

MGConnect ADM ↔ Albert-View

V4.7 · 2026-07-31 · Contrato canônico (pós-JWT / API Externa)

SPEC unificada — MGConnect ADM ↔ Albert View / API Externa

Versão: 4.7 · Data: 2026-07-31

Status: Contrato canônico para desenvolvimento

Fontes: V3.1 sem JWT (§4.7–4.9 obrigatórios no login), SPEC cadastro v1.8, DIFF ADM, TODO motoristas · View 2026-07

Foco: login perms; listas e livres; perfil completo; vínculo View; mídia; histórico; busca; mode driver; aceite de política; robustez operacional.

Documento irmão (só View): CONTRATO_API_Albert_View_V4.md

0. Prompt de sistema (para a equipe)

Você implementa a integração MGConnect ADM ↔ Albert View via API Externa Supabase.

REGRAS INEGOCIÁVEIS:

- 1. NÃO usar JWT como entrada na View. JWT foi ELIMINADO. Entrada = POST /auth/login (Bearer mgc_live_...) + sessão interna Flask no View (cookie, TTL ~7d).
- 2. ADM autentica e é fonte de cadastro (frota, empresa, usuário, motorista).
- 3. View NÃO autentica por si; NÃO cria motorista; só consome API + telemetria local.
- 4. Tamanhos de string = limites Albert hoje. Overflow → HTTP 400 em PT; nunca truncar.
- 5. NÃO ampliar schema IoT Albert nesta fase.
- 6. Série IoT / modelo IoT / HW = Tacita (read-only no ADM). SW = servidor (dinâmica).
- 7. Composição cavalo/carretas = Albert Admin (read-only no ADM frota).
- 8. Vínculo veículo↔motorista = bidirecional (View e ADM). Cadastro motorista = só ADM.
- 9. Login OBRIGATÓRIO: mode + permissions_all + permissions[] (catálogo V3.1 §4.7 exato). Sem isso o login é INCOMPLETO para a View (não aceitar como contrato final).
- 10. View PODE listar veículos, motoristas, vínculos, livres; consultar perfil completo do veículo; VINCULAR e DESVINCULAR motorista↔veículo. View NÃO cadastra motorista.
- 11. view_plan / plan_name complementam mode; não substituem permissions[].
- 12. Alias EN legado ok se shape e max forem os mesmos.

Base API: https://gacequockmkpeidzbota.supabase.co/functions/v1/external-api
Auth: Authorization: Bearer <mgc_live_...>

1. Arquitetura (pós-JWT)

- [Browser] → entrada View (entrada.html)
 - POST External /auth/login (View backend com API-key)
 - Flask session (sub, mode/plan, escopo, permissões, plan_*)
 - UI albertviewui_v2_1 (+ PWA)
 - /view/api/* (telemetria / warehouse Albert)

[MGConnect ADM] → CRUD frota/empresa/usuário/motorista → External API
[Tacita] ← bridge Albert → series-iot (listar/casar)
[Albert Admin] → composição_veículo + SW dinâmica → espelho na External API

Componente	Responsabilidade
MGConnect ADM	Credenciais; cadastro frota/empresa/usuário/motorista; casar série Tacita; emitir payload de login coerente (company, plan, scope, perms, vehicles)
API Externa	Contrato HTTP; validação de max; login; listas; vínculos; bridge Tacita/composição
Albert View	Sessão interna; UI Pro/Max; filtro sigla/placa; (fase 2) enforcement perms; telemetria
Tacita	Acervo série + MAC + modelo_iot + versao_hw
Albert Admin / servidor	SW corrente; composição cavalo/carretas; UI “Alberts disponíveis para instalação” (lista Tacita via bridge — não cria identidade)

Eliminado (V3.1): /mgconnect/auth/jwt, claims JWT, MGCONNECT_JWT_SECRET como entrada, iss/aud/exp de token, redirect ?token=. **Mantido da V3.1 (semântica):** modo operacional, catálogo de permissões de UI, escopo sigla/placa, sessão interna, status_cmd / campos operacionais compartilhados.

2. Entrada autenticada (substituto do JWT)

OBRIGATÓRIO no login: mode (§4.8) + permissions_all + permissions[] (§4.7) + escopo. Exemplos de perfil §4.9 definem combinações válidas. **Sem mode / permissions_all / permissions no formato View, a resposta de login NÃO atende esta SPEC.** JWT permanece eliminado.

2.1 Fluxo

- Usuário digita e-mail/senha na **View** (/entrada.html).
- Backend View (POST /view/api/ui_login ou form) chama External:
- POST /auth/login body {"email","password"} + Bearer da **chave de serviço**.
- ADM/API devolve usuário + empresa + **mode** + **permissions_all** + **permissions[]** (§2.5–2.7) + escopo + frota.
- View grava **sessão Flask** e abre UI. Continuação = cookie (não revalidar login a cada tela).
- Exceção: mgconnect1 httpasswd (manutenção) → mode=max, permissions_all=true, escopo total.

2.2 Contrato mínimo do POST /auth/login (resposta 200)

Campos **obrigatórios para a View** (além de autenticar):

Campo	Obrig.	Descrição		
authenticated	sim	true		
user.id	sim*	company_user UUID (*quando houver vínculo de empresa)		
user.auth_user_id	sim	UUID auth		
user.full_name/email	sim	exibição + sub		
mode	sim	modo operacional §2.6 (exatos; sem alias)		
permissions_all	sim	boolean		
permissions	sim	array de strings do catálogo §2.5; vazio se permissions_all=true		
siglas_all/siglas	sim	escopo sigla		
placas_all/placas	sim	escopo placa		
company	cond.	empresa + plan_id / plan_name / opcional view_plan		
vehicles	sim	array (pode []; View faz GET se vazio)		
source	rec.	plan_role_permissions\	role_permissions\	none

```
{
  "authenticated": true,
  "source": "plan_role_permissions",
  "ver": 4,
  "user": {
    "id": "03b97de6-19e2-4348-bd59-ed108f73e563",
    "auth_user_id": "40224d06-2b27-4a08-b737-8d2e1a6daa94",
    "full_name": "Klaus Liesenberg",
    "email": "liesenbergk@gmail.com",
    "status": "ativo",
    "profile_role": "operacional",
    "cpf": null,
    "phone": null
  },
  "mode": "pro",
  "permissions_all": false,
  "permissions": [
    "view_dashboard",
    "view_charts",
    "view_routes",
    "view_summary"
  ],
  "siglas_all": false,
  "siglas": ["MGC"],
  "placas_all": true,
  "placas": [],
  "company": {
```

```

    "id": "bf0d8c75-5405-476d-9ad6-9ad8ff53c8d9",
    "cnpj": "12345678000190",
    "sigla": "MGC",
    "nome_empresa": "MGConnect",
    "razao_social": "MGConnect Telemetria IA Premium",
    "nome_fantasia": "MGConnect",
    "status": "ativo",
    "is_blocked": false,
    "plan_id": "1dcba507-e051-4a9d-94c1-20ff396f79b2",
    "plan_name": "MGConnect Pro",
    "view_plan": "pro"
  },
  "vehicles": []
}

```

Aliases de escopo aceitos na leitura (View normaliza): escopo_frota.siglas_todas ≡ siglas_all; lista_siglas ≡ siglas; idem placas. Preferir o shape **plano** acima (como claims V3.1).

