

MoskoGás

Manual Técnico — v5.0

Versão do Manual: v5.0 (24/02/2026)

Worker Backend: v2.38.2

Empresa: Mosko Ltda — Campo Grande/MS

Sistema: MoskoGás — Gestão de Pedidos Gás/Água

Infraestrutura: Cloudflare Workers + D1 + R2 + GitHub Pages

■ Prioridade absoluta: Velocidade operacional (15-30s por pedido). Poucos cliques. UX para atendente.

■■ 1. ARQUITETURA E INFRAESTRUTURA

Stack Tecnológico

Componente	Tecnologia	Detalhe
Backend	Cloudflare Workers (ES Module)	worker.js — moskogas-backend-v2
Banco de Dados	Cloudflare D1 (SQLite)	moskogas_ops
Storage Fotos	Cloudflare R2	moskogas-comprovantes
Frontend	GitHub Pages	moskogas-app.pages.dev
Repositório	GitHub	github.com/luismosko/moskogas-app
Backup	Cloudflare Worker + Google Drive	moskogas-backup.luismosko.workers.dev
E-mail	Resend API	Relatório diário automático
WhatsApp	IzChat API	chatapi.izchat.com.br
ERP	Bling v3 API	www.bling.com.br/Api/v3
PIX	PushInPay	api.pushinpay.com.br
Contratos	Assinafy	api.assinafy.com.br/v1

URLs Principais

Backend API: <https://api.moskogas.com.br>

Frontend: <https://moskogas-app.pages.dev>

Backup Worker: <https://moskogas-backup.luismosko.workers.dev>

Secrets Cloudflare — Worker Principal

```
BLING_CLIENT_ID, BLING_CLIENT_SECRET, IZCHAT_TOKEN, APP_API_KEY, JWT_SECRET, PUSHINPAY_TOKEN,  
RESEND_API_KEY
```

Secrets Cloudflare — Worker Backup

```
GDRIVE_CLIENT_ID, GDRIVE_CLIENT_SECRET, APP_API_KEY
```

■ 2. AUTENTICAÇÃO E ROLES

Roles do Sistema

Role	Acesso	Páginas
admin	Total	Todas + usuarios.html + config permissões
atendente	Pedidos/gestão	pedido, gestao, pagamentos, relatorio, config, dashboard, consulta, contratos
entregador	Entregas	entregador.html

Segurança

- Senhas com PBKDF2 (100.000 iterações, SHA-256)
- Rate limiting: 5 falhas login / 15min por IP → bloqueio automático
- Token JWT de sessão (24h) salvo em localStorage
- Desativar usuário invalida todas as sessões imediatamente
- Permissões dinâmicas via app_config (admin configura o que atendente pode)
- Atendente pode criar/editar atendentes e entregadores, mas NÃO admins

■ 3. SISTEMA DE PAGAMENTOS

Tipo	Cria Bling?	Marca Pago?	Em Pagamentos?
■ Dinheiro	■ ao entregar	■	■
■ PIX à vista	■ ao entregar	■	■
■ PIX a receber	■ ao entregar	■	■
■ Débito	■ ao entregar	■	■
■ Crédito	■ ao entregar	■	■
■ Mensalista	■ só no lote	■	■
■ Boleto/Órgão	■ só no lote	■	■

■■ REGRA CRÍTICA: Bling só é criado ao marcar ENTREGUE — nunca ao criar o pedido.

Lógica do Worker (ao marcar entregue)

```
const criarBling = ['dinheiro','pix_vista','pix_receber','debito','credito'].includes(tipo); const pago = ['dinheiro','pix_vista','debito','credito'].includes(tipo) ? 1 : 0;
```

PIX PushInPay (v2.34.0+)

Substituiu o sistema Cora PIX. Secret: PUSHINPAY_TOKEN no Cloudflare.

```
POST /api/pix/cashIn → gera QR Code imediato POST /api/pix/webhook → confirma pagamento automaticamente
```

■ 4. STATUS DO PEDIDO

Status	Cor	Significado
NOVO	■ Vermelho	Sem entregador atribuído
ENCAMINHADO	■ Amarelo	Entregador escolhido
WHATS ENVIADO	■ Verde	IzChat confirmou envio
ENTREGUE	■ Azul	Finalizado — cria Bling, exige foto
CANCELADO	■ Cinza	Cancelado (motivo obrigatório)

- Sem restrição rígida de transição: editar, trocar entregador e reverter funciona em qualquer status. • Cancelar/reverter exigem motivo + log de auditoria. • Cancelamento pós-entrega ou reversões por não-admin → alerta WhatsApp ao admin.

