a-dia real

Este capítulo existe para:

- impedir regressões
  - proteger o que já funciona
  - permitir troca de desenvolvedor ou IA sem perda de lógica
  - alinhar **gestão, tecnologia e modelo de negócio** em um único guia
- 

### 12.2 PRINCÍPIO FUNDAMENTAL (REGRA DE OURO)

**O SISTEMA EXECUTA O MANUAL.  
O MANUAL NÃO SE ADAPTA AO SISTEMA.**

Durante o Modo Congelamento:

- ✗ Código não cria regra
- ✗ Interface não inventa lógica
- ✗ IA não interpreta o negócio
- ✗ Desenvolvedor não “melhora” o modelo

- ✓ O código **apenas implementa fielmente o que está neste manual**
-

## 12.3 O QUE É O MODO CONGELAMENTO

O **Modo Congelamento** é o estado oficial do projeto quando:

- o Manual V8 está validado
- o sistema está funcional
- o foco é **estabilidade e operação**

Nesse modo:

- alterações são **permitidas apenas para corrigir divergências**
  - qualquer mudança fora do escopo é **proibida**
  - toda entrega exige validação externa
- 

## 12.4 PROIBIÇÕES ABSOLUTAS NO MODO CONGELAMENTO

Durante o Modo Congelamento, é **expressamente proibido**:

- refatorar código funcional
- reorganizar arquivos
- alterar layout aprovado
- renomear variáveis que já funcionam
- “otimizar” performance
- alterar comportamento sem erro comprovado

 Mesmo que a mudança pareça “melhor” tecnicamente, ela é proibida.

---

## 12.5 FONTE ÚNICA DA VERDADE

A **única fonte da verdade** do projeto é este Manual V8, incluindo:

- capítulos de regras de negócio
- explicação de colunas do Google Sheets
- lógica de estoque, rotas, preços e permissões
- capítulos complementares (ex.: WhatsApp V8.1)

 Supabase não define regra  
 Front-end não define regra  
 ERP não define regra

Todos apenas **executam o que está documentado aqui**.

---

## 12.6 REGRAS IMUTÁVEIS DO SISTEMA (RESUMO EXECUTIVO)

### 12.6.1 Pedido

- Todo pedido nasce como **PEDIDO ENVIADO**
  - Não existe status “PENDENTE”
  - Fluxo único:
    1. PEDIDO ENVIADO
    2. PEDIDO CONFIRMADO
    3. SEU PEDIDO SAIU PARA ENTREGA
    4. PEDIDO ENTREGUE
- 

### 12.6.2 Estoque

- Estoque é sempre **por data**
- Comprometem estoque:
  - PEDIDO ENVIADO
  - PEDIDO CONFIRMADO
  - SAIU PARA ENTREGA
- PEDIDO ENTREGUE:
  - não compromete
  - vira apenas histórico
- PEDIDO CANCELADO:
  - devolve estoque automaticamente

 Nunca exibir código/SKU no lugar da quantidade disponível

---

### 12.6.3 Catálogo

- Produto só aparece se **COLUNA AX = TRUE**
  - Ordenação padrão:
    - maior estoque disponível primeiro
  - Preço exibido:
    - sempre TAB3
    - cliente vê apenas **R\$/KG + benefício (%)**
    - nunca vê nome de tabela
- 

### 12.6.4 Permissões

- Usuários nível 1–5:
    - editar/cancelar **somente em PEDIDO ENVIADO**
  - Usuários nível 6–10:
    - confirmar
    - avançar status
    - cancelar em qualquer status
- 

## 12.6.5 Cancelamento

- Cancelamento exige:
    - motivo
    - usuário responsável
    - data e hora
  - Deve ser registrado no banco
  - Status final: **CANCELADO**
- 

## 12.6.6 Peso Médio

- Produto com peso médio = 0:
    - bloqueia envio do pedido
    - mensagem clara ao usuário
  - Bloqueio ocorre no **envio**, não apenas visual
- 

## 12.6.7 WhatsApp Obrigatório (V8.1)

- Todo pedido digitado gera WhatsApp automático ao cliente
  - WhatsApp é **ciência**, não confirmação
  - Não altera pedido
  - Mesmo fluxo para B2B e B2C
- 

## 12.6.8 Realtime

- Websocket é opcional
  - Se falhar:
    - sistema continua funcionando
    - fallback via fetch
    - sem travar interface
    - sem erro em loop
-

## 12.7 FORMA OBRIGATÓRIA DE ENTREGA (DEV / IA)

Qualquer alteração durante o Modo Congelamento deve conter:

- A) Lista completa de arquivos alterados
- B) Patch/diff completo (antes/depois)
- C) Checklist de aderência ao Manual V8
- D) Teste de aceite passo a passo

 Publicação automática é proibida  
 Ajustes posteriores não são aceitos

---

## 12.8 VALIDAÇÃO E RESPONSABILIDADE

**Nenhuma alteração entra em produção sem validação externa.**

Se houver regressão:

- a versão é revertida
  - o Modo Congelamento permanece ativo
  - a correção ocorre de forma pontual
- 

## 12.9 DECLARAÇÃO FINAL

“Estabilidade é prioridade absoluta.

A evolução do sistema só é válida  
quando não apaga o que já funciona.”

Você está atuando como DESENVOLVEDOR(A) DE SOFTWARE do projeto  
SCHLOSSER PRO — [sercarne.com](http://sercarne.com).

 **REGRA ABSOLUTA:**

Você NÃO TEM autonomia para alterar regras de negócio.

Você executa fielmente o MANUAL OPERACIONAL SCHLOSSER PRO V8  
e o CAPÍTULO WHATSAPP V8.1.

=====  
REGRAS FUNDAMENTAIS (IMUTÁVEIS)  
=====

- 1) Google Sheets é a fonte da verdade

- 2) Pedido é o primeiro compromisso (não existe reserva)
- 3) UND ≠ KG
- 4) Preço sempre R\$/KG
- 5) Cliente nunca vê tabela
- 6) WhatsApp NÃO é sistema

=====

## WHATSAPP — REGRA OBRIGATÓRIA (V8.1)

=====

- Todo pedido digitado por VENDEDOR para um CLIENTE DEVE disparar WhatsApp automático de ciência ao cliente.
- Esse disparo é OBRIGATÓRIO.
- O vendedor NÃO pode desativar o envio.

WhatsApp:

- NÃO valida pedido
- NÃO confirma pedido
- NÃO altera pedido
- NÃO muda status
- NÃO escreve no banco

A ausência de resposta do cliente:

- NÃO invalida o pedido
- NÃO bloqueia o fluxo

=====

## REGRAS DE DISPARO

=====

- Pedido criado por VENDEDOR → WhatsApp obrigatório (template B2B)
- Pedido criado por CLIENTE → WhatsApp informativo (template B2C)
- Disparo ocorre após:
  - PEDIDO ENVIADO (obrigatório)
  - PEDIDO CONFIRMADO (opcional)

Falha no envio do WhatsApp:

- NÃO cancela pedido
- NÃO altera status
- NÃO afeta estoque

=====

## PROIBIÇÕES EXPRESSAS

=====

Você NÃO pode:

- Criar botão “Confirmar pelo WhatsApp”
- Alterar status por resposta do WhatsApp

- Esperar resposta do cliente para seguir fluxo
  - Criar lógica de retorno WhatsApp → sistema
  - Alterar templates sem autorização
- =====

## METODOLOGIA OBRIGATÓRIA

=====

Sempre responda em 6 blocos:

- 1) Entendimento do pedido
- 2) Escopo permitido
- 3) Escopo proibido
- 4) Arquivos que serão alterados
- 5) Patch/diff completo
- 6) Teste de aceite

Se houver qualquer dúvida ou conflito:

PARE e peça confirmação antes de implementar.

=====

## DECLARAÇÃO FINAL OBRIGATÓRIA

=====

Antes de entregar qualquer código, você deve afirmar:

“Esta implementação está 100% aderente ao Manual V8  
e ao Capítulo WhatsApp V8.1, sem regressão.”

🌐 FLUXO VISUAL — WHATSAPP OBRIGATÓRIO (APRESENTAÇÃO INTERNA)

👉 Esse fluxo é ideal para:

apresentar para vendedores

alinhar supervisão

mostrar que WhatsApp protege a empresa

evitar discussão futura

🧱 FLUXO MACRO (VISÃO EXECUTIVA)

VENDEDOR

|

| cria pedido no sistema

v

PEDIDO ENVIADO (sercarne.com)

|

| sistema identifica: pedido B2B?

v

SIM —————► DISPARA WHATSAPP (CIÊNCIA)

|

| (cliente recebe aviso)

|

| (resposta não altera nada)

|

v

## PEDIDO SEGUE FLUXO NORMAL

(CONFIRMADO → SAIU → ENTREGUE)

✖ WhatsApp não trava, não valida, não decide.

✖ FLUXO DETALHADO (OPERACIONAL)

1) Vendedor lança pedido

2) Sistema salva pedido (PEDIDO ENVIADO)

3) Sistema verifica:

- pedido criado por vendedor? → SIM

4) Sistema monta mensagem padrão B2B

5) Sistema envia WhatsApp ao cliente

6) Pedido continua normalmente

7) Supervisor revisa / confirma

➡ FLUXO DE RESPOSTA DO CLIENTE (IMPORTANTE)

Cliente responde no WhatsApp

|

v

NADA MUDA AUTOMATICAMENTE

|

v

Ação HUMANA (se necessário):

- contato com vendedor

- ajuste/cancelamento NO SISTEMA

✖ FLUXOS PROIBIDOS (PARA APRESENTAÇÃO)

✖ Pedido aguardando “OK” do WhatsApp

✖ WhatsApp confirmando pedido

✖ WhatsApp alterando pedido

✖ WhatsApp mudando status

✖ WhatsApp virando prova única

✖ FRASE-CHAVE PARA O TIME (USE ESSA)

“O pedido é feito no sistema.

O WhatsApp serve apenas para deixar o cliente ciente.”

 V8.0  CHECKLIST TÉCNICO



# CHECKLIST TÉCNICO

## SCHLOSSER PRO — VERSÃO V8 (DEV / IA)

### Objetivo:

Garantir que qualquer implementação técnica esteja **100% aderente ao Manual Operacional V8**, sem perda de regra de negócio.

---

### BLOCO 0 — REGRAS ABSOLUTAS (SE FALHAR → REPROVA)

Antes de qualquer commit, responder **SIM** para todas:

- Google Sheets é a **única fonte da verdade**
- UND ≠ KG em todo o sistema
- Preço é sempre tratado como **R\$/KG**
- Cliente **nunca** vê tabela
- Peso é sempre **estimativo**
- Não existe reserva de estoque
- Pedido é o primeiro compromisso
- Manual V8 foi lido integralmente

Se algum item acima for “**NÃO**”, **interromper desenvolvimento**.

---

### BLOCO 1 — CATÁLOGO DE PRODUTOS

#### 1.1 Visibilidade (OBRIGATÓRIO)

- Produto só aparece se **Coluna AX = TRUE**
- Nenhum filtro por código (ex.:  $\geq 400000$ ) existe
- Front-end não ignora AX
- Back-end não ignora AX

#### 1.2 Ordenação inicial

- Produtos são exibidos **por maior estoque disponível primeiro**
- Nenhum outro critério interfere antes disso
- Filtros futuros (marca, espécie, embalagem) **não estão ativos**



## BLOCO 2 — PRECIFICAÇÃO (IMUTÁVEL)

### 2.1 Fórmula

- Valor estimado = `UND × PESO MÉDIO × PREÇO/KG`
- Peso médio vem da planilha
- UND vem do carrinho
- Preço vem da tabela aplicada

### 2.2 Centralização

- Nenhuma tela calcula preço diretamente
- Toda precificação passa por `schlosserRules.js`
- Funções obrigatórias existem:
  - `getTabelaAplicada()`
  - `getPrecoKg()`
  - `calcularValorEstimado()`
  - `calcularBeneficioPercentual()`



## BLOCO 3 — TABELAS E VOLUME

- Regra por UND total:
  - 1 UND → TAB1
  - 2–9 UND → TAB0
  - $\geq 10$  UND → TAB4
- Preço exibido ao cliente é sempre TAB3
- Cliente vê apenas benefício em %
- Nome de tabela nunca aparece



## BLOCO 4 — ROTAS, DATAS E CUTOFF (CRÍTICO)

### 4.1 Fonte da verdade

- Aba usada: `Rotas Dias De Entrega`
- Supabase não define rota
- Front-end não cria regra de data

### 4.2 Regras obrigatórias

- Rota definida **pela cidade**
  - Datas permitidas respeitam os dias da rota
  - Cutoff aplicado **no dia anterior à entrega**
  - Após cutoff, a entrega mais próxima é bloqueada
  - Usuário não consegue forçar data inválida
- 



## BLOCO 5 — ESTOQUE DISPONÍVEL POR DATA

### 5.1 Fórmula obrigatória

Para cada data **D**:

$$\begin{aligned}\text{Disponível}(D) &= \text{Estoque\_Base\_Hoje} \\ &+ \text{Entradas}(\leq D) \\ &- \text{Pedidos\_Comprometidos}(\leq D)\end{aligned}$$

### 5.2 Origem dos dados

- Estoque base = Sheets (coluna H)
- Entradas = Supabase (entradas\_estoque)
- Pedidos = Supabase (pedidos)

### 5.3 Status que COMPROMETEM estoque

- PEDIDO ENVIADO
- PEDIDO CONFIRMADO
- SEU PEDIDO SAIU PARA ENTREGA

### 5.4 Status que NÃO comprometem

- CANCELADO
  - PEDIDO ENTREGUE
- 



## BLOCO 6 — PEDIDOS E STATUS

### 6.1 Status válidos

- PEDIDO ENVIADO
- PEDIDO CONFIRMADO
- SEU PEDIDO SAIU PARA ENTREGA
- PEDIDO ENTREGUE

- CANCELADO

## 6.2 Permissões (OBRIGATÓRIO)

### Níveis 1–5

- Criam pedido
- Editam/cancelam **somente** se status = PEDIDO ENVIADO
- Não alteram após confirmado

### Níveis 6–10

- Confirmam pedido
  - Avançam status
  - Cancelam em qualquer status (com motivo)
  - Ajustam preço (ver bloco 7)
- 



## BLOCO 7 — VOUCHER E AJUSTE DE PREÇO

### 7.1 Piso inegociável

- Preço final por KG ≥ TAB5 (coluna W)

### 7.2 Voucher (Vendedor)

- Só pode ser aplicado em PEDIDO ENVIADO
- Apenas níveis 1–5
- Respeita piso TAB5

### 7.3 Ajuste (Supervisor)

- Pode ocorrer após PEDIDO CONFIRMADO
- Apenas níveis 6–10
- Sempre exige:
  - motivo
  - usuário
  - data/hora
- Respeita piso TAB5

Cliente **nunca** vê:

- tabela
  - voucher
  - ajuste
-

## BLOCO 8 — CANCELAMENTO

- Cancelamento exige motivo
  - Registra usuário e data/hora
  - Pedido CANCELADO:
    - não compromete estoque
    - devolve disponibilidade automaticamente
- 

## BLOCO 9 — ANTI-REGRESSÃO

Antes de publicar:

- Nenhuma regra do Manual V8 foi alterada
  - Nenhuma melhoria “aproveitou” outro ajuste
  - Nenhuma lógica foi duplicada fora do núcleo
  - Código respeita o fluxo mental do vendedor:  
**Escolhe UND → sistema decide o resto**
- 

## BLOCO 10 — TESTE DE ACEITE (OBRIGATÓRIO)

- Produto com AX=FALSE não aparece
  - Ordenação por estoque funciona
  - Cutoff bloqueia data corretamente
  - Pedido ENVIADO compromete estoque
  - Cancelar pedido devolve estoque
  - PEDIDO ENTREGUE não compromete
  - Preço nunca fica abaixo da TAB5
- 

## DECLARAÇÃO FINAL (DEV / IA)

**Se qualquer item deste checklist falhar, a implementação não está conforme a V8 e deve ser corrigida antes de ir para produção.**

# V9 - KARDEX Evolução para Kardex (V9)

# Capítulo Oficial — Evolução para Kardex (V9)

## Objetivo

Migrar o controle de estoque para um modelo **100% rastreável por movimentações**, suportando:

- transformação (produto mãe → produtos filhos)
- desossa, produção e embutidos
- transferências entre unidades
- ajustes e inventário
- auditoria completa por usuário/motivo

## Fonte da Verdade (V9)

- **Supabase será a fonte única do saldo.**
- Sheets permanece como **cadastro/precificação/visibilidade** (catálogo), não como saldo oficial.

## Conceito

Toda mudança de estoque vira um lançamento no **Kardex**:

**Saldo por SKU e data = soma(entradas) – soma(saídas)**

## Estrutura recomendada (tabelas)

### 1) `stock_movements` (Kardex)

Campos mínimos:

- `id`
- `sku`
- `movement_date` (data/hora)
- `qty_und` (positivo = entrada / negativo = saída)
- `type` (enum)
- `origin_ref` (ex.: pedido\_id, nf\_id, transferencia\_id, producao\_id)
- `unit` (ex.: matriz, filial)
- `notes`

- `created_by`
- `created_at`

## 2) `stock_balance` (opcional, cache)

- `sku`
- `unit`
- `balance_und`
- `updated_at`

(Atualizado por trigger/edge function para performance.)

## Tipos de movimento (enum sugerido)

### ENTRADAS

- `COMPRA`
- `DEVOLUCAO`
- `PRODUCAO_ENTRADA` (produto filho entra)
- `AJUSTE_POSITIVO`
- `TRANSFERENCIA_ENTRADA`

### SAÍDAS

- `VENDA`
- `BONIFICACAO`
- `PERDA`
- `PRODUCAO_SAIDA` (produto mãe sai)
- `AJUSTE_NEGATIVO`
- `TRANSFERENCIA_SAIDA`

## Transformação (produto mãe → produtos filhos)

Exemplo: 5 meia res (400010) viram cortes embalados

- Lança `PRODUCAO_SAIDA` em 400010: `-5 UND`
- Lança `PRODUCAO_ENTRADA` nos filhos (picanha, maminha...): `+X UND`  
Tudo com `origin_ref = production_batch_id`

## Como isso afeta o SerCarne

- O “disponível por data” vira consulta direta no kardex (muito mais fiel)
- Cancelamento devolve estoque com um movimento inverso rastreável
- Transferência entre Zaleski matriz/filial fica automática

- Auditoria real para gestão e contador

## Prioridade após GO LIVE

1. Criar `stock_movements`
  2. Ajustar cálculo disponível por data para ler do kardex
  3. Criar fluxo de produção/transformação (produto mãe/filho)
  4. Transferências entre unidades
  5. Inventário e ajustes com permissão nível 8–10
- 

## Pra fechar: o que você me pediu “validar”

- Sim — é **mais vantajoso** ter o controle de estoque no Supabase, porque você precisa Kardex por causa de transformação e transferências.
- Mas **NÃO antes do GO LIVE**. A decisão correta é: **lança V8 congelado** e entra em **Kardex V9** como prioridade máxima.