Regras de escopo (V3.1 §4.6, mantidas):

Caso	Payload
Todas as siglas	"siglas_all": true, "siglas": []
Siglas restritas	"siglas_all": false, "siglas": ["MGT", "SP0"]
Todas as placas	"placas_all": true, "placas": []
Placas restritas	"placas_all": false, "placas": ["ABC1D23"]
Todas as perms View	"permissions_all": true, "permissions": []
Perms parciais	"permissions_all": false, "permissions": ["view_routes", "view_predictive"]

2.5 Catálogo oficial de permissões View — V3.1 §4.7 (ESSENCIAL)

O conjunto **oficial e exclusivo** reconhecido pelo Albert View:

Permissão (string exata)	Módulo UI
view_dashboard	painel de instrumentos
view_charts	gráficos
view_routes	jornadas & rotas
view_diagnostics	diagnóstico do ciclo
view_predictive	predição ML
view_alerts	eventos & notificações
view_summary	resumo / mural
view_driving	condução / score / motorista (UI)

Regras absolutas (ADM deve emitir exatamente estes nomes):

- **Sem variações**
- **Sem aliases** (não enviar dashboard, routes, view_driver no lugar de view_driving)
- **Sem abreviações**
- Se permissions_all === true → array permissions **vazio** (View libera todos os módulos do catálogo, sujeito ao mode)
- Se permissions_all === false → só entram strings presentes em permissions[]; o resto da UI de módulo fica oculto / bloqueado
- String **fora do catálogo** → View ignora e pode logar; ADM não deve emitir lixo

Mapeamento UI legado (já implantado na View):

Permissão	Botão / área
view_dashboard	painel de instrumentos
view_charts	gráficos
view_routes	jornadas & rotas
view_diagnostics	diagnóstico do ciclo

Permissão	Botão / área
view_predictive	predição ML
view_alerts	eventos & notificações
view_summary	resumo
view_driving	página / score motorista

Enforcement de backend nos endpoints sensíveis (ML, diagnóstico, alertas, agente) = **obrigatório** na View em fase seguinte; UI já deve ocultar.

2.6 Regras dos modos operacionais — V3.1 §4.8 (ESSENCIAL)

Campo login: mode (string exata). Valores válidos:

mode	Comportamento
super	Acesso completo de visualização . Sem edição.
super_agent	Igual super + agente Albert.
admin	Acesso completo + edição autorizada na View.
admin_agent	Igual admin + agente .
pro	Página inicial, cards, minicards, seletores, mapa . Clique no veículo → abre mapa . Não abre modal do veículo. Diagnóstico capa bloqueado .
pro_agent	Igual pro + agente .
max	Acesso ao modal do veículo. Módulos dentro do modal = permissions / permissions_all. Diagnóstico permitido se permissão/módulo ok.
max_agent	Igual max + agente .
driver	Abre direto Página do Motorista / Gamificação. Sem navegação tradicional (capa/frota/modal).

Interação mode × permissions:

- mode define **casca** de produto (pro/max/driver/super/admin).
- permissions[] define **módulos** disponíveis **dentro** do max (e, no futuro, granulosidade em outros modos).
- mode=pro **não** libera modal mesmo com permissions_all=true.
- mode=driver ignora navegação tradicional mesmo com permissões de view_*.
- mode=super|admin + permissions_all=true = visão (e edição no admin) completa.

Plano comercial vs mode:

Fonte	Uso
mode	Autoridade de UX View (obrigatório no login)
company.view_plan / plan_name	Comercial ADM; se mode ausente, View deriva pro/max de view_plan/plan_name como fallback
Conflito	Prevalece mode se presente e válido

2.7 Exemplos de perfis — V3.1 §4.9 (ESSENCIAL)

Forma antiga no JWT (ver/sub/name/...). **Mesmos campos no login** (sem iat/exp/iss/aud).

Super usuário MGConnect

```
{
  "authenticated": true,
  "ver": 4,
  "user": { "id": "mgc_admin", "full_name": "Administrador MGConnect", "email":
"admin@mgconnect.online" },
  "mode": "super",
  "siglas_all": true,
  "siglas": [],
  "placas_all": true,
  "placas": [],
  "permissions_all": true,
  "permissions": [],
  "company": null,
  "vehicles": []
}
```

Proprietário de frota (max parcial)

```
{
  "authenticated": true,
  "ver": 4,
  "user": { "id": "owner_1", "full_name": "Proprietário", "email": "owner@frota.com" },
  "mode": "max",
  "siglas_all": false,
  "siglas": ["MGT"],
  "placas_all": true,
  "placas": [],
  "permissions_all": false,
  "permissions": [
    "view_routes",
    "view_predictive",
    "view_summary",
    "view_driving"
  ],
  "company": {
    "sigla": "MGT",
    "nome_empresa": "MGTech",
    "plan_name": "MGConnect Max",
    "view_plan": "max"
  },
  "vehicles": []
}
```

Mecânico (max, tudo liberado nos módulos)

```
{
  "authenticated": true,
  "ver": 4,
  "user": { "id": "mec_1", "full_name": "Mecânico" },
  "mode": "max",
  "siglas_all": false,
  "siglas": ["MGT"],
  "placas_all": true,
  "placas": [],
  "permissions_all": true,
  "permissions": []
}
```

Motorista (só gamificação)

```
{
  "authenticated": true,
  "ver": 4,
  "user": { "id": "drv_1", "full_name": "Motorista" },
  "mode": "driver",
  "siglas_all": false,
  "siglas": ["MGT"],
  "placas_all": false,
  "placas": ["ABC1D23"],
  "permissions_all": false,
  "permissions": []
}
```

Pro (operacional MGConnect — espelho live)

```
{
  "authenticated": true,
  "ver": 4,
  "user": {
    "id": "03b97de6-...",
    "full_name": "Klaus Liesenberg",
    "email": "liesenbergk@gmail.com"
  },
  "mode": "pro",
  "siglas_all": false,
  "siglas": ["MGC"],
  "placas_all": true,
  "placas": [],
  "permissions_all": false,
  "permissions": [
    "view_dashboard",
    "view_charts",
    "view_summary"
  ],
  "company": {
```

```

    "plan_name": "MGConnect Pro",
    "view_plan": "pro",
    "sigla": "MGC",
    "nome_empresa": "MGConnect"
  },
  "vehicles": []
}

```

2.8 Sessão interna View (mapeamento a partir do login)

Sessão Flask / UI	Campo login	
sub	user.email ou user.id	
name	user.full_name	
mode	mode (§2.6)	
plan	derivado: pro se mode ∈ {pro,pro_agent}; max se max/max_agent/super/admin...; driver separado	
permissions_all	permissions_all	
permissions	permissions[] (catálogo §2.5)	
siglas / siglas_all	idem	
placas / placas_all	idem	
adm_plan_*	company.plan_*	
av_login	external\	htpasswd

TTL cookie ~7 dias. Mesma sessão /app e PWA.