■ 5. WHATSAPP SAFETY LAYER (v2.27.0)

■ REGRA ABSOLUTA: TODA mensagem WhatsApp DEVE usar sendWhatsApp(env, to, msg, {category}). NUNCA chamar fetch() direto para a API IzChat.

6 Barreiras de Proteção

#	Barreira	Default	Motivo
1	Circuit Breaker	30min pausa	Se 429/bloqueio → PARA TUDO
2	Horário Comercial	8h-18h BRT	Fora = reportado como spam
3	Rate Limit Global	25/min, 100/h, 200/dia	Envio rápido = ban
4	Intervalo entre msgs	4 segundos	Simula comportamento humano
5	Cooldown por número	12h entre lembretes	Não bombardear cliente
6	Variação de mensagem	Auto em lembretes	Msgs idênticas = spam

Categorias de Mensagem

Categoria	Safety?	Onde usa
entrega	Normal	Notificação ao entregador
lembrete_pix	Normal + cooldown + variação	Cobrança PIX a receber
contrato	Normal	Links de assinatura comodato
admin_alerta	Skip Safety	Alertas cancelamento/reversão
teste	Skip Safety	Endpoint de teste manual

Endpoints Admin

```
GET /api/whatsapp/safety-config – configuração atual  
POST /api/whatsapp/safety-config – alterar limites  
GET /api/whatsapp/stats – envios hoje, bloqueios, circuit breaker
```

■ 6. RELATÓRIO DIÁRIO POR E-MAIL (v2.29.0)

Visão Geral

Envia resumo HTML + CSV dos pedidos do dia anterior por e-mail, automaticamente via cron. Utiliza a API Resend (resend.com). Configurável pelo config.html → seção "Relatório Diário por E-mail".

Configurações Disponíveis

Campo	Descrição	Padrão
Envio Automático	Ativa/desativa o cron diário	Ativo
Hora do Envio	Horário de Campo Grande/MS	03:00 BRT (06:00 UTC)
E-mails de Destino	Um por linha	moskogas@gmail.com
Incluir CSV	Anexa CSV com todos os pedidos	Ativo
Incluir Cancelados	Inclui pedidos cancelados no CSV	Ativo
API Key Resend	Chave re_xxx... da Resend API	Salvo no app_config

Endpoints

```
GET /api/relatorio/email-config – config atual (admin) POST /api/relatorio/email-config – salvar config (admin) POST /api/relatorio/enviar-teste – disparo manual imediato
```

■ RESEND_API_KEY pode ser configurada como secret do Cloudflare OU salva via POST /api/relatorio/email-config com o campo resend_api_key. O worker usa qualquer uma das duas fontes.

■ 7. LEMBRETES PIX (v2.26.0+)

Modos de Envio

- Manual: botão ■ por pedido individual na tela Pagamentos
- Bulk: selecionar vários pedidos PIX → "Enviar Lembretes"
- Cron: automático diário na hora configurada

Melhorias v2.28.1

- Saudação variada automática (bom dia / boa tarde / boa noite)
- Variáveis de template: {nome}, {id}, {itens}, {valor}, {data_entrega}, {ontem}, {chave_pix}
- Delay de 60s entre envios (anti-ban)

Config padrão (app_config key: lembrete_pix)

```
ativo: true/false intervalo_horas: 24 max_lembretes: 3 cron_ativo: true/false cron_hora_utc: 14 (=  
11h BRT) mensagem: template customizável
```