HORIZONS ENTENDEU

 PERFEITO! AQUI ESTÁ TUDO QUE VOCÊ DISSE QUE É REGRA (NÃO DISCUTÍVEL):



## REGRAS CRÍTICAS - MANUAL DO SISTEMA

### 1 ACESSO SEM LOGIN (B2C - CONSUMIDOR FINAL)

Catálogo:

-  Banner: "CATÁLOGO PÚBLICO"
-  Preço exibido: TAB3 (preço público)
-  Sem desconto % visível
-  Mostra estoque disponível por data de entrega

Estoque Disponível (Fórmula):

-  disponível(data) = base(H) + entradas\_até(data) - pedidos\_confirmados\_até(data)
-  Base vem de coluna H (Estoque Inicial) do Catálogo
-  Entradas vêm de tabela entradas\_estoque (Supabase)
-  Pedidos vêm de tabela pedidos (Supabase) com status = 'CONFIRMADO'
-  Exemplo: Produto 400010 em 30/01 =  $2 + 10 - 7 = 5$  UND disponível

Carrinho (sem login):

-  NÃO pode aparecer lista de clientes (Lei LGPD)

- Campo obrigatório: Cidade de Entrega (dropdown com todas as cidades da aba "Rotas Dias De Entrega" coluna B)
- Opção adicional: "RETIRAR NO FRIGORÍFICO" (independente de cidade/rota)
- Se cidade está em Rotas: mostra datas de entrega disponíveis
- Se cidade NÃO está em Rotas: mostra "Enviar Solicitação"
- Campos de cadastro (obrigatórios):

- 
- Validação: Bloqueia se qtd > estoque disponível (mostra em vermelho)
- Peso Estimado:  $qtd \times peso\_médio$  (sem NaN)
- Subtotal Estimado:  $peso\_total \times preço\_kg$  (sem NaN)

Descrição do Produto:

- Ordem de prioridade: AK → AL → E (fallback)
- Descrição principal (AK): negrito
- Descrição técnica (AL): cinza/menor
- Nunca exibir #N/A

Pedido (sem login):

- Status ao criar: 'SOLICITADO' (não PENDENTE)

- Salva dados do cliente: nome, CPF/CNPJ, WhatsApp, cidade

- Mensagem após confirmar: "Pedido enviado! Aguarde confirmação"

- Limpa carrinho após confirmar

---

## 2) ACESSO COM LOGIN (B2B - VENDEDOR/CLIENTE)

Nível de Usuário (Supabase > usuarios > COLUNA 8):

- Nível 1-5: Cliente/Vendedor (cria pedido com status 'SOLICITADO')

- Nível 6-10: Vendedor Senior (pode confirmar pedidos → status 'CONFIRMADO')

Catálogo (com login):

- Banner: "CATÁLOGO PARCEIROS"

- Preço exibido: TAB0, TAB1 ou TAB4 (conforme nível/cliente)

- Mostra desconto % vs TAB3 (em verde, ex: "15% OFF")

- Mostra estoque disponível por data de entrega

- Descrição: AK (negrito) + AL (cinza)

Dashboard (Vendedor nível 6+):

- Lista pedidos com status 'SOLICITADO' (aguardando confirmação)

- Lista pedidos com status 'CONFIRMADO' (já confirmados)

- Botão "Confirmar" (apenas para nível 6+)

- Ao confirmar: status muda para 'CONFIRMADO' (permanente, não volta)
- Atualização otimista (não refetch automático)
- Sugestão futura: Filtro por ROTA + Gráfico "Top 10 Clientes do Mês"

Pedido (com login):

- Se nível 1-5: status = 'SOLICITADO'
- Se nível 6+: status = 'CONFIRMADO' (direto)

---

### 3 DESCRIÇÃO DO PRODUTO (AMBOS OS ACESSOS)

Estrutura no Google Sheets (2026 Base Catalogo Precifica V2):

- Coluna AK: Descrição principal (usar primeiro)
- Coluna AL: Descrição técnica (usar se AK vazio)
- Coluna E: Descrição fallback (usar se AK e AL vazios)
- Nunca exibir: #N/A, #N/A!, espaços em branco

Exibição:

- AK: negrito, maior
- AL: cinza, menor

## 4 PREÇOS (AMBOS OS ACESSOS)

Sem Login (B2C):

- Preço: TAB3 (público)
- Sem desconto visível

Com Login (B2B):

- Preço: TAB0, TAB1 ou TAB4 (conforme cliente/nível)
- Desconto %:  $((\text{TAB3} - \text{preço_cliente}) / \text{TAB3}) \times 100$
- Exibição: "15% OFF" (em verde, apenas logado)

---

## 5 PESO E SUBTOTAL

Cálculo:

- Peso Estimado:  $\text{qtd} \times \text{peso_médio}$  (coluna do Sheets)
- Subtotal Estimado:  $\text{peso_total} \times \text{preço_kg}$
- Nunca exibir NaN (validar antes de calcular)
- Se dados inválidos: mostrar "Peso não disponível" ou "0"

---

## 6 ROTAS E DATAS DE ENTREGA

Fonte: Google Sheets - Aba "Rotas Dias De Entrega"

- Coluna A: Rota de entrega
- Coluna B: Município/Cidade
- Coluna C+: Dias de entrega por data

Fluxo:

1. Cliente seleciona Cidade
2. Sistema busca Rota correspondente (coluna A)
3. Sistema mostra datas de entrega disponíveis
4. Cliente seleciona data
5. Sistema calcula estoque disponível naquela data

Se cidade NÃO está em Rotas:

- Mostrar: "Enviar Solicitação" (cliente preenche formulário)

Opção adicional:

- "RETIRAR NO FRIGORÍFICO" (sem rota, sem data específica)

---

## 7] VALIDAÇÕES E BLOQUEIOS

## Estoque:

- Se `qtd_solicitada` > `estoque_disponivel`: bloqueia em vermelho
- Mensagem: "Apenas X UND disponíveis em [data]"

## Pedido sem login:

- Bloqueia se campos obrigatórios vazios (Nome, CPF/CNPJ, WhatsApp, Cidade)
- Bloqueia se não selecionou data de entrega (ou "RETIRAR NO FRIGORÍFICO")

## Pedido com login:

- Bloqueia se estoque insuficiente
- Permite confirmar se nível 6+

---

## 8 VÍDEO/MÍDIA

- Remover Google Drive (causa loop infinito)
- Usar YouTube (<https://youtu.be/NVvHTm4e-N4>) ou remover

---

## 9 ERROS A EVITAR

- Nunca exibir NaN (peso, subtotal, preço)

-  Nunca exibir #N/A (descrição)
  -  Nunca mostrar lista de clientes (sem login - LGPD)
  -  Nunca deixar status voltar de CONFIRMADO para SOLICITADO
  -  Nunca permitir qtd > estoque sem avisar
  -  Nunca carregar vídeo do Google Drive
- 

## RESUMO FINAL:

Aspecto	Sem Login (B2C)	Com Login (B2B)
Banner	CATÁLOGO PÚBLICO	CATÁLOGO PARCEIROS
Preço	TAB3	TAB0/TAB1/TAB4
Desconto %	Não	Sim (verde)
Status Pedido	SOLICITADO	SOLICITADO (1-5) / CONFIRMADO (6+)
Carrinho	Cidades + Cadastro	Dados do usuário
Estoque	Desconta pedidos confirmados	Desconta pedidos confirmados
Descrição	AK (negrito) + AL (cinza)	AK (negrito) + AL (cinza)

---

AGORA VOCÊ PODE TESTAR COM CONFIANÇA! 

**V7.6 ROTAS + ESTOQUE  
DISPONIVEL**

♦ **1 TEXTO OFICIAL — para colar no REQUISITOS.md**

**Este é o bloco definitivo.**

Se alguém descumprir isso, está **fora do modelo Schlosser PRO**.

## # 9. ESTOQUE POR DATA DE ENTREGA (REGRA CRÍTICA)

No Schlosser PRO, o estoque **\*\*NÃO\*\*** é validado pela data atual.  
Toda validação ocorre **\*\*exclusivamente pela DATA DE ENTREGA\*\***.

O cliente compra **\*\*UNIDADE (UND)\*\***.

O peso é sempre **\*\*estimado\*\***, apenas informativo, até a expedição.

---

### ## 9.1 Fontes de Dados

- Google Sheets:

- Estoque físico base (câmara fria)
- Entradas futuras de estoque (aba ENTRADAS\_ESTOQUE)

- Supabase:

- Pedidos confirmados
- Reservas ativas
- Saídas (expedição)

---

### ## 9.2 Fórmula Oficial de Estoque

Para cada produto e data de entrega:

```
estoque_disponivel_na_data =  
estoque_fisico_base
```

- pedidos\_confirmados\_ate\_a\_data
- entradas\_previstas\_ate\_a\_data

---

## ## 9.3 Regra de Bloqueio

O sistema **\*\*DEVE bloquear\*\*** a seleção de uma data de entrega quando:

estoque\_disponivel\_na\_data < quantidade\_solicitada

Mesmo que:

- existe rota logística
- existe dia de entrega configurado
- existe estoque hoje

---

## ## 9.4 Regra de Sugestão Automática

Quando uma data estiver bloqueada por falta de estoque, o sistema deve:

1. Calcular a próxima data da rota do cliente
2. Verificar novamente o estoque por data
3. Sugerir automaticamente a **\*\*primeira data futura viável\*\***

---

## ## 9.5 Caso de Uso Oficial (Exemplo)

Data atual: 26/01/2026

Produto: 400010 – Meia Rês

Quantidade solicitada: 4 UND

- Estoque físico atual: 2 UND
- Pedidos já confirmados: 1 UND
- Entrada futura: +10 UND em 28/01/2026

Resultado:

- Entrega 27/01 → BLOQUEADA (estoque insuficiente)
- Entrega 29/01 → LIBERADA (estoque disponível: 11 UND)

---

```
## 9.6 Regra de Divisão de Pedido (Opcional)
```

```
O sistema **PODE sugerir**, mas nunca impor:
```

- Entrega parcial imediata (se houver estoque)
  - Novo pedido agendado para a próxima data viável
- 

## ◆ ② PSEUDOCÓDIGO (para Horizons, dev ou IA)

Isso elimina 100% das ambiguidades que estão te travando hoje.

```
function calcularEstoqueDisponivel(produtoid, dataEntrega) {  
    estoqueBase = sheets.getEstoqueFisico(produtoid)  
    entradas = sheets.getEntradasAteData(produtoid, dataEntrega)  
    pedidos = supabase.getPedidosConfirmadosAteData(produtoid, dataEntrega)  
  
    return estoqueBase + entradas - pedidos  
}  
  
function validarDataEntrega(produtoid, quantidade, dataEntrega) {  
    estoque = calcularEstoqueDisponivel(produtoid, dataEntrega)  
  
    if (estoque >= quantidade) {  
        return { valido: true }  
    }  
  
    return {  
        valido: false,  
        motivo: `Estoque insuficiente para ${dataEntrega}`  
    }  
}  
  
function sugerirProximaData(produtoid, quantidade, rota, dataAtual) {  
    datasRota = rota.getProximasDatas(dataAtual)  
  
    for (data of datasRota) {  
        estoque = calcularEstoqueDisponivel(produtoid, data)  
        if (estoque >= quantidade) {
```

```
    return data
}
}

return null
}

-----
```

## ◆ **3 AJUSTES DE UI (simples, mas fundamentais)**

### O que REMOVER

- Bloco duplicado explicando rota no carrinho  
 A rota já está clara no calendário
- 

### O que MANTER

- Calendário de datas:
    -  Disponível (rota + estoque suficiente)
    -  Bloqueado (rota existe, mas estoque insuficiente)
- 

### O que ADICIONAR (texto exato)

#### Quando data estiver bloqueada:

Estoque insuficiente para esta data.

Disponível: 1 UND

#### Quando sugerir nova data:

Sugestão automática:

Entrega completa disponível em 29/01/2026

---



## Carrinho (padrão final)

Exibição obrigatória:

4 UND × 180 kg × R\$ 24,40/kg

Peso total estimado: 720 kg

Entrega prevista: 29/01/2026



# MANUAL COMPLETO V7.5

## 2) MANUAL TÉCNICO OFICIAL — Schlosser PRO V7.5

---

# MANUAL TÉCNICO OFICIAL — SCHLOSSER PRO V7.5

**Plataforma:** sercarne.com | 2026

**Status:** Documento Oficial (Fonte Única da Verdade)

**Autoridade máxima:** Cássio Zaleski

**Stack oficial:** Google Sheets (verdade) + Supabase (operação) + Horizons (front)

---

## 0) REGRAS ABSOLUTAS (NÃO NEGOCIÁVEIS)

1. **Google Sheets = fonte da verdade** para catálogo, preço e rotas.
  2. **UND ≠ KG** (UND é unidade de compra; KG é base de cálculo).
  3. **Preço sempre é R\$/KG** (nunca “R\$/UND”).
  4. **Cliente nunca vê tabela (TAB0/TAB1/etc)** — ele vê **benefício %**.
  5. **Produto com código < 400000 NÃO aparece** no catálogo.
  6. **Pedido só existe se originado de reserva válida** (TTL 2h).
  7. **Nenhuma page recalcula preço** fora de `src/domain/schlosserRules.js`.
- 

## 1) ARQUITETURA (VISÃO GERAL)

### 1.1 Google Sheets (Fonte Primária)

- Catálogo, descrições, peso médio, imagens, tabelas de preço (TABs)
- Rotas, dias de entrega, cutoff
- Clientes (quando aplicável)
- Estoque em câmara fria por produto

### 1.2 Supabase (Banco Operacional)

- Autenticação simples (usuarios)
- Reservas (travamento do estoque por 2h)
- Pedidos (pedido confirmado)
- Entradas futuras de estoque (entradas\_estoque)

### **1.3 Frontend (Hostinger Horizons)**

- Catálogo B2C público (sem login)
  - Login inline (sem trocar URL base)
  - B2B (cliente / vendedor)
  - Carrinho + reserva + pedido
  - Estoque por data/rota (timeline)
- 

## **2) GOOGLE SHEETS — FONTE PRIMÁRIA**

### **2.1 Planilha Oficial**

**Nome:** Precificação Mix Frigo Schlosser / Zaleski 2026 (IA APP)

**Aba oficial de catálogo:** [2026 Base Catalogo Precifica V2](#)

### **2.2 Filtro obrigatório de cadastro válido**

- Somente produtos com código  $\geq 400000$
  - Código do produto: **COLUNA D** ([codigo\\_produto](#))
- 

## **3) DATA DICTIONARY — ABA “2026 Base Catalogo Precifica V2”**

### **3.1 Identidade do Produto**

- **COLUNA D — [codigo\\_produto](#)**
  - Tipo: number (int)
  - Regra:  $\geq 400000$
  - Chave primária no app

### **3.2 Descrições (fallback obrigatório)**

- **COLUNA AK — [descricao\\_nf](#)** (principal)
- **COLUNA AL — [descricao\\_tecnica](#)** (secundária)
- **COLUNA E — [descricao\\_fallback](#)**

**Ordem obrigatória de exibição:**

1. AK
2. AL (abaixo, cinza)

3. Se AK/AL vazias ou **#N/A**, usar E

 Nunca exibir **#N/A**

### 3.3 Unidade comercial e peso médio

- COLUNA AC — **tipo\_venda**: UND ou CX
- COLUNA I — **peso\_medio\_kg** (estimativa)

Regra oficial de CX:

- **tipo\_venda == CX** ⇒ peso\_medio\_kg assumido = **10kg** (padrão)
- Variação permitida: ±30%
- Sempre estimativo (peso real ajusta na expedição)

### 3.4 Tabelas de preço (R\$/KG)

Preço sempre é por KG.

- **TAB0**: COLUNA V — **preco\_tab0\_kg**
- **TAB5**: COLUNA W — **preco\_tab5\_kg**
- **TAB4**: COLUNA X — **preco\_tab4\_kg**
- **TAB1**: COLUNA Y — **preco\_tab1\_kg**
- **TAB2**: COLUNA Z — **preco\_tab2\_kg** (se existir/ativar)
- **TAB3**: COLUNA AA — **preco\_tab3\_kg** (público / base)

### 3.5 Imagens (ordem obrigatória)

1. **AE img\_principal**
2. **AF img\_alt\_1**
3. **AG img\_alt\_2**
4. **AH img\_marca** (se cair aqui: badge “Imagen da Marca”)
5. fallback: imagem padrão local Schlosser

 Não buscar imagem externa

 Não alterar ordem de fallback

---

## 4) ROTAS E DIAS DE ENTREGA — GOOGLE SHEETS

### 4.1 Aba oficial

**Rotas Dias De Entrega**

### 4.2 Colunas

- A — `descricao_grupo_rota`
- B — `municipio` (chave de busca)
- C — `dias_entrega` (texto)
- D — `corte_ate` (cutoff)
- E — `rota_logistica_sysmo` (auditoria)
- F — `codigo_cidade` (IBGE/controle)

### 4.3 Regras obrigatórias (execução)

- Rota é por `cidade` (coluna B)
- Datas válidas só nos dias da coluna C
- Cutoff da coluna D bloqueia a entrega “mais próxima” quando excedido

Exemplo (obrigatório):

- Entrega: terça/quinta
- Cutoff: 17:30

Pedido	Próxima entrega
Segunda 17:20	Terça
Segunda 17:31	Quinta

---

## 5) SUPABASE — BANCO OPERACIONAL

### 5.1 Tabela `usuarios`

Campos:

- `usuario`
- `login`
- `senha_hash` (*hoje pode estar texto; futuro migrar p/ hash real*)
- `tipo_de_Usuario`
- `ativo` (bool)
- `app_login` (rota pós-login)
- `nivel` (int2) — hierarquia:
  - 1–3 cliente
  - 4–7 vendedor/supervisor
  - 8–10 gestão/admin
- `tabelas` (text) — quais TABs pode usar (interno)

Regras:

- login case-insensitive
- bloqueia se `ativo=false`
- redireciona via `app_login`

## 5.2 Tabela `reservas` (TTL 2h)

Campos recomendados:

- `id, created_at, expires_at`
- `status: RESERVADO|EXPIRADO|CONVERTIDO_PEDIDO|CANCELADO`
- cliente/digitador
- `codigo_produto`
- `quantidade_und`
- `tipo_venda`
- `peso_medio_kg`
- `peso_total_estimado_kg`
- `preco_kg_aplicado`
- `tab_aplicada` (interno)
- `valor_total_estimado`

## 5.3 Tabela `pedidos`

Campos recomendados:

- `id, created_at, status`
- cliente/digitador/supervisor
- rota, cidade
- `data_entrega_prevista` (ou `delivery_date`)
- `itens` (jsonb)
- `valor_total_estimado`

## 5.4 Tabela `entradas_estoque` (entradas futuras)

Campos recomendados:

- `codigo_produto`
  - `data_disponivel`
  - `quantidade_total_und`
  - (opcional) `rota_nome`
-

## 6) REGRAS DE PRECIFICAÇÃO (EXECUTÁVEL)

### 6.1 Preço visível (sempre)

- Sempre mostrar TAB3 (AA) como “preço público”

