

Documentação da API

Sistema de gestão Wester

[Autenticação](#) · [Endpoints](#) · [Modelos](#) · [Integrações](#) · [OpenAPI](#)

Versão de 2026-10-15

Índice

1. Visão geral da API
2. Arquitetura
3. Autenticação e autorização
4. Integrações (inclui TUTOM)
5. Tratamento de erros
6. Índice de endpoints
7. Modelos de dados
8. OpenAPI / Swagger
9. Acesso, usuários e perfis
- .0. Produtos
 - .1. Categorias de produtos
 - .2. Setores e tipos de material
 - .3. Corredores, blocos e níveis
 - .4. Itens e transferências
 - .5. Moldes
 - .6. Materiais fixos e slots
 - .7. Listas de materiais
 - .8. Consultas de estoque e mapa
 - .9. Histórico e auditoria
- !0. Movimentações
 - !1. Alertas de estoque
 - !2. Relatórios de estoque
 - !3. PCP e produção (MES)
 - !4. Integração industrial Edge

Documentação técnica das APIs — Sistema Wester

Backend `Storage_Wester` (Spring Boot) do sistema de gestão da Wester: WMS/almoxarifado, tinturaria, produtos e moldes, histórico/auditoria, usuários e perfis, relatórios e o módulo PCP/Produção (MES).

Base da análise: código local em 15/10/2026 — backend `Storage_Wester_UPDATED/Storage_Wester` e frontend `storage-app`, incluindo alterações ainda não commitadas (por exemplo, categorias de produtos). **Método:** leitura estática do código. Nenhuma chamada foi executada contra servidor.

Números

ITEM	QUANTIDADE
Módulos funcionais da API	16
Operações (métodos de controller) documentadas	145
Combinações método + rota (inclui aliases PT/EN)	211
Esquemas de dados (<code>components.schemas</code> no OpenAPI)	138
Controllers (REST + MES + Edge)	23

Índice

ARQUIVO	CONTEÚDO
<code>arquitetura.md</code>	Visão geral, camadas, modelo de dados e relacionamentos, tempo real, frontend
<code>autenticacao.md</code>	Login, JWT, perfis e telas (<code>SCREEN_*</code>), regras do MES, autenticação do Edge, CORS, limites
<code>endpoints.md</code>	Índice das 211 rotas e dos 16 módulos; cada rota leva à ficha completa
<code>modulos/</code>	Fichas por endpoint (17 itens pedidos: método, URL, parâmetros, headers, autorização, corpo, resposta, exemplos, códigos HTTP, validações, regras, dependências, cURL)
<code>modelos.md</code>	Campos de todos os DTOs e contratos, com limites e obrigatoriedade
<code>integracoes.md</code>	REST, STOMP, Edge, importação de PDF, OCR e possibilidades com o TUTOM
<code>erros.md</code>	Formatos de erro, mapeamento exceção → HTTP, tratamento no frontend
<code>openapi.yaml</code>	Especificação OpenAPI 3.0.3, 211 operações, <code>bearerAuth</code> e <code>edgeKey</code>

Módulos

MÓDULO	OPERAÇÕES	ROTAS	FICHA
Acesso, usuários e perfis	16	27	acesso.md
Produtos	10	32	produtos.md
Categorias de produtos	6	12	categorias.md
Setores e tipos de material	11	32	setores.md
Corredores, blocos e níveis	21	21	estruturas.md
Itens e transferências de estoque	5	5	itens-estoque.md
Moldes	7	13	moldes.md
Materiais fixos e slots legados	15	15	materiais-fixos.md
Listas de materiais	10	10	listas.md
Consultas de estoque e mapa	3	3	estoque-consolidado.md
Histórico e auditoria	4	4	historico.md
Movimentações operacionais	3	3	movimentacoes.md
Alertas de estoque	1	1	alertas.md
Relatórios de estoque	4	4	relatorios.md
PCP e produção (MES)	26	26	producao.md
Integração industrial Edge	3	3	edge.md

Como usar

1. **Base da URL:** `BASE_URL` = endereço do backend (sem `/api`). Todas as rotas começam com `/api`.
2. **Autenticar:** `POST /api/auth/login` → usar `token` em `Authorization: Bearer <token>` ([autenticacao.md](#)).
3. **Localizar a rota** em [endpoints.md](#) pelo identificador `API-nnn` e abrir a ficha do módulo.
4. **Importar no Swagger/Postman:** carregue `openapi.yaml`. Os exemplos são **sintéticos**; troque IDs e códigos por dados reais do ambiente.

Convenções confirmadas no código

- Prefixo geral `/api`; JSON em UTF-8; datas em ISO-8601.
- Duas famílias de permissão: `SCREEN_*` (em que tela o perfil entra) e `FULL_ACCESS` (pode alterar). O MES usa o bean `mesAccess`.
- Listagens grandes são paginadas: Spring Data (`content`, `number`, `size`, `totalElements` ...) nas rotas WMS, `page` / `size` no MES e `pagination` + `items` nos relatórios. **Não trocar um formato pelo outro** ([modelos.md](#)).

- Aliases PT/EN equivalentes: `/api/areas` = `/api/setores` , `/api/produtos` = `/api/products` , `/api/perfis` = `/api/profiles` , `/api/categorias-produto` = `/api/product-categories` , `/api/auth/login` = `/api/usuarios/login` .
 - Concorrência otimista com campo `version` em `Fileira` , `Grade` , `Nivel` e `ItemEstoque` (conflito → `409` com prefixo `OPTIMISTIC_LOCK:`).
-

O que é existente e o que é planejado

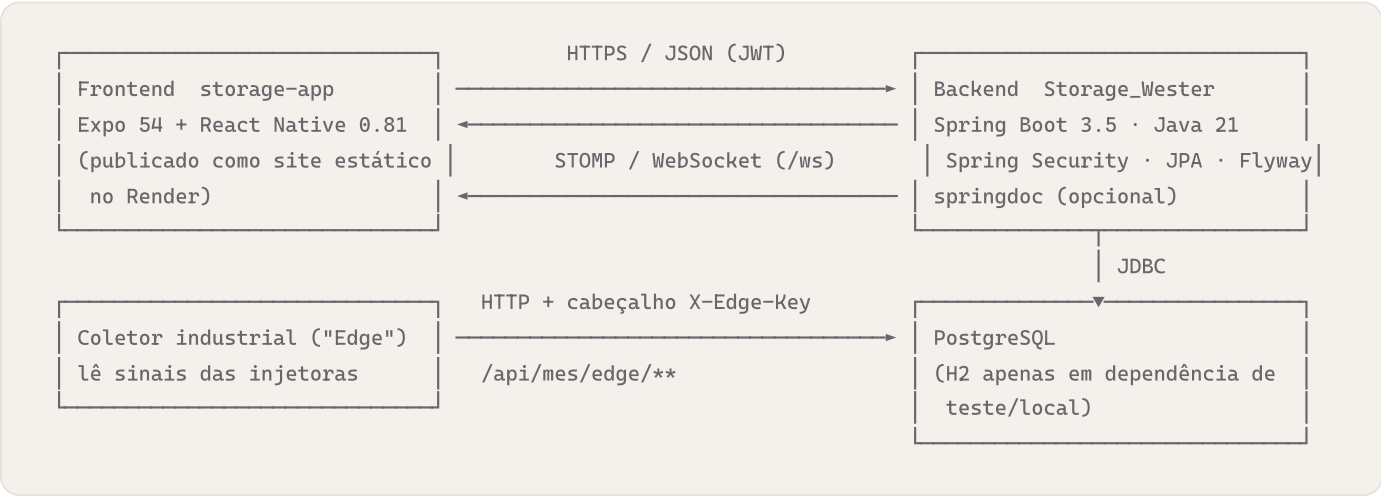
Resumo: Alertas, Movimentações e Relatórios têm backend e código de tela, mas **estão desativados no menu atual**; integração direta com o TUTOM, cadastro de ferramentas, OEE/MTBF/MTTR e recuperação de senha por e-mail **não existem**.

Arquitetura

← Índice da documentação da API

Escopo: o que está comprovado no código dos dois projetos — frontend `storage-app` (Expo / React Native Web) e backend `Storage_Wester` (Spring Boot).

1. Visão geral



CAMADA	TECNOLOGIA (COMPROVADA EM <code>POM.XML</code> / <code>PACKAGE.JSON</code>)
Frontend	Expo <code>^54</code> , React <code>19.1</code> , React Native <code>0.81</code> , React Navigation (Drawer), React Native Paper, Axios, <code>@stomp/stompjs</code> , Three.js + <code>@react-three/fiber</code> (visão 3D), <code>@dnd-kit</code> (arrastar e soltar)
Backend	Spring Boot <code>3.5.0</code> , Java <code>21</code> , Spring Web, Spring Security (JWT com <code>jjwt</code>), Spring Data JPA, Bean Validation, Spring WebSocket (STOMP), Actuator, springdoc-openapi, Apache PDFBox <code>3.0.3</code>
Banco	PostgreSQL (driver e dialeto <code>PostgreSQLDialect</code>), migrações Flyway em <code>src/main/resources/db/migration</code>
Implantação	Frontend: <code>render.yaml</code> (site estático, <code>npm run build:web</code> , reescreve <code>/*</code> para <code>/index.html</code>); backend: <code>Dockerfile</code> e <code>Dockerfile.hardened</code> , <code>docker-compose.yml</code>

2. Backend: camadas e módulos

O backend segue o padrão `controller` → `service` → `repository` → `entidade`, com DTOs de entrada/saída em `dto/` :

PACOTE	CONTEÚDO
controller	21 controllers REST + exception/GlobalExceptionHandler
mes	módulo de produção (MES/PCP): MesController , MesEdgeController , MesService , MesRules , regras de acesso, leitura de PDF (MesOrderPdfParser , MesOrderPdfOcr)
service	24 classes de serviço e apoio com as regras de negócio
repository	repositórios JPA (alguns com consultas customizadas)
model	17 entidades JPA e 3 enums (MaterialType , TipoAcessoPerfil , TipoEstruturaArea)
config	segurança, JWT, CORS, limite de taxa, OpenAPI
realtime	WebSocket/STOMP, publicação de eventos após commit

2.1 Módulos funcionais da API

A API foi agrupada em **16 módulos** (ver **endpoints.md**):

1. Acesso, usuários e perfis
2. Produtos
3. Categorias de produtos
4. Setores e tipos de material
5. Corredores, blocos e níveis (estrutura física)
6. Itens e transferências de estoque
7. Moldes
8. Materiais fixos e slots legados
9. Listas de materiais ("encerto")
- .0. Consultas de estoque e mapa
- .1. Histórico e auditoria
- .2. Movimentações operacionais
- .3. Alertas de estoque
- .4. Relatórios de estoque
- .5. PCP e produção (MES)
- .6. Integração industrial Edge

3. Modelo de dados e relacionamentos

Os campos de cada DTO estão em **modelos.md**. A relação entre as entidades persistidas é:

```

AreaEstoque (setor) —1—N— Fileira (corredor/estante) —1—N— Grade (bloco) —1—N— Nivel (gaiola/ordem/posição)
    |
    | 1—N
    ▼
AreaMaterialTag (tipos de material aceitos no setor: MOLDE | PRODUTO | FERRAMENTA)
    |
    | 1—N ItemEstoque —N—1
    | 1—N NivelMaterial (n
    | 1—N SlotNivel —1—N—
    | (listas de materiais)
AreaEstoque —1—N— TabelaEncerto —1—N— TabelaEncertoItem —N—1— Produto

HistoricoMovimentacao — referência opcional a Usuario, Produto, ItemEstoque, Nivel (atual/origem/destino)
                        e ids avulsos (área, corredor, bloco, slot, molde, entidade)
Usuario —(texto `perfil` = `Perfil.codigo`, sem chave estrangeira)— Perfil

Módulo MES (tabelas criadas em V10): máquina · ordem de serviço · execução · evento · ciclo · resultado de ci
apontamento de qualidade · parada