■ 8. CONTRATOS COMODATO (v2.25.0)

Fluxo Completo

Etapa	Ação	Detalhe
1	Criar contrato	Status: draft — dados do comodatário e itens
2	Gerar PDF	A partir do template HTML configurável
3	Upload Assinafy	Envia PDF → recebe URLs de assinatura
4	Envio WhatsApp	Links de assinatura para signatários via IzChat
5	Assinatura	Signatários assinam online pelo link
6	Webhook	Assinafy notifica: document_ready
7	Download	PDF assinado baixado e salvo no R2

Status

```
draft → pending_signatures → signed | canceled
```

Fix v2.28.5

Se um signatário já existe no Assinafy (mesmo e-mail), o worker reutiliza o signer existente em vez de criar um novo — evitando erro 400.

■ 9. IMPORTAÇÃO GLP MASTER (24/02/2026)

O que foi feito

15.981 clientes importados do banco PostgreSQL do GLP Master para customers_cache do D1. 313 endereços de PREFEITURA SAS e SEMED importados para customer_addresses.

Item	Quantidade	Observação
Clientes importados	15.981	Com telefone válido e normalizado
Descartados sem telefone	672	phone_digits é obrigatório (PK)
Descartados fone inválido	90	Menos de 10 dígitos numéricos
Endereços PREFEITURA SAS	66	customer_addresses — phone: 556733144482
Endereços SEMED	247	customer_addresses — phone placeholder: 5567999000001

Normalização de Telefones

(67) 9555-3333 → 8 dígitos após DDD → inserir 9 → 5567995553333 (67) 99555-3333 → 11 dígitos, ok → 5567995553333 Sem telefone ou < 10 dígitos → descartado

Novos Campos em customers_cache

```
ultima_compra_glp TEXT DEFAULT '' -- data última compra no GLP Master origem TEXT DEFAULT 'manual' --
valores: manual | glp_master | bling
```

Uso Futuro

- Filtrar clientes inativos (sem compra há X dias) para campanhas WhatsApp
- Dashboard: "clientes GLP Master que nunca compraram no MoskoGás" = prospecção
- Disparo automático: cliente sem compra há 60 dias → lembrete personalizado
- SEMED: corrigir telefone placeholder 5567999000001 para o número real

■■ 10. SCHEMA DO BANCO D1

Tabelas Principais

Tabela	Função
orders	Pedidos — tabela central do sistema
app_users	Usuários do sistema (admin/atendente/entregador)
auth_sessions	Tokens de sessão JWT
customers_cache	Cache de clientes (phone_digits PK) + campos GLP Master
customer_addresses	Endereços alternativos dos clientes
contracts	Contratos de comodato
contract_signers	Signatários dos contratos
contract_attachments	PDFs dos contratos no R2
contract_events	Log de eventos dos contratos
payment_reminders	Histórico de lembretes PIX enviados
whatsapp_send_log	Log de todos os envios WhatsApp
app_products	Produtos com preços sugeridos MoskoGás
app_config	Configurações dinâmicas (JSON por chave)
bling_tokens	Tokens OAuth Bling (refresh automático)
integration_audit	Auditoria de integrações com Bling
order_status_log	Histórico de mudanças de status dos pedidos
audit_snapshots	Snapshots diários para conciliação
streets_cg	Ruas de Campo Grande/MS para autocomplete
login_attempts	Tentativas de login (rate limiting)

Campos Importantes — customers_cache

```
phone_digits TEXT PRIMARY KEY name TEXT address_line TEXT bairro TEXT complemento TEXT referencia  
TEXT bling_contact_id TEXT cpf_cnpj TEXT email TEXT ultima_compra_glp TEXT DEFAULT '' -- NOVO v2.38.2  
origem TEXT DEFAULT 'manual' -- NOVO v2.38.2
```