2.9 Fallback pós-login (View)

Situação	Ação
vehicles=[]	GET /vehicles
mode ausente	derivar de company.view_plan / plan_name; senão max
permissions ausente e permissions_all ausente	tratar como permissions_all=true só se mode ∈ {super,admin,max} e política local; senão lista vazia + log
company/plan fraco	GET /permissions?user_id= (enriquecer mode / permissions View se a API expuser; não inventar view_* — ADM deve corrigir o login)

3. Resumo mode × permissions (referência rápida)

mode	Modal veículo	Agente	Perms de módulo
pro / pro_agent	não	só *_agent	opcional (capa/mapa)
max / max_agent	sim	só *_agent	sim — filtra abas do modal
super / super_agent	sim (view)	só *_agent	all se permissions_all
admin / admin_agent	sim + edição	só *_agent	all se permissions_all
driver	N/A (vai p/ motorista)	não (salvo evolução)	—

4. Fontes da verdade

Domínio	Fonte	Quem edita
Frota (placa, marca, modelo, sigla, frota, tipo caminho...)	MGConnect ADM	ADM
Série IoT + MAC + modelo IoT + HW	Tacita	provisionamento / set-meta
Versão SW	Servidor Albert	OTA / contr / Admin
Composição cavalo/carretas	Albert Admin	Albert Admin
Cadastro motorista (Nome+CPF)	MGConnect ADM	só ADM

Domínio	Fonte	Quem edita
Vínculo veículo↔motorista	API Externa	ADM e View (last-write-wins)
Telemetria	warehouse Albert	—
Max de strings	Albert atual	ADM adapta; não mudar IoT

5. Política de tamanhos (canônica)

API e ADM: **recusar > max (400 PT)**. Não truncar. Não ampliar colunas IoT.

Campo	Max	Notas
número de série	20	só Tacita
sigla	3	upper
placa	8	upper [A-Z0-9] 7–8
marca	10	ex. Mercedes-Benz inválido
modelo (veículo)	10	≠ modelo IoT
frota	20	
nome_empresa	40	ao lado da sigla; ≠ razão social
razão social	120	só legal
nome fantasia	80	só legal
classe / grupo / ano / motor / tração	20 / 30 / 4 / 20 / 15	subcampos; View compõe classe-grupo-ano-motor-tracao
modelo_iot / descrição equip.	15	Tacita
versao_hw / versao_sw	10	HW Tacita; SW servidor
chassi / cor / tipo_veiculo / nº motor	30 / 20 / 30 / 40	ADM
nome completo motorista / CPF	80 / 14	
e-mail / senha / UUID / CNPJ	120 / 128 / 36 / 18	
veículos no login	500	
permissões	50	

Normalização: placa/série/sigla upper; unicidade série entre ativos; placa única por empresa.

6. Objetos principais

6.1 Veículo operacional

Campos editáveis ADM: placa, marca, modelo, ano_fabricacao, ano_modelo, chassi, cor, tipo_veiculo, numero_motor, status_veiculo, sigla, nome_empresa, frota, classe, grupo, ano, motor, tracao. status_veiculo: ativo|inativo|bloqueado|manutencao. Marca/modelo obrigatórios se ativo.

Equipamento (read-mostly): numero_serie (casar Tacita), modelo_iot/descricao, versao_hw (Tacita), versao_sw (servidor), status.

Composição (composicao_veiculo): ativa, cavalo {papel, ordem, placa≤8, identificador≤20, tipo_unidade≤20, descricao≤40}, carretas[] (0–5), atualizado_em, observacao≤80. Fonte Albert Admin; ADM/View **somente leem**.

6.2 Empresa

id, cnpj, razao_social, nome_fantasia, sigla, nome_empresa, status, is_blocked, plan_id, plan_name, view_plan (desejável), frota padrão.

6.3 Usuário

id (company_user), auth_user_id, full_name, email, status, profile_role, cpf, phone.

6.4 Motorista (cadastro = Nome Completo + CPF)

Campo	Obrig.	Max		
nome_completo	sim	80		

Campo	Obrig.	Max		
cpf	sim	14		
sigla	sim	3		
status	sim	20 ('ativo	inativo	bloqueado')
id	resposta	36		

Fora do contrato mínimo de cadastro: e-mail, telefone, perfil (podem existir no ADM interno). Unicidade: (sigla, cpf) entre ativos.

6.5 Vínculo

1 veículo ↔ no máximo 1 motorista vigente; 1 motorista ↔ no máximo 1 veículo; **mesma sigla**. origem: mgconnect_adm | albert_view. Last-write-wins.

Campos: id, sigla, id_veiculo/placa, id_motorista, espelhos nome/cpf, vigente, origem, atualizado_em, atualizado_por.

7. Endpoints — View (leitura + vínculo) e ADM

Paths canônicos preferidos em **PT**. EN live (/vehicles, /drivers, ...) aceitos como **alias** se o shape for o desta SPEC. Toda listagem respeita o **escopo da sessão** (siglas/placas do login). API-key de serviço + contexto de usuário conforme implementação (sessão View no backend).

7.0 Pré-condições gerais (todas as listas / perfil / vínculo)

Condição	Regra
Autenticado	Sessão View válida ou chamada ADM com chave + usuário autorizado
Escopo	Filtrar por sigla / placas do login; siglas_all / placas_all ampliam
sigla	Query obrigatória nas listas "por empresa/frota", max 3 , upper
Paginação	limite (1-500, default 100) + deslocamento (≥0) ou limit/offset
403	Fora do escopo ou sem permissão de modo
422	sigla inexistente / placa não da sigla

7.1 Listas que a View deve poder pedir

A) Lista de veículos (completa da sigla)

GET /veiculos?sigla=MGC&status=ativo&limite=100&deslocamento=0
GET /vehicles?sigla=MGC&status=ativo&limit=100&offset=0 # alias

Query	Obrig.	Notas	
sigla	sim	escopo	
status	não	default ativo (ou todos se omitido — documentar default)	
tem_vinculo	não	true \	false filtra só vinculados / só livres
limite / deslocamento	não	paginação	

Cada item inclui: campos cadastrais do veículo (§6.1 + §7.4) + tem_vinculo (boolean) + motorista_vinculado { id, nome_completo, cpf } \ null.

B) Lista de motoristas (completa da sigla)

GET /motoristas?sigla=MGC&status=ativo&limite=100&deslocamento=0
GET /drivers?sigla=MGC&status=ativo&limit=100&offset=0 # alias

Query	Obrig.	Notas	
sigla	sim		
status	não	default ativo	
tem_vinculo	não	true \	false

Query	Obrig.	Notas	
limite/ deslocamento	não		

Cada item: id, nome_completo, cpf, sigla, status + tem_vinculo + veiculo_vinculado { id, placa } (+ opcional marca/modelo) \} null.

C) Lista de vínculos veículo ↔ motorista

GET /vinculos-motorista?sigla=MGC&vigente=true

Query	Obrig.	Notas
sigla	rec.	filtrar empresa
placa	não	
id_motorista	não	
id_veiculo	não	
vigente	não	default true

Item: objeto §6.5 (id, sigla, id_veiculo, placa, id_motorista, nome_completo, cpf, vigente, origem, atualizado_em, ...).