```

Observações confirmadas no código:

- **Dois tipos de setor** (`TipoEstruturaArea`): `DINAMICO` (níveis podem ser criados, movidos e resequenciados) e `FIXO` (posições criadas na configuração e editadas pontualmente). O frontend mostra uma tela diferente para cada um (`Warehouse2DView` para dinâmico, `WarehouseFixedView` para fixo) e uma visão 3D para os dois.
- **Rótulos configuráveis por setor**: `labelEstruturaPrimaria`, `labelEstruturaSecundaria`, `labelNivel` e `labelSlot` — por padrão o frontend usa "Corredor", "Bloco" e "Ordem/Gaiola"; nos setores fixos podem aparecer "Estante", "Posição" etc.
- **Controle de concorrência otimista** (`@Version`) em `Fileira`, `Grade`, `Nivel` e `ItemEstoque`.
- **Perfil do usuário não é chave estrangeira**: o usuário guarda o código do perfil como texto; se o perfil for inativado ou apagado, o usuário fica sem permissões (ver `autenticacao.md`).
- **Auditoria**: o `HistoricoMovimentacao` é gravado na **mesma transação** da alteração de negócio (exigido por `Propagation.MANDATORY` em `registrarEventoGenericoNaTransacaoAtual`): se a auditoria falhar, a mudança é desfeita.
- **Migrações Flyway**: 19 arquivos `V*_*.sql` (V1–V16, V21–V23; não existem V17–V20 no repositório). `ddl-auto` é `update` no perfil padrão e `validate` no perfil `prod`.

4. Tempo real

`WebSocketConfig` publica STOMP em `/ws` (SockJS) e `/ws-native`, com corretor simples em `/topic` e prefixo de aplicação `/app`. `EntityChangeListener` publica um `RealtimeEvent` **depois do commit** para as entidades que alteram estoque, estrutura, histórico, cadastro etc. Tópicos: `/topic/estoque`, `/topic/area/{areaId}` e `/topic/entidade/{entidade}` (detalhes em `integracoes.md`). O frontend usa isso para atualizar listas e mapa sem recarregar a página.

5. Frontend: organização

PASTA (<code>SRC/</code>)	FUNÇÃO
<code>navigation/</code>	menu lateral (Drawer), rotas e <i>deep links</i> (<code>linking.ts</code>)
<code>auth/</code>	contexto de autenticação (<code>AuthContext</code>), proteção de rotas (<code>RequireAuth</code>), retorno pós-login
<code>security/permissions.tsx</code>	mapeia o perfil para telas e ações; mostra "Sem permissão" quando não liberado
<code>screens/</code>	telas: login/cadastro, Dashboard, Armazém, Setores, Produtos, Moldes, Perfis, Histórico, Usuários, PCP (<code>mes/</code>) e as telas desativadas Alertas, Movimentações, Relatórios
<code>components/</code> , <code>components/warehouse2d/</code>	componentes do armazém 2D, setor fixo, mapa V4, modais
<code>twin/</code>	"Wester Twin" — visão 3D do armazém
<code>services/</code>	clientes HTTP por módulo; <code>services/http/</code> trata erros
<code>realtime/</code> , <code>services/realtime/</code>	assinatura dos tópicos STOMP
<code>axios.ts</code> , <code>config/apiConfig.ts</code>	cliente HTTP, anexo do token e URL da API

- **URL da API:** `EXPO_PUBLIC_API_BASE_URL` (ou `VITE_API_BASE_URL`); se ausente, `http://localhost:8080/` / `127.0.0.1` quando a página roda em localhost, senão a URL de produção definida em `apiConfig.ts` . Barras finais e um sufixo `/api` são normalizados.
- **Sessão:** token e dados do usuário ficam em `AsyncStorage` (no navegador, armazenamento local). Um `401` com token anexado limpa a sessão e leva ao Login.
- **Temas:** Light, Dark e Console (seletor no cabeçalho de toda tela).
- **3D:** disponível apenas no navegador (WebGL); a opção não é oferecida em build nativo.

6. Pontos de extensão e restrições conhecidas

- A API tem **aliases duplicados** em português e inglês (`/api/areas` = `/api/setores` , `/api/produtos` = `/api/products` ...). Eles contam como rotas, mas são a mesma operação (211 combinações método+rota para 145 operações).
- Alertas, Movimentações e Relatórios têm **backend e código de tela prontos, mas estão desativados no menu atual** (comentário em `DrawerNavigator.tsx`). Ver documento interno de lacunas.
- Swagger/OpenAPI da aplicação só liga com `SWAGGER_ENABLED=true` ; o `openapi.yaml` deste diretório é uma análise estática independente, com exemplos sintéticos.

Autenticação e autorização

← Índice da documentação da API

Tudo nesta página foi conferido em `config/SecurityConfig.java`, `config/JwtAuthenticationFilter.java`, `config/JwtSecurityConfig.java`, `config/LoginRateLimitingFilter.java`, `realtime/WebSocketConfig.java`, `mes/MesAccess.java`, `mes/MesEdgeSecurity.java`, `service/AuthenticationService.java` e nas anotações `@PreAuthorize` de cada controller.

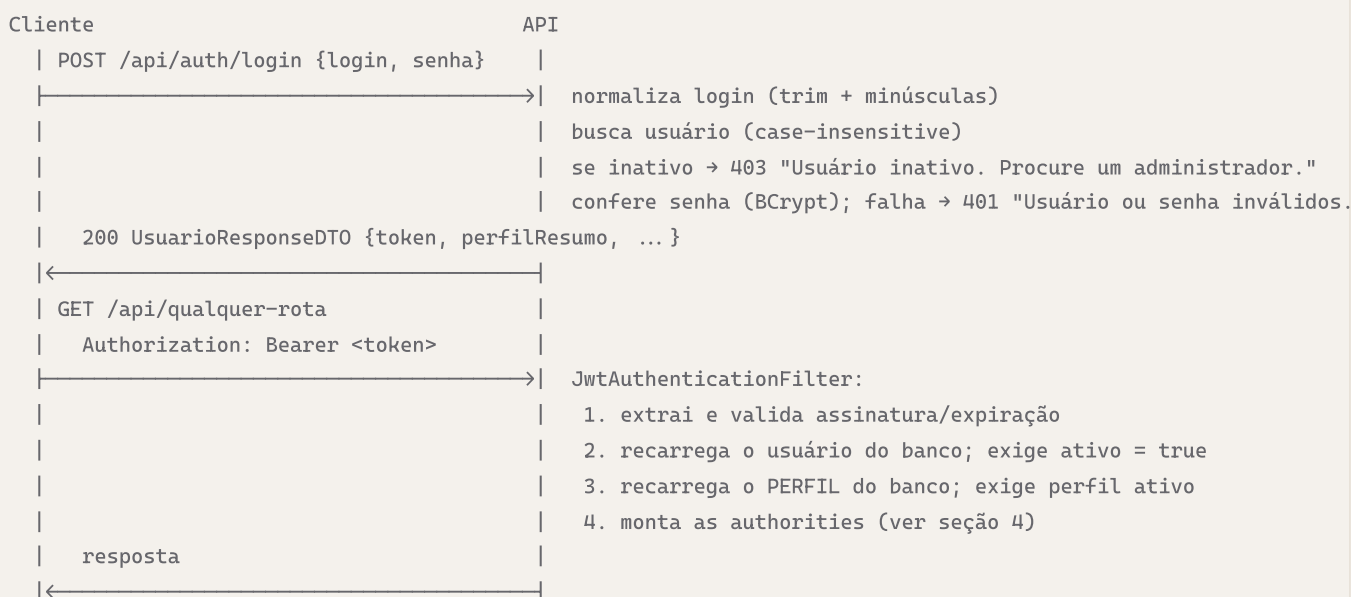
1. Visão geral

A API é **stateless**: não há sessão de servidor (`SessionCreationPolicy.STATELESS`) e o CSRF está desligado, coerente com um cliente que envia o token em toda chamada. Existem **duas cadeias de segurança independentes**:

CADEIA	ALCANCE	COMO AUTENTICA
Principal (<code>SecurityConfig</code>)	toda a API, exceto <code>/api/mes/edge/**</code>	JWT no cabeçalho <code>Authorization: Bearer <token></code>
Edge do MES (<code>MesEdgeSecurity</code> , <code>@Order(1)</code>)	<code>/api/mes/edge/**</code>	chave compartilhada no cabeçalho <code>X-Edge-Key</code>

São cadeias separadas de propósito: o coletor de chão de fábrica não é uma pessoa e não faz login.

2. Fluxo de autenticação (usuários)



Pontos confirmados no código:

- **Rotas de login:** `POST /api/auth/login` e `POST /api/usuarios/login` (mesma operação, dois caminhos).
- **Corpo do login:** `{"login": "...", "senha": "..."} (LoginDTO`, validado com `@Valid`).
- **Sessão atual:** `GET /api/auth/me` devolve o usuário autenticado **sem** token novo, com `perfilResumo`.
- **Token:** JWT assinado com HS256. O `subject` é o login do usuário. **Não há claims de perfil nem de permissões no token.** Validade padrão: `28800000` ms (8 horas), configurável pela variável `SECURITY_JWT_EXPIRATION_MS`.
- **Segredo:** `SECURITY_JWT_SECRET` (ou `JWT_SECRET`); o serviço **não sobe** sem ele nem com menos de 32 caracteres. A chave efetiva é o SHA-256 do segredo informado.
- **Não existe** refresh token, logout no servidor, revogação de token nem troca de senha obrigatória. "Sair" é apenas descartar o token no cliente (ver documento interno de lacunas).
- **Efeito imediato de mudanças:** como usuário e perfil são relidos do banco a cada requisição, inativar um usuário, inativar um perfil ou mudar as telas de um perfil vale **na próxima chamada**, sem invalidar tokens.
- **Perfil ausente, inativo ou desconhecido** gera **zero authorities**: o usuário continua autenticado, mas toda rota que exige uma permissão específica responde `403`.

3. Rotas públicas (sem token)

Declaradas em `SecurityConfig`:

ROTA	OBSERVAÇÃO
<code>OPTIONS /**</code>	pré-voo de CORS
<code>POST /api/auth/login</code> , <code>POST /api/usuarios/login</code>	login (limitado a 10 tentativas por minuto por IP)
<code>POST /api/usuarios</code>	autocadastro (limitado a 5 por minuto por IP) — ver seção 6
<code>/swagger-ui.html</code> , <code>/swagger-ui/**</code> , <code>/v3/api-docs/**</code> , <code>/api-docs/**</code>	só respondem se <code>SWAGGER_ENABLED=true</code> (padrão <code>false</code>)
<code>/actuator/**</code> , <code>/error</code>	liberados na cadeia; o que o Actuator expõe depende da configuração de deploy (não verificada)
<code>/ws/**</code> , <code>/ws-native/**</code>	<i>handshake</i> de WebSocket liberado; a autenticação é exigida no comando STOMP <code>CONNECT</code> (seção 7)

Todo o resto cai em `anyRequest().authenticated()` (sem token → `401`).

4. Autorização: perfil, telas e tipo de acesso

O modelo é o de **perfil** (`perfil`): cada usuário tem um código de perfil (campo `perfil` do usuário, texto, **não é chave estrangeira**) que aponta para um registro com:

- `tipoAcesso`: `FULL_ACCESS` (pode alterar) ou `READ_ONLY` (só consulta);
- `telasPermitidas`: lista de "telas" (ex.: `WAREHOUSE`, `PRODUCTS`, `MES_PCP`);
- `ativo`.