Chaves Relevantes — app_config

Chave	Conteúdo
relatorio_email	JSON: ativo, destinos[], hora_utc, incluir_csv, incluir_cancelados
resend_api_key	API key Resend (fallback se secret não disponível)
lembrete_pix	JSON: ativo, intervalo_horas, max_lembretes, cron_ativo, mensagem
google_review_url	URL do perfil Google para QR de avaliação
permissoes	JSON: o que atendente pode fazer (cancelar, reverter, etc.)
contrato_comodante_*	Dados da empresa comodante para contratos

■ 11. BACKUP GOOGLE DRIVE

Item	Valor
Worker	moskogas-backup (Cloudflare Worker separado)
URL	https://moskogas-backup.luismosko.workers.dev
Cron	06:00 UTC = 03:00 BRT (diário)
Conta Google	luismosko@gmail.com
Pasta	MoskoGás Backup
Retenção	180 dias (pastas antigas deletadas automaticamente)

O que exporta

- 12 tabelas D1: orders, customers_cache, customer_addresses, app_users, app_config, order_events, order_status_log, integration_audit, audit_snapshots, bling_tokens, streets_cg, products • Todas as fotos de comprovantes do R2 (prefixo comprovantes/)

Estrutura no Google Drive

```
MoskoGás Backup/ 2026-02-24/ banco_d1_2026-02-24.json resumo_2026-02-24.json fotos/  
pedido_123_1708234567.webp
```

Endpoints

```
GET /google/auth – autorizar Google (1x) GET /google/status – verificar autorização GET  
/backup/executar – forçar manual (requer APP_API_KEY no header) GET /backup/status – último backup  
executado
```

■ 12. VERSÕES DOS ARQUIVOS (24/02/2026)

Arquivo	Versão	Função
pedido.html	v2.11.4	Inserção de pedido
gestao.html	v2.9.0	Gestão + resumo produtos
pagamentos.html	v1.9.6	Pagamentos pendentes + lembretes PIX
config.html	v2.11.3	Configurações gerais + relatório e-mail
contratos.html	v1.1.8	Contratos comodato + Assinafy
entregador.html	v2.6.3	Painel entregador + venda externa
relatorio.html	v1.2.3	Relatórios de entregador
dashboard.html	v1.0.1	Dashboard visual KPIs
consulta-pedidos.html	v1.0.7	Consulta avançada 15+ filtros
auditoria.html	v1.1.0	Auditoria Bling + logs
login.html	v1.2.0	Login
usuarios.html	v1.4.1	Gestão usuários (admin)
shared.js	v1.10.0	Auth, API, toast, nav, loading
worker.js	v2.38.2	Backend completo (Cloudflare Worker)
backup-worker.js	v1.1.0	Backup Google Drive (worker separado)

■■■ VERSIONAMENTO OBRIGATÓRIO: Ao editar qualquer HTML, atualizar versão em 3 LUGARES: , e badge. Em JS, atualizar comentário // vX.Y.Z no topo do arquivo.

■ 13. CHANGELOG WORKER.JS

Versão	Mudanças
v2.38.2	customers_cache: campos ultima_compra_glp + origem para importação GLP Master
v2.38.1	PushInPay PIX (substituiu Cora) + webhook + force lembrete
v2.34.0	PushInPay — integração inicial PIX (substituição do Cora)
v2.32.x	Índices customers_cache (performance 10k+ clientes) + reset circuit breaker
v2.31.0	Cora PIX — cobrança automática, QR code, webhook pagamento
v2.30.0	WhatsApp troca entregador + Venda externa + QR avaliação Google
v2.29.0	Relatório Diário por E-mail via Resend + CSV anexo + cron automático
v2.28.5	Fix Assinafy — reusa signer existente se e-mail já cadastrado (evita erro 400)
v2.28.3	Fix WhatsApp — formatPhoneWA automático em sendWhatsApp + erro detalhado
v2.28.1	Lembretes PIX — saudação variada, {ontem}/{chave_pix}, delay 60s anti-ban
v2.28.0	Produtos — icon_key (upload ícone R2) + reorder endpoint + serve público
v2.27.1	Remove assinatura/fechamento e opt-out das mensagens WhatsApp
v2.27.0	WhatsApp Safety Layer — anti-ban, circuit breaker, rate limit, categorias
v2.26.0	Lembretes PIX — payment_reminders, envio manual/bulk/cron, config admin
v2.25.x	Módulo Contratos Comodato — schema, endpoints, Assinafy, IzChat
v2.24.0	Último pedido cliente + app_products (preços sugeridos MoskoGás)
v2.23.x	Produtos favoritos — tabela product_favorites + endpoints
v2.22.x	Dashboard endpoint (KPIs, status, produtos, pagamentos, hora)
v2.21.0	Rate limiting login, troca de senha, permissões atendente expandidas
v2.20.0	Consulta avançada (15+ filtros, paginação, resumo)
v2.19.x	Permissões dinâmicas + foto WebP defaults (1200px 85%)
v2.18.0	Config dinâmica (app_config)
v2.17.x	Bling só ao ENTREGAR + Consumidor Final padrão
v2.16.x	Reabrir/cancelar com motivo + auditoria status + WhatsApp admin
v2.15.0	Entrega com foto obrigatória (R2)