Consulta pontual (alias live): GET /driver-vehicle-link?driver_cpf=&plate=&cnpj= → { linked, assignment, driver, vehicle, company }. Útil para checagem 1:1; **não substitui** a lista §C.

D) Lista de veículos livres (sem motorista vigente)

GET /veiculos?sigla=MGC&tem_vinculo=false&status=ativo
ou, se preferirem path dedicado:
GET /veiculos/livres?sigla=MGC

Semântica: veículos da sigla no escopo com tem_vinculo=false (disponíveis para vincular). Mesmo shape da lista A, com motorista_vinculado: null.

E) Lista de motoristas livres (sem veículo vigente)

GET /motoristas?sigla=MGC&tem_vinculo=false&status=ativo
ou:
GET /motoristas/livres?sigla=MGC

Semântica: motoristas ativos da sigla com tem_vinculo=false. Mesmo shape da lista B, com veiculo_vinculado: null.

Contrato mínimo: path dedicado **ou** filtro tem_vinculo=false — um dos dois **obrigatório** e estável. Preferência da SPEC: **filtro na lista** (+ path dedicado opcional como atalho).

7.2 Perfil completo do caminhão / veículo (query de detalhe)

A View deve obter **todos os dados cadastrais do ADM**, inclusive campos novos, em uma leitura.

GET /veiculos/{id}
GET /vehicles/{id} # alias
opcional por placa+sigla:
GET /veiculos?sigla=MGC&placa=NS02GBF # retorna 1 item completo

Resposta data — objeto veículo completo:

Bloco	Campos (não exaustivo se ADM acrescentar; API versiona)
Identidade	id, id_empresa, placa, status_veiculo / status
Cadastro frota	marca, modelo, ano_fabricacao, ano_modelo, chassi, cor, tipo_veiculo, numero_motor
Empresa/frota	sigla, nome_empresa, frota
Tipo caminhão	classe, grupo, ano, motor, tracao
Equipamento IoT	equipamento: numero_serie, modelo_iot/descricao, versao_hw, versao_sw, status
Composição	composicao_veiculo (§1A / §6.1)
Vínculo	tem_vinculo, motorista_vinculado
Operacional	status_cmd se exposto (0/1/2)
Meta	created_at, updated_at

Regra: campos novos do ADM no cadastro de veículo **devem aparecer neste GET** (forward-compatible: View ignora desconhecidos; ADM documenta no OpenAPI). Max de strings = \$5. Overflow na escrita = 400; na leitura devolve o valor persistido.

7.3 Vincular e desvincular (Albert View e ADM)

O ADM **já** faz vínculo. A View **deve** usar os **mesmos** endpoints de escrita.

Criar / substituir vínculo vigente

PUT /vinculos-motorista
Content-Type: application/json

```
{
  "sigla": "MGC",
  "id_veiculo": "...",
  "placa": "NS02GBF",
  "id_motorista": "...",
  "origem": "albert_view"
}
```

Campo	Obrig.	Notas
sigla	sim	mesma sigla veículo e motorista
id_veiculo ou placa+sigla	sim	
id_motorista	sim	
origem	sim	View → albert_view; ADM → mgconnect_adm

Efeitos:

- encerra vínculo vigente anterior do veículo e do motorista;
- cria o novo;
- last-write-wins;
- 409 se entidades de siglas diferentes / inativos / fora de escopo.

Remover vínculo

DELETE /vinculos-motorista
Content-Type: application/json

```
{
  "sigla": "MGC",
  "id_veiculo": "...",
  "origem": "albert_view"
}
```

Alternativa: body/query com id_motorista + sigla (em vez de veículo). Resultado: veículo e motorista ficam **livres** (tem_vinculo=false).

Quem pode o quê

Ação	Albert View	MGConnect ADM
Listar veículos / motoristas / vínculos / livres	sim	sim
Perfil completo do veículo	sim (leitura)	sim
Vincular / desvincular	sim	sim (já existe)
CRUD cadastro motorista (Nome+CPF)	não	sim
CRUD cadastro veículo	não	sim

Escopos de chave: vehicles:read, drivers:read, assignments:read, assignments:write (View e ADM). drivers:write **somente** ADM (cadastro).

7.4 Alias live (estado atual da API — não é o contrato final)

Live hoje	Uso
POST /auth/login	existe; ainda sem mode/permissions §4.7 — gap P0

Live hoje	Uso
GET /vehicles, /drivers	listas parciais EN; falta sigla+tem_vinculo canônico
GET /driver-vehicle-link	consulta pontual OK
PUT/DELETE /vinculos-motorista	a implementar se não existir (ADM pode ter UI local — View precisa da API)

7.5 Tacita / série (ADM; View só lê no perfil)

Método	Path	Função
GET	/series-iot	inventário
GET/POST	/series-iot/verificar	pode_casar
POST	/series-iot/casar · /descasar	só ADM

7.6 Diagrama de uso na View

Login (mode + permissions §4.7-4.9 + escopo)

```

■ GET /veiculos?sigla= (frota + tem_vinculo)
■ GET /motoristas?sigla= (equipe + tem_vinculo)
■ GET /veiculos?tem_vinculo=false (livres p/ vincular)
■ GET /motoristas?tem_vinculo=false
■ GET /vinculos-motorista?sigla=
■ GET /veiculos/{id} (perfil completo ADM)
■ PUT /vinculos-motorista (vincular) origem=albert_view
■ DELETE /vinculos-motorista (desvincular) origem=albert_view

```

8. Tacita e casamento de série (bridge)

8.1 Princípios

#	Regra
1	Tacita = fonte da verdade do acervo (série existe? MAC? modelo IoT? HW?)
2	Casamento série↔veículo = estado de negócio somente no Supabase (ADM/API) . Não grava placa/veículo no SQLite Tacita
3	External API (Supabase) nunca abre <code>tacita.db</code> nem vê <code>nyx/fek</code>
4	Bridge no servidor Albert (rede privada) é o único processo que lê Tacita para a API (inventário)
5	Série inventada no formulário ADM = proibido ; só escolher da lista / verificar
6	Só casar se <code>pode_casar=true</code> ou política de substituição no mesmo veículo (§8.6)
7	View só lê <code>numero_serie</code> + metadados no perfil; não casa/descasar
8	Recasar no mesmo ou outro veículo: substitui o vínculo anterior (não 409 por "já tem série" no alvo)
9	Descasar zera <code>numero_serie</code> no veículo (Supabase). O IoT continua no Tacita e pode ser recondicionado → voltar a disponível
10	Estado reservado existe já na v1 (hold ADM com TTL)
11	ADM pode eliminar IoT (quebrado / roubado / extraviado): série nunca mais casável; se estava casada, descasa e zera o veículo