O filtro JWT transforma isso em duas famílias de *authorities*:

AUTHORITY	ORIGEM
<code>ROLE_FULL_ACCESS</code> ou <code>ROLE_READ_ONLY</code>	<code>tipoAcesso</code> do perfil
<code>SCREEN_<TELA></code>	uma por item de <code>telasPermitidas</code> (maiúsculas)

As anotações `@PreAuthorize` combinam os dois eixos. O padrão que se repete é:

- **leitura:** exige só `SCREEN_<teLa>`;
- **criar / alterar / excluir:** exige `SCREEN_<teLa>` e `hasRole('FULL_ACCESS')`.

Exemplo real (`AreaEstoqueController`): `GET /api/areas` aceita `SCREEN_DASHBOARD` ou `SCREEN_WAREHOUSE`; `POST /api/areas` e `DELETE /api/areas/{id}` exigem `SCREEN_WAREHOUSE` e `FULL_ACCESS`.

4.1 Telas (`SCREEN_*`) e o que liberam

AUTHORITY	TELA / USO NO FRONTEND	O QUE O BACKEND EXIGE
<code>SCREEN_DASHBOARD</code>	Dashboard	leitura de setores (<code>/api/areas</code>) e do estoque consolidado (<code>/api/estoque/itens</code> , <code>/api/estoque/itens-consolidado</code>), sempre "ou" <code>SCREEN_WAREHOUSE</code>
<code>SCREEN_WAREHOUSE</code>	Armazém, Setores, estruturas, itens de estoque, listas (encerto)	leitura com a tela; criar/alterar/excluir com a tela e <code>FULL_ACCESS</code>
<code>SCREEN_PRODUCTS</code>	Produtos e categorias; Moldes	leitura com a tela; escrita com a tela e <code>FULL_ACCESS</code>
<code>SCREEN_MOLDS</code>	Moldes	aceita-se <code>SCREEN_MOLDS</code> ou <code>SCREEN_PRODUCTS</code> nas rotas de moldes
<code>SCREEN_USERS</code>	Usuários	listar/consultar com a tela; alterar com a tela e <code>FULL_ACCESS</code>
<code>SCREEN_PROFILES</code>	Perfis	criar/alterar/ativar/inativar com a tela e <code>FULL_ACCESS</code> ; listar também aceita <code>SCREEN_USERS</code>
<code>SCREEN_HISTORY</code>	Histórico	consulta; o histórico por entidade também aceita <code>SCREEN_USERS</code> , <code>SCREEN_PROFILES</code> e <code>SCREEN_PRODUCTS</code>
<code>SCREEN_REPORTS</code>	Relatórios de estoque (tela desativada no menu atual)	consulta
<code>SCREEN_ALERTS</code>	Alertas (tela desativada no menu atual)	consulta
<code>SCREEN_MOVEMENTS</code>	Movimentações (tela desativada no menu atual)	consulta; registrar exige também <code>FULL_ACCESS</code>

Cada ficha de endpoint em [modulos/](#) traz a expressão exata. A lista dos perfis iniciais está na seção 5.

4.2 MES / PCP: outro esquema

`MesController` não usa `@PreAuthorize("SCREEN_ ... ")` direto. Usa o bean `mesAccess` (`mes/MesAccess.java`), que olha quatro authorities de tela e o papel `FULL_ACCESS` :

VERIFICAÇÃO	REGRA NO CÓDIGO
<code>read</code>	qualquer uma entre <code>SCREEN_MES_ADMIN</code> , <code>SCREEN_MES_OPERATOR</code> , <code>SCREEN_MES_MECHANIC</code> , <code>SCREEN_MES_PCP</code>
<code>admin</code>	<code>FULL_ACCESS</code> e <code>SCREEN_MES_ADMIN</code>
<code>operate</code>	<code>admin</code> ou (<code>FULL_ACCESS</code> e <code>SCREEN_MES_OPERATOR</code>)
<code>maintain</code>	<code>admin</code> ou (<code>FULL_ACCESS</code> e <code>SCREEN_MES_MECHANIC</code>)
<code>plan</code>	<code>admin</code> ou (<code>FULL_ACCESS</code> e <code>SCREEN_MES_PCP</code>)

Consequência importante: um perfil `READ_ONLY` com `MES_PCP` só consulta — não planeja. As fichas do módulo `producao.md` indicam qual verificação cada rota usa; `GET /api/mes/permissions` devolve `{read, operate, maintain, plan, admin}` do usuário logado, e é o que o frontend usa para mostrar/ocultar ações.

4.3 Autenticação do Edge (coletor industrial)

`MesEdgeSecurity` cria uma cadeia própria para `/api/mes/edge/**` que compara o cabeçalho `X-Edge-Key` com a variável de ambiente `MES_EDGE_KEY` (comparação em tempo constante, `MessageDigest.isEqual`):

SITUAÇÃO	RESPOSTA
<code>MES_EDGE_KEY</code> ausente ou com menos de 16 bytes	503
cabeçalho ausente ou diferente da chave configurada	401
chave correta	segue para o controller (principal <code>mes-edge</code> , authority <code>MES_EDGE</code>)

O corpo é `{"message":"Integracao edge indisponivel ou credencial invalida"}` nos dois casos — deliberadamente igual, para não revelar qual deles ocorreu. **JWT de usuário não substitui a chave**, e a chave não abre nenhuma rota fora de `/api/mes/edge/**` . A chave é **global** (não por máquina).

5. Perfis iniciais (migração `V2__insert_default_data.sql`)

CÓDIGO	TIPO	TELAS
<code>ADMINISTRADOR</code>	<code>FULL_ACCESS</code>	<code>DASHBOARD</code> , <code>ALERTS</code> , <code>MOVEMENTS</code> , <code>REPORTS</code> , <code>WAREHOUSE</code> , <code>PRODUCTS</code> , <code>USERS</code> , <code>HISTORY</code> , <code>PROFILES</code>
<code>OPERADOR</code>	<code>FULL_ACCESS</code>	<code>DASHBOARD</code> , <code>ALERTS</code> , <code>MOVEMENTS</code> , <code>REPORTS</code> , <code>WAREHOUSE</code> , <code>PRODUCTS</code> , <code>HISTORY</code>
<code>CONSULTOR</code>	<code>READ_ONLY</code>	<code>DASHBOARD</code> , <code>ALERTS</code> , <code>REPORTS</code> , <code>WAREHOUSE</code> , <code>PRODUCTS</code> , <code>HISTORY</code>

Os perfis podem ser editados depois pela API/tela de Perfis (`/api/perfis`); o banco de produção pode estar diferente da semente. Essa semente não contém `MOLDS` nem as telas `MES_*` ; elas passam a existir quando um perfil é criado ou

editado com elas. A tela `MOLDS` foi criada depois: o frontend libera Moldes por `MOLDS` ou `PRODUCTS`, e o backend também aceita `SCREEN_PRODUCTS` ou `SCREEN_MOLDS` nas rotas de moldes.

6. Autocadastro: `POST /api/usuarios`

A rota é pública (usada pela tela "Criar conta"), mas o campo `perfil` do corpo **só é respeitado quando quem chama é um usuário autenticado com `SCREEN_USERS` e `FULL_ACCESS` (`chamadorPodeEscolherPerfil`)**. Nos demais casos o perfil é **forçado para `CONSULTOR` (`PERFIL_PADRAO_AUTOCADASTRO`)**, qualquer que seja o valor enviado. Por isso a pergunta "um anônimo consegue se criar como administrador?" tem resposta **não** no código atual.

7. WebSocket / STOMP

Endpoints `/ws` (com SockJS) e `/ws-native` (WebSocket puro). O *handshake* é sempre aceito, mas o comando STOMP `CONNECT` **exige JWT válido** (cabeçalho `Authorization` / `authorization` / `X-Authorization` / `token` do frame STOMP, ou parâmetro `?token=` no handshake); sem isso a conexão é rejeitada com `SecurityException`. Usuário inativo ou token inválido também são rejeitados. Detalhes dos tópicos em [integracoes.md](#).

8. CORS

`CorsConfig` aceita apenas origens de navegador autorizadas (o frontend publicado, ambientes locais de desenvolvimento e hosts internos da Wester).

Métodos permitidos: `GET, POST, PUT, PATCH, DELETE, OPTIONS`; cabeçalhos permitidos **apenas** `Authorization` e `Content-Type`; credenciais habilitadas; cache de pré-voo de 3600 s. Um cliente de servidor (cURL, TUTOM, coletor) não é afetado por CORS; um navegador em outra origem só consegue chamar a API se a origem estiver nessa lista.

`X-Edge-Key` **não** está na lista de cabeçalhos permitidos: o coletor deve chamar de servidor/dispositivo, não de navegador.

9. Limites de taxa

`LoginRateLimitingFilter` (em memória, por IP, janela deslizante):

ROTA	LIMITE
<code>POST /api/auth/login</code>	10 por minuto
<code>POST /api/usuarios/login</code>	10 por minuto
<code>POST /api/usuarios</code>	5 por minuto

Excedido: `429` com `{"status":429,"error":"Too Many Requests","message":"Muitas tentativas. Aguarde um momento antes de tentar novamente."}`. O IP vem do primeiro valor de `X-Forwarded-For` quando existe, senão do endereço remoto. O contador **não sobrevive a reinício e não é compartilhado entre instâncias**.

10. Exemplos

bash

```
# 1) login - devolve o token em .token
curl -s -X POST "${BASE_URL}/api/auth/login" \
  -H "Content-Type: application/json" \
  -d '{"login": "usuario@exemplo.com", "senha": "<SENHA>"}'

# 2) chamada autenticada
curl -s "${BASE_URL}/api/auth/me" -H "Authorization: Bearer ${TOKEN}"

# 3) chamada do coletor (não usa JWT)
curl -s "${BASE_URL}/api/mes/edge/config?machineId=<TAG_DA_INJETORA>" -H "X-Edge-Key: ${EDGE_KEY}"
```

Resposta típica de login (estrutura de `UsuarioResponseDTO` ; valores ilustrativos):

json

```
{
  "id": 12,
  "login": "usuario@exemplo.com",
  "nome": "Usuário Exemplo",
  "perfil": "OPERADOR",
  "ativo": true,
  "token": "<JWT>",
  "perfilResumo": {
    "id": 2,
    "code": "OPERADOR",
    "description": "Operador",
    "type": "FULL_ACCESS",
    "allowedScreens": ["DASHBOARD", "ALERTS", "MOVEMENTS", "REPORTS", "WAREHOUSE", "PRODUCTS", "HISTORY"],
    "active": true
  }
}
```

Exemplos acima não carregam credenciais nem endereços reais de produção.

Integrações

← [Índice da documentação da API](#)

Este documento separa o que existe e foi comprovado no código do que seria **possibilidade futura** (marcada como "não implementado"). Nada em "Possibilidades" funciona hoje sem desenvolvimento.