### 6.2 Tabela aplicada (interno) por volume total UND do carrinho

UND total carrinho    Tabela

1	TAB1
2–9	TAB0
≥10	TAB4

### 6.3 Fórmula obrigatória (fundamental)

$$\text{VALOR} = \text{UND} \times \text{PESO\_MÉDIO} \times \text{PREÇO\_POR\_KG}$$

Exemplo:

- $2 \text{ UND} \times 180 \text{ kg} \times \text{R\$ } 24,40/\text{kg} = \text{R\$ } 8.784,00$

### 6.4 Benefício %

Cliente vê:

- “Benefício do seu cadastro + volume: X%”  
Comparação: preço aplicado vs TAB3

---

## 7) ESTOQUE ONLINE (MODELO OFICIAL)

### 7.1 Conceitos

- **Estoque câmara fria (hoje):** base em UND (Sheets: coluna H)
- **Entradas futuras:** (Supabase): UND por data ([entradas\\_estoque](#))
- **Comprometido:** reservas válidas + pedidos não cancelados (por data entrega)
- **Disponível por data:**  
 $\text{disponível(d)} = \text{base} + \text{entradas\_ate(d)} - \text{comprometido\_ate(d)}$

### 7.2 UI obrigatória

- No ProductCard:
  - Badge “Disponível p/ data selecionada”
  - Timeline 30 dias (compacta)

---

## 8) STATUS DO PROJETO (AGORA)

### 8.1 Já rodando no Horizons / Hostinger (OK)

- Catálogo carregando do Sheets (filtro  $\geq 400000$ )
  - Login vendedor
  - Seleção de clientes
  - Tabelas de preço por regra central
  - Lançamento de pedidos no Supabase
  - Badge de disponibilidade por data selecionada
- 

## 9) PRÓXIMOS PASSOS (SEM QUEBRAR O QUE ESTÁ OK)

### Passo 2 — Entradas futuras (timeline real)

- Garantir que `entradas_estoque` esteja preenchida com:
  - `codigo_produto`
  - `data_disponivel`
  - `quantidade_total_und`

### Passo 3 — Reservas 2h (job)

- Job/cron: expirar reservas e liberar comprometido
- Bloquear pedido sem reserva válida

### Passo 4 — Rotas/cutoff “travando” seleção de data (UX)

- Calendário só mostra dias válidos
  - Após cutoff, pula próxima entrega
- 

## 10) LINKS E CREDENCIAIS (GOVERNANÇA)

### 10.1 Links oficiais

- Site: <https://sercarne.com/>
- Catálogo/cliente: <https://sercarne.com/cliente>

- Planilha Google (ID fixo no código):  
`12wPGal_n7PKYFGz9W__bXgK4mly2NbrEEGwTrIDCzcI`

## 10.2 Credenciais (NÃO colocar em repo/documento público)

Para ficar “profissional e seguro”, o padrão é:

- **Documento oficial:** apenas *onde encontrar* (ex.: “1Password / cofre / variável de ambiente”)
- **Arquivo privado interno** (não versionado): valores reais

**Sugestão de seção (preencher internamente):**

- Supabase URL: \_\_\_\_\_
  - Supabase ANON KEY: \_\_\_\_\_
  - Service Role (somente backend): \_\_\_\_\_
  - Login admin Supabase: \_\_\_\_\_
  - Senhas operacionais: \_\_\_\_\_
- 

## Observação importante sobre “senhas no manual”

Eu **não recomendo** colocar senha explícita em manual que pode circular (mesmo internamente), porque vira risco real. O mais profissional é:

- manual aponta “onde está” (cofre/1Password/Secrets do Hostinger/Vercel)
- e quem tem permissão acessa lá.

Se você quiser, eu preparam um **ANEXO SECRETO** (separado) “Credenciais Operacionais” pra você guardar offline.

---

No Schlosser PRO, o estoque \*\*NÃO\*\* é validado pela data atual.  
Toda validação ocorre \*\*exclusivamente pela DATA DE ENTREGA\*\*.

O cliente compra \*\*UNIDADE (UND)\*\*.  
O peso é sempre \*\*estimado\*\*, apenas informativo, até a expedição.

---

## 9.1 Fontes de Dados

- Google Sheets:  
- Estoque físico base (câmara fria)  
- Entradas futuras de estoque (aba ENTRADAS\_ESTOQUE)

- Supabase:  
- Pedidos confirmados  
- Reservas ativas  
- Saídas (expedição)

---

## ## 9.2 Fórmula Oficial de Estoque

Para cada produto e data de entrega:

```
estoque_disponivel_na_data =  
estoque_fisico_base  
  
• pedidos_confirmados_ate_a_data  
• entradas_previstas_ate_a_data
```

---

## ## 9.3 Regra de Bloqueio

O sistema **DEVE bloquear** a seleção de uma data de entrega quando:

estoque\_disponivel\_na\_data < quantidade\_solicitada

Mesmo que:

- existe rota logística
- existe dia de entrega configurado
- existe estoque hoje

---

## ## 9.4 Regra de Sugestão Automática

Quando uma data estiver bloqueada por falta de estoque, o sistema deve:

1. Calcular a próxima data da rota do cliente
2. Verificar novamente o estoque por data
3. Sugerir automaticamente a \*\*primeira data futura viável\*\*

---

#### ## 9.5 Caso de Uso Oficial (Exemplo)

Data atual: 26/01/2026

Produto: 400010 – Meia Rês

Quantidade solicitada: 4 UND

- Estoque físico atual: 2 UND
- Pedidos já confirmados: 1 UND
- Entrada futura: +10 UND em 28/01/2026

Resultado:

- Entrega 27/01 → BLOQUEADA (estoque insuficiente)
- Entrega 29/01 → LIBERADA (estoque disponível: 11 UND)

---

#### ## 9.6 Regra de Divisão de Pedido (Opcional)

O sistema \*\*PODE sugerir\*\*, mas nunca impõe:

- Entrega parcial imediata (se houver estoque)
- Novo pedido agendado para a próxima data viável

---

## ◆ ② PSEUDOCÓDIGO (para Horizons, dev ou IA)

Isso elimina 100% das ambiguidades que estão te travando hoje.

```
function calcularEstoqueDisponivel(produtoid, dataEntrega) {  
    estoqueBase = sheets.getEstoqueFisico(produtoid)
```

```

entradas = sheets.getEntradasAteData(produtoid, dataEntrega)
pedidos = supabase.getPedidosConfirmadosAteData(produtoid, dataEntrega)

return estoqueBase + entradas - pedidos
}

function validarDataEntrega(produtoid, quantidade, dataEntrega) {
  estoque = calcularEstoqueDisponivel(produtoid, dataEntrega)

  if (estoque >= quantidade) {
    return { valido: true }
  }

  return {
    valido: false,
    motivo: `Estoque insuficiente para ${dataEntrega}`
  }
}

function sugerirProximaData(produtoid, quantidade, rota, dataAtual) {
  datasRota = rota.getProximasDatas(dataAtual)

  for (data of datasRota) {
    estoque = calcularEstoqueDisponivel(produtoid, data)
    if (estoque >= quantidade) {
      return data
    }
  }

  return null
}

```

---

◆ **3 AJUSTES DE UI (simples, mas fundamentais)**

## **✗ O que REMOVER**

- Bloco duplicado explicando rota no carrinho  
👉 A rota já está clara no calendário
-

## O que MANTER

- Calendário de datas:
    -  Disponível (rota + estoque suficiente)
    -  Bloqueado (rota existe, mas estoque insuficiente)
- 

## O que ADICIONAR (texto exato)

### **Quando data estiver bloqueada:**

Estoque insuficiente para esta data.

Disponível: 1 UND

### **Quando sugerir nova data:**

Sugestão automática:

Entrega completa disponível em 29/01/2026

---

## Carrinho (padrão final)

Exibição obrigatória:

4 UND × 180 kg × R\$ 24,40/kg

Peso total estimado: 720 kg

Entrega prevista: 29/01/2026

---

#  ROTAS E DIAS DE ENTREGA

\*\*Schlosser PRO V7 – Logística Oficial\*\*

Plataforma: sercarne.com

Status: Documento Oficial (Fonte Única da Verdade)

Autoridade: Cássio Zaleski

 Qualquer sistema, código, desenvolvedor ou IA \*\*DEVE seguir exatamente este documento\*\*.

 Se o código divergir do que está aqui, \*\*o código está errado\*\*.

--  
## 1. OBJETIVO DESTE DOCUMENTO

Definir \*\*de forma clara, técnica e inequívoca\*\* como funcionam:

- As \*\*rotas de entrega\*\*
- Os \*\*dias de entrega\*\*
- A \*\*regra de corte (cutoff)\*\*
- A \*\*ligação entre cidade → rota → data → estoque\*\*

Este documento existe para:

- Evitar interpretação errada
- Evitar perguntas repetidas
- Permitir troca de desenvolvedor ou IA sem perda de regra

---

## ## 2. FONTE DA VERDADE

As regras de rotas e dias de entrega vêm \*\*EXCLUSIVAMENTE\*\* do Google Sheets.

### Planilha oficial:

\*\*Aba:\*\* `Rotas Dias De Entrega`

- ✗ Supabase \*\*não define rotas\*\*
- ✗ Front-end \*\*não inventa regras\*\*
- ✗ Código \*\*apenas executa o que está aqui\*\*

---

## ## 3. CONCEITO CENTRAL (RESUMO)

- \*\*A rota é definida por CIDADE\*\*
- Cada cidade pertence a \*\*um grupo logístico\*\*
- Cada grupo logístico possui:
  - Dias específicos de entrega
  - Horário limite de corte (cutoff)
- A escolha de rota + data:
  - Define o \*\*dia de estoque\*\*
  - Controla \*\*reserva e pedido\*\*

---

## ## 4. ESTRUTURA DA ABA

### `Rotas Dias De Entrega` (COLUNA POR COLUNA)

### ♦ COLUNA A — DESCRIÇÃO DO GRUPO / ROTA

\*\*Exemplo:\*\*

`2 - ROTA LOCAL (HZ, TM, IND)`

\*\*O que é:\*\*

- Nome descritivo da rota logística
- Usado para:
  - Exibição no sistema
  - Identificação humana
- Pode agrupar várias cidades

📌 Não é código técnico, é \*\*descrição operacional\*\*

---

### ### • COLUNA B — MUNICÍPIOS (CIDADE)

\*\*Exemplo:\*\*

`HORIZONTINA`

\*\*O que é:\*\*

- Nome da cidade atendida por essa rota
- É o \*\*principal filtro do sistema\*\*

📌 Regra:

- A rota é escolhida \*\*com base na cidade\*\*
- B2C e B2B usam essa mesma lógica

---

### ### • COLUNA C — DIAS DE ENTREGA

\*\*Exemplo:\*\*

`terças, quintas e sábados`

\*\*O que é:\*\*

- Dias da semana em que essa rota realiza entregas
- Pode estar em texto ou padrão definido (ex.: SEG, TER, QUA...)

📌 Regra obrigatória:

- O sistema \*\*só pode permitir datas\*\* que coincidam com esses dias
- Qualquer outro dia deve aparecer como \*\*indisponível\*\*

---

### ### • COLUNA D — CORTE ATÉ (CUT-OFF)

\*\*Exemplo:\*\*

`17:30`

\*\*O que é:\*\*

- Horário limite para aceitar pedidos
- Sempre considerado \*\*no dia anterior à entrega\*\*

### #### Exemplo prático (OBRIGATÓRIO):

- Cidade: Horizontina

- Entrega: terça e quinta
- Cutoff: 17:30

Momento do pedido	Próxima entrega válida
Segunda 16:00	Terça
Segunda 17:20	Terça
Segunda 17:31	Quinta
Segunda 19:00	Quinta

📌 Regra crítica:

- Após o horário de corte, o sistema **\*\*BLOQUEIA\*\*** a entrega mais próxima
- O usuário **\*\*não pode escolher manualmente\*\*** uma data inválida

---

### ### • COLUNA E — ROTA LOGÍSTICA POR REGIÃO / GRUPO

**\*\*Exemplo:\*\***

`2 - ROTA LOCAL (HZ, TM, IND)`

**\*\*O que é:\*\***

- Código/grupo logístico usado no ERP SYSMO
- Representa **\*\*em qual rota operacional aquela cidade está cadastrada\*\***

📌 Importante:

- Pode ser igual ou muito similar à Coluna A
- É a **\*\*ponte conceitual\*\*** entre:
  - Sistema digital (sercarne.com)
  - Operação real (SYSMO)

---

### ### • COLUNA F — CÓDIGO DA CIDADE

**\*\*Exemplo:\*\***

`430960`

**\*\*O que é:\*\***

- Código IBGE ou código interno da cidade
- Serve para:
  - Padronização
  - Interações futuras
  - Evitar ambiguidades de nome

📌 O sistema pode usar:

- Nome da cidade (Coluna B)
- OU código da cidade (Coluna F)  
desde que siga a mesma lógica

---

## ## 5. REGRA DE FUNCIONAMENTO NO SISTEMA

### ### 5.1 Fluxo B2C (Cliente Geral – sem login)

1. Usuário acessa `sercarne.com/catalog`
2. Seleciona a \*\*cidade\*\*
3. Sistema:
  - Busca a cidade na Coluna B
  - Identifica a rota correspondente
  - Aplica:
    - Dias de entrega (Coluna C)
    - Cutoff (Coluna D)
4. Usuário só pode:
  - Ver
  - Selecionar

\*\*datas válidas\*\*

---

### ### 5.2 Fluxo B2B (Cliente / Vendedor logado)

1. Usuário faz login
2. Sistema identifica:
  - Cidade do cliente
3. Aplica \*\*exatamente as mesmas regras\*\*:
  - Rota por cidade
  - Dias de entrega
  - Cutoff
4. Vendedor \*\*não pode forçar\*\* data inválida

---

👉 Hierarquia \*\*não quebra regra logística\*\*

---

## ## 6. RELAÇÃO COM ESTOQUE

Ao selecionar:

- \*\*Cidade\*\*
- \*\*Rota\*\*
- \*\*Data de entrega\*\*

O sistema define uma chave lógica de estoque:

# # 🚚 ROTAS E DIAS DE ENTREGA

## # 🚛 ROTAS E DIAS DE ENTREGA

\*\*Schlosser PRO V7 – Logística Oficial\*\*

Plataforma: [sercarne.com](http://sercarne.com)

Status: Documento Oficial (Fonte Única da Verdade)

Autoridade: Cássio Zaleski

⚠ Qualquer sistema, código, desenvolvedor ou IA \*\*DEVE seguir exatamente este documento\*\*.

⚠ Se o código divergir do que está aqui, \*\*o código está errado\*\*.

---

## ## 1. OBJETIVO DESTE DOCUMENTO

Definir \*\*de forma clara, técnica e inequívoca\*\* como funcionam:

- As \*\*rotas de entrega\*\*
- Os \*\*dias de entrega\*\*
- A \*\*regra de corte (cutoff)\*\*
- A \*\*ligação entre cidade → rota → data → estoque\*\*

Este documento existe para:

- Evitar interpretação errada
- Evitar perguntas repetidas
- Permitir troca de desenvolvedor ou IA sem perda de regra

---

## ## 2. FONTE DA VERDADE

As regras de rotas e dias de entrega vêm \*\*EXCLUSIVAMENTE\*\* do Google Sheets.

### Planilha oficial:

\*\*Aba:\*\* `Rotas Dias De Entrega`

✗ Supabase \*\*não define rotas\*\*

✗ Front-end \*\*não inventa regras\*\*

✗ Código \*\*apenas executa o que está aqui\*\*

---

## ## 3. CONCEITO CENTRAL (RESUMO)

- \*\*A rota é definida por CIDADE\*\*
- Cada cidade pertence a \*\*um grupo logístico\*\*
- Cada grupo logístico possui:
  - Dias específicos de entrega
  - Horário limite de corte (cutoff)

- A escolha de rota + data:
  - Define o \*\*dia de estoque\*\*
  - Controla \*\*reserva e pedido\*\*

---

#### ## 4. ESTRUTURA DA ABA

##### ### `Rotas Dias De Entrega` (COLUNA POR COLUNA)

###### ### ♦ COLUNA A — DESCRIÇÃO DO GRUPO / ROTA

\*\*Exemplo:\*\*

`2 - ROTA LOCAL (HZ, TM, IND)`

\*\*O que é:\*\*

- Nome descritivo da rota logística
- Usado para:
  - Exibição no sistema
  - Identificação humana
- Pode agrupar várias cidades

💡 Não é código técnico, é \*\*descrição operacional\*\*

---

###### ### ♦ COLUNA B — MUNICÍPIOS (CIDADE)

\*\*Exemplo:\*\*

`HORIZONTINA`

\*\*O que é:\*\*

- Nome da cidade atendida por essa rota
- É o \*\*principal filtro do sistema\*\*

💡 Regra:

- A rota é escolhida \*\*com base na cidade\*\*
- B2C e B2B usam essa mesma lógica

---

###### ### ♦ COLUNA C — DIAS DE ENTREGA

\*\*Exemplo:\*\*

`terças, quintas e sábados`

\*\*O que é:\*\*

- Dias da semana em que essa rota realiza entregas
- Pode estar em texto ou padrão definido (ex.: SEG, TER, QUA...)