■ 14. REGRAS DE UX E ERROS A EVITAR

Regras de UX (SEMPRE SEGUIR)

- Modais NUNCA fecham ao clicar fora — Só por botão X, Cancelar ou Salvar
- Toasts grandes — Fundo colorido, animação slide-in, duração 3s
- Tooltips — title em todos os botões de ação
- Redirect após salvar pedido → gestao.html (delay 1.2s)
- Consumidor Final — Sem endereço obrigatório na edição
- Versão visível — Badge em todas as páginas
- Cidade hardcoded — NUNCA exibir campos cidade/UF (sempre Campo Grande/MS)
- Foto obrigatória — Entregador precisa de foto (admin pode pular)
- Formato foto — WebP 1200px 85% quality + sharpen
- Check Bling — Silencioso a cada 60s com auto-recovery
- Loading overlay — showLoading() em TODA ação assíncrona
- Botão X — Todos os modais devem ter botão X no canto

Erros Conhecidos (NÃO REPETIR)

■ Usar /nfce ou /nfces — NÃO EXISTE na API Bling v3

- Esquecer versão → Atualizar nos 3 LUGARES (title, h1, badge)
- Usar pedido_numero ao invés de bling_pedido_id
- Form reset sem null check → sempre if (el) el.value = "
- Modal fecha ao clicar fora → usar shared.js
- Token Bling sem recovery → implementar check silencioso
- Sem parseInt/parseFloat → items_json pode ter strings
- Comparar created_at (epoch) com string texto → usar epoch com offset BRT
- Service Account Google em conta gratuita → não funciona, usar OAuth pessoal
- Chamar API IzChat diretamente → SEMPRE usar sendWhatsApp() com Safety Layer
- Enviar WhatsApp sem category → SEMPRE passar { category: 'xxx' }
- Loop de envio sem verificar blocked → SEMPRE checar result.blocked e parar
- RESEND_API_KEY só como secret → salvar também via endpoint email-config

■ 15. WORKFLOW DE DEPLOY

Passos do Deploy

Passo	Ação	Detalhe
1	Incrementar versão	Todos os arquivos editados — 3 lugares no HTML!
2	Testar local	wrangler dev (backend) + Live Server (frontend)
3	Deploy Worker	Cloudflare Quick Edit ou wrangler deploy worker.js --name moskogas-backend-v2
4	Git push	Claude faz push via HTTPS+token (ghp_xxx) — INCLUIR worker.js
5	Verificar	Checar badge de versão no sistema em produção

Wrangler — Comandos Úteis

```
# Login (1x) wrangler login # Deploy worker wrangler deploy worker.js --name moskogas-backend-v2 #
Executar SQL no D1 wrangler d1 execute moskogas_ops --file=script.sql --remote # Ver quem está logado
wrangler whoami
```

■ worker.js no GitHub = fonte da verdade. Sempre clonar o repo para saber a versão atual em produção ao iniciar uma nova sessão.