1. Integrações existentes

#	INTEGRAÇÃO	SENTIDO	PROTOCOLO	AUTENTICAÇÃO	SITUAÇÃO
1	Frontend → API REST	cliente → servidor	HTTPS/JSON	JWT (<code>Authorization: Bearer</code>)	Em uso
2	API → Frontend (tempo real)	servidor → cliente	STOMP sobre WebSocket	JWT no <code>CONNECT</code>	Em uso
3	Coletor industrial (Edge) → API	dispositivo → servidor	HTTP/JSON	<code>X-Edge-Key</code>	Implementado (3 rotas)
4	PDF de Ordem de Serviço → API (PCP)	usuário envia arquivo	<code>multipart/form-data</code>	JWT com permissão de planejamento	Implementado (prévia, não grava)
5	OCR (Tesseract)	API → programa local	linha de comando	n/a	Opcional, exige instalação no servidor
6	Swagger / OpenAPI da aplicação	consulta	HTTP	pública quando habilitada	Desligado por padrão (<code>SWAGGER_ENABLED=false</code>)

1.1 Frontend → API REST

- URL base: `EXPO_PUBLIC_API_BASE_URL` (ver [arquitetura.md](#) §5). Todas as rotas começam com `/api`.
- O frontend chama as rotas em português (`/api/areas` etc.) e em inglês (`/api/products` etc.) conforme o módulo; os dois aliases são equivalentes ([endpoints.md](#)).
- Login: ver [autenticacao.md](#). Todas as chamadas seguintes levam o token.

1.2 Tempo real (STOMP)

ITEM	VALOR
Endpoints	<code>/ws</code> (SockJS) e <code>/ws-native</code> (WebSocket puro — é o que o frontend usa)
Prefixo de destino de aplicação	<code>/app</code> configurado, mas não existe nenhum <code>@MessageMapping</code> no backend: o cliente só recebe, não envia mensagens
Tópicos de assinatura	<code>/topic/estoque</code> (geral), <code>/topic/area/{areaId}</code> , <code>/topic/entidade/{entidade}</code>
Autenticação	JWT no frame <code>CONNECT</code> (<code>Authorization: Bearer ...</code> , <code>authorization</code> , <code>X-Authorization</code> , <code>token</code>) ou <code>?token=</code> no handshake

Corpo do evento (`RealtimeEvent`, JSON):

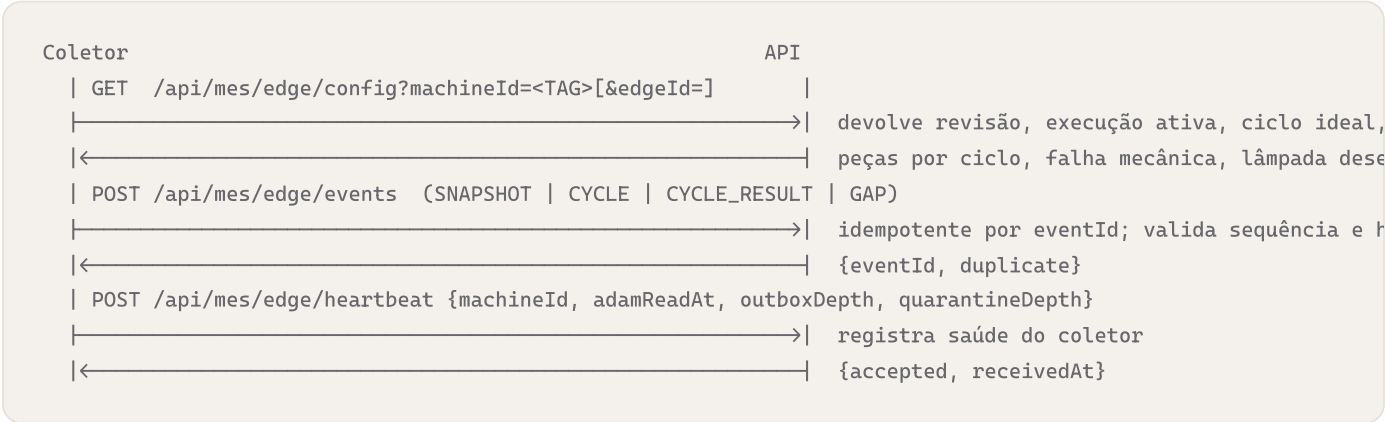
json

```
{
  "tipo": "ITEMESTOQUE_ATUALIZADO",
  "entidade": "ItemEstoque",
  "entidadeId": 123,
  "areaEstoqueId": 4,
  "payload": {},
  "origemUsuario": "usuario@exemplo.com",
  "momento": "2026-10-15T12:00:00Z"
}
```

- `tipo` = `<ENTIDADE em maiúsculas>_<AÇÃO>`, com ação `CRIADO`, `ATUALIZADO` ou `REMOVIDO` (`EntityChangeListener`). Também publicam eventos próprios `NivelMaterialService` e `TabelaEncertoService` (listas), com tipos definidos por esses serviços.
- O módulo MES/PCP não publica eventos STOMP (nenhum `RealtimeEvent` sob `mes/`); a atualização da tela PCP é por consulta (botão "Atualizar" e consulta automática a cada 1 segundo enquanto a aba está em foco, conforme `MesScreen.tsx`).
- O evento é publicado **depois do commit** da transação.
- `origemUsuario` é "best effort" e pode ser nulo.

1.3 Edge (coletor industrial) → API

Fluxo comprovado (`MesEdgeController`, `MesService`, `MesRules`):



Regras confirmadas (detalhes na ficha [modulos/edge.md](#)):

- Cabeçalho `X-Edge-Key` obrigatório. Chave global (`MES_EDGE_KEY` , mínimo 16 bytes); sem ela configurada, as rotas respondem `503` .
- `machineId` deve ser a **Tag** da injetora cadastrada no sistema (a tela de Injetoras diz: "A Tag deve ser a mesma configurada no serviço edge; não muda depois de criada"). Máquina sem vínculo → erro `Edge nao vinculado a maquina` .
- **Idempotência**: o mesmo `eventId` com o mesmo corpo é aceito como duplicado (`duplicate: true`); com corpo diferente, `409` . Reutilizar a `sequence` com outro `eventId` também gera conflito.
- **Sequência**: saltos exigem validação manual e **não contabilizam peças daquele ciclo**.
- **Horário**: eventos com mais de 5 minutos no futuro são rejeitados.
- **Tipos**: `CYCLE` exige execução original (`executionId`) e tempos; `schemaVersion ≥ 3` exige `namedSignals` (`BOMBA` , `MOLDE_FECHADO` , `ENCOSTE_BICO` , `INJECAO` , `ALARME`).
- **Lâmpada**: `lamp` ∈ `OFF | YELLOW | RED | BOTH` ; `outputStatus` ∈ `DISABLED | CONFIRMED | ERROR` .
- **Meta da OS**: o servidor continua recebendo eventos depois da meta (`TARGET_REACHED` no histórico), mas **não soma peças nem ciclos** aos indicadores; o bloqueio é de contabilização, **não envia comando de parada física** (ver [docs/PCP-META-E-SINALIZACAO-2026-09-15.md](#)).
- O firmware/serviço do coletor **não faz parte** dos dois projetos analisados.

1.4 Importação de OS por PDF

POST `/api/mes/orders/import-preview` (multipart, campo `file` , permissão de planejamento `mesAccess.plan`):

1. Extrai o texto do PDF (PDFBox). Se o PDF não tem camada de texto, tenta **OCR da primeira página** com Tesseract (idioma `por` , 300 dpi, limite de 30 s). O executável é procurado em `MES_TESSERACT_PATH` , em dois caminhos fixos do Windows e no `PATH` .
2. Reconhece rótulos fixos: `OS:` , `Data:` , `Produto:` , `Código:` , `Múltiplo:` , `Lote:` , `N° Máquina:` , `Ciclo:` e as linhas de produto/ferramenta.
3. Devolve uma **prévia** (não cria a ordem). Sem código de OS reconhecido, devolve erro.
4. O usuário confere e confirma; só então o frontend chama **POST** `/api/mes/orders` .

No frontend, a tela de Ordem de Serviço aceita vários PDFs por vez e avisa quando produto ou molde do PDF não existem no cadastro ("garanta que o produto e o molde já estejam cadastrados no sistema").

2. TUTOM — situação atual e possibilidades

O que o código afirma: a palavra "TUTOM" **não aparece** no código-fonte dos dois projetos. Ela aparece nas propostas e especificações do projeto (`02_PROPOSTA_V3_1_OPERACAO_PCP.md` , `docs/superpowers/specs/2026-09-15-pcp-operacao-design.md`), que descrevem o PDF importado como sendo do **TUTOM**, e afirmam que **não há integração direta (API) com o TUTOM**. Portanto:

ITEM	SITUAÇÃO
Importação de OS do TUTOM via PDF	Implementada (item 1.4), dependente do layout do PDF
Chamada do sistema Wester ao TUTOM (saída)	Não implementada
Chamada do TUTOM à API Wester (entrada)	Não implementada como integração dedicada; a API não tem chave de integração para sistemas externos além do Edge
Webhooks / eventos de saída para sistemas externos	Não implementado (o único canal de saída é o STOMP para o frontend)

2.1 Possibilidades (não implementadas) — com o que o código já oferece

POSSIBILIDADE	COMO SERIA, COM BASE NAS ROTAS EXISTENTES	PONTOS A DECIDIR
Cadastro automático de OS (sem PDF)	O TUTOM, ou um serviço intermediário, chama POST <code>/api/mes/orders</code> com <code>MesOrderInput</code> (<code>orderCode</code> , <code>productCode</code> , <code>productName</code> , <code>quantityPieces</code> , <code>piecesPerCycle</code> , <code>idealCycleMs</code> , <code>toolCode</code> , <code>toolSettings.lot</code> , <code>notes</code>). A rota já existe e exige permissão de planejamento	Não há chave de API para terceiros: seria preciso um usuário de serviço com perfil <code>MES_PCP</code> + <code>FULL_ACCESS</code> que faz login e renova o token a cada 8 h. Definir quem é a fonte da verdade quando a OS já existe (a tela do PCP avisa "A OS ... já existe na injetora ..."; a regra equivalente no backend não foi isolada neste levantamento)
Conferir produto/molde antes de enviar a OS	GET <code>/api/produtos/codigo/{codigoWester}</code> e GET <code>/api/molde</code> para validar códigos; é o mesmo cruzamento que o PDF faz hoje	Permissões <code>SCREEN_PRODUCTS</code> / <code>SCREEN_MOLDS</code> para o usuário de serviço
Devolver produção ao TUTOM	Consulta periódica a GET <code>/api/mes/history</code> e GET <code>/api/mes/reports</code> (permissão de leitura MES) para ler peças boas, refugo, ciclos e paradas por OS	Frequência de consulta, formato esperado pelo TUTOM, e que indicadores são confiáveis: o próprio sistema registra que OEE, MTBF e MTTR exigem validação
Notificação em tempo real	Um cliente STOMP autenticado assinando <code>/topic/entidade/...</code> poderia reagir a mudanças	Definir o controle de acesso por tópico; sem eventos específicos do MES documentados neste levantamento
Reconhecimento mais robusto do PDF	Evoluir <code>MesOrderPdfParser</code> com amostras reais do TUTOM	A própria proposta lista "variações de layout" como risco

Nenhuma dessas possibilidades foi testada. A tabela mostra apenas o que as rotas existentes permitiriam.

3. Segurança das integrações

- Credenciais de integração (`MES_EDGE_KEY` , `SECURITY_JWT_SECRET`) vêm de variáveis de ambiente; **não estão no código** e **não devem** ser copiadas para documentação ou repositório.
- O Edge e a API principal usam cadeias de segurança separadas; a chave do Edge não abre outras rotas.
- Qualquer integração de servidor para servidor não depende de CORS; um navegador de outra origem depende da lista de origens em `autenticacao.md` §8.

Tratamento de erros

← Índice da documentação da API

Fontes: `controller/exception/GlobalExceptionHandler.java` , `mes/MesErrors.java` ,
`mes/MesEdgeSecurity.java` , `config/LoginRateLimitingFilter.java` ,
`config/PreflightAwareAuthenticationEntryPoint.java` , `util/MensagemErro.java` , e, no frontend,
`src/axios.ts` e `src/utils/userFacingError.ts` .