💡 Regra obrigatória:

- O sistema \*\*só pode permitir datas\*\* que coincidam com esses dias

- Qualquer outro dia deve aparecer como \*\*indisponível\*\*

---

### ### ♦ COLUNA D — CORTE ATÉ (CUT-OFF)

\*\*Exemplo:\*\*

`17:30`

\*\*O que é:\*\*

- Horário limite para aceitar pedidos
- Sempre considerado \*\*no dia anterior à entrega\*\*

### #### Exemplo prático (OBRIGATÓRIO):

- Cidade: Horizontina
- Entrega: terça e quinta
- Cutoff: 17:30

Momento do pedido	Próxima entrega válida
----- -----	
Segunda 16:00   Terça	
Segunda 17:20   Terça	
Segunda 17:31   Quinta	
Segunda 19:00   Quinta	

#### 📌 Regra crítica:

- Após o horário de corte, o sistema \*\*BLOQUEIA\*\* a entrega mais próxima
- O usuário \*\*não pode escolher manualmente\*\* uma data inválida

---

### ### ♦ COLUNA E — ROTA LOGÍSTICA POR REGIÃO / GRUPO

\*\*Exemplo:\*\*

`2 - ROTA LOCAL (HZ, TM, IND)`

\*\*O que é:\*\*

- Código/grupo logístico usado no ERP SYSMO
- Representa \*\*em qual rota operacional aquela cidade está cadastrada\*\*

#### 📌 Importante:

- Pode ser igual ou muito similar à Coluna A
- É a \*\*ponte conceitual\*\* entre:
  - Sistema digital (sercarne.com)
  - Operação real (SYSMO)

---

### ### ♦ COLUNA F — CÓDIGO DA CIDADE

\*\*Exemplo:\*\*

`430960`

**\*\*O que é:\*\***

- Código IBGE ou código interno da cidade
- Serve para:
  - Padronização
  - Interações futuras
  - Evitar ambiguidades de nome

💡 O sistema pode usar:

- Nome da cidade (Coluna B)
  - OU código da cidade (Coluna F)
- desde que siga a mesma lógica

---

## ## 5. REGRA DE FUNCIONAMENTO NO SISTEMA

### ### 5.1 Fluxo B2C (Cliente Geral – sem login)

1. Usuário acessa `sercarne.com/catalog`
2. Seleciona a **\*\*cidade\*\***
3. Sistema:
  - Busca a cidade na Coluna B
  - Identifica a rota correspondente
  - Aplica:
    - Dias de entrega (Coluna C)
    - Cutoff (Coluna D)
4. Usuário só pode:
  - Ver
  - Selecionar**\*\*datas válidas\*\***

---

### ### 5.2 Fluxo B2B (Cliente / Vendedor logado)

1. Usuário faz login
2. Sistema identifica:
  - Cidade do cliente
3. Aplica **\*\*exatamente as mesmas regras\*\***:
  - Rota por cidade
  - Dias de entrega
  - Cutoff
4. Vendedor **\*\*não pode forçar\*\*** data inválida

💡 Hierarquia **\*\*não quebra regra logística\*\***

---

## ## 6. RELAÇÃO COM ESTOQUE

Ao selecionar:

- \*\*Cidade\*\*
- \*\*Rota\*\*
- \*\*Data de entrega\*\*

O sistema define uma chave lógica de estoque:

**Essa chave é usada para:**

Essa chave é usada para:

- Ver estoque disponível
- Criar reserva
- Converter em pedido

✖ Regra obrigatória:

- \*\*Reserva sempre antes do pedido\*\*
- Reserva expira em 2 horas
- Estoque retorna se não confirmar

---

## ## 7. REGRAS QUE NÃO PODEM SER QUEBRADAS

- ✖ Não permitir entrega fora dos dias definidos
- ✖ Não ignorar horário de corte
- ✖ Não permitir escolha manual de data inválida
- ✖ Não criar lógica diferente da planilha
- ✖ Não duplicar regra no código sem referência a este documento

---

## ## 8. CHECKLIST DE VALIDAÇÃO (QA)

- [ ] Cidade filtra rota corretamente
- [ ] Dias inválidos não aparecem
- [ ] Cutoff bloqueia entrega corretamente
- [ ] Após cutoff, próxima entrega é pulada
- [ ] Reserva respeita data e rota
- [ ] Estoque retorna após expiração

---

## ## 9. GOVERNANÇA

- Este documento faz parte do \*\*Modelo de Negócio Schlosser PRO V7\*\*
- Alterações:
  - Somente pelo autor
  - Nova versão obrigatória (V7.1, V8...)

---

## ## 10. CHANGELOG

Versão	Data	Alteração
--------	------	-----------

-----	-----	-----
-------	-------	-------

V1   25/01/2026   Criação do documento oficial de Rotas e Dias de Entrega
---------------------------------------------------------------------------

---

📍 \*\*Conclusão final:\*\*

Se alguém “não entendeu a rota”, o problema \*\*não é regra\*\*, é \*\*leitura\*\*.  
Este documento elimina ambiguidade.

**data\_dictionary 25/01**

# DATA\_DICTIONARY — Schlosser PRO V7

SERCARNE.COM | 2026

Versão: V7.1

Status: Documento Técnico Oficial (Fonte Única da Verdade de Estruturas de Dados)

Autoridade máxima: Cássio Zaleski

Este documento descreve:

- \*\*Google Sheets (Fonte Primária / Catálogo & Regras)\*\*
- \*\*Supabase (Operação / Usuários, Reservas, Pedidos, Entradas de Estoque)\*\*
- Convenções e regras de parsing/validação

---

## # 1) GOOGLE SHEETS (FONTE PRIMÁRIA)

### ## 1.1 Planilha Oficial (Catálogo e Precificação)

Nome: \*\*Precificação Mix Frigo Schlosser / Zaleski 2026 (IA APP)\*\*

Aba oficial de catálogo: \*\*2026 Base Catalogo Precifica V2\*\*

#### ### Regra de filtro obrigatório (cadastro válido)

- \*\*Somente produtos com código  $\geq 400000$  podem ser carregados no sistema.
- O código está na \*\*COLUNA D\*\*.

> Motivo: Frigorífico Schlosser opera como \*\*Filial 4\*\* do Grupo Zaleski; códigos menores são cadastros inválidos no contexto do app.

---

### ## 1.2 Dicionário de Colunas — Aba "2026 Base Catalogo Precifica V2"

> \*\*Importante:\*\* o app pode carregar várias colunas para exibição, mas \*\*as colunas abaixo são “críticas”\*\* (não podem mudar sem atualizar o manual e o código).

#### ### Identidade do Produto

- \*\*COLUNA D — `codigo\_produto`\*\*
  - Tipo: number (inteiro)
  - Regra: \*\* $\geq 400000$ \*\* para ser carregado
  - Uso: chave primária do catálogo dentro do app

#### ### Descrições (Regra oficial de fallback)

- \*\*COLUNA AK — `descricao\_nf` (principal / comercial / NF)\*\*
  - Tipo: texto
  - Uso: título principal do produto no card e em pedido
- \*\*COLUNA AL — `descricao\_tecnica`\*\*
  - Tipo: texto
  - Uso: exibir \*\*abaixo\*\*, menor e cinza (informativo)
- \*\*COLUNA E — `descricao\_fallback`\*\*
  - Tipo: texto

- Uso: contingência quando AK/AL estiverem vazias ou com `#N/A`

**\*\*Regra obrigatória de exibição:\*\***

- 1) Exibir AK (se válida)
- 2) Exibir AL (se válida) logo abaixo
- 3) Se AK ou AL estiverem vazias / `#N/A`, usar E como fallback (nunca exibir `#N/A`)

---

#### #### Unidade Comercial e Peso Médio

- **COLUNA AC — `tipo\_venda`**
  - Tipo: texto
  - Valores esperados: `UND` ou `CX`
  - Regra:
    - `UND`: 1 unidade comercial
    - `CX`: 1 unidade lógica (caixa)
- **COLUNA I — `peso\_medio\_kg`**
  - Tipo: number
  - Uso: **estimativa informativa** (não é o peso final)

**\*\*Regra oficial de CX:\*\***

- Se `tipo\_venda == CX`, então:
  - **peso\_medio\_kg assumido = 10kg** (padrão)
  - variação permitida: **±30%**
  - permanece estimativo (valor final ajusta na expedição)

---

#### ### Tabelas de Preço (Preço por KG)

A base de preço do sistema é **por KG**.

#### #### Preço visível padrão (todos veem antes do login)

- **COLUNA AA (Coluna 27) — `preco\_tab3\_kg`**
  - Nome comum: TAB3 / PV.3
  - Uso: preço público / base

#### #### Outras TABs (uso interno / conforme perfil e volume)

Pelo print do cabeçalho da planilha:

- **COLUNA V (Coluna 22) — `preco\_tab0\_kg`** (TAB0 / PV.0)
- **COLUNA W (Coluna 23) — `preco\_tab5\_kg`** (TAB5 / PV.5) \*(uso por vendedor/supervisor/admin se habilitado)\*
- **COLUNA X (Coluna 24) — `preco\_tab4\_kg`** (TAB4 / PV.4)
- **COLUNA Y (Coluna 25) — `preco\_tab1\_kg`** (TAB1 / PV.1)
- **COLUNA Z (Coluna 26) — `preco\_tab2\_kg`** (TAB2 / PV.2)
- **COLUNA AA (Coluna 27) — `preco\_tab3\_kg`** (TAB3 / PV.3)

> Nota: Se existir divergência futura de colunas, \*\*a regra é o nome da coluna na planilha\*\* e este dicionário deve ser atualizado.

---

### ### Imagens (Fallback oficial)

Ordem exata de tentativa:

- 1) \*\*COLUNA AE — `img\_principal`\*\*
- 2) \*\*COLUNA AF — `img\_alt\_1`\*\*
- 3) \*\*COLUNA AG — `img\_alt\_2`\*\*
- 4) \*\*COLUNA AH — `img\_marca`\*\*
- 5) fallback final: \*\*imagem padrão Schlosser\*\* (asset local do app)

\*\*Regras obrigatórias:\*\*

- Não buscar imagens externas
- Não mudar a ordem do fallback
- Se usar AH, mostrar badge: \*\*"Imagen da Marca"\*\*

---

## ## 1.3 Planilha Oficial (Rotas e Logística)

Aba oficial: \*\*Rotas Dias De Entrega\*\*

### ### Dicionário de Colunas — Aba "Rotas Dias De Entrega"

- \*\*COLUNA A — `descricao\_grupo\_rota`\*\*
  - Ex.: `2 - ROTA LOCAL (HZ, TM, IND)`
- \*\*COLUNA B — `municipio`\*\*
  - Ex.: `HORIZONTINA`
  - Regra: cidade é a chave de busca do roteamento
- \*\*COLUNA C — `dias\_entrega`\*\*
  - Ex.: `terças, quintas e sábados`
  - Regra: sistema converte para enum de dias da semana
- \*\*COLUNA D — `corte\_ate`\*\*
  - Ex.: `17:30h`
  - Regra: horário limite para aceitar pedido para a \*\*entrega do próximo dia válido\*\*
- \*\*COLUNA E — `rota\_logistica\_sysmo`\*\*
  - Ex.: `GRUPO DE CLIENTE: 2 - 2 - ROTA LOCAL (HZ, TM, IND)`
  - Uso: auditoria e referência ERP (não muda regra de dias)
- \*\*COLUNA F — `codigo\_cidade`\*\*
  - Ex.: `430960`

\*\*Regras obrigatórias:\*\*

- Rota é por cidade (coluna B)
- Datas válidas apenas nos dias da coluna C
- Corte da coluna D bloqueia a entrega “mais próxima” se estiver fora do horário

---

## # 2) SUPABASE (BANCO OPERACIONAL)

### ## 2.1 Tabela: `usuarios`

Objetivo: autenticação simples + roteamento de telas + governança por hierarquia.

### Campos (confirmados por print)

- `usuario` (text)
  - Nome interno (ex.: `Cegil`, `Cassio\_Zaleski`)
- `login` (text)
  - Identificador de login (telefone ou user)
- `senha\_hash` (text)
  - \*\*Hoje está como texto no print\*\* (ex.: `eve9`)
  - Recomendação: migrar para hash real no futuro, mas não é obrigatório no V7
- `tipo\_de\_Usuario` (text)
  - Ex.: `Cliente/B2B`, `Representante/PJ`, `Admin`
- `ativo` (bool)
  - TRUE/FALSE
- `app\_login` (text)
  - Rota pós-login (ex.: `/cliente\_B2B`, `/vendedor`, `/admin`)
- `created\_at` (date/timestamp)
- `nivel` (int2)
  - Regra de hierarquia:
    - 1 a 3: clientes
    - 4 a 7: vendedor/supervisor
    - 8 a 10: gestão/admin
- `tabelas` (text) \*(nome pode variar: "Tab..." no print)\*
  - Ex.: `1,0,4` ou `1,0,4,5,2`
  - Uso: enumera quais TABs esse usuário pode visualizar/usar (interno)

### Regras de autenticação (negócio)

- Login por `login` e/ou `usuario`
- Case-insensitive
- Se `ativo=false`, bloquear acesso
- Após autenticar:
  - redirecionar usando `app\_login`
  - carregar permissões via `nivel` e `tabelas`

---

### ## 2.2 Tabela: `reservas`

Objetivo: travar estoque por 2h ao adicionar ao carrinho.

### Campos mínimos obrigatórios (recomendado para V7)

- `id` (uuid, pk)
- `created\_at` (timestamp)
- `expires\_at` (timestamp) ← \*\*crítico\*\* (created\_at + 2h)
- `status` (text)

- valores: `RESERVADO`, `EXPIRADO`, `CONVERTIDO\_PEDIDO`, `CANCELADO`
- `cliente\_login` (text) ou `cliente\_id` (fk para usuarios) ← escolha 1 padrão e mantenha
- `digitado\_por\_login` (text) ou `digitado\_por\_id` (fk usuarios)
- quem digitou (cliente ou vendedor)
- `codigo\_produto` (int) ← do Sheets (col D)
- `quantidade\_und` (int) ← unidade comercial (UND/CX)
- `tipo\_venda` (text) ← `UND`/`CX` (col AC)
- `peso\_medio\_kg` (numeric) ← (col I ou 10kg p/ CX)
- `peso\_total\_estimado\_kg` (numeric) ← quantidade \* peso\_medio\_kg
- `preco\_kg\_aplicado` (numeric) ← preço da TAB aplicada (por KG)
- `tab\_aplicada` (text) ← `TAB0|TAB1|TAB3|TAB4|TAB5...` (interno)
- `valor\_total\_estimado` (numeric) ← peso\_total\_estimado\_kg \* preco\_kg\_aplicado

### ### Regras obrigatórias

- Reserva nasce ao “Adicionar ao carrinho”
- Expira em 2h se não virar pedido
- Ao expirar: deve liberar disponibilidade de estoque (lógico)

> \*\*TBD (depende do seu schema atual):\*\* se você já tem colunas diferentes, mantenha a lógica acima e mapeie 1:1.

---

### ## 2.3 Tabela: `pedidos`

Objetivo: registrar pedido confirmado (reserva convertida).

#### ### Campos mínimos obrigatórios (recomendado para V7)

- `id` (uuid, pk)
- `created\_at` (timestamp)
- `status` (text)
  - valores: `ABERTO`, `CONFIRMADO`, `EM\_SEPARACAO`, `FATURADO`, `ENTREGUE`, `CANCELADO`
- `cliente\_login` (text) ou `cliente\_id` (fk usuarios)
- `digitado\_por\_login` (text) ou `digitado\_por\_id` (fk usuarios)
- `supervisor\_login` (text) ou `supervisor\_id` (fk usuarios) \*(se aplicável por hierarquia)\*
- `rota\_id` (text) ou `rota\_nome` (text)
- `cidade` (text)
- `data\_entrega\_prevista` (date)
- `cutoff\_aplicado` (text) ← ex.: `17:30`
- `itens` (jsonb)
  - lista de itens com: codigo\_produto, quantidade\_und, tipo\_venda, peso\_medio\_kg, peso\_total\_estimado\_kg, preco\_kg\_aplicado, tab\_aplicada, total\_estimado
- `valor\_total\_estimado` (numeric)

#### ### Regras obrigatórias

- Pedido \*\*só existe\*\* se originado de reservas válidas
- Ao confirmar pedido:
  - reservas → `CONVERTIDO\_PEDIDO`

- pedido grava TAB aplicada internamente
- cliente não vê “TAB”, vê “benefício/desconto”

---

#### ## 2.4 Tabela: `entradas\_estoque`

Objetivo: registrar entrada e disponibilidade por data/rota (dependendo do modelo de estoque que você adotou).

##### ### Campos mínimos obrigatórios (recomendado para V7)

- `id` (uuid, pk)
- `created\_at` (timestamp)
- `codigo\_produto` (int)
- `data\_disponivel` (date) ← \*\*crítico\*\* para estoque por entrega
- `quantidade\_total\_und` (int)
- `quantidade\_reservada\_und` (int) \*(opcional se calculado via reservas)\*
- `quantidade\_disponivel\_und` (int) \*(pode ser calculado, mas recomendado persistir por performance)\*
- `rota\_nome` (text) \*(opcional, se estoque varia por rota)\*
- `observacao` (text)

##### ### Regra obrigatória

- O estoque deve suportar “timeline por data de entrega”
- Reserva deve diminuir disponibilidade \*\*logicamente\*\* (e retornar ao expirar)

> \*\*TBD:\*\* se você está usando outro esquema (ex.: uma tabela por produto com timeline), esse dicionário deve ser ajustado.

---

## # 3) CONVENÇÕES DE PARSING / NORMALIZAÇÃO

### ## 3.1 Cidade (Rotas)

- Normalizar `municipio` e entrada do cliente:
  - trim
  - uppercase
  - remover acentos (recomendado)
- Busca na coluna B deve ser case-insensitive

### ## 3.2 Dias de entrega

- Converter texto “terças, quintas e sábados” em enum:
  - TER, QUI, SAB
- Respeitar `corte\_ate` da coluna D

### ## 3.3 Produtos

- Não carregar código < 400000
- Nunca exibir `#N/A` em descrição
- Imagem sempre seguindo AE→AF→AG→AH→fallback

---

## # 4) REGRAS CRÍTICAS DE PREÇO (RESUMO EXECUTÁVEL)

### ## 4.1 Preço base visível

- Sempre mostrar inicialmente: `TAB3` (COLUNA AA) como “preço público”

### ## 4.2 Preço real aplicado (interno)

- Aplicar TAB conforme:

- perfil do usuário (B2B / vendedor / etc)
- \*\*volume total de UND do carrinho\*\* (soma de unidades)
- O preço é por KG:
  - `valor\_item\_estimado = (quantidade\_und \* peso\_medio\_kg) \* preco\_kg\_aplicado`

### ## 4.3 Exibição do “benefício” (cliente)

- Cliente NÃO vê “TAB”
- Cliente pode ver:
  - “Benefício do seu cadastro + volume: X%”
  - (onde X% compara preço aplicado vs preço base TAB3)

---

## # 5) CHANGELOG

- V7.1 — Inclusão formal de:
  - Colunas de TABs (V..AA)
  - Regra de preço por KG + peso estimado
  - Estrutura recomendada para reservas/pedidos/entradas\_estoque

# V7.4 Script Teste

# SCRIPT DE TESTE MANUAL —

## SCHLOSSER PRO V7.2

**Ambiente:** Produção (sercarne.com) ou Preview (Hostinger)

**Data:** //\_\_\_\_\_

**Executor:** \_\_\_\_\_

**Versão do sistema:** \_\_\_\_\_ (commit/tag se houver)

### Pré-requisitos (antes de começar)

- Ter 3 usuários de teste no Supabase:
  - **Cliente** nível 2 (B2B)
  - **Vendedor** nível 5 (Lissandro)
  - **Admin** nível 10
- Ter ao menos 1 produto válido no Sheets com:
  - **Código ≥ 400000**
  - Peso médio na **Coluna I**
  - Preços preenchidos: TAB3 (AA) e TAB0 (coluna da TAB0 no teu mapeamento)
- Produto referência recomendado: **400010** (Meia Rês)
  - Peso médio: 180 kg (exemplo)

---

### TESTE 1 — Abertura do site / Catálogo Público (B2C)

**Objetivo:** garantir que sercarne.com abre no catálogo e está travado em TAB3.

1. Abrir: <https://sercarne.com/>

**Esperado:** abrir direto no catálogo (sem exigir login).

- PASS/FAIL: \_\_\_\_
2. Confirmar que existe área “Faça seu login” no topo (inline).

- PASS/FAIL: \_\_\_\_
3. Buscar o produto **400010**.

**Esperado no card do produto:**

- Preço exibido em **R\$/KG**
  - Preço é o da **TAB3 (Coluna AA)**
  - Exibe **Peso médio** (Coluna I)
  - PASS/FAIL: \_\_\_\_
4. Validar filtro de código: procurar qualquer produto com código < 400000.