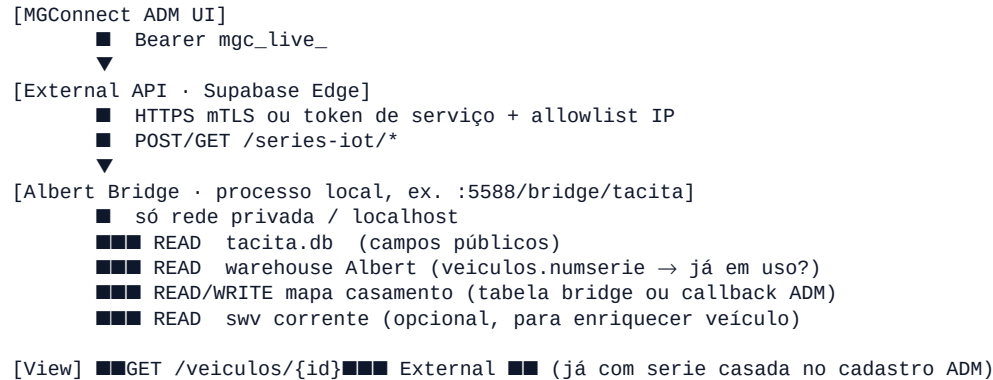
8.2 O que existe hoje (ops)

Item	Valor
Host	servidor Albert (~ / tacita /)
DB	<code>tacita.db</code> · tabela <code>tacita_identity</code>
CLI	<code>tacita</code> → <code>Tacita_v_1_2.py</code>
Domínio default	Albert2
Colunas públicas	<code>num_serie</code> , <code>mac</code> , <code>domain</code> , <code>modelo_iot</code> , <code>versao_hw</code> , <code>created_at</code>
Colunas secretas (nunca API)	<code>nyx</code> , <code>fek</code> , <code>expected_hash</code> , <code>fek_burned_at</code> , <code>k2_version</code>
Escala atual	~87 séries provisionadas (ex. 37GBF, modelo Albert 1, HW HWv 1.0)

Provisionamento (ops, **não** via External):

```
tacita --num-serie XXX --mac aa:bb:cc:dd:ee:ff --modelo-iot "Albert 1" --versao-hw "HWv 1.0"
tacita set-meta --num-serie XXX --modelo-iot "Albert 1" --versao-hw "HWv 1.0"
tacita --show XXX
```

8.3 Arquitetura do bridge



Componente	Responsabilidade
External /series-iot/* + DB Supabase	Auth; única fonte do casamento (série↔id_veículo); chama bridge só para inventário Tacita
Bridge Albert	Lê Tacita (público); não é dono do mapa de casamento; nunca devolve secrets
ADM	UI seletor/reserva/casar/descasar via External
Albert warehouse	Telemetria por série depois que o ADM casou; se a View/core usar numserie local, pode espelhar limpeza no descasar (opcional ops) — não é fonte do casamento
View	Só lê equipamento no perfil

Segurança bridge: bind 127.0.0.1 ou VPC; BRIDGE_TOKEN; timeout curto; sem porta pública.

8.4 Estados (v1 inclui reservado)

Estado	Significado	pode_casar
disponivel	No Tacita (domínio ok); sem casamento; sem reserva; não eliminada	true
casado	Casamento ativo no Supabase	false
reservado	Hold ADM vigente (TTL)	false*
eliminado	Baixa definitiva no Supabase (quebrado/roubado/extraviado/...); não casável	false
indisponivel	Fora do Tacita, domínio errado, ou bloqueio ops genérico	false

*Exceção: dono da reserva → pode_casar_para_mim=true.

Cálculo (v1):

```
se registro Supabase.eliminado      → eliminado (prioridade)
se num_serie ∉ Tacita (domínio ok)   → indisponivel
senão se casamento Supabase ativo    → casado
senão se reserva não expirada        → reservado
senão                                → disponivel
```

Reserva (v1):

Campo	Regra
Quem	ADM (serials:write)
TTL	default 15 min (configurável 5–60)
Efeito	some da lista somente_disponiveis para outros
Liberar	timeout, POST /series-iot/liberar-reserva, ou casar com sucesso
Body reserva	{ "numero_serie", "id_veiculo?" , "ttl_segundos?" }

8.5 Objeto Série IoT (resposta pública)

```
{
  "numero_serie": "37GBF",
  "mac": "70:4b:ca:69:3d:48",
  "dominio": "Albert2",
  "modelo_iot": "Albert 1",
  "versao_hw": "HWv 1.0",
  "estado": "disponivel",
  "pode_casar": true,
  "id_veiculo": null,
  "placa": null,
  "sigla": null,
  "nome_empresa": null,
  "criado_em": "2026-07-09T16:45:39.740934+00:00"
}
```

Se estado=casado, preencher id_veiculo, placa, sigla, nome_empresa quando conhecidos. **Nunca:** nyx, fek, hashes, paths de output FEK.

8.6 Endpoints External (→ bridge)

Método	Path	Escopo	Função
GET	/series-iot	serials:read	Listar inventário
GET/POST	/series-iot/verificar	serials:read	Existe? pode_casar?
POST	/series-iot/reservar	serials:write	Hold v1 (TTL)
POST	/series-iot/liberar-reserva	serials:write	Solta hold
POST	/series-iot/casar	serials:write	Casa / substitui no veículo
POST	/series-iot/descascar	serials:write	Zera série no veículo; IoT segue casável
POST	/series-iot/eliminar	serials:write	Baixa definitiva (quebra/roubo); ADM only

GET /series-iot

Query	Obrig.	Default	Notas
somente_disponiveis	não	false	true → só pode_casar=true
dominio	não	Albert2	
numero_serie	não	—	filtro exact/prefixo (documentar)
limite	não	100	1–500
deslocamento	não	0	

Resposta: { "total", "limite", "deslocamento", "dados": [SérieIoT, ...] }

Verificar

GET /series-iot/verificar?numero_serie=37GBF&dominio=Albert2
ou POST { "numero_serie", "dominio" }

```
{
  "existe_tacita": true,
  "pode_casar": true,
  "estado": "disponivel",
  "serie": { "numero_serie": "37GBF", "modelo_iot": "Albert 1", "versao_hw": "HWv 1.0", "mac":
  "..."},
  "motivo": null
}
```

Se casada: pode_casar=false, motivo: already_linked, detalhe do veículo.

Reservar (v1)

POST /series-iot/reservar
{ "numero_serie": "37GBF", "id_veiculo": "...", "ttl_segundos": 900 }

- 200: { "estado": "reservado", "reserva_id", "expira_em" }

- 409 se já casado ou reservado por outro

```
POST /series-iot/liberar-reserva
{ "numero_serie": "37GBF" } // ou reserva_id
```

Casar (substitui se já houver série no veículo)

```
POST /series-iot/casar
{
  "numero_serie": "37GBF",
  "id_veiculo": "5c4155cf-...",
  "sigla": "MGC"
}
```

Política fechada (decisão produto):

Situação	Comportamento
Série disponível (ou reservada por mim)	Casa no id_veiculo
Veículo já tem outra série	Substitui: descasa a série antiga (fica disponível se no Tacita) e casa a nova
Série já casada em outro veículo	409 already_linked (não rouba série de outro carro sem descasar antes)
Série não está no Tacita	404 series_not_in_tacita

Passos (transação lógica no Supabase; inventário via bridge):

- Bridge: série existe no Tacita? metadados modelo_iot/versao_hw.
- Supabase: se veículo tinha numero_serie anterior ≠ nova → limpar casamento antigo (série antiga → livre).
- Supabase: gravar casamento **novo** (id_veiculo ↔ numero_serie); zerar reserva.
- Preencher equipamento do veículo: série, modelo_iot, versao_hw (read-only UI).
- **Não** gravar casamento no SQLite Tacita; **não** mexer FEK/NYX.
- 200 com veículo+equipamento atualizados.