1. Formato padrão de erro (`ErrorResponseDTO`)

A maior parte dos erros sai pelo `GlobalExceptionHandler` neste formato:

```
{
  "timestamp": "2026-10-15T09:41:07.123456",
  "status": 400,
  "error": "Bad Request",
  "message": "Descrição legível do problema.",
  "path": "/api/produtos"
}
```

json

CAMPO	TIPO	OBSERVAÇÃO
<code>timestamp</code>	<code>LocalDateTime</code> ISO-8601	hora do servidor, sem fuso
<code>status</code>	inteiro	mesmo valor do status HTTP
<code>error</code>	texto	descrição padrão do status HTTP em inglês (<code>Bad Request</code> , <code>Conflict</code> ...)
<code>message</code>	texto	em português , é o texto mostrado ao usuário pelo frontend; se vier vazio vira <code>Erro interno inesperado.</code>
<code>path</code>	texto	URI da requisição

2. Mapeamento exceção → status (cadeia global)

SITUAÇÃO / EXCEÇÃO	STATUS	MESSAGE
Bean Validation falhou (@Valid)	400	mensagens dos campos separadas por ; (ou Dados inválidos.)
IllegalArgumentException	400	a mensagem da exceção (regra de negócio violada)
DataIntegrityViolationException	400	Já existe um produto com este código Wester ... (violação de codigo_sistema_wester); Registro duplicado detectado. Verifique os campos únicos.; Não foi possível concluir a operação por vínculo com outros registros.; OU Violação de integridade de dados. Verifique os campos enviados.
InvalidCredentialsException	401	Usuário ou senha inválidos. / Informe login e senha para continuar.
AuthenticationCredentialsNot FoundException	401	mensagem da exceção
InactiveUserException	403	Usuário inativo. Procure um administrador.
AuthorizationDeniedException (falha de @PreAuthorize)	403	Você não tem permissão para executar esta ação.
RuntimeException cuja mensagem contém "não encontrad(o/a)"	404	a mensagem (comparação sem acento e sem gênero, via MensagemErro)
HttpRequestMethodNotSupportedException	405	mensagem do Spring
IllegalStateException	409	a mensagem (conflito de estado)
Conflito de versão otimista (ObjectOptimisticLockingFailureException , OptimisticLockException)	409	OPTIMISTIC_LOCK: O registro foi alterado por outro usuário. Atualize os dados e tente novamente.
ResponseStatusException	o status da exceção	reason da exceção
Qualquer outra RuntimeException / Exception	500	mensagem da exceção; erro registrado em log
Cliente desconectou no meio da resposta (AsyncRequestNotUsableException)	—	apenas log em nível debug ; não gera corpo

Pontos a saber:

- O prefixo `OPTIMISTIC_LOCK:` na mensagem é um contrato com o frontend, que abre o `ConflictDialog` ("alterado por outro usuário"). Entidades com controle de versão (`@Version`) no código: `Fileira`, `Grade`, `Nivel` e `ItemEstoque` (campo `version`).
- **404 depende do texto da mensagem.** Uma exceção genérica só vira `404` se a mensagem contiver "não encontrad..."; se o texto mudar, o status passa a `500`. Está documentado no próprio utilitário `MensagemErro`.
- O texto de `IllegalArgumentException` e `IllegalStateException` vai direto ao cliente. Por isso as mensagens dos serviços são escritas em português para o usuário final.

3. Formatos de erro diferentes do padrão

Nem todo erro usa o `ErrorResponseDTO`. Integradores devem tratar estes formatos:

ORIGEM	STATUS	CORPO
<code>UsuarioController</code> (criar, atualizar, alterar senha, status, excluir)	<code>400</code> , <code>403</code> , <code>404</code> , <code>500</code>	texto puro (<code>String</code>) no corpo, por exemplo <code>Senha é obrigatória.</code> e <code>Você não tem permissão para alterar a senha de outra conta.</code>
MES — <code>MesErrors</code> (apenas rotas de <code>MesController</code>)	<code>400</code> , status de <code>ResponseStatusException</code> , <code>403</code>	<code>{"message": "..."} (Dados invalidos, Operacao recusada, Sem permissao quando não há texto)</code>
Edge — <code>MesEdgeSecurity</code>	<code>401</code> ou <code>503</code>	<code>{"message":"Integracao edge indisponivel ou credencial invalida"}</code>
Limite de taxa — <code>LoginRateLimitingFilter</code>	<code>429</code>	<code>{"status":429,"error":"Too Many Requests","message":"Muitas tentativas. Aguarde um momento antes de tentar novamente."}</code>
Falta de autenticação na cadeia principal (<code>PreflightAwareAuthenticationEntryPoint</code>)	<code>401</code>	sem corpo (apenas o status); um pré-voos CORS recebe <code>200</code>

`PreflightAwareAuthenticationEntryPoint` define apenas o status: `401` sem corpo quando falta autenticação em uma rota protegida (e `200` para pré-voos CORS). Um cliente **não deve** esperar `ErrorResponseDTO` nesse caso.

Os controllers do MES que respondem `ResponseStatusException` seguem `MesErrors`, não o handler global, porque o `@RestControllerAdvice` do MES tem `@Order(0)` e escopo `basePackageClasses=MesController.class`.

4. Códigos HTTP que a API pode devolver

STATUS	QUANDO
200	consulta ou alteração com corpo
201	criação (<code>POST</code> <code>/api/usuarios</code> e outras criações que declaram <code>201</code>)
204	alteração sem corpo (por exemplo, troca de senha)
400	validação, regra de negócio, integridade de dados, JSON inválido
401	token ausente/inválido/expirado, usuário ou senha inválidos, chave do Edge incorreta
403	usuário inativo, falta de permissão (<code>SCREEN_*</code> / <code>FULL_ACCESS</code> /MES)
404	recurso não encontrado
405	método HTTP não suportado na rota
409	conflito de estado ou de versão (otimista); MES também usa para sequência de eventos do coletor
429	excesso de tentativas de login/cadastro por IP
500	erro não tratado
503	integração Edge indisponível (chave não configurada)

Cada ficha de endpoint (`modulos/*.md`) lista os códigos declarados e as mensagens de validação identificadas no serviço. Ela **não** garante que todas as mensagens possíveis estejam listadas: as mensagens dinâmicas e as que vêm de métodos auxiliares aparecem em documento interno de regras de negócio.

5. Como o frontend trata os erros (para o manual e para integradores)

- 401 com token anexado:** o `axios` dispara o *handler* de não autorizado, que faz logout e volta para a tela de Login (o token vale 8 horas).
- Mensagem exibida:** `getUserFacingErrorMessage` usa o `message` da API, **descartando** textos técnicos (`network error`, `internal server error`, `not found`, `timeout`, `404`, `500`, HTML, `stack trace` ...) e caindo na mensagem padrão da tela. Para `401` / `403` sem texto aproveitável, mostra `Você não tem permissão para concluir esta operação.`
- Conflito 409:** mostra a mensagem da API ou `Os dados foram alterados por outro usuário. Atualize a tela e tente novamente.`
- Login:** `401` → `E-mail ou senha inválidos.`; `403` → `Seu usuário não pode acessar o sistema.`; sem resposta do servidor → `Não foi possível conectar ao servidor. Tente novamente.`

6. Recomendações para quem integra

- Trate o corpo de erro como **texto** ou **JSON** dependendo da rota (seção 3); verifique o `Content-Type`.
- Use o `status` HTTP como contrato; o `message` é para pessoas e pode mudar.
- Em `409` com prefixo `OPTIMISTIC_LOCK:`, releia o recurso (incluindo `version`) antes de tentar de novo.

4. Em **429**, espere um minuto antes de repetir.

5. Para o Edge, **503** significa que a chave **não está configurada no servidor** (não é erro do coletor), e **401** que a chave enviada não confere.

Endpoints implementados

211 combinações de método e rota, correspondentes a 145 métodos de controller, em 16 agrupamentos funcionais. Aliases contam como rotas expostas, não como funcionalidades diferentes. Não inclui rotas automáticas do Spring, recursos estáticos, OPTIONS/HEAD implícitos nem transporte SockJS. URL de cada chamada = `BASE_URL` + rota completa abaixo. Não há context-path adicional declarado nos arquivos ativos consultados; a implantação pode adicionar prefixo de proxy.

MÓDULO	OPERAÇÕES LÓGICAS	MÉTODO + ROTA
Acesso, usuários e perfis	16	27
Produtos	10	32
Categorias de produtos	6	12
Setores e tipos de material	11	32
Corredores, blocos e níveis	21	21
Itens e transferências de estoque	5	5
Moldes	7	13
Materiais fixos e slots legados	15	15
Listas de materiais	10	10
Consultas de estoque e mapa	3	3
Histórico e auditoria	4	4
Movimentações operacionais	3	3
Alertas de estoque	1	1
Relatórios de estoque	4	4
PCP e produção	26	26
Integração industrial Edge	3	3

Índice de todas as rotas

ID	MÉTODO	ROTA	CONTRATO
API-001	POST	/api/alertas-estoque	consultarAlertas
API-002	GET	/api/areas	listarTodasAreas
API-002	GET	/api/setores	listarTodasAreas
API-003	GET	/api/areas/{id}	buscarAreaPorId
API-003	GET	/api/setores/{id}	buscarAreaPorId
API-004	GET	/api/areas/nome/{nome}	buscarPorNome
API-004	GET	/api/setores/nome/{nome}	buscarPorNome
API-005	POST	/api/areas	criarArea
API-005	POST	/api/setores	criarArea
API-006	PUT	/api/areas/{id}	atualizarArea
API-006	PUT	/api/setores/{id}	atualizarArea
API-007	PATCH	/api/areas/{id}/labels	atualizarLabels
API-007	PATCH	/api/setores/{id}/labels	atualizarLabels
API-008	PATCH	/api/areas/{id}/activate	ativarArea
API-008	POST	/api/areas/{id}/activate	ativarArea
API-008	PATCH	/api/areas/{id}/ativar	ativarArea
API-008	POST	/api/areas/{id}/ativar	ativarArea
API-008	PATCH	/api/setores/{id}/activate	ativarArea
API-008	POST	/api/setores/{id}/activate	ativarArea
API-008	PATCH	/api/setores/{id}/ativar	ativarArea
API-008	POST	/api/setores/{id}/ativar	ativarArea
API-009	PATCH	/api/areas/{id}/inactivate	inativarArea
API-009	POST	/api/areas/{id}/inactivate	inativarArea
API-009	PATCH	/api/areas/{id}/inativar	inativarArea
API-009	POST	/api/areas/{id}/inativar	inativarArea
API-009	PATCH	/api/setores/{id}/inactivate	inativarArea
API-009	POST	/api/setores/{id}/inactivate	inativarArea
API-009	PATCH	/api/setores/{id}/inativar	inativarArea
API-009	POST	/api/setores/{id}/inativar	inativarArea