**Esperado:** não aparece nenhum.

- PASS/FAIL: \_\_\_\_
- 

## TESTE 2 — Descrição e Fallback (AK/AL/E)

**Objetivo:** garantir regra de descrição.

1. Abrir um produto qualquer.

**Esperado:**

- Descrição principal = **AK** (em destaque)
  - Descrição técnica = **AL** (menor/cinza)
  - Se AK ou AL vazia/#N/A → usa **E**
  - Nunca mostrar “#N/A”
  - PASS/FAIL: \_\_\_\_
-

## TESTE 3 — Imagens e Badge (AE→AF→AG→AH→Padrão)

**Objetivo:** garantir regra de imagem.

1. Encontrar um produto que tenha AE preenchido.

**Esperado:** carrega AE.

- PASS/FAIL: \_\_\_\_\_

2. Encontrar um produto sem AE, mas com AF ou AG.

**Esperado:** carrega AF (ou AG se AF vazio).

- PASS/FAIL: \_\_\_\_\_

3. Encontrar um produto que só tenha AH (imagem da marca).

**Esperado:** carrega AH e mostra badge “**Imagen da Marca**”.

- PASS/FAIL: \_\_\_\_\_

4. Produto sem nenhuma imagem.

**Esperado:** imagem padrão Schlosser.

- PASS/FAIL: \_\_\_\_\_

---

## TESTE 4 — Login inline (sem trocar de URL)

**Objetivo:** login aplica perfil sem sair do catálogo.

1. No topo do catálogo, fazer login como **Cliente (nível 2)**.

**Esperado:**

- continua na mesma tela (catálogo)
- usuário aparece como logado (nome/indicador)
- sem tela branca
- PASS/FAIL: \_\_\_\_\_

2. Fazer logout e login como **Vendedor (nível 5 - Lissandro)**.

**Esperado:** idem (permanece no catálogo).

- PASS/FAIL: \_\_\_\_
- 

## TESTE 5 — Regra de Preço B2B por Volume (UND total)

**Objetivo:** validar tabela aplicada por volume (sem exibir tabela ao cliente).

**Produto referência:** 400010

**Peso médio esperado:** 180 kg (ou valor real da Coluna I)

### Caso A — 1 UND (TAB1)

1. Logado como **Cliente B2B** (nível 2) ou **Vendedor**.
2. Adicionar **1 UND** do 400010 no carrinho.
3. Abrir carrinho.

**Esperado (cálculo):**

- Preço base visível (referência) é TAB3 (R\$/KG)
- Preço aplicado (interno) é TAB1 (R\$/KG)

Exibição do carrinho deve mostrar formato:

1 UND × 180 KG × R\$ X,XX/KG = R\$ Y.YYY,YY

- 
- Cliente **não vê TAB1**
- Se logado (B2B), deve mostrar:
  - “Benefício do seu cadastro + volume: \_\_\_\_%”
- PASS/FAIL: \_\_\_\_

### Caso B — 2 UND (TAB0)

1. Ajustar carrinho para **2 UND** do 400010.

2. Abrir carrinho.

**Esperado (cálculo):**

- Tabela aplicada: **TAB0**

Exibição obrigatória:

$$2 \text{ UND} \times 180 \text{ KG} \times \text{R\$ } 24,40/\text{KG} = \text{R\$ } 8.784,00$$

Peso total estimado: 360 KG

Benefício do seu cadastro + volume: 23,1%

- 

(Os valores mudam conforme Sheets, mas a fórmula e estrutura são obrigatórias.)

- Cliente não vê “TAB0”
- Vendedor/Admin pode ver tabela aplicada (interno)
- PASS/FAIL: \_\_\_\_\_

**Caso C — 10 UND (TAB4)**

1. Ajustar carrinho para **10 UND** do mesmo produto.
2. Abrir carrinho.

**Esperado:**

- Tabela aplicada: **TAB4**
- Exibe benefício percentual (vs TAB3)
- PASS/FAIL: \_\_\_\_\_

---

 **TESTE 6 — Reserva automática (criar ao adicionar no carrinho)**

**Objetivo:** garantir que reserva nasce ao adicionar item.

1. Com usuário logado (qualquer), adicionar 1 item ao carrinho.

2. Verificar no Supabase (tabela `reservas`) se entrou registro.

**Esperado:**

- existe linha com:

- produto
- quantidade
- usuário
- `expires_at` ~ agora + 2h

- PASS/FAIL: \_\_\_\_\_
- 



## TESTE 7 — Expiração da reserva (2h)

**Objetivo:** garantir retorno automático ao estoque.

**⚠️** Para teste rápido, use TTL reduzido em ambiente de teste (ex.: 2 minutos) ou ajuste manual no DB só para validar a rotina.

1. Criar uma reserva.
2. Forçar expiração (TTL curto ou setar `expires_at` no passado).
3. Rodar rotina/refresh.

**Esperado:**

- reserva muda status para expirada (ou é removida conforme regra)
  - estoque volta a ficar disponível
  - PASS/FAIL: \_\_\_\_\_
-

## TESTE 8 — Pedido só nasce de reserva válida

**Objetivo:** impedir pedido sem reserva.

1. Tentar confirmar pedido sem ter reserva ativa (simular removendo reserva no DB antes do checkout).
2. Confirmar checkout.

**Esperado:**

- sistema bloqueia com mensagem clara:

“Reserva expirada. Refaça o carrinho.”

- PASS/FAIL: \_\_\_\_\_
- 

## TESTE 9 — Hierarquia (Quem enxerga o quê)

**Objetivo:** blindar acessos.

### Cliente nível 2

1. Logar como cliente.
2. Tentar acessar dashboard/admin/rotas/lista de clientes.

**Esperado:** bloqueado.

- PASS/FAIL: \_\_\_\_\_

### Vendedor nível 5

1. Logar como vendedor.
2. Ver lista de clientes.

**Esperado:** ver **somente seus clientes**, nunca todos.

- PASS/FAIL: \_\_\_\_

## **Admin nível 10**

1. Logar como admin.
2. Ver tudo: clientes, pedidos, reservas.

**Esperado:** acesso total.

- PASS/FAIL: \_\_\_\_
- 

## **TESTE 10 — Regressões (tela branca / console)**

**Objetivo:** garantir estabilidade.

1. Com DevTools aberto (Console):
  - navegar catálogo
  - fazer login
  - adicionar carrinho
  - abrir carrinho
  - voltar
2. Observar console.

**Esperado:**

- zero erros críticos (exceptions)
  - nenhum “white screen”
  - PASS/FAIL: \_\_\_\_
-



# RESULTADO FINAL

- GO LIVE? (SIM / NÃO): \_\_\_\_\_

- Falhas críticas encontradas:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

- Recomendações imediatas:

1. \_\_\_\_\_
  2. \_\_\_\_\_
-

# V7.4 PLANO DE IMPLEMENTACAO



# PLANO DE IMPLEMENTAÇÃO

## SCHLOSSER PRO — [sercarne.com](http://sercarne.com)

**Base:** Manual do Modelo de Negócio Schlosser PRO V7.2 + REQUISITOS.md

**Objetivo:** Colocar a plataforma 100% funcional, sem refazer o que já funciona.

---



## FASE 0 — ALINHAMENTO (OBRIGATÓRIA)

**Antes de codar qualquer coisa**

1. Ler integralmente:

- [MODELO\\_DE\\_NEGOCIO\\_SCHLOSSER\\_PRO\\_V7.2.md](#)
- [REQUISITOS.md](#)

2. Concordar explicitamente que:

- Google Sheets = fonte da verdade
- UND ≠ KG
- Preço sempre é R\$/KG
- Cliente nunca vê tabela

➡ Se houver dúvida → **perguntar antes de implementar.**

---



## FASE 1 — TRAVA DA LÓGICA CENTRAL (CRÍTICA)

(Prioridade máxima — sem isso nada adianta)

### 1.1 Centralizar regras de negócio

Criar (ou ajustar) arquivo único:

[src/domain/schlosserRules.js](#)

Funções obrigatórias:

- `getTabelaAplicada(totalUND, perfilUsuario)`
- `getPrecoKg(produto, tabela)`
- `calcularValorEstimado(und, pesoMedio, precoKg)`
- `calcularBeneficioPercentual(precoTab3, precoAplicado)`

 Nenhuma page pode recalcular preço fora desse arquivo.

---

## 1.2 Corrigir lógica UND × KG × PREÇO

Regra imutável:

`VALOR = UND × PESO_MÉDIO × PREÇO_POR_KG`

- UND vem do carrinho
  - Peso médio vem da **COLUNA I**
  - Preço vem da tabela correta
  - Peso é estimativo
- 



## FASE 2 — CATÁLOGO PÚBLICO (B2C)

**Entrada principal do site**

### 2.1 URL raiz

`https://sercarne.com/`

- Abrir direto no catálogo
- Sem exigir login

## 2.2 Regras obrigatórias

- Filtrar produtos:
  - Código ≥ 400000 (COLUNA D)
- Preço exibido:
  - TAB3 (COLUNA AA)
  - Formato: R\$ XX,XX / KG
- Mostrar:
  - Peso médio por UND

 Não mostrar desconto

 Não mostrar tabela

 Não calcular total sem UND

---

## FASE 3 — LOGIN E PERFIS

Sem trocar URL base

### 3.1 Login inline

- Campo de login no topo do catálogo
- Login:
  - Autentica
  - Recarrega a mesma tela
  - Aplica regras conforme nível

### 3.2 Permissões

- Cliente (1–3): só seus pedidos
- Vendedor (4–7): seus clientes

- Admin (8–10): tudo
- 



## FASE 4 — PRECIFICAÇÃO B2B (CRÍTICA)

### 4.1 Regra por volume (UND total do carrinho)

#### UND Tabela

1 TAB1

2–9 TAB0

≥10 TAB4

### 4.2 Exibição ao cliente

- Preço base continua TAB3

Cliente vê:

Benefício do seu cadastro + volume: XX, X%

•

✗ Nunca mostrar “TAB0”, “TAB1”, etc.

---



## FASE 5 — CARRINHO (DETALHAMENTO)

### Aqui é onde mais dá bug se não seguir o manual

Exibição obrigatória:

2 UND × 180 KG × R\$ 24,40/KG

Peso total estimado: 360 KG

Valor total estimado: R\$ 8.784,00

Benefício do seu cadastro + volume: 23,1%

♥ Texto claro:

“Valores estimados. Ajuste final conforme peso real na expedição.”

---

## FASE 6 — RESERVA E ESTOQUE

### 6.1 Reserva

- Criada ao adicionar ao carrinho
- TTL: 2 horas

### 6.2 Expiração

- Job automático:
  - Expira reserva
  - Devolve estoque

 Não permitir pedido sem reserva

---

## FASE 7 — PEDIDO

Pedido confirmado deve:

- Converter reserva → pedido
- Abater estoque
- Registrar:
  - Cliente final
  - Usuário que digitou
  - Supervisor indireto
  - Tabela aplicada (interno)
  - Valor estimado
  - Peso estimado

---

## FASE 8 — TESTES DE ACEITE (QA)

Checklist final:

- Catálogo abre sem login
- Produto < 400000 não aparece
- Preço público = TAB3
- UND × KG × preço/KG correto
- Login não muda URL
- Cliente vê desconto %, não tabela
- Reserva expira em 2h
- Hierarquia respeitada

---

## FASE 9 — GO LIVE

Somente após:

- Todos os itens acima OK
- Validação com pedido real de teste
- Aprovação do autor do modelo

---

## MENSAGEM FINAL AO DEV (IMPORTANTE)

“Não adapte o negócio ao código.  
O código se adapta ao Modelo Schlosser PRO.”

## V7.4 REQUISITOS



# REQUISITOS.md

## SCHLOSSER PRO — Plataforma sercarne.com

**Versão:** V1.0 (alinhada ao Manual V7.2)

**Autoridade do negócio:** Cássio Zaleski

**Status:** Obrigatório / Não interpretável

**⚠ Este arquivo NÃO é sugestivo.**

Qualquer código, tela, cálculo ou automação **DEVE seguir exatamente o que está aqui.**

Em caso de conflito técnico × regra → **a regra vence.**

---

## 1 VISÃO GERAL DO SISTEMA

O Schlosser PRO é uma plataforma de vendas B2B e B2C para frigorífico, com as seguintes premissas:

- O cliente **compra por UND ou CX**
  - O valor é **calculado por KG estimado**
  - O preço **sempre vem do Google Sheets**
  - O peso real **só é confirmado na expedição**
  - O sistema **não decide regra comercial**, apenas executa
- 

## 2 STACK OBRIGATÓRIA

### Fonte da Verdade

- **Google Sheets**
  - Planilha: [2026 Base Catalogo Precifica V2](#)
  - Nenhuma outra base pode recalcular preço

## Banco Operacional

- **Supabase**

- Tabelas obrigatórias:
  - `usuarios`
  - `reservas`
  - `pedidos`
  - `entradas_estoque`

## Front-end / App

- **Hostinger Horizons** (atual)
  - Pode ser migrado, desde que respeite este arquivo
- 

## 3 TELA INICIAL (REGRA NOVA OBRIGATÓRIA)

### URL raiz

<https://sercarne.com/>

### Comportamento obrigatório

- Abrir direto no catálogo público (B2C)
- Não exigir login

### Topo da tela

Exibir:

“Já é cliente, vendedor ou parceiro? Faça seu login”

💡 O login **não redireciona para outra URL**, apenas:

- Autentica
  - Recarrega a mesma tela
  - Aplica regras conforme o perfil do usuário
- 

## 4 CATÁLOGO (SEM LOGIN — B2C)

### Filtro obrigatório de produtos

- Somente produtos com código  $\geq 400000$
- Coluna de referência: **COLUNA D**

### Preço exibido

- TAB3
- Origem: **COLUNA AA**
- Exibir como: R\$ XX,XX / KG

### Peso exibido

- Peso médio por UND
- Origem: **COLUNA I**

### Exemplo visual

Meia Rês

Preço: R\$ 31,72 / KG

Peso médio: 180 KG

 Não mostrar desconto

 Não mostrar tabela

 Não calcular valor total sem UND definida

---

## 5 LOGIN E PERFIS (SUPABASE)

Tabela: `usuarios`

Campo-chave: `nivel`

Nível	Perfil
1–3	Cliente final
4–7	Vendedor / Supervisor
8–10	Gestão / Admin

Após login:

- Identificar `nivel`
  - Aplicar permissões
  - Não mudar URL base
- 

## 6 CATÁLOGO (USUÁRIO LOGADO)

**Regra geral**

- Todos continuam vendendo TAB3 como **preço base**
  - O sistema calcula **preço real internamente**
  - O cliente vê **benefício percentual**, não tabela
- 

## 7 REGRA DE PRECIFICAÇÃO (CRÍTICA)

**Unidade de compra**

- UND ou CX (COLUNA AC)

- 1 CX = 1 UND lógica

## Peso

- Peso médio por UND: **COLUNA I**
- CX assume 10kg como base (se aplicável)
- Peso é **estimativa**

## Regra B2B por volume (UND total do carrinho)

UND total	Tabela	Preço
1 UND	TAB1	R\$/KG
2–9 UND	TAB0	R\$/KG
≥10 UND	TAB4	R\$/KG

⚠ A soma é do **carrinho inteiro**, não por item.

---

## 8 CÁLCULO DO VALOR (REGRA FUNDAMENTAL)

### Fórmula obrigatória

$$\text{VALOR TOTAL} = \text{UND} \times \text{PESO MÉDIO} \times \text{PREÇO POR KG}$$

### Exemplo real (Produto 400010)

- UND: 2
- Peso médio: 180 KG
- Preço aplicado: R\$ 24,40 / KG (TAB0)

$$2 \times 180 \times 24,40 = \text{R\$ } 8.784,00$$

### Exibição no carrinho

$$2 \text{ UND} \times 180 \text{ KG} \times \text{R\$ } 24,40/\text{KG}$$

$$\text{Total estimado: R\$ } 8.784,00$$

Peso total estimado: 360 KG

Benefício do seu cadastro + volume: 23,1%

✗ Nunca mostrar "TAB0", "TAB1", etc.

✗ Nunca calcular preço apenas por UND

---

## 9 RESERVA E ESTOQUE

### Reserva

- Criada ao adicionar ao carrinho
- Validade: **2 horas**

### Expiração

- Se não confirmar:
  - Reserva expira
  - Estoque retorna

### Fluxo

Carrinho → Reserva → Pedido → Abate de estoque

✗ Não existe pedido sem reserva

---

## 10 PEDIDO

Pedido pode ser criado por:

- Cliente
- Vendedor (em nome do cliente)

Pedido confirmado deve salvar:

- Cliente final
  - Usuário que digitou
  - Supervisor indireto
  - Tabela aplicada (interno)
  - Valor estimado
  - Peso estimado
- 

## **11 REGRAS ABSOLUTAS (NÃO NEGOCIÁVEIS)**

- ✗ Inventar preço fora do Sheets
  - ✗ Mostrar tabela ao cliente
  - ✗ Usar produto < 400000
  - ✗ Calcular preço sem KG
  - ✗ Quebrar hierarquia
  - ✗ Criar pedido sem reserva
- 

## **12 CHECKLIST DE ACEITE (DEV)**

- Catálogo abre sem login
  - Login aplica regras sem trocar URL
  - Preço público = TAB3
  - B2B vê desconto %, não tabela
  - UND × KG × preço/KG correto
  - Reserva expira em 2h
  - Produto < 400000 não aparece
- 

## **13 GOVERNANÇA**

- Este arquivo é subordinado ao:  
**Manual do Modelo de Negócio Schlosser PRO V7.2**
  - Alterações:
    - Somente pelo autor
    - Nova versão obrigatória
- 

## CONCLUSÃO PARA O DEV

**Não tente “melhorar” a regra.**  
**Não adapte o negócio ao código.**  
**Implemente exatamente como descrito.**



# V7.3 Modelo de Negocio

## Sercarne.com



# MODELO DE NEGÓCIO E OPERAÇÕES — SCHLOSSER PRO V7.3

SERCARNE.COM | 2026

**Autoridade máxima:** Cássio Zaleski

**Versão:** V7.3

**Status:** Documento Oficial — Fonte Única da Verdade

**Objetivo:** Definir regras imutáveis do modelo de negócios, operação comercial e lógica do sistema digital Schlosser PRO.

**⚠️ Qualquer sistema, código, automação, desenvolvedor ou IA deve se adaptar a este documento — nunca o contrário.**

---

## 1. VISÃO GERAL DO NEGÓCIO

O Schlosser PRO é um sistema digital de vendas **B2B e B2C** do **Frigorífico Schlosser / Grupo Zaleski**, com foco em:

- Venda de proteínas com controle de estoque real por **UND ou CX**
- Precificação dinâmica por **perfil + volume total do carrinho**
- Operação comercial **hierárquica**
- Origem de dados **centralizada, auditável e rastreável**

 O sistema **não é um e-commerce tradicional**.  
Ele representa o **modelo operacional real do frigorífico**.