Descasar (zera série no veículo; IoT permanece no mundo)

```
POST /series-iot/descasar
{ "numero_serie": "37GBF" }
# ou { "id_veiculo": "..." }
```

Política fechada:

- Supabase: remove casamento; no veículo numero_serie / **equipamento série = null** (zerado).
- Registro no **Tacita continua existindo** (hardware físico / identidade).
- Estado da série no inventário API → disponível (pronta para novo casamento), **salvo** se ops marcar recondicionamento:
 - - fluxo de **conserto**: ops pode colocar série em hold interno ou manter disponível; produto default após descasar = disponível.
 - - “Recondicionado” = processo físico/ops; quando o device volta à frota, a série **já está** no Tacita e pode ser casada de novo.
- 404 se não havia casamento.
- Albert warehouse: se houver espelho numserie só para telemetria, **recomenda-se** limpar no mesmo fluxo (best-effort) para não misturar dados — **fonte do casamento continua só Supabase**.

Eliminar IoT / baixa definitiva (só ADM) — quebrado, roubado, extraviado

Para série **já usada** (ou no inventário) que **não deve voltar** a frota.

```
POST /series-iot/eliminar
Authorization: Bearer ... # serials:write · perfil ADM

{
  "numero_serie": "37GBF",
  "motivo": "roubado",
  "observacao": "B0 123/2026",
  "confirmado": true
}
```

Campo	Obrig.	Notas				
numero_serie	sim	max 20				
motivo	sim	quebrado\	roubado\	extraviado\	fim_vida\	outro
observacao	não	max 200				

Campo	Obrig.	Notas				
confirmado	sim	deve ser true				

Efeitos:

- Se **casada** → **descasar** (zera série no veículo).
- Supabase: marca **eliminado** (motivo, observacao, eliminado_em, eliminado_por).
- Estado API → eliminado; pode_casar=false sempre.
- Fora de somente_disponiveis.
- Tacita: identidade **pode permanecer** (auditoria); API **não** apaga NYX/FEK. Wipe físico = ops/CLI se política exigir.
- casar / reservar → **409** series_eliminated.

Reabilitar (opcional, super-ops): POST /series-iot/reabilitar — fora da UI normal do ADM.

	Descasar	Eliminar
Objetivo	soltar device (reuso/conserto)	baixar da frota (roubo/quebra definitiva)
Depois	disponivel → pode casar	eliminado → não casa
Quem	ADM	só ADM (View nunca)

8.7 Endpoints internos do Bridge (Albert)

Sugestão de serviço albert_tacita_bridge (Flask/FastAPI em loopback):

Método	Path interno	Função
GET	/bridge/tacita/series	lista pública Tacita (sem estado de casamento)
GET	/bridge/tacita/series/{num}	existe? mac, modelo_iot, versao_hw
GET	/bridge/tacita/health	db legível, contagem

Auth: X-Bridge-Token. SQLite **read-only**. **Casar / descasar / reservar / eliminar = só Supabase (External)**; bridge **não** grava esses estados. External, ao listar, **enriquece** cada série com estado (casado/reservado/eliminado/disponivel) a partir do Supabase + inventário do bridge.

8.8 Fluxo ADM (UI)

1. Cadastrar veículo (placa≤8, marca≤10, modelo≤10, sigla≤3, ...) → ADM
2. GET /series-iot?somente_disponiveis=true&dominio=Albert2 → seletor
3. (opcional) GET/POST verificar série escolhida
4. POST /series-iot/casar { numero_serie, id_veiculo }
5. GET /veiculos/{id} → equipamento preenchido (série, modelo_iot, versao_hw)
6. View/telemetria usam numero_serie / numserie

Proibido na UI ADM: input free-text de série que não passou por lista/verificar.

8.9 Fluxo View

- **Não** chama casar / descasar / eliminar / reservar.
- Lê equipamento no perfil completo.
- Telemetria e mapas usam série casada já refletida no warehouse.

8.9A Albert Admin — UI de identidade (mudança obrigatória)

A função legada “**Configurar identidade do dispositivo**” / “Nova identidade” / “Identificar dispositivo” **deixa de criar** série, MAC, NYX ou FEK na UI.

Antes (legado)	Depois (v1 bridge)
Nome: Configurar identidade do dispositivo	Nome: “ Alberts disponíveis para instalação ”
Ação: provisionar / inventar identidade	Ação: somente listar IoTs do inventário
Fonte	Tacita via bridge (+ estado Supabase)
Estados mostrados	disponivel e reservado (e opcionalmente ver casados em outra aba read-only)
Criar série / digitar numSerie livre	Proibido
Casar série↔veículo	MGConnect ADM via /series-iot/casar (não no painel legado de identidade)
Provisionamento FEK/NYX	Só CLI Tacita (ops de fábrica/bancada)

Comportamento da nova tela (Albert Admin):

- Chama inventário (bridge local ou External com chave de serviço, conforme rede):
- GET .../series-iot?somente_disponiveis=true **e/ou** incluir estado=reservado do operador.
- Lista colunas: numero_serie, modelo_iot, versao_hw, mac (opcional), estado, expira_em se reservado.
- **Não** oferece formulário de create identity.
- Ações opcionais na v1 Admin: copiar série; link “abrir no MGConnect ADM para casar”; **não** eliminar daqui (eliminar = ADM).
- Se bridge indisponível → mensagem clara; sem fallback para inventar série.

Depreciação: código/UI de “Nova identidade” no Albert Admin marca-se **deprecated**; remoção após B5/B6 do bridge.
Composição do veículo e SW continuam no Albert Admin (outras telas) — esta mudança é **só** a de identidade/série.