ID	MÉTODO	ROTA	CONTRATO
API-010	DELETE	/api/areas/{id}	deletarArea
API-010	DELETE	/api/setores/{id}	deletarArea
API-011	GET	/api/area-material-tags/area/{areaId}	listar
API-012	PUT	/api/area-material-tags/area/{areaId}	configurar
API-013	POST	/api/auth/login	login
API-013	POST	/api/usuarios/login	login
API-014	GET	/api/auth/me	me
API-015	GET	/api/categorias-produto	listar
API-015	GET	/api/product-categories	listar
API-016	POST	/api/categorias-produto	criar
API-016	POST	/api/product-categories	criar
API-017	PUT	/api/categorias-produto/{id}	atualizar
API-017	PUT	/api/product-categories/{id}	atualizar
API-018	DELETE	/api/categorias-produto/{id}	excluir
API-018	DELETE	/api/product-categories/{id}	excluir
API-019	PUT	/api/categorias-produto/{id}/produtos	vincular
API-019	PUT	/api/product-categories/{id}/produtos	vincular
API-020	DELETE	/api/categorias-produto/{id}/produtos	desvincular
API-020	DELETE	/api/product-categories/{id}/produtos	desvincular
API-021	GET	/api/estoque/itens	listarItensDashboard
API-022	GET	/api/estoque/itens-consolidado	listarItensConsolidado
API-023	GET	/api/estoque/posicoes/area/{areaId}	buscarMapaPorArea
API-024	GET	/api/fileiras/area/{areaId}	listarFileirasPorArea
API-025	GET	/api/fileiras/{id}	buscarFileiraPorId
API-026	POST	/api/fileiras/area/{areaId}	criarFileira
API-027	PUT	/api/fileiras/{id}	atualizarFileira
API-028	PATCH	/api/fileiras/{id}/identificador	atualizarIdentificador
API-029	DELETE	/api/fileiras/{id}	deletarFileira
API-030	GET	/api/grades/fileira/{fileiraId}	listarGradesPorFileira
API-031	GET	/api/grades/{id}	buscarGradePorId
API-032	POST	/api/grades/fileira/{fileiraId}	criarGrade

ID	MÉTODO	ROTA	CONTRATO
API-033	PUT	/api/grades/{id}	atualizarGrade
API-034	PATCH	/api/grades/{id}/identificador	atualizarIdentificador
API-035	DELETE	/api/grades/{id}	deletarGrade
API-036	GET	/api/historico-movimentacao	listar
API-037	POST	/api/historico-movimentacao/filter	filtrar
API-038	GET	/api/historico-movimentacao/{id}	buscarPorId
API-039	GET	/api/historico-movimentacao/entidade/{entidade}/{entidadeId}	listarPorEntidade
API-040	GET	/api/itens-estoque/nivel/{nivelId}	buscarPorNivel
API-041	PUT	/api/itens-estoque/nivel/{nivelId}	upsertPorNivel
API-042	POST	/api/itens-estoque/nivel/{nivelId}	criarPorNivel
API-043	POST	/api/itens-estoque/movimentacao	movimentar
API-044	DELETE	/api/itens-estoque/nivel/{nivelId}	deletePorNivel
API-045	GET	/api/moldes	listar
API-046	GET	/api/moldes/{id}	buscarPorId
API-047	POST	/api/moldes	criar
API-048	PUT	/api/moldes/{id}	atualizar
API-049	PATCH	/api/moldes/{id}/ativar	ativar
API-049	POST	/api/moldes/{id}/ativar	ativar
API-049	PATCH	/api/moldes/{id}/activate	ativar
API-049	POST	/api/moldes/{id}/activate	ativar
API-050	PATCH	/api/moldes/{id}/inativar	inativar
API-050	POST	/api/moldes/{id}/inativar	inativar
API-050	PATCH	/api/moldes/{id}/inactivate	inativar
API-050	POST	/api/moldes/{id}/inactivate	inativar
API-051	DELETE	/api/moldes/{id}	deletar
API-052	GET	/api/movimentacoes/contexto-inicial	buscarContextoInicial
API-053	POST	/api/movimentacoes	registrarMovimentacao
API-054	GET	/api/movimentacoes/recentes	listarRecentes
API-055	GET	/api/niveis/grade/{gradeId}	listarNiveisPorGrade

ID	MÉTODO	ROTA	CONTRATO
API-056	GET	/api/niveis/{id}	buscarNivelPorId
API-057	POST	/api/niveis/grade/{gradeId}	criarNivel
API-058	POST	/api/niveis/{nivelId}/move	moverNivel
API-059	PUT	/api/niveis/{id}	atualizarNivel
API-060	PATCH	/api/niveis/{id}/identificador	atualizarIdentificador
API-061	DELETE	/api/niveis/{nivelId}/resequence	deleteAndResequence
API-062	DELETE	/api/niveis/{nivelId}	deleteNivelOnly
API-063	DELETE	/api/niveis/{nivelId}/desativar	desativarNivel
API-064	GET	/api/nivel-materiais/nivel/{nivelId}	listar
API-065	POST	/api/nivel-materiais/nivel/{nivelId}/alocar-batch	alocarBatch
API-066	DELETE	/api/nivel-materiais/{nivelMaterialId}	remover
API-067	PATCH	/api/nivel-materiais/{nivelMaterialId}/quantidade	atualizarQuantidade
API-068	DELETE	/api/nivel-materiais/nivel/{nivelId}/limpar	limpar
API-069	GET	/api/perfis	listarPerfis
API-069	GET	/api/profiles	listarPerfis
API-070	GET	/api/perfis/ativos	listarPerfisAtivos
API-070	GET	/api/profiles/ativos	listarPerfisAtivos
API-071	POST	/api/perfis	criarPerfil
API-071	POST	/api/profiles	criarPerfil
API-072	PUT	/api/perfis/{id}	atualizarPerfil
API-072	PUT	/api/profiles/{id}	atualizarPerfil
API-073	PATCH	/api/perfis/{id}/ativar	ativarPerfil
API-073	PATCH	/api/perfis/{id}/activate	ativarPerfil
API-073	PATCH	/api/profiles/{id}/ativar	ativarPerfil
API-073	PATCH	/api/profiles/{id}/activate	ativarPerfil
API-074	PATCH	/api/perfis/{id}/inativar	inativarPerfil
API-074	PATCH	/api/perfis/{id}/inactivate	inativarPerfil
API-074	PATCH	/api/profiles/{id}/inativar	inativarPerfil
API-074	PATCH	/api/profiles/{id}/inactivate	inativarPerfil

ID	MÉTODO	ROTA	CONTRATO
API-075	GET	/api/products	listarTodosProdutos
API-075	GET	/api/produtos	listarTodosProdutos
API-076	GET	/api/products/{id}	buscarProdutoPorId
API-076	GET	/api/produtos/{id}	buscarProdutoPorId
API-077	GET	/api/products/codigo/{codigoWester}	buscarPorCodigoWester
API-077	GET	/api/produtos/codigo/{codigoWester}	buscarPorCodigoWester
API-078	POST	/api/products	criarProduto
API-078	POST	/api/produtos	criarProduto
API-079	PUT	/api/products/{id}	atualizarProduto
API-079	PUT	/api/produtos/{id}	atualizarProduto
API-080	DELETE	/api/products/{id}	deletarProduto
API-080	DELETE	/api/produtos/{id}	deletarProduto
API-081	PATCH	/api/products/{id}/activate	ativarProduto
API-081	POST	/api/products/{id}/activate	ativarProduto
API-081	PATCH	/api/products/{id}/ativar	ativarProduto
API-081	POST	/api/products/{id}/ativar	ativarProduto
API-081	PATCH	/api/produtos/{id}/activate	ativarProduto
API-081	POST	/api/produtos/{id}/activate	ativarProduto
API-081	PATCH	/api/produtos/{id}/ativar	ativarProduto
API-081	POST	/api/produtos/{id}/ativar	ativarProduto
API-082	PATCH	/api/products/{id}/inactivate	inativarProduto
API-082	POST	/api/products/{id}/inactivate	inativarProduto
API-082	PATCH	/api/products/{id}/inativar	inativarProduto
API-082	POST	/api/products/{id}/inativar	inativarProduto
API-082	PATCH	/api/produtos/{id}/inactivate	inativarProduto
API-082	POST	/api/produtos/{id}/inactivate	inativarProduto
API-082	PATCH	/api/produtos/{id}/inativar	inativarProduto
API-082	POST	/api/produtos/{id}/inativar	inativarProduto
API-083	GET	/api/products/ativos	listarAtivos
API-083	GET	/api/produtos/ativos	listarAtivos
API-084	GET	/api/products/buscar	buscarProdutos

ID	MÉTODO	ROTA	CONTRATO
API-084	GET	/api/produtos/buscar	buscarProdutos
API-085	POST	/api/relatorios/produtos	gerarRelatorioProdutos
API-086	POST	/api/relatorios/estoques	gerarRelatorioEstoques
API-087	POST	/api/relatorios/itens-estoque	gerarRelatorioItensEstoque
API-088	POST	/api/relatorios/historico-movimentacao	gerarRelatorioHistoricoMovimentacao
API-089	GET	/api/slots/nivel/{nivelId}	listarPorNivel
API-090	GET	/api/slots/area/{areaId}	buscarPorArea
API-091	GET	/api/slots/estante/{fileiraId}	buscarEstante
API-092	POST	/api/slots/nivel/{nivelId}/inicializar	inicializarSlots
API-093	POST	/api/slots/nivel/{nivelId}	adicionarSlot
API-094	PATCH	/api/slots/{slotId}	atualizarSlot
API-095	DELETE	/api/slots/{slotId}	removerSlot
API-096	POST	/api/slots/{slotId}/alocar	alocar
API-097	DELETE	/api/slots/{slotId}/moldes/{moldeAlocadoId}	removerMolde
API-098	DELETE	/api/slots/{slotId}/limpar	limparSlot
API-099	GET	/api/encerto	listarPorArea
API-100	GET	/api/encerto/{id}	buscarPorId
API-101	POST	/api/encerto	criar
API-102	PATCH	/api/encerto/{id}	atualizar
API-103	DELETE	/api/encerto/{id}	excluir
API-104	POST	/api/encerto/{tabelaId}/itens	adicionarItem
API-105	DELETE	/api/encerto/itens/{itemId}	removerItem
API-106	PATCH	/api/encerto/itens/{itemId}/quantidade	atualizarQuantidade
API-107	PATCH	/api/encerto/itens/{itemId}/observacao	atualizarObservacao
API-108	PATCH	/api/encerto/itens/{itemId}/ordem	reordenar
API-109	GET	/api/usuarios	listarTodosUsuarios
API-110	GET	/api/usuarios/id/{id}	buscarUsuarioPorId
API-111	GET	/api/usuarios/by-login/{login}	buscarPorLogin
API-112	POST	/api/usuarios	criarUsuario
API-113	PUT	/api/usuarios/id/{id}	atualizarUsuario

ID	MÉTODO	ROTA	CONTRATO
API-114	POST	/api/usuarios/id/{id}/alterar-senha	alterarSenha
API-115	PATCH	/api/usuarios/id/{id}/status	atualizarStatusUsuario
API-116	DELETE	/api/usuarios/id/{id}	deletarUsuario
API-117	GET	/api/mes/permissions	permissions
API-118	GET	/api/mes/machines	machines
API-119	POST	/api/mes/machines	createMachine
API-120	PUT	/api/mes/machines/{id}	machine
API-121	GET	/api/mes/machines/{id}/status	status
API-122	GET	/api/mes/orders	orders
API-123	POST	/api/mes/orders	createOrder
API-124	DELETE	/api/mes/orders/{id}	deleteOrder
API-125	PUT	/api/mes/orders/{id}	updateOrder
API-126	POST	/api/mes/orders/import-preview	importPreview
API-127	PUT	/api/mes/orders/{id}/position	position
API-128	PUT	/api/mes/orders/{id}/machine	assignMachine
API-129	POST	/api/mes/orders/{id}/prepare	prepare
API-130	POST	/api/mes/executions/{id}/unprepare	unprepare
API-131	GET	/api/mes/executions/{id}	execution
API-132	POST	/api/mes/executions/{id}/start	start
API-133	POST	/api/mes/executions/{id}/finish	finish
API-134	POST	/api/mes/executions/{id}/reallocate	reallocate
API-135	POST	/api/mes/executions/{id}/quality	quality
API-136	GET	/api/mes/executions/{id}/quality	qualities
API-137	POST	/api/mes/machines/{id}/downtimes	downtime
API-138	POST	/api/mes/downtimes/{id}/release	release
API-139	GET	/api/mes/machines/{id}/downtimes	downtimes
API-140	GET	/api/mes/machines/{id}/downtimes/active	activeDowntime
API-141	GET	/api/mes/reports	reports
API-142	GET	/api/mes/history	history
API-143	GET	/api/mes/edge/config	config

ID	MÉTODO	ROTA	CONTRATO
API-144	POST	/api/mes/edge/events	events
API-145	POST	/api/mes/edge/heartbeat	heartbeat

Modelos e contratos de dados

Os campos abaixo derivam dos DTOs, entidades serializadas, getters Jackson e contratos dinâmicos do MES. Colunas internas do banco não são automaticamente campos de entrada. `obrigatório` representa as anotações do DTO; só é aplicado automaticamente quando o argumento recebe `@Valid`. Regras condicionais estão nos serviços. Campos não obrigatórios não significam que `null` e omissão tenham o mesmo efeito. Exemplos nos endpoints usam dados fictícios e exigem referências reais do ambiente. Paginação Spring Data usa `content`, `number`, `size`, `totalElements`, `totalPages`, `first`, `last`, `empty`, `numberOfElements` e metadados `pageable` / `sort`. O MES usa `content`, `page`, `size`, `totalElements`, `totalPages`. Relatórios WMS usam `pagination` e `items`; não trocar os formatos.

AdicionarItemEncertoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AdicionarItemEncertoRequestDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>produtoId</code>	integer	Sim	
<code>quantidade</code>	integer	Não	
<code>observacao</code>	string	Não	

AlertaEstoqueFiltroDTO

Fonte: `src/main/java/com/wester/storage/dto/AlertaEstoqueFiltroDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>tipoAlerta</code>	string	Não	Filtros para consulta de alertas de estoque
<code>produtoId</code>	integer	Não	Identificador do produto
<code>textoLivre</code>	string	Não	Busca livre por nome, código ou descrição do produto
<code>codigoWester</code>	string	Não	Código Wester exato
<code>areaId</code>	integer	Não	Identificador da área/setor
<code>criticidade</code>	string	Não	Criticidade do alerta
<code>produtoAtivo</code>	boolean	Não	Status do produto
<code>diasSemMovimentacao</code>	integer	Não	Quantidade mínima de dias sem movimentação para gerar alerta
<code>ordenarPor</code>	string	Não	Campo de ordenação
<code>direcaoOrdenacao</code>	string	Não	Direção da ordenação
<code>pagina</code>	integer	Não	Página da consulta
<code>tamanhoPagina</code>	integer	Não	Tamanho da página

AlertaEstoqueItemDTO

Fonte: `src/main/java/com/wester/storage/dto/AlertaEstoqueItemDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>tipoAlerta</code>	string	Não	
<code>criticidade</code>	string	Não	
<code>mensagem</code>	string	Não	
<code>produtoId</code>	integer	Não	
<code>produtoNomeModelo</code>	string	Não	
<code>produtoCodigoSistemaWester</code>	string	Não	
<code>produtoCor</code>	string	Não	
<code>produtoAtivo</code>	boolean	Não	
<code>quantidadeTotal</code>	integer	Não	
<code>estoqueMinimo</code>	integer	Não	
<code>estoqueMaximo</code>	integer	Não	
<code>areaId</code>	integer	Não	
<code>areaNome</code>	string	Não	
<code>itemEstoqueId</code>	integer	Não	
<code>nivelId</code>	integer	Não	
<code>localizacaoResumo</code>	string	Não	
<code>ultimaMovimentacaoEm</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>diasSemMovimentacao</code>	integer	Não	

AlertaEstoqueResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/AlertaEstoqueResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>itens</code>	array de <code>AlertaEstoqueItemDTO</code>	Não	
<code>pagina</code>	integer	Não	
<code>tamanhoPagina</code>	integer	Não	
<code>totalItens</code>	integer	Não	
<code>totalPaginas</code>	integer	Não	
<code>resumo</code>	<code>AlertaEstoqueResumoDTO</code>	Não	

AlertaEstoqueResumoDTO

Fonte: `src/main/java/com/wester/storage/dto/AlertaEstoqueResumoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>totalAlertas</code>	integer	Não	
<code>produtosSemEstoque</code>	integer	Não	
<code>produtosAbaixoMinimo</code>	integer	Não	
<code>produtosAcimaMaximo</code>	integer	Não	
<code>produtosSemMovimentacao</code>	integer	Não	
<code>criticidadeAlta</code>	integer	Não	
<code>criticidadeMedia</code>	integer	Não	
<code>criticidadeBaixa</code>	integer	Não	

AlocarMateriaisRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AlocarMateriaisRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>materiais</code>	array de AlocarMaterialRequestDTO	Sim	

AlocarMaterialRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AlocarMaterialRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>materialId</code>	integer	Sim	
<code>materialType</code>	string	Sim	
<code>ordem</code>	integer	Não	
<code>quantidade</code>	integer	Não	
<code>quantidadeAproximada</code>	boolean	Não	

AlocarMoldeRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AlocarMoldeRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>moldeId</code>	integer	Sim	DTO para alocar um molde em um slot de nível
<code>ordem</code>	integer	Não	Ordem do molde dentro do slot

ApiErrorDTO

Fonte: `src/main/java/com/wester/storage/dto/ApiErrorDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>message</code>	string	Não	

AreaLabelsUpdateRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AreaLabelsUpdateRequestDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>labelEstruturaPrimaria</code>	string	Não	Payload para atualizar labels visuais do setor Aliases de entrada: {"labelFileira", "label_fileira", "nomenclaturaFileira"}
<code>labelEstruturaSecundaria</code>	string	Não	Aliases de entrada: {"labelGrade", "label_grade", "nomenclaturaGrade"}
<code>labelNivel</code>	string	Não	Aliases de entrada: {"nomenclaturaNivel"}
<code>labelSlot</code>	string	Não	Aliases de entrada: {"labelPosicao", "label_posicao", "nomenclaturaPosicao"}

AreaMaterialTagsRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AreaMaterialTagsRequestDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>materialTypes</code>	array de string	Sim	

AtualizarObservacaoEncertoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AtualizarObservacaoEncertoRequestDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>observacao</code>	string	Não	

AtualizarQuantidadeEncertoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AtualizarQuantidadeEncertoRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>quantidade</code>	integer	Sim	

AtualizarQuantidadeNivelMaterialRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AtualizarQuantidadeNivelMaterialRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>quantidade</code>	integer	Sim	
<code>quantidadeAproximada</code>	boolean	Não	

AtualizarTabelaEncertoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/AtualizarTabelaEncertoRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>titulo</code>	string	Não	
<code>colunas</code>	array de string	Não	

CategoriaProdutoDTO

Fonte: `src/main/java/com/wester/storage/dto/CategoriaProdutoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>codigo</code>	string	Não	
<code>descricao</code>	string	Não	
<code>totalProdutos</code>	integer	Não	Quantidade de produtos vinculados a esta categoria

CategoriaProdutoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/CategoriaProdutoRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>codigo</code>	string	Sim	Dados para criar/atualizar categoria de produto Não pode ser vazio ou conter apenas espaços.
<code>descricao</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.

CategoriaProdutoVinculoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/CategoriaProdutoVinculoRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>produtoIds</code>	array de integer	Sim	Produtos a vincular/desvincular de uma categoria

CriarFileiraRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/CriarFileiraRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>identificador</code>	string	Não	Payload para cria\u00e7\u00e3o de fileira
<code>ordem</code>	integer	Não	Opcional. O backend recalcula a ordem final com base na \u00e9rea de estoque.

CriarTabelaEncertoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/CriarTabelaEncertoRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>areaId</code>	integer	Sim	
<code>titulo</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>colunas</code>	array de string	Não	

ErrorResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/ErrorResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>timestamp</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>status</code>	integer	Não	
<code>error</code>	string	Não	
<code>message</code>	string	Não	
<code>path</code>	string	Não	

NivelFixoResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/EstanteFixaResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>nivelId</code>	integer	Não	
<code>nivelIdentificador</code>	string	Não	
<code>nivelOrdem</code>	integer	Não	
<code>slots</code>	array de SlotNivelResponseDTO	Não	
<code>materiais</code>	array de NivelMaterialResponseDTO	Não	

GradeFixaResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/EstanteFixaResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>gradeId</code>	integer	Não	
<code>gradeIdentificador</code>	string	Não	
<code>gradeOrdem</code>	integer	Não	
<code>niveis</code>	array de NivelFixoResponseDTO	Não	

EstanteFixaResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/EstanteFixaResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>fileiraId</code>	integer	Não	
<code>fileiraIdentificador</code>	string	Não	
<code>fileiraOrdem</code>	integer	Não	
<code>grades</code>	array de GradeFixaResponseDTO	Não	

EstoqueConsolidadoRowDTO

Fonte: `src/main/java/com/wester/storage/dto/EstoqueConsolidadoRowDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>registroId</code>	integer	Não	
<code>tipoEstrutura</code>	string	Não	
<code>areaId</code>	integer	Não	
<code>areaNome</code>	string	Não	
<code>labelEstruturaPrimaria</code>	string	Não	
<code>labelEstruturaSecundaria</code>	string	Não	
<code>labelNivel</code>	string	Não	
<code>labelSlot</code>	string	Não	
<code>fileiraId</code>	integer	Não	
<code>fileiraIdentificador</code>	string	Não	
<code>gradeId</code>	integer	Não	
<code>gradeIdentificador</code>	string	Não	
<code>nivelId</code>	integer	Não	
<code>nivelIdentificador</code>	string	Não	
<code>nivelOrdem</code>	integer	Não	
<code>slotId</code>	integer	Não	
<code>slotIdentificador</code>	string	Não	
<code>tipoMaterial</code>	string	Não	
<code>itemId</code>	integer	Não	
<code>codigo</code>	string	Não	
<code>nome</code>	string	Não	
<code>descricao</code>	string	Não	
<code>cor</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>quantidade</code>	integer	Não	
<code>quantidadeAproximada</code>	boolean	Não	
<code>estoqueMinimo</code>	integer	Não	
<code>estoqueMaximo</code>	integer	Não	
<code>observacao</code>	string	Não	

EstoqueItemConsolidadoDTO

Fonte: `src/main/java/com/wester/storage/dto/EstoqueItemConsolidadoDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>registroId</code>	integer	Não	Record id — <code>itemEstoqueId</code> (DINAMICO) or <code>nivelMaterialId</code> (FIXO)
<code>tipoEstrutura</code>	string	Não	Sector structure type
<code>areaId</code>	integer	Não	Area identifier
<code>areaNome</code>	string	Não	Area name
<code>labelEstruturaPrimaria</code>	string	Não	Label for primary structure (fileira)
<code>labelEstruturaSecundaria</code>	string	Não	Label for secondary structure (grade)
<code>labelNivel</code>	string	Não	Label for nivel
<code>labelSlot</code>	string	Não	Label for slot — null for DINAMICO
<code>localizacaoFormatada</code>	string	Não	Pre-formatted location string built from area labels
<code>fileiraId</code>	integer	Não	Fileira identifier
<code>fileiraIdentificador</code>	string	Não	Fileira code
<code>gradeId</code>	integer	Não	Grade identifier
<code>gradeIdentificado</code>	string	Não	Grade code
<code>nivelId</code>	integer	Não	Nivel identifier
<code>nivelIdentificado</code>	string	Não	Nivel code
<code>nivelOrdem</code>	integer	Não	Nivel order
<code>slotId</code>	integer	Não	Slot identifier — null for DINAMICO
<code>slotIdentificador</code>	string	Não	Slot code — null for DINAMICO
<code>tipoMaterial</code>	string	Não	Material type: PRODUTO or MOLDE — always PRODUTO for DINAMICO
<code>itemId</code>	integer	Não	Item id — <code>produtId</code> (DINAMICO) or <code>moldeId/produtId</code> (FIXO)
<code>codigo</code>	string	Não	Item code — <code>codigoSistemaWester</code> (DINAMICO) or <code>codigo/codigoSistemaWester</code> (FIXO)
<code>nome</code>	string	Não	Item name — <code>nomeModelo</code> (DINAMICO); FIXO uses <code>descricao</code> for moldes, <code>nomeModelo</code> for produtos
<code>descricao</code>	string	Não	Item description