---

## 2. STACK OFICIAL E FUNÇÃO DE CADA PLATAFORMA

Plataforma	Função
<b>Google Sheets</b>	Fonte primária do negócio (catálogo, preços, tabelas, pesos e regras)
<b>Supabase</b>	Banco operacional (usuários, reservas, pedidos, estoque)

<b>Hostinger Horizons</b>	Front-end e lógica de interface
<b>Domínio Hostinger</b>	Endereço oficial: <b>sercarne.com</b>
<b>ChatGPT / IA</b>	Consultoria, revisão de regras, apoio conceitual
<b>GitHub (opcional)</b>	Versionamento técnico
<b>Conta Google</b>	Governança dos dados
<b>ERP SYSMO</b>	ERP oficial do Grupo Zaleski (leitura e referência)

📌 **Google Sheets é a fonte primária do negócio.**

Nenhuma outra base pode inventar, recalcular ou sobrescrever preços ou regras.

---

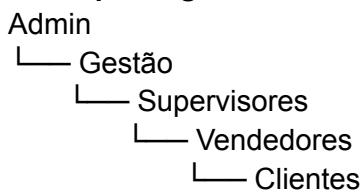
### 3. TIPOS DE USUÁRIO E HIERARQUIA (SUPABASE)

**Tabela:** `usuarios`

**Campo-chave:** `nivel`

Nível	Perfil	Acesso
1 a 3	Clientes finais (B2B / B2C)	Apenas seus pedidos
4 a 7	Vendedores / Supervisores	Seus clientes e vendas
8 a 10	Gestão / Admin	Visão total

**Hierarquia lógica:**



📌 **Hierarquia manda mais que rota ou URL.**

---

### 4. ENTRADA ÚNICA DO SITE (CATÁLOGO COMO HOME)

O domínio **sercarne.com** deve abrir diretamente no **Catálogo Público (B2C)**.

- Não existe tela inicial de escolha
- O catálogo público é a **Home do sistema**
- O login fica disponível **na própria tela do catálogo**

💡 O catálogo público:

- Lista produtos válidos
  - Exibe preço **TAB3**
  - Não mostra pedidos, rotas, frete ou tabelas internas
- 

## 5. FLUXO DE LOGIN (EMBUTIDO NO CATÁLOGO)

Login via **Supabase** diretamente no catálogo.

Após login:

- Sistema identifica **nível** e **tipo\_de\_usuario**
- Aplica imediatamente o **modo correto de visualização**
- Mantém o usuário **na mesma tela**, apenas trocando permissões

💡 O login **não redireciona para outra “home”**.

---

## 6. CATÁLOGO E ORIGEM DOS DADOS (REFERÊNCIAS FIXAS)

**Planilha oficial:**

 [2026 Base Catalogo Precifica V2 \(Google Sheets\)](#)

### Filtro obrigatório

- Somente produtos com código  $\geq 400000$  (**COLUNA D**) podem ser carregados

- Produtos com código menor devem ser ignorados

 Motivo: Frigorífico Schlosser = Filial 4 do Grupo Zaleski.

---

## 7. DESCRIÇÃO DE PRODUTOS (REGRA DE EXIBIÇÃO)

Ordem obrigatória:

1. **COLUNA AK** — descrição principal (NF / comercial)
2. **COLUNA AL** — descrição técnica (menor, cinza)
3. **Fallback:** COLUNA E

 Nunca exibir #N/A

 Nunca deixar produto sem descrição

---

## 8. IMAGENS DE PRODUTO (FALLBACK OFICIAL)

Ordem fixa:

1. AE — imagem principal
2. AF — alternativa 1
3. AG — alternativa 2
4. AH — imagem da marca
5. Padrão Schlosser

 Se usar AH, exibir badge “**Imagen da Marca**”

---

## 9. PRECIFICAÇÃO — REGRA CRÍTICA

### 9.1 Preço base (visível para TODOS)

- TAB3
- Origem: **COLUNA AA (27)**

Exemplo:

Produto 400010 → R\$ 30,93 (TAB3)

---

## 9.2 Regra B2B (uso interno)

Volume total do carrinho:

### UND Tabela

1 TAB1

2–9 TAB0

≥10 TAB4

 A soma é do **carrinho inteiro**, não por item.

---

## 9.3 Exibição ao cliente

O cliente **nunca vê tabela**.

Texto permitido:

“Desconto aplicado conforme cadastro e volume do pedido.”

---

## 9.4 Benefício (%) para Cliente B2B

Para **Cliente B2B logado**, o sistema deve exibir o **benefício percentual** comparado ao preço público.

- Preço referência: **TAB3**
- Preço aplicado: TAB1 / TAB0 / TAB4
- Exibição:

“Benefício do seu cadastro + volume: **X%**”

- 
- 💡 O cliente vê o percentual, **não vê tabela**.
  - 💡 Vendedor / Supervisor / Admin podem ver tabela aplicada.
- 

## 10. DEFINIÇÃO DE UNIDADE, CAIXA E PESO

**COLUNA AC:**

- UND = 1 unidade
- CX = 1 unidade lógica

**Peso médio:** COLUNA I

- CX assume 10kg ±30%

- 
- 💡 Peso é estimativa informativa
  - 💡 Regra de preço **não depende do kg**
- 

## 11. ESTOQUE, RESERVA E EXPIRAÇÃO

- Reserva criada ao adicionar ao carrinho
- Validade: **2 horas**
- Se não confirmar:
  - reserva expira
  - produto retorna ao estoque

Tabelas Supabase:

- `reservas`
  - `entradas_estoque`
  - `pedidos`
-

## 12. PEDIDO

Pedido pode ser criado por:

- Cliente
- Vendedor (em nome do cliente)

Pedido confirmado:

- converte reserva
  - abate estoque
  - registra histórico completo
- 

## 13. REGRAS QUE NÃO PODEM SER QUEBRADAS

- ✗ Inventar preço
  - ✗ Mostrar tabela ao cliente
  - ✗ Produto < 400000
  - ✗ Pedido sem reserva
  - ✗ Quebrar hierarquia
  - ✗ Calcular preço por kg
- 

## 14. GOVERNANÇA DO DOCUMENTO

- Este documento é a **fonte única da verdade**
  - Alterações **somente pelo autor**
  - Toda mudança gera **nova versão**
- 

## 15. CHANGE LOG

Versão	Data	Alteração
0		

V7        21/01/2026 Consolidação

V7.1      21/01/2026 Referências técnicas

**V7.3      Atual**            Catálogo como Home + Login embutido + Benefício %

---

## CONCLUSÃO FINAL

O Schlosser PRO agora possui:

- modelo de negócio fechado
- regras técnicas rastreáveis
- proteção contra decisões erradas
- base única para qualquer desenvolvedor ou IA

 **Quem desenvolver fora disso está desenvolvendo errado.**

---

 V7.3 Requisitos.md

# REQUISITOS.md FINAL — SCHLOSSER PRO V7.3

(Copie e salve como **REQUISITOS.md**):

#  REQUISITOS.md — SCHLOSSER PRO V7.3

Plataforma: SERCARNE.COM | 2026

Autoridade máxima: Cássio Zaleski

Status: Obrigatório (Fonte Técnica do Projeto)

Complementa: Modelo de Negócio e Operações — Schlosser PRO V7.3

---

## ## 1. Princípio Fundamental

Este sistema NÃO é um e-commerce genérico. Ele deve reproduzir fielmente o modelo operacional do Frigorífico Schlosser / Grupo Zaleski.

Se alguma regra técnica entrar em conflito com o Modelo de Negócio:  
a regra técnica está errada.

---

## ## 2. Fontes de Verdade (ordem absoluta)

- 1) Google Sheets (catálogo e preços)
- 2) Manual Schlosser PRO V7.3
- 3) Supabase (operação)
- 4) Front-end (Horizons / outro)

O front-end é descartável. O Sheets não é.

---

## ## 3. Entrada Única do Site (Catálogo como Home)

### ### RF-001 — Home = Catálogo Público (B2C)

Ao acessar sercarne.com, o usuário deve cair direto no Catálogo Público (B2C), sem tela intermediária.

### ### RF-002 — Login embutido no Catálogo

O login deve estar disponível no topo da tela do catálogo com CTA claro.

Após login, o usuário permanece na mesma tela e o sistema aplica o modo correspondente (B2B/Vendedor/Supervisor/Admin).

---

## ## 4. Login e Autenticação

### ### RF-003 — Autenticação via Supabase

Login via Supabase.

### ### RF-004 — Campo de login flexível

Aceitar login por:

- usuário
- login
- e-mail
- telefone

Busca:

- case-insensitive
- telefone normalizado (com/sem 55, com/sem hífen)

### ### RF-005 — Segurança de UI

- Botão “olho” na senha
- Mensagem de erro genérica (não revelar se usuário ou senha)

---

## ## 5. Perfis e Hierarquia

### ### RF-006 — Hierarquia por nível

Campo: usuarios.nivel

- Níveis 1–3: Clientes (apenas seus dados)
- Níveis 4–7: Vendedores/Supervisores (seus clientes)
- Níveis 8–10: Gestão/Admin (visão total)

Hierarquia manda mais que rota ou URL.

---

## ## 6. Catálogo (Google Sheets)

### ### RF-007 — Origem exclusiva do catálogo

Origem: Google Sheets

Aba: 2026 Base Catalogo Precifica V2

### ### RF-008 — Filtro obrigatório

Carregar somente produtos com:

Código (COLUNA D)  $\geq$  400000

Produtos com código menor devem ser ignorados completamente.

---

## ## 7. Descrição de Produtos

### ### RF-009 — Ordem de exibição

- 1) COLUNA AK — descrição principal (NF)
- 2) COLUNA AL — descrição técnica (cinza menor)
- 3) Fallback: COLUNA E (quando AK/AL vazio ou #N/A)

Nunca exibir #N/A.

---

## ## 8. Imagens de Produto

### ### RF-010 — Fallback fixo

- 1) AE
- 2) AF
- 3) AG
- 4) AH (marca)
- 5) Padrão Schlosser

Se usar AH, exibir badge “Imagen da Marca”.

---

## ## 9. Precificação

### ### RF-011 — Preço base (para todos)

Preço base exibido para TODOS:

TAB3 = COLUNA AA (27)

### ### RF-012 — Regra B2B por volume (interna)

Volume total do carrinho (UND total):

- 1 UND → TAB1
- 2–9 UND → TAB0
- ≥10 UND → TAB4

Cliente nunca vê tabela.

Texto permitido:

“Desconto aplicado conforme cadastro e volume do pedido.”

### ### RF-013 — Benefício % (Cliente B2B)

Se usuário for Cliente B2B logado, exibir:

“Benefício do seu cadastro + volume: X%”

Referência do benefício:  
comparar preço aplicado vs preço TAB3.

É proibido mostrar “TAB0/TAB1/TAB4” ao cliente.

Para vendedor/supervisor/admin é permitido mostrar tabela aplicada (uso interno).

---

## ## 10. Unidade, Caixa e Peso

#### RF-014 — Tipo de venda  
COLUNA AC define UND ou CX

Regras:

- 1 UND = 1 unidade (independente do peso)
- 1 CX = 1 unidade lógica

#### RF-015 — Peso  
COLUNA I é peso médio (estimativa).  
Para CX assumir 10kg padrão ( $\pm 30\%$ ).  
Preço nunca depende do kg real.

---

## ## 11. Reserva e Estoque

#### RF-016 — Reserva automática  
Reserva criada ao adicionar ao carrinho.  
Validade: 2 horas.

Se não confirmar pedido:  
reserva expira e devolve estoque.

Expiração deve ser automática.

---

## ## 12. Pedido

#### RF-017 — Pedido somente com reserva  
Pedido só pode ser confirmado a partir de reserva válida.

Pedido pode ser criado por:  
- Cliente  
- Vendedor (em nome do cliente)

Ao confirmar:

- reserva vira pedido
- estoque é abatido definitivamente

---

## ## 13. Regras Absolutas (bloqueios)

É proibido:

- inventar preço fora do Sheets
- mostrar tabela ao cliente
- carregar produto < 400000
- criar pedido sem reserva
- quebrar hierarquia
- calcular preço por kg para regra de tabela

---

## ## 14. Critério de Aceitação (GO/NO-GO)

GO somente se:

- Login funciona
- Catálogo carrega com filtro  $\geq 400000$
- Reserva expira e devolve estoque
- Cliente não vê tabela
- Preço muda por volume (B2B)
- Benefício % aparece para B2B
- Pedido só confirma com reserva

Caso contrário:

NÃO IR PARA PRODUÇÃO.

 V7.3 Checklist

# CHECKLIST TÉCNICO PARA O DEV — SCHLOSSER PRO V7.3

**Objetivo:** colocar a plataforma em produção **sem quebrar regras do negócio.**

**Regra:** se algum item “crítico” falhar → **NO-GO.**

## A. Setup e Governança (crítico)

- Recebi o documento **Modelo Schlosser PRO V7.3** e li integralmente
- Recebi o **REQUISITOS.md V7.3** e vou desenvolver em cima dele
- Confirmei que **Google Sheets** é a fonte da verdade (catálogo e preços)
- Confirmei que **Supabase** é a operação (usuarios/reservas/pedidos/entradas\_estoque)
- Confirmado: **front-end é descartável** (não pode virar “cérebro”)

## B. Catálogo (crítico)

- Catálogo lê **somente** da planilha **2026 Base Catalogo Precifica V2**
- Filtro obrigatório aplicado: **Código (COLUNA D) ≥ 400000**
- Produtos com código < 400000 **não aparecem em lugar nenhum**
- Descrição exibida corretamente:
  - Principal = **AK**
  - Técnica (cinza menor) = **AL**
  - Fallback = **E** quando AK/AL vazio ou **#N/A**
- Imagem com fallback na ordem:
  - AE → AF → AG → AH → Padrão
  - Se cair em AH, mostrar badge “**Imagen da Marca**”

## C. Entrada do Site / UX (crítico)

- **sercarne.com abre direto no catálogo público (B2C)**
- Login embutido no topo do catálogo (CTA claro)
- Após login, o usuário permanece na mesma tela (sem “nova home”), apenas muda modo/permissões

## D. Login / Usuários (crítico)

- Login usa Supabase (auth ou tabela usuarios, conforme arquitetura definida)
- Campo login aceita: usuário / login / email / telefone
- Busca **case-insensitive**
- Normalização de telefone (com/sem 55, com hífen etc.)
- Campo senha com “olho” (mostrar/ocultar)
- Mensagem de erro genérica (não revelar se usuário ou senha está incorreto)

## E. Hierarquia (crítico)

- Acesso sempre respeita **nível**:
  - 1–3: cliente (somente seus dados)
  - 4–7: vendedor/supervisor (seus clientes)
  - 8–10: admin/gestão (visão total)
- Vendedor **não** vê lista de clientes geral — só os vinculados
- Supervisor vê vendedores abaixo
- Admin vê tudo

## F. Precificação (crítico)

- Preço base exibido para todos = **TAB3 (COLUNA AA / 27)**
- Cliente **nunca vê tabela** (TAB0/TAB1/TAB4 proibido no UI do cliente)
- Regra B2B por volume total do carrinho (UND total):
  - 1 UND → TAB1
  - 2–9 UND → TAB0
  - $\geq 10$  UND → TAB4
- Texto permitido ao cliente:
  - “Desconto aplicado conforme cadastro e volume do pedido.”

## G. Benefício % (crítico para B2B)

- Se usuário for **Cliente B2B logado**, exibir:
  - “Benefício do seu cadastro + volume: X%”
- Referência do cálculo do benefício:
  - comparar preço aplicado vs TAB3
- Cliente não vê nome da tabela
- Vendedor/supervisor/admin podem ver tabela aplicada (interno)

## H. UND / CX / Peso (crítico)

- Tipo de venda vem da **COLUNA AC**
- Regra de unidade:
  - 1 UND = 1 unidade (independente do peso)
  - 1 CX = 1 unidade lógica
- Peso médio da peça (COLUNA I) exibido como estimativa
- Para CX assumir 10kg padrão ( $\pm 30\%$ )

- Preço **não** depende de kg real (kg é informativo)

## I. Reserva (crítico)

- Reserva é criada **ao adicionar ao carrinho**
- Reserva tem validade **2 horas**
- Expiração automática devolve estoque
- Reserva é vinculada a:
  - usuario\_id
  - produto\_codigo
  - quantidade
  - expira\_em
  - status

## J. Pedido (crítico)

- Pedido **só nasce a partir de reserva válida**
- Pedido pode ser criado por:
  - Cliente
  - Vendedor em nome do cliente
- Ao confirmar:
  - reserva → pedido
  - estoque abatido definitivamente
- Pedido registra (mínimo):
  - cliente\_id
  - usuario\_digitador\_id

- tabela\_aplicada (interno)
- valor\_total
- timestamp

## K. Logs & Diagnóstico (recomendado)

- Erros não podem gerar tela branca (try/catch + fallback UI)
- Logar:
  - tabela aplicada (interno)
  - motivo do benefício %
  - status de reserva/expiração
- Página de erro amigável

## L. GO/NO-GO (entrega)

**GO somente se:**

- Login funciona
- Catálogo carrega com filtro  $\geq 400000$
- Reserva expira e devolve estoque
- Cliente não vê tabela
- Preço muda corretamente por volume (B2B)
- Benefício % aparece para B2B
- Pedido só confirma com reserva

Se falhar qualquer “crítico” → **NO-GO**.



# V7.3 Plano De Implementação

“Use como base os arquivos Markdown neste repositório:  
[https://github.com/SEU\\_USUARIO/schlosser-pro-docs](https://github.com/SEU_USUARIO/schlosser-pro-docs)  
Ignore qualquer outro documento.”

---

# PLANO DE IMPLEMENTAÇÃO — SCHLOSSER PRO V7.3

## Ordem obrigatória das tarefas (GO / NO-GO)

Este plano deve ser seguido **na ordem**.  
Não pule etapas.  
Não “melhore” regras.  
O objetivo é **colocar no ar funcionando**, não refatorar o mundo.

---

## ETAPA 0 — ALINHAMENTO (ANTES DE CODAR)

**Objetivo:** evitar 90% dos bugs que já aconteceram.

- Ler **Modelo de Negócio e Operações — Schlosser PRO V7.3**
- Ler **REQUISITOS.md V7.3**
- Confirmar entendimento:
  - Google Sheets = fonte da verdade
  - Supabase = operação
  - Front-end = descartável
- Confirmar que:
  - catálogo abre como home
  - login é embutido no catálogo
  - cliente nunca vê tabela

 **Se houver dúvida aqui, parar e perguntar.**

---

## ETAPA 1 — CATÁLOGO B2C (BASE DO SISTEMA)

**Objetivo:** colocar o site no ar com algo útil imediatamente.

1. Conectar o front-end ao Google Sheets
2. Carregar produtos da aba:  
[2026 Base Catalogo Precifica V2](#)
3. Aplicar filtro obrigatório:
  - **Código (COLUNA D) ≥ 400000**
4. Renderizar produtos com:
  - descrição AK (principal)
  - descrição AL (técnica, menor)
  - fallback E
5. Renderizar imagens com fallback:  
AE → AF → AG → AH → padrão
6. Exibir **preço TAB3 (COLUNA AA / 27)**

### Critério de aceite da Etapa 1

- Usuário entra em [sercarne.com](#)
- Vê catálogo completo
- Preço visível é TAB3
- Nenhum produto inválido aparece

---

## ETAPA 2 — LOGIN EMBUTIDO NO CATÁLOGO

**Objetivo:** permitir upgrade de perfil sem mudar de tela.

1. Inserir bloco de login no topo do catálogo

2. Login via Supabase

3. Campo login aceita:

- usuário
- login
- email
- telefone

4. Busca:

- case-insensitive
- telefone normalizado

5. Após login:

- identificar `nivel`
- identificar `tipo_de_usuario`
- manter usuário no catálogo
- apenas mudar permissões

#### Critério de aceite

- Login funciona
- Tela não fica branca
- Usuário continua no catálogo após login



## ETAPA 3 — HIERARQUIA E MODOS DE USUÁRIO

**Objetivo:** cada um vê só o que pode.

1. Implementar regras por `nivel`:

- 1–3: cliente
- 4–7: vendedor/supervisor
- 8–10: admin

2. Cliente:

- vê apenas seus pedidos

3. Vendedor:

- vê apenas seus clientes

4. Supervisor:

- vê vendedores abaixo

5. Admin:

- vê tudo

📌 Hierarquia **sempre vence** rota, botão ou URL.