8.10 Versão SW vs HW

Campo	Fonte no casamento	Depois
modelo_iot	Tacita	estável até set-meta
versao_hw	Tacita	estável até set-meta
versao_sw	não no Tacita	bridge/warehouse swv dinâmico (OTA) no GET veículo

8.11 Erros bridge / series-iot

HTTP	reason	Quando
400	validation_error	série >20, body inválido
404	series_not_in_tacita	não existe no inventário
404	vehicle_not_found	id_veiculo inválido
409	already_linked	série já casada
409	vehicle_has_series	veículo já tem série (se política impedir overwrite)
422	domain_mismatch	domínio ≠ Albert2
503	bridge_unavailable	External não alcança Albert bridge

8.12 Aceite (QA bridge + casamento)

- [] Listar somente_disponiveis = pode_casar=true (exclui casado **e** reservado de outros)
- [] Reserva v1 com TTL; some da lista; libera no timeout/casar/liberar
- [] Série inexistente no Tacita → 404
- [] Série casada em **outro** veículo → 409 no casar
- [] Casar em veículo que **já tinha** série → **substitui** (antiga vira disponível)
- [] Descasar → veículo com série **null**; inventário **disponível** (reuso/conserto ok)
- [] Eliminar (quebrado/roubado) → se casada, descasa; estado **eliminado**; casar → 409
- [] Eliminar **não** expõe/apaga FEK via API; só ADM com confirmado=true
- [] Casamento **só** no Supabase
- [] Resposta **sem** nyx/fek
- [] External sem filesystem Tacita
- [] View sem serials:write
- [] Provisionamento só CLI Tacita

8.13 Ordem de implementação sugerida

Fase	Entrega
B1	Bridge health + list/verify Tacita RO
B2	External: list/verify + join estado Supabase (casado/reservado/disponível)
B3	reservar / liberar-reserva (Supabase)
B4	casar (replace) / descasar (zera série no veículo) no Supabase
B5	eliminar (baixa definitiva) + UI ADM com confirmação
B6	UI ADM seletor + reserva
B7	View consome equipamento no perfil

9. Diff ADM (só o que o Admin deve mudar)

- Marca/modelo **30**→**10**; placa **8**; sigla **3**; frota **20**; nome_empresa **40** (separado de razão social).
- Subcampos tipo caminhão com max §5; **não** gravar só blob composto.
- Série: seletor Tacita, não free-text.
- HW/SW/composição: **read-only** no form frota.
- Motorista: form Nome Completo + CPF; CRUD só ADM.
- Vínculo: PUT/DELETE API com origem: mgconnect_adm.
- Validar na UI **antes** do save; API 400 se passar.

Fora do diff ADM (MGConnect): alterar IoT; inventar série; CRUD composição.

Albert Admin (paralelo): renomear tela de identidade → “**Alberts disponíveis para instalação**”; só lista disponível/reservado da Tacita (§8.9A).

10. Integração administrativa / status_cmd (V3.1, sem JWT)

Campos operacionais compartilhados (banco/bridge/API — não claim JWT):

status_cmd, num_serie, sigla, empresa/nome_empresa, placa, marca, modelo, geral/frota + conjuntos motorista: registro, nome, placa vinculada, num_serie.

status_cmd	Significado
0	liberado
1	deprecado (só GPS / básico)
2	bloqueado

Sinalizar mudanças (fila/evento/flag) para evitar poll contínuo.

11. Erros HTTP (diagnóstico)

Código	Uso
400	params ausentes, overflow de campo, body inválido
401	credencial inválida / não autorizado
403	sem permissão / modo incompatível
404	recurso/usuário/empresa/motorista não encontrado
409	série já casada / conflito de unicidade
422	inconsistência semântica (placa≠sigla, perm desconhecida)
500	erro interno

Mensagens de overflow em **português** com nome do campo.

12. Critérios de aceite (checklist dev/QA)

Auth / sessão / permissões V3.1 §4.7–4.9

- [] Login External sem JWT; View cria sessão e abre UI
- [] Login traz mode válido (§2.6)
- [] Login traz permissions_all + permissions[] só com strings do catálogo §2.5
- [] Sem aliases (view_routes ok; routes / dashboard **não** no array View)
- [] Exemplos §2.7 comportam-se na UI (super / owner max / mecânico / driver / pro)
- [] mode=pro → sem modal; Diagnóstico capa bloqueado
- [] mode=max + perms parciais → só módulos listados no modal
- [] company + plan_id/plan_name quando houver empresa
- [] user.id = company_user; /permissions resolve gestores e motoristas
- [] httpswd mgconnect1 → mode max + permissions_all (se mantido)

Cadastro / max

- [] ADM e API recusam marca/modelo >10, placa>8, sigla>3, frota>20, nome_empresa>40
- [] Sigla + nome_empresa no payload; razão social separada

- [] Subcampos tipo caminhão; View compõe string

Tacita / composição

- [] Série só do inventário; modelo_iot + versao_hw na lista
- [] SW dinâmica no veículo após casamento
- [] composição_veiculo no GET/login (ou ativa:false)

Motoristas / vínculo / listas View

- [] Login **sem** mode/permissions §4.7 → **não** considerar contrato fechado
- [] GET veículos por sigla com tem_vinculo + motorista_vinculado
- [] GET motoristas por sigla com tem_vinculo + veiculo_vinculado
- [] GET vínculos vigentes por sigla
- [] Veículos livres e motoristas livres (tem_vinculo=false ou path dedicado)
- [] GET perfil completo do veículo (todos campos ADM + IoT + composição + vínculo)
- [] View **PUT** vincular e **DELETE** desvincular com origem=albert_view
- [] ADM continua com as mesmas operações de vínculo
- [] Cadastro motorista = Nome+CPF **só** ADM
- [] 1:1 vigente; mesma sigla; last-write-wins

View UI

- [] Filtro sigla/placa
- [] Pro: sem Diagnóstico capa / sem modal; Max: modal + perms
- [] Aplicar permissions[] (view_*) na UI e bloquear backend sensível

13. To-do prioritário (ordem)

Prio	Item	Dono
P0	Login com mode + permissions_all + permissions[] (§4.7–4.9) — obrigatório	ADM/API
P0	Login company+plan+user.id; mode alinhado a pro/max	ADM/API
P0	Validação max marca/modelo/placa/sigla/f rota/nome_empresa	ADM+API
P0	View: sessão + mode/Pro-Max + Diagnóstico	View ✓ parcial
P0	GET veículos/motoristas por sigla + tem_vinculo + livres	API
P0	GET perfil completo veículo	API
P0	PUT/DELETE vinculos-motorista (View + ADM)	API+View+ADM
P1	CRUD motorista só ADM (Nome+CPF)	ADM+API
P1	series-iot + bridge Tacita	API+Albert
P1	Enforcement view_* na UI e backend View	View
P2	composição + SW no payload	Albert+API
P2	status_cmd + sinalização de mudança	ambos
P1	Histórico motoristas + busca autocomplete	API+View
P1	mode=driver E2E	View+API
P2	Mídia veículo + aceite termos	API+ADM+View
P2	Robustez §16 (concorrência, reasons, ver, sinalização)	API+View
P1	Albert Admin: UI “ Alberts disponíveis para instalação ” (só lista Tacita)	Albert
P3	Remover httpasswd; apagar código legado “Nova identidade”	View/Albert

14. Exemplos rápidos de validação

Login completo: contém mode, permissions_all, permissions só com view_*. **Pro:** mode=pro → badge P, sem modal, Diagnóstico bloqueado. **Max + perms:** mode=max, permissions_all=false, lista parcial → só módulos listados. **Livres:** tem_vinculo=false em veículos e motoristas. **Vínculo View:** PUT com origem=albert_view reflete no ADM na próxima leitura.

15. O que este documento não é

- Não restaura JWT.
 - Não autoriza ampliar schema IoT.
 - Não detalha telemetria/warehouse.
 - Shape e max desta SPEC prevalecem sobre OpenAPI live incompleto; paths EN = alias.
-

16. Robustez operacional (contrato)

16.1 Concorrência no vínculo

- **Last-write-wins** no estado vigente, com atualizado_em sempre presente.
- Se o cliente enviar se_coincide_com / expected_atualizado_em (opcional) e divergir → **409** + corpo com vínculo atual.
- Dois writes simultâneos View+ADM: o último atualizado_em prevalece; o outro cliente relê.

16.2 Auditoria

Todo vínculo gravado registra: origem (albert_view|mgconnect_adm), atualizado_por (user id/e-mail), atualizado_em. Histórico: ver §17.2.

16.3 Bloqueios de negócio (vínculo recusado com 422/409)

Não permitir PUT de vínculo se:

- veículo status ∈ {inativo,bloqueado} ou status_cmd=2;
- motorista status ∈ {inativo,bloqueado};
- siglas diferentes;
- fora do escopo do usuário autenticado.

reason estável: vehicle_blocked, driver_blocked, sigla_mismatch, scope_denied.

16.4 Multi-sigla

Se siglas_all=false e len(siglas)>1, **toda** lista exige sigla na query. View: seletor de sigla antes de listar/vincular.

16.5 Refresh de permissões

- Login traz snapshot; sessão View pode ficar stale.
- GET /auth/me ou GET /permissions?user_id= (shape View: mode + permissions) para renovar **sem** senha, se a sessão ainda for válida.
- Se perfil revogado → 401/403 e logout View.

16.6 Sinalização de mudança (anti-poll)

Mecanismo a combinar (flag frota_versao, evento, webhook interno): alteração de frota/vínculo/cadastro incrementa versão por sigla. View: GET ...&desde_versao= ou header If-None-Match → 304 se igual.

16.7 Paginação e limites

- limite 1–**500**, default **100**.
- Resposta inclui total / count + limite + deslocamento.
- Rate limit razoável por chave/usuário (ex. 60 req/min listas); 429 + Retry-After.

16.8 Versionamento do contrato de auth

Campo ver: **4** no login. View rejeita ou entra em modo degradado se ver ausente ou < 4 (log + mensagem “atualize o ADM/API”).

16.9 Erros padronizados

Corpo JSON: { "error": "texto PT", "reason": "snake_case_estavel" }. Reasons mínimas: invalid_credentials, user_not_found, company_blocked, scope_denied, driver_not_found, vehicle_not_found, already_linked, vehicle_blocked, driver_blocked, sigla_mismatch, validation_overflow, permission_unknown, contract_version_unsupported.

16.10 LGPD / CPF

- API pode devolver cpf completo (uso operacional autenticado) **ou** cpf_mascarado (ex. *.456.789-).
- Campo cpf no vínculo/lista: se política = mascarar, documentar no OpenAPI; View **nunca** loga CPF em claro em analytics.

16.11 Falha de rede / offline

View **não** persiste vínculo local como fonte da verdade. Falha no PUT/DELETE → mensagem de erro; estado = última leitura da API.

16.12 OpenAPI

Um OpenAPI gerado a partir desta SPEC (paths PT + x-alias-en). Fonte da verdade de paths = esta SPEC quando live divergir.

17. Extensões de produto (View + API)

17.1 Mídia do veículo (foto / documentos)

Opcional no cadastro ADM; se existir, a View **lê**.

```
GET /veiculos/{id}/midias
GET /veiculos/{id}/midias/{midia_id}
```

Campo	Tipo	Notas		
id	UUID			
tipo	enum	foto_veiculo\	documento\	outro
titulo	string ≤80			
content_type	string	ex. image/jpeg, application/pdf		
url	string ≤500	URL assinada ou path; TTL se assinada		
criado_em	datetime			

- View: **somente leitura**. Upload = **só ADM**.
- Perfil completo (§7.2) pode incluir midias: [] resumido ou contagem midias_count.

17.2 Histórico de motoristas no veículo (não só o vigente)

```
GET /veiculos/{id}/historico-motoristas?limite=50
GET /vinculos-motorista?id_veiculo={id}&vigente=false&incluir_encerrados=true
```

Cada item:

Campo	Notas	
id_vinculo		
id_motorista, nome_completo, cpf	espelho	
inicio_em	= assigned_at / criado	
fim_em	null se vigente	
origem_inicio / origem_fim	albert_view\	mgconnect_adm
vigente	boolean	

Ordenação default: mais recente primeiro. View usa para auditoria operacional / “quem dirigiu este caminhão”.

17.3 Busca / autocomplete (placa, CPF, nome)

```
GET /busca?sigla=MGC&q=NS02&tipos=veiculo,motorista&limite=20
```

Query	Obrig.	Notas
sigla	sim*	*exceto siglas_all
q	sim	min 2 caracteres; placa, CPF (dígitos) ou nome
tipos	não	veiculo, motorista ou ambos
limite	não	default 20, max 50

Resposta:

```
{
  "resultados": [
    { "tipo": "veiculo", "id": "...", "placa": "NS02GBF", "marca": "Volvo", "modelo": "FH",
      "tem_vinculo": true },
    { "tipo": "motorista", "id": "...", "nome_completo": "Ze do Pedal", "cpf": "1234567890",
      "tem_vinculo": true }
  ]
}
```

Respeita escopo. Rate limit mais agressivo (autocomplete).

17.4 Modo driver *de ponta a ponta*

Quando mode=driver no login:

Regra	Comportamento
Entrada UI	Não abrir capa/frota/modal; redirect para Página do Motorista / Gamificação
Escopo	Preferir placas do motorista (1 placa) ou vínculo vigente
Permissões view_*	Ignoradas para navegação tradicional; só módulos da página motorista
Listas frota	403 ou não expostas na UI
Vínculo	Motorista não altera vínculo (só ADM/View operacional)
Agente	Fora do escopo salvo driver_agent futuro

Aceite: login driver → deep-link/gamificação; logout volta a entrada.

17.5 Aceite de política / primeiro login

Campos no login (ou GET /auth/me):

```
{
  "termos": {
    "aceite_necessario": true,
    "versao_politica": "2026-07-01",
    "url_documento": "https://.../politica.pdf",
    "aceito_em": null
  }
}

POST /auth/aceite-termos
{ "versao_politica": "2026-07-01", "aceito": true }
```

- Se aceite_necessario=true, View **bloqueia** navegação até POST com sucesso.
- Registrar user_id, versao_politica, aceito_em, IP/user-agent se política exigir.
- Nova versão de política → aceite_necessario=true de novo.

17.6 Resumo de prioridades das extensões

Item	Prio
§16 robustez (concorrência, bloqueios, ver, errors, paginação)	P1
§17.2 histórico motoristas	P1
§17.3 busca autocomplete	P1
§17.4 mode driver E2E	P1
§17.5 aceite termos	P2
§17.1 mídia veículo	P2
§16.6 sinalização mudança	P2

Fim SPEC V4.4 — contrato completo ADM↔View via API Externa · zero JWT · login 4.7–4.9 · frota/motoristas/vínculos · perfil · mídia · histórico · busca · driver · termos.