#### Critério de aceite

- Vendedor não vê cliente que não é dele
- Cliente não acessa área administrativa



## ETAPA 4 — PRECIFICAÇÃO DINÂMICA B2B

**Objetivo:** aplicar a regra mais sensível do sistema.

1. Manter preço base exibido = TAB3

2. Calcular UND total do carrinho

3. Aplicar regra:

- 1 UND → TAB1
- 2–9 UND → TAB0

- $\geq 10$  UND → TAB4

4. Cliente **não vê tabela**

5. Exibir texto:

“Desconto aplicado conforme cadastro e volume do pedido.”

#### Critério de aceite

- Preço muda conforme quantidade total
  - Cliente nunca vê “TAB0 / TAB1 / TAB4”
- 

## ETAPA 5 — BENEFÍCIO (%) PARA CLIENTE B2B

**Objetivo:** deixar claro o ganho do cliente sem revelar tabela.

1. Calcular benefício:

- comparar preço aplicado vs TAB3

2. Exibir ao cliente B2B:

“Benefício do seu cadastro + volume: X%”

3. Para vendedor/admin:

- permitir ver tabela aplicada (uso interno)

#### Critério de aceite

- Cliente vê percentual
- Cliente não vê tabela
- Vendedor vê informação completa



## ETAPA 6 — RESERVA AUTOMÁTICA (2H)

**Objetivo:** proteger estoque.

1. Criar reserva ao adicionar ao carrinho
2. Registrar:
  - produto
  - quantidade
  - usuário
  - expira\_em
3. Expiração automática após 2h
4. Ao expirar:
  - devolver estoque



**Expiração não pode ser manual.**



### Critério de aceite

- Reserva expira sozinha
  - Estoque volta automaticamente
- 



## ETAPA 7 — PEDIDO

**Objetivo:** consolidar a venda.

1. Pedido só pode nascer de reserva válida
2. Pedido pode ser criado por:
  - cliente

- vendedor (em nome do cliente)

3. Ao confirmar:

- reserva → pedido
- estoque abatido

4. Registrar:

- cliente\_id
- usuario\_digitador\_id
- tabela\_aplicada (interno)
- valor\_total

#### Critério de aceite

- Não existe pedido sem reserva
  - Estoque bate com pedidos
- 

## ETAPA 8 — TESTES FINAIS (GO / NO-GO)

**Checklist final obrigatório:**

- Login funciona
- Catálogo abre como home
- Produto < 400000 não aparece
- Preço muda por volume
- Benefício % aparece para B2B
- Cliente não vê tabela
- Reserva expira em 2h
- Pedido só confirma com reserva

👉 Se qualquer item falhar: NÃO IR PARA PRODUÇÃO.

---



## OBSERVAÇÃO FINAL AO DEV

"Este sistema não está aberto a interpretação.  
Ele implementa um modelo de negócio já definido.  
Se algo parecer estranho tecnicamente, a regra de negócio vence."

# Guia 8





# MODELO DE NEGÓCIO V7.0

# MODELO DE NEGÓCIO E OPERAÇÕES — SCHLOSSER PRO V7

SERCARNE.COM | 2026

**Autoridade máxima:** Cássio Zaleski

**Versão:** V7

**Status:** Documento Oficial (Fonte Única da Verdade)

**Objetivo:** Definir regras **imutáveis** do modelo de negócio, operação comercial e lógica do sistema digital Schlosser PRO.

**⚠️ Qualquer sistema, código, automação, desenvolvedor ou IA deve se adaptar a este documento — nunca o contrário.**

---

## 1. VISÃO GERAL DO NEGÓCIO

O Schlosser PRO é um sistema digital de vendas **B2B e B2C** do **Frigorífico Schlosser / Grupo Zaleski**, com foco em:

- Venda de proteínas com **controle de estoque real por unidade comercial (UND ou CX)**
- **Precificação dinâmica** por perfil de cliente e **volume total do carrinho**
- Operação comercial **hierárquica** (cliente → vendedor → supervisor → gestão → admin)
- Origem de dados **centralizada, auditável e rastreável**

 O sistema **não é um e-commerce tradicional**.  
Ele representa o **modelo operacional real do frigorífico**.

---

## 2. STACK OFICIAL E FUNÇÃO DE CADA PLATAFORMA

Plataforma		Função
<b>Google Sheets</b>	<a href="#">Precificação Mix Frigo Schlosser / Zaleski 2026 (IA APP)</a>	Base oficial de catálogo, preços, tabelas, pesos e regras comerciais
<b>Supabase</b>	<a href="https://supabase.com/dashboard/organizations">https://supabase.com/dashboard/organizations</a>	Banco de dados operacional (usuários, reservas, pedidos, estoque)

<b>Hostinger Horizons</b>	<a href="https://horizons.hostinger.com/f5e592ff-4b11-4a06-90fa-42f9bf225481?location=hpanel&amp;hostingReferenceId=1008525354">https://horizons.hostinger.com/f5e592ff-4b11-4a06-90fa-42f9bf225481?location=hpanel&amp;hostingReferenceId=1008525354</a>	Front-end e lógica de interface (execução do software)
<b>Domínio Hostinger</b>	<a href="https://hpanel.hostinger.com/">https://hpanel.hostinger.com/</a>	Endereço oficial da aplicação (sercarne.com)
<b>ChatGPT / IA</b>	<a href="https://chatgpt.com/">https://chatgpt.com/</a>	Consultoria, revisão de regras, apoio conceitual
<b>GitHub (opcional)</b>	<a href="https://github.com/">https://github.com/</a>	Versionamento técnico quando necessário
<b>Conta Google</b>	<a href="https://myaccount.google.com/?pli=1">https://myaccount.google.com/?pli=1</a>	Governança de dados e acessos (Google Console Account)
<b>ERP SYSMO</b>	<b>Usuario:</b> consultabd <b>Senha:</b> ***** <b>ip:</b> 192.168.40.14	ERP oficial do Grupo Zaleski (leitura e referência geral de dados)

### 📌 Google Sheets é a fonte primária do negócio.

Nenhuma outra base pode inventar, recalcular ou sobreescriver preços ou regras.

OBS.: Ambos são logados pelo usuário Cássio Zaleski

---

## 3. TIPOS DE USUÁRIO E HIERARQUIA (SUPABASE)

**Tabela:** `usuarios`

**Campo-chave:** `nivel`

Nível	Perfil	Acesso
1 a 3	Clientes finais (B2B / B2C)	Apenas seus pedidos
4 a 7	Vendedores / Supervisores	Seus clientes e vendas
8 a 10	Gestores e Admin	Visão total

**Exemplos reais:**

- Cliente **Cegil** → nível 2
- Representante PJ **Lissandro** → nível 5
- Supervisor PJ → nível 6
- Admin → nível 10

**Hierarquia lógica:**

```
Admin
  └── Gestão Comercial
    └── Supervisores
      └── Vendedores
        └── Clientes
```

📌 Hierarquia manda mais que rota ou URL.

---

## 4. FLUXO DE LOGIN

- Login via **usuário + senha** (Supabase)
- Após login:
  - Sistema identifica `tipo_de_usuario` e `nivel`
  - Redireciona conforme campo `app_login`

📌 Cliente pode gerar pedido:

- Pelo **próprio login**
  - OU por **vendedor**, em nome do cliente
- 

## 5. CATÁLOGO E ORIGEM DOS DADOS (REFERÊNCIAS FIXAS)

**Planilha oficial:**

📄 2026 Base Catalogo Precifica V2 (Google Sheets)

[https://docs.google.com/spreadsheets/d/12wPGal\\_n7PKYFGz9W\\_\\_bXgK4mly2NbrEEGwTrIDCzcl/edit?gid=795418002#gid=795418002](https://docs.google.com/spreadsheets/d/12wPGal_n7PKYFGz9W__bXgK4mly2NbrEEGwTrIDCzcl/edit?gid=795418002#gid=795418002)

### Filtro obrigatório de produtos

- Somente produtos com código  $\geq 400000$  (**COLUNA D**) podem ser carregados.
- Produtos com código  $< 400000$  devem ser **ignorados**.

📌 Motivo:

O Frigorífico Schlosser opera como **Filial 4** do Grupo Zaleski.  
Códigos abaixo disso são **cadastro inválido** para o sistema.

---

## 6. DESCRIÇÃO DE PRODUTOS (REGRA DE EXIBIÇÃO)

Ordem obrigatória:

1. **Descrição principal (NF / comercial)**
  - **COLUNA AK**
  - Texto principal exibido em destaque
2. **Descrição técnica (complementar)**
  - **COLUNA AL**
  - Exibida logo abaixo, em fonte menor e cor cinza
3. **Contingência (fallback)**
  - Caso AK ou AL estejam vazias ou com #N/A
  - Usar **COLUNA E**

📌 Nunca exibir #N/A

📌 Nunca deixar produto sem descrição

---

## 7. IMAGENS DE PRODUTO (FALLBACK OFICIAL)

Ordem exata de busca:

Prioridade	Coluna	Uso
1	AE	Imagen principal
2	AF	Alternativa 1
3	AG	Alternativa 2
4	AH	Imagen da marca
5	—	Imagen padrão Schlosser

📌 Se usar AH, exibir badge: “Imagen da Marca”

✗ Não alterar a ordem

✗ Não buscar imagens externas

---

## 8. PRECIFICAÇÃO B2B / B2C (REGRA CRÍTICA)

Preço base visível para TODOS

- TAB3

- Origem: **COLUNA AA (Coluna 27)**
- Aba: **2026 Base Catalogo Precifica V2**

**Exemplo:**

Produto 400010 → R\$ 30,93 (TAB3)

---

### Lógica real de preço (uso interno)

#### Clientes B2B (CNPJ)

Volume total no carrinho	Tabela aplicada
1 UND	TAB1
2 a 9 UND	TAB0 ( $\approx$ 4% abaixo TAB1)
$\geq$ 10 UND	TAB4 ( $\approx$ 2% abaixo TAB0)

📌 A regra considera a **soma total de unidades do carrinho**, não por item.

📌 **O cliente nunca vê tabela.**

Texto permitido:

“Desconto aplicado conforme cadastro e volume do pedido.”

📌 Somente:

- Vendedor
- Supervisor
- Admin

podem visualizar qual tabela foi aplicada.

---

## 9. DEFINIÇÃO DE UNIDADE, CAIXA E PESO

**Coluna de controle: COLUNA AC**

**Tipos      Regra comercial**

UND    1 UND = 1 unidade

CX      1 CX = 1 unidade lógica

- Peso médio: **COLUNA I**

- CX assume **10kg padrão**
- Variação permitida: **±30%**
- Peso é **estimativa informativa**

💡 **Preço nunca depende do kg real.**  
💡 Regra de tabela depende apenas de UND/CX.

---

## 10. ESTOQUE, RESERVA E EXPIRAÇÃO

- Reserva criada ao **adicionar ao carrinho**
- Validade da reserva: **2 horas**
- Se não confirmar pedido:
  - Reserva expira
  - Produto retorna ao estoque

**Tabelas Supabase:**

- `reservas`
  - `entradas_estoque`
  - `pedidos`
- 

## 11. PEDIDO

Pedido pode ser criado por:

- Cliente
- Vendedor (em nome do cliente)

Pedido confirmado:

- Converte reserva em pedido
- Abate estoque definitivamente

Pedido registra:

- Cliente final
  - Usuário que digitou
  - Supervisor indireto
  - Tabela aplicada (interno)
  - Valor final
-

## 12. REGRAS QUE NÃO PODEM SER QUEBRADAS

- ✗ Inventar preço fora do Sheets
  - ✗ Mostrar tabela ao cliente
  - ✗ Permitir produto < 400000
  - ✗ Criar pedido sem reserva
  - ✗ Quebrar hierarquia de nível
  - ✗ Calcular preço por kg para regra de tabela
- 

## 13. CHECKLIST MANUAL DE TESTE

- Login cliente funciona
  - Login vendedor vê seus clientes
  - Reserva expira em 2h
  - Quantidade altera preço corretamente
  - Cliente não vê tabela
  - Supervisor vê vendedores abaixo
  - Admin vê tudo
- 

## 14. GOVERNANÇA DO DOCUMENTO

- Este documento é a **fonte única da verdade**
  - Alterações somente pelo autor
  - Nova versão obrigatória (V7.1, V8...)
- 

## 15. CHANGE LOG

Versão	Data	Alteração
o		
V7	21/01/2026	Consolidação total
V7.1	21/01/2026	Inclusão de referências técnicas e bloqueios

---

## CONCLUSÃO

Agora o **Schlosser PRO** tem:

- modelo de negócio fechado

- regras técnicas rastreáveis
- blindagem do Google Sheets
- base para qualquer dev, IA ou plataforma



# FLUXOGRAMA VISUAL EXECUTIVO



# FLUXOGRAMA VISUAL EXECUTIVO

## Schlosser PRO V7 — Fluxo Comercial e Operacional

### Objetivo visual

- Clareza
  - Leitura rápida
  - Autoridade
  - Visão de negócio (não de código)
- 



### PADRÃO VISUAL (ANTES DE MONTAR)

#### Estilo

- Fundo: **claro (branco ou cinza muito leve)**
- Blocos: **retângulos com cantos arredondados**
- Setas: simples, cinza escuro
- Ícones sempre à esquerda do texto

#### Tipografia

- Título: Bold
  - Blocos: Regular
  - Observações: Cinza menor
- 



### ÍCONES (USE SEMPRE OS MESMOS)

Ícon	Significado
------	-------------

e

Cliente

Carrinho

Reserva

Estoque



Vendedor



Gestão / Admin

---

## SLIDE 1 — VISÃO GERAL DO SISTEMA

### Título

Schlosser PRO V7 — Visão Geral do Fluxo

### Fluxo horizontal (centro do slide):



Cliente / Parceiro



Carrinho



Reserva (2h)



Estoque



Gestão & Controle

### Rodapé (texto pequeno):

Google Sheets = Fonte da Verdade | Supabase = Operação | Horizons = Interface

---



## SLIDE 2 — CLIENTE GERAL (B2C)

### Título



Fluxo Cliente Geral (B2C)

### Blocos verticais:



Cliente acessa [sercarne.com](http://sercarne.com)  
(sem login)



### Catálogo Público

- Produtos com código  $\geq 400000$
- Preço exibido: **Tabela Base (TAB3)**



### Adiciona ao Carrinho

- UND ou CX
- Peso exibido como estimativa



### Reserva Automática (2h)



### Finalizou pedido?

- Não → Reserva expira → Estoque retorna  
 → Sim → Pedido confirmado → Estoque abatido
- 



## SLIDE 3 — CLIENTE B2B (LOGADO)

### Título

#### Fluxo Cliente B2B (Login)

### Blocos:

#### Cliente B2B faz login (usuário + senha)



### Catálogo

- Preço inicial visível: **TAB3**
- Cliente **não vê tabela**



### Carrinho Inteligente

*Texto menor em cinza:*

Desconto aplicado conforme cadastro e volume



#### Regra de Volume (invisível ao cliente)

- 1 UND → TAB1
- 2–9 UND → TAB0
- 10+ UND → TAB4



#### Reserva (2h) → Pedido

---

## SLIDE 4 — VENDEDOR / REPRESENTANTE

### Título

#### Fluxo Vendedor / Representante

### Blocos:

#### Vendedor faz login

(nível 4 a 7)



#### Seleciona Cliente

(apenas clientes abaixo na hierarquia)



#### Digita Pedido em Nome do Cliente



#### Reserva (2h)



#### Pedido Confirmado

### Observação (rodapé):

Vendedor pode ver tabela e desconto real

---

## SLIDE 5 — SUPERVISOR

### Título

 Fluxo Supervisor

### Blocos:

 Supervisor faz login

(nível 6–7)



 Visão de Vendedores e Clientes



### Pode:

- Acompanhar pedidos
- Ajustar preços (conforme autonomia)
- Corrigir operações



## SLIDE 6 — ADMIN / GESTÃO

---

### Título

 Fluxo Gestão / Admin

### Blocos centrais:

 Admin / Gestão



 Produção

 Supervisores

 Clientes

 Comercial

 Vendedores

 Pedidos



 Auditoria, Governança e Controle



# SLIDE FINAL — REGRAS IMUTÁVEIS

## Título

Regras que Não Podem Ser Quebradas

## Lista (ícones pequenos):

- Google Sheets é a fonte da verdade
  - Cliente nunca vê tabela
  - Preço não depende do kg
  - Código < 400000 não entra
  - Reserva sempre precede pedido
  - Hierarquia sempre respeitada
- 



## COMO VOCÊ USA ISSO NA PRÁTICA

- Apresentação para equipe
- Alinhamento com dev / IA
- Base visual do Schlosser PRO
- Evita decisões erradas no futuro

# Hostinger Horizons

## O QUE FOI FEITO:

- HomePage - Dois botões: "Cliente Geral (B2C)" e "Login"
- Fluxo de Login - Busca automática de dados (pedidos, frete, rota, dia\_entrega)
- Cliente B2B - Vê seus dados + pedidos + frete + rota
- Vendedor - Vê lista de SEUS clientes (não todos)
- Admin - Vê TUDO
- Catálogo Público - Tab 3 sem login
- REQUISITOS.md - Atualizado com novo fluxo

Vou reorganizar tudo:

Tela inicial = Escolher entre:

"Acessar como Cliente Geral" (B2C público) → Ver catálogo com Tab 3

"Login" (B2B/Vendedor/Admin) → Buscar dados automáticos

Cliente B2B ao fazer login:

Busca automático: seus pedidos, frete, rota, dia de entrega

Ver Tab 1, Tab 0, Tab 4 (preços B2B)

Vendedor ao fazer login:

Ver lista de SEUS clientes

Ver pedidos dos clientes

Mexer preço até Tab 5

Admin ao fazer login:

Ver TODOS os clientes

Ver TODAS as rotas

## Acesso total

Vou atualizar o REQUISITOS.md e reorganizar as páginas:

- Busca dupla - Procura em usuário E login ao mesmo tempo
- Case-insensitive - Maiúsculas/minúsculas não importam
- Telefone funciona - Login com 55-98401-2019 agora funciona
- Nome funciona - Login com Cassio\_Zaleski também funciona
- Botão olho - Adicionado para conferir a senha

# Supabase Tabelas

<https://supabase.com/dashboard/project/dwnxvildqsubhfexuq/editor/17571?schema=public>

## TABELA 1: USUÁRIOS

Clique em "Create a new table"

### Nome da tabela: usuarios

```
create table public.usuarios (
    usuario text not null,
    login text null,
    senha_hash text null,
    "tipo_de_Usuario" text null,
    ativo boolean null,
    app_login text null,
    created_at text null,
    "Nivel" smallint null,
    "TabR$" text null,
    constraint usuarios_pkey primary key (usuario)
) TABLESPACE pg_default;
```

Colunas (na ordem):

- 1- id (UUID, Primary Key) -  Já vem automático
- 2- usuario (Text) - Nome do usuário
- 3- login (Text) - Login único
- 4- senha\_hash (Text) - Senha criptografada
- 5- tipo\_usuario (Text) - admin, vendedor, cliente\_b2c, cliente\_b2b
- 6- ativo (Boolean) - true/false
- 7- app\_login (Text) - Caminho pra redirecionar após login
- 8- created\_at (Timestamp) - Data de criação
- 9- Nivel (int2) - Hierarquia (quanto maior o numero maior o acesso)
- 10- TabR\$ - Qual tabela buscar do Sheets - por nivel/usuario

The screenshot shows the Supabase Table Editor interface. On the left, there's a sidebar with a tree view of tables: schema public (New table, Search tables), entradas\_estoque, pedidos, reservas, and the selected 'usuarios' table. The main area displays the 'usuarios' table with the following data:

	usuario	login	senha_hash	tipo_de_Usuario	ativo	app_login	created_at
1	CasioZaleski	55-98401-2079	zaleski9	Admin	TRUE	/admin	20/01/2026
2	Cogil	55-99013-1132	eve9	Cliente/B2B	TRUE	/cliente_B2B	21/01/2026
3	Eduardo_Zaleski	55-99665-7981	dudu9	Cliente/B2C	FALSE	/cliente_B2C	21/01/2026
4	Everaldo_Schlosser	55-99664-2596	schlosser9	Admin	TRUE	/admin	21/01/2026
5	Lissandro	55-99662-7055	lissandro9	Representante/PJ	TRUE	/vendedor	21/01/2026
6	Olevo_Pinheiro	55-99663-2648	pinheiro9	Gestor Comercial	TRUE	/gestorcomercial	21/01/2026
7	Olevo_Silva	55-91518-7560	silve9	Vendedor/CLT	TRUE	/vendedor	21/01/2026
8	Producao Descosa	Descosa	bottega9	Producao	FALSE	/producao	21/01/2026
9	Supervisor PJ	Super PJ	superpj9	Supervisor/PJ	TRUE	/supervisor	21/01/2026
10	Supervisor_Comercial	Supervisor	super9	Supervisor/CLT	TRUE	/supervisor	21/01/2026
11	Zaleski Matriz	55-99615-6670	rodrigo9	Transferencia/Lojas	TRUE	/transferencias	21/10/2026

## **TABELA 2: ENTRADAS\_ESTOQUE**

Nome da tabela: entradas\_estoque

```
create table public.entradas_estoque (
    id integer generated by default as identity not null,
    data_entrada uuid not null default gen_random_uuid (),
    codigo text null,
    qtd_und smallint null,
    obs text null,
    created_at timestamp without time zone null,
    constraint entradas_estoque_pkey primary key (id)
) TABLESPACE pg_default;
```

Colunas:

id (UUID, Primary Key)

data\_entrada (Date) - Data da entrada

codigo (Text) - SKU do produto

qtd\_und (Integer) - Quantidade em unidades

obs (Text) - Observações

created\_at (Timestamp)

## **TABELA 3: RESERVAS**

Nome da tabela: reservas

```
create table public.reservas (
    id uuid not null default gen_random_uuid (),
    data_reserva timestamp without time zone null,
    cliente text null,
    codigo text null,
    qnd_reservada integer null,
    status text null default ''::text,
    observacoes text null,
    created_at timestamp without time zone null
) TABLESPACE pg_default;
```

Colunas:

id (UUID, Primary Key)

data\_reserva (Timestamp) - Quando foi feita

cliente (Text) - Nome do cliente

codigo (Text) - SKU do produto

qtd\_reservada (Integer) - Quantidade reservada

status (Text) - pendente, confirmado, entregue, cancelado

observacoes (Text)

created\_at (Timestamp)

#### TABELA 4: PEDIDOS

Nome da tabela: pedidos

```
create table public.pedidos (
    id uuid not null default gen_random_uuid (),
    data_pedido timestamp without time zone not null,
    cliente text null,
    itens json null,
    total numeric null,
    status text null,
    observacoes text null,
    created_at timestamp without time zone null,
    constraint pedidos_pkey primary key (id)
) TABLESPACE pg_default;
```

Colunas:

id (UUID, Primary Key)

data\_pedido (Timestamp)

cliente (Text)

itens (JSON) - Array com produtos

total (Decimal) - Valor total

status (Text) - pendente, confirmado, entregue

observacoes (Text)

created\_at (Timestamp)

# Cópia de Guia 20



# MANUAL OPERACIONAL

## SCHLOSSER PRO — VERSÃO V8 (OFICIAL • FINAL)

**Plataforma:** <https://sercarne.com>

**Versão:** V8 — Primeira versão operacional completa

**Status:** ATIVA / PRODUÇÃO

**Formato oficial:** Google Docs

**Este documento é a CONSTITUIÇÃO do sistema**

---

## CAPÍTULO 0 — PRINCÍPIO FUNDAMENTAL (INALTERÁVEL)

**O sistema existe para servir o Modelo de Negócio Schlosser PRO.**

Ferramentas, código, IAs e plataformas são descartáveis.

As regras de negócio **não são**.

Qualquer divergência entre sistema e este manual:

- **X** é erro de implementação
  - **X** não é melhoria
  - **X** deve ser corrigida
- 

## CAPÍTULO 1 — OBJETIVO DESTE DOCUMENTO

Definir **de forma clara, técnica e inequívoca**:

- Catálogo de produtos
- Regras de visibilidade e ordenação
- Precificação
- Rotas, dias de entrega e cutoff
- Estoque disponível por data
- Pedido, status, cancelamento
- Autonomia do vendedor e do supervisor

Este documento existe para:

- Evitar interpretação errada
- Evitar perguntas repetidas
- Permitir troca de desenvolvedor, IA ou tecnologia **sem perda de regra**

---

## CAPÍTULO 2 — FONTES DA VERDADE (REGRA INEGOCIÁVEL)

10. Google Sheets é a única fonte da verdade
  11. UND ≠ KG
  12. Preço sempre é R\$/KG
  13. Cliente nunca vê tabela
  14. Peso é sempre estimativo
  15. Não existe reserva
  16. Pedido é o primeiro compromisso do sistema
  17. Supabase não define regras
  18. Código apenas executa o que está no Sheets
- 

## CAPÍTULO 3 — ARQUITETURA OFICIAL (INDEPENDENTE DE TECNOLOGIA)

A arquitetura lógica do projeto está documentada na planilha:

🔗 Legenda / Objetivos sercarne.com

[https://docs.google.com/spreadsheets/d/12wPGal\\_n7PKYFGz9W\\_bXgK4mly2NbrEEGwTrIDCzcl/edit](https://docs.google.com/spreadsheets/d/12wPGal_n7PKYFGz9W_bXgK4mly2NbrEEGwTrIDCzcl/edit)

A aba “Legenda/Objetivos sercarne.com” define oficialmente:

- Stakeholders
- Responsabilidades
- Onde cada dado nasce
- Onde cada dado é consumido

👉 Esta aba faz parte do sistema.

---

## CAPÍTULO 4 — CATÁLOGO DE PRODUTOS (VISIBILIDADE)

### REGRA DEFINITIVA V8

Quem decide se o produto aparece na plataforma é a planilha.

- ♦ COLUNA AX (COLUNA 50) — EXIBIR NA PLATAFORMA

- **TRUE / VERDADEIRO** → produto aparece
- **FALSE / FALSO** → produto não aparece

✖ Regras:

- **Somente produtos com AX = TRUE aparecem**
  - Código do produto **não é critério**
  - Front-end e back-end **não podem ignorar essa coluna**
- 

## CAPÍTULO 5 — ORDEM DE EXIBIÇÃO NO CATÁLOGO

### REGRA AUTOMÁTICA (ANTES DE QUALQUER FILTRO)

Produtos são exibidos do **MAIOR** para o **MENOR** estoque disponível.

Objetivos:

- Vender o que tem mais disponibilidade
- Evitar ruptura
- Ajudar o vendedor sem ele decidir

Filtros futuros (não implementados ainda):

- Marca
- Espécie
- Tipo de embalagem

⚠ Até existirem, **somente a ordenação por estoque é aplicada**.

---

## CAPÍTULO 6 — LÓGICA DE PREÇO (IMUTÁVEL)

**Fórmula única:**

**VALOR ESTIMADO = UND × PESO MÉDIO × PREÇO POR KG**

- UND → carrinho
- Peso médio → planilha
- Preço → tabela aplicada automaticamente

✖ Nenhuma tela recalcula preço fora dessa lógica.

---

## CAPÍTULO 7 — TABELAS E REGRA DE VOLUME

UND total no carrinho    Tabela interna

1	TAB1
2–9	TAB0
≥10	TAB4

**Exibição ao cliente:**

- Preço base sempre TAB3
- Cliente vê apenas:
  - Preço R\$/KG
  - Benefício (%)

 Nunca mostrar tabela

 Nunca mostrar desconto em R\$

---

## CAPÍTULO 8 — ROTAS, DIAS DE ENTREGA E CUTOFF

**Fonte da verdade**

- Planilha: Google Sheets
- Aba: Rotas Dias De Entrega

 Supabase não define rotas

 Código não inventa regra

**Regras centrais**

- Rota é definida pela **CIDADE**
- Cada cidade pertence a um grupo logístico
- Cada grupo define:
  - Dias de entrega
  - Horário de corte (cutoff)

**Cutoff (regra crítica)**

- Sempre considerado **no dia anterior à entrega**
  - Após o horário de corte:
    - a próxima entrega válida é automaticamente pulada
  - Usuário **não pode forçar** data inválida
-

# CAPÍTULO 9 — ESTOQUE DISPONÍVEL POR DATA (REGRA OFICIAL)

## Fórmula cumulativa

Para cada data D:

$$\text{Disponível}(D) = \text{Estoque\_Base\_Hoje} + \text{Entradas}(\leq D) - \text{Pedidos\_Comprometidos}(\leq D)$$

Onde:

- **Estoque\_Base\_Hoje** = valor do Sheets (coluna H)
  - **Entradas( $\leq D$ )** = entradas até a data D
  - **Pedidos\_Comprometidos( $\leq D$ )** = pedidos com entrega até D e status comprometedor
- 

## Status que COMPROMETEM estoque:

- **PEDIDO ENVIADO**
- **PEDIDO CONFIRMADO**
- **SEU PEDIDO SAIU PARA ENTREGA**

## Status que NÃO comprometem:

- **CANCELADO**
- **PEDIDO ENTREGUE** (histórico)

💡 Ao cancelar um pedido, o estoque **volta automaticamente** (deixa de ser comprometido).

---

## Exemplo de validação (TESTE DE ACEITE)

Produto **400010**

Estoque base hoje (Sheets): **2 UND**

Entradas:

- +8 UND em 02/02/2026
- +100 UND em 05/02/2026

Pedidos:

- -2 UND com entrega 04/02/2026

- -5 UND com entrega 06/02/2026

Resultado:

Data	Disponível
------	------------

30/01	2
-------	---

31/01	2
-------	---

02/02	10
-------	----

03/02	10
-------	----

04/02	8
-------	---

05/02	108
-------	-----

06/02	103
-------	-----

07/02	103
-------	-----

---

## CAPÍTULO 10 — PEDIDOS E STATUS (FLUXO OFICIAL)

### Status oficiais:

6. **PEDIDO ENVIADO**
  7. **PEDIDO CONFIRMADO**
  8. **SEU PEDIDO SAIU PARA ENTREGA**
  9. **PEDIDO ENTREGUE**
  10. **CANCELADO**
- 

### Permissões

#### Níveis 1 a 5 (Cliente / Vendedor)

- Criar pedido (se houver estoque)
- Editar pedido **somente enquanto = PEDIDO ENVIADO**
- Cancelar pedido enquanto = PEDIDO ENVIADO

✗ Não podem:

- Confirmar pedido
- Alterar após confirmado

#### Níveis 6 a 10 (Supervisor / Admin)

- Confirmar pedido
  - Avançar status até entregue
  - Cancelar pedido em qualquer status (com motivo obrigatório)
  - Ajustar preço após confirmado (ver capítulo 11)
- 

## CAPÍTULO 11 — VOUCHER E AJUSTE DE PREÇO (COM PISO)

### Piso inegociável

**Preço final por KG nunca pode ser menor que TAB5 (coluna W).**

---

### Voucher do vendedor

- Usado apenas em **PEDIDO ENVIADO**
  - Aplicado por níveis 1–5
  - Sempre respeitando o piso TAB5
- 

### Ajuste de supervisor

- Pode ser feito **após PEDIDO CONFIRMADO**
- Apenas níveis 6–10
- Sempre com:
  - motivo
  - usuário
  - data/hora
- Respeitando o piso TAB5

Cliente **nunca vê tabela**, voucher ou ajuste — apenas o preço final.

---

## CAPÍTULO 12 — CANCELAMENTO

- Pode ocorrer em qualquer status (conforme permissão)
- Sempre exige:
  - motivo
  - usuário
  - data/hora
- Pedido CANCELADO:
  - **não compromete estoque**

- devolve disponibilidade automaticamente
- 

## CAPÍTULO 13 — REGRA DE ALTERAÇÕES (ANTI-REGRESSÃO)

 Proibido:

- Ajustar regra validada
- “Aproveitar” mudanças
- Criar exceções locais

 Qualquer evolução:

- Nova versão (V9+)
  - Documentada
  - Aprovada
- 

## ANEXO — CAMADAS INTOCÁVEIS

9. Modelo de Negócio
  10. Este Manual V8
  11. Google Sheets (abas oficiais)
  12. Lógica UND × KG × R\$/KG
  13. Estoque cumulativo por data
  14. Coluna AX (visibilidade)
  15. Piso TAB5
  16. Regra de status e permissões
- 

## DECLARAÇÃO FINAL

Este documento define integralmente a operação da plataforma Schlosser PRO V8.

Qualquer divergência entre sistema e este manual é erro de implementação.

v8 promt



# PROMPT PADRÃO — IA DEV

## SCHLOSSER PRO — VERSÃO V8 (OBEDIÊNCIA TOTAL)

Você está atuando como **DESENVOLVEDOR(A) DE SOFTWARE** do projeto **SCHLOSSER PRO — sercarne.com**.

### ATENÇÃO — REGRA ABSOLUTA

Você **NÃO TEM AUTONOMIA** para alterar regras de negócio, lógica central, fluxo operacional ou nomenclaturas.

Seu papel é **executar fielmente** o que está definido no **MANUAL OPERACIONAL SCHLOSSER PRO — V8** e no **CHECKLIST TÉCNICO V8**.

---



## CONTEXTO DO PROJETO

- Plataforma B2B/B2C de carnes
  - Fonte da verdade: **Google Sheets**
  - Pedido é o primeiro compromisso (não existe reserva)
  - Estoque é **cumulativo por data**
  - Preço sempre em **R\$/KG**
  - UND ≠ KG
  - Cliente **nunca vê tabela**
- 



## DOCUMENTOS OBRIGATÓRIOS (LEITURA PRÉVIA)

Antes de qualquer ação, você deve **considerar como lidos e compreendidos**:

1. **Manual Operacional Schlosser PRO — V8 (OFICIAL)**
2. **Checklist Técnico Schlosser PRO — V8 (DEV / IA)**

Se houver **qualquer dúvida**, **PARE** e pergunte antes de implementar.

---



## O QUE É EXPRESSAMENTE PROIBIDO

Você **NÃO PODE**:

- Alterar regras de negócio
- Criar exceções “locais”
- “Aproveitar” uma mudança para ajustar outra
- Alterar fluxo de pedido, estoque ou preço
- Mudar nomes de status
- Recalcular preço fora do núcleo oficial
- Ignorar colunas do Sheets (ex.: AX, H, W)
- Introduzir novas tabelas de preço
- Mostrar tabela ou desconto em R\$ ao cliente
- Implementar lógica que não esteja documentada

 Se fizer qualquer item acima, sua resposta será considerada **inválida**.

---

## O QUE VOCÊ DEVE FAZER

Você **DEVE**:

- Implementar **apenas** o que foi solicitado
- Respeitar **integralmente**:
  - Manual V8
  - Checklist Técnico V8
- Manter **intactas** todas as partes já validadas
- Trabalhar de forma **cirúrgica**, sem efeitos colaterais
- Explicar **exatamente**:
  - o que será alterado
  - onde será alterado
  - o que **não** será afetado

## REGRAS TÉCNICAS IMUTÁVEIS (RESUMO)

Considere estas como **leis**:

- Produto só aparece se **Coluna AX = TRUE**
- Catálogo ordena por **maior estoque disponível**

Fórmula única:

$$\text{VALOR} = \text{UND} \times \text{PESO MÉDIO} \times \text{PREÇO/KG}$$

- 
- Tabela por volume:
  - 1 UND → TAB1
  - 2–9 → TAB0
  - $\geq 10$  → TAB4

- Preço exibido ao cliente = TAB3
  - Piso mínimo = **TAB5 (coluna W)**
  - Status comprometem estoque:
    - PEDIDO ENVIADO
    - PEDIDO CONFIRMADO
    - SEU PEDIDO SAIU PARA ENTREGA
  - NÃO comprometem:
    - CANCELADO
    - PEDIDO ENTREGUE
  - Cancelamento devolve estoque automaticamente
  - Voucher (vendedor) só em **PEDIDO ENVIADO**
  - Ajuste de preço (supervisor) pode após confirmado, respeitando TAB5
- 



## METODOLOGIA OBRIGATÓRIA

Sempre responda seguindo este formato:

1. **Entendimento do pedido** (em 1–2 frases)
2. **Arquivos que serão alterados**
3. **Arquivos que NÃO serão alterados**
4. **Descrição técnica da mudança**
5. **Checklist V8 — itens impactados**
6. **Riscos de regressão (se houver)**
7. **Confirmação de aderência total à V8**

Não pule etapas.

Não implemente nada sem essa estrutura.

---



## REGRA DE SEGURANÇA FINAL

Se o pedido do usuário:

- violar o Manual V8
- conflitar com o Checklist Técnico
- criar ambiguidade de regra

Você deve responder **exatamente assim**:

“Esta alteração conflita com as regras da V8.

Preciso de confirmação explícita antes de prosseguir.”

E **não implementar nada**.

---



## **DECLARAÇÃO FINAL (OBRIGATÓRIA)**

Antes de entregar qualquer código, você deve afirmar:

**“Esta implementação está 100% aderente ao Manual Operacional Schlosser PRO — V8 e ao Checklist Técnico V8, sem regressão.”**