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>cor</code>	string	Não	Color (DINAMICO or FIXO produto) or category (FIXO molde)
<code>ativo</code>	boolean	Não	Whether the item is active
<code>quantidade</code>	integer	Não	Quantity in stock — always 1 for FIXO moldes
<code>quantidadeAproximada</code>	boolean	Não	Whether the quantity is approximate — always false for FIXO moldes
<code>estoqueMinimo</code>	integer	Não	Minimum stock threshold — null for FIXO moldes
<code>estoqueMaximo</code>	integer	Não	Maximum stock threshold — null for FIXO moldes
<code>observacao</code>	string	Não	Additional observation

EstoqueItemConsolidadoPageResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/EstoqueItemConsolidadoPageResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>content</code>	array de EstoqueItemConsolidadoDTO	Não	Current page content
<code>page</code>	integer	Não	Current page index
<code>size</code>	integer	Não	Page size
<code>totalElements</code>	integer	Não	Total amount of matching rows
<code>totalPages</code>	integer	Não	Total amount of pages

EstoqueItemDashboardDTO

Fonte: `src/main/java/com/wester/storage/dto/EstoqueItemDashboardDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>itemEstoqueId</code>	integer	Não	Stock item identifier
<code>areaId</code>	integer	Não	Area identifier
<code>areaNome</code>	string	Não	Area name
<code>fileiraId</code>	integer	Não	Fileira identifier
<code>fileiraIdentificador</code>	string	Não	Fileira code
<code>gradeId</code>	integer	Não	Grade identifier
<code>gradeIdentificador</code>	string	Não	Grade code
<code>nivelId</code>	integer	Não	Nível identifier
<code>nivelIdentificador</code>	string	Não	Nível code
<code>nivelOrdem</code>	integer	Não	Nível order
<code>produtoId</code>	integer	Não	Produto identifier
<code>codigoSistemaWester</code>	string	Não	Wester system product code
<code>nomeModelo</code>	string	Não	Product model name
<code>descricao</code>	string	Não	Product description
<code>observacao</code>	string	Não	Item-specific note for this stock position
<code>cor</code>	string	Não	Product color
<code>produtoAtivo</code>	boolean	Não	Indicates whether the product is active
<code>estoqueMinimo</code>	integer	Não	Minimum stock threshold configured for the product
<code>estoqueMaximo</code>	integer	Não	Maximum stock threshold configured for the product
<code>quantidadeAproximada</code>	boolean	Não	Indicates whether the stored quantity is approximate
<code>quantidade</code>	integer	Não	Stored quantity

EstoqueItemDashboardPageResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/EstoqueItemDashboardPageResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>content</code>	array de EstoqueItemDashboardDTO	Não	Current page content
<code>page</code>	integer	Não	Current page index
<code>size</code>	integer	Não	Page size
<code>totalElements</code>	integer	Não	Total amount of matching rows
<code>totalPages</code>	integer	Não	Total amount of pages

EstoquePosicaoDTO

Fonte: `src/main/java/com/wester/storage/dto/EstoquePosicaoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>fileiraId</code>	integer	Não	ID da fileira
<code>fileiraIdentificado</code> <code>r</code>	string	Não	Identificador da fileira
<code>fileiraOrdem</code>	integer	Não	Ordem numerica da fileira
<code>gradeId</code>	integer	Não	ID da grade
<code>gradeIdentificador</code>	string	Não	Identificador da grade
<code>gradeOrdem</code>	integer	Não	Ordem numerica da grade
<code>nivelId</code>	integer	Não	ID do nivel
<code>nivelVersion</code>	integer	Não	Versao atual do nivel
<code>nivelIdentificador</code>	string	Não	Identificador do nivel
<code>nivelOrdem</code>	integer	Não	Ordem numerica do nivel
<code>itemEstoqueId</code>	integer	Não	ID do item em estoque (pode ser nulo quando vazio)
<code>itemEstoqueVersion</code>	integer	Não	Versao atual do item em estoque
<code>quantidade</code>	integer	Não	Quantidade armazenada no nivel
<code>quantidadeAproximad</code> <code>a</code>	boolean	Não	Indica se a quantidade armazenada e aproximada
<code>observacao</code>	string	Não	Observacao especifica do item nesta posicao do estoque
<code>produtoId</code>	integer	Não	ID do produto associado
<code>codigoSistemaWester</code>	string	Não	Codigo do produto no sistema Wester
<code>nomeModelo</code>	string	Não	Nome ou modelo do produto
<code>cor</code>	string	Não	Cor do produto
<code>descricao</code>	string	Não	Descrição do produto

FileiraResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/FileiraResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Resposta da criação/consulta de fileira
<code>identificador</code>	string	Não	Identificador da fileira
<code>ordem</code>	integer	Não	Ordem da fileira
<code>areaEstoqueId</code>	integer	Não	ID da área de estoque

GradeCreateRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/GradeCreateRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>identificador</code>	string	Não	Payload para criar uma grade vinculada a uma fileira
<code>ordem</code>	integer	Não	Opcional. O backend recalcula a ordem final com base na fileira.

GradeResequeneceResultDTO

Fonte: `src/main/java/com/wester/storage/dto/GradeResequeneceResultDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>gradeId</code>	integer	Não	Resultado do processo de remoção e resequenciamento de níveis de uma grade
<code>niveis</code>	array de NivelDTO	Não	Lista de níveis após o resequenciamento

GradeResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/GradeResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Resposta de grade criada
<code>identificador</code>	string	Não	
<code>ordem</code>	integer	Não	
<code>fileiraId</code>	integer	Não	

HistoricoMovimentacaoFilterRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/HistoricoMovimentacaoFilterRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>dataInicio</code>	string	Não	Filtro para consulta avançada do histórico de movimentação
<code>dataFim</code>	string	Não	Data/hora final ISO-8601 (aceita yyyy-MM-dd ou yyyy-MM-ddTHH:mm:ss)
<code>tipoOperacao</code>	string	Não	Tipo de operação
<code>usuarioId</code>	integer	Não	ID do usuário
<code>produtoId</code>	integer	Não	ID do produto
<code>moldeId</code>	integer	Não	ID do molde
<code>nivelId</code>	integer	Não	ID de um nível (nível principal, origem ou destino)
<code>textoLivre</code>	string	Não	Texto livre para busca em detalhes e campos de produto
<code>areaEstoqueId</code>	array de integer	Não	IDs das áreas de estoque para filtro (setores)

s

HistoricoMovimentacaoResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/HistoricoMovimentacaoResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>timestamp</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>tipoOperacao</code>	string	Não	
<code>usuarioId</code>	integer	Não	
<code>usuarioLogin</code>	string	Não	
<code>usuarioNome</code>	string	Não	
<code>produtoId</code>	integer	Não	
<code>produtoNomeModelo</code>	string	Não	
<code>produtoDescricao</code>	string	Não	
<code>produtoCor</code>	string	Não	
<code>produtoCodigoSistemaWester</code>	string	Não	
<code>itemEstoqueId</code>	integer	Não	
<code>areaId</code>	integer	Não	
<code>areaNome</code>	string	Não	
<code>setorId</code>	integer	Não	
<code>setorNome</code>	string	Não	
<code>tipoEstruturaArea</code>	string	Não	
<code>labelEstruturaPrimaria</code>	string	Não	
<code>labelEstruturaSecundaria</code>	string	Não	
<code>labelFileira</code>	string	Não	
<code>labelGrade</code>	string	Não	
<code>labelNivel</code>	string	Não	
<code>labelSlot</code>	string	Não	
<code>labelPosicao</code>	string	Não	
<code>fileiraId</code>	integer	Não	
<code>fileiraIdentificador</code>	string	Não	
<code>gradeId</code>	integer	Não	
<code>gradeIdentificador</code>	string	Não	
<code>nivelId</code>	integer	Não	
<code>nivelIdentificador</code>	string	Não	
<code>nivelOrigemId</code>	integer	Não	

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
nívelOrigemIdentificador	string	Não	
nívelDestinoId	integer	Não	
nívelDestinoIdentificador	string	Não	
slotId	integer	Não	
slotIdentificador	string	Não	
slotPosicao	integer	Não	
posicao	string	Não	
moldeId	integer	Não	
moldeCodigo	string	Não	
entidade	string	Não	
entidadeId	integer	Não	
campo	string	Não	
valorAnterior	string	Não	
valorNovo	string	Não	
payloadJson	string	Não	
quantidadeAlterada	integer	Não	
quantidadeAnterior	integer	Não	
quantidadeNova	integer	Não	
detalhesAlteracao	string	Não	

IdentifierUpdateRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/IdentifierUpdateRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
identificador	string	Sim	Payload para atualizar identificador estrutural Não pode ser vazio ou conter apenas espaços.
version	integer	Não	Versao esperada para entidades com concorrência otimista

ItemEstoqueDTO

Fonte: `src/main/java/com/wester/storage/dto/ItemEstoqueDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Stock item representation
<code>version</code>	integer	Não	Entity version for optimistic locking
<code>nívelId</code>	integer	Não	Nível ID
<code>nívelIdentificador</code>	string	Não	Nível identifier
<code>nívelCompletoIdentificador</code>	string	Não	Full identifier of nível (if applicable)
<code>produtoId</code>	integer	Não	Product ID
<code>produtoCodigoWester</code>	string	Não	Product code
<code>produtoNomeModelo</code>	string	Não	Product model name
<code>produtoCor</code>	string	Não	Product color
<code>produtoDescricao</code>	string	Não	Product description
<code>observacao</code>	string	Não	Item-specific note for this warehouse position
<code>quantidade</code>	integer	Não	Quantity
<code>quantidadeAproximada</code>	boolean	Não	Indicates whether the quantity is approximate
<code>dataAtualizacao</code>	string	Não	Last update date/time

ItemEstoqueLookupDTO

Fonte: `src/main/java/com/wester/storage/dto/ItemEstoqueLookupDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>itemEstoqueId</code>	integer	Não	
<code>produtoId</code>	integer	Não	
<code>codigo</code>	string	Não	
<code>nome</code>	string	Não	
<code>cor</code>	string	Não	
<code>localizacao</code>	string	Não	
<code>quantidade</code>	integer	Não	

ItemEstoqueMovimentacaoRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/ItemEstoqueMovimentacaoRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>nivelOrigemId</code>	integer	Sim	Payload para movimentar item entre níveis
<code>nivelOrigemVersion</code>	integer	Não	Versao atual do nivel de origem
<code>nivelDestinoId</code>	integer	Sim	ID do nível de destino
<code>nivelDestinoVersion</code>	integer	Não	Versao atual do nivel de destino
<code>itemOrigemVersion</code>	integer	Não	Versao atual do item de origem
<code>itemDestinoVersion</code>	integer	Não	Versao atual do item de destino
<code>quantidade</code>	integer	Sim	Quantidade a movimentar

ItemEstoqueResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/ItemEstoqueResponseDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Resposta do item de estoque por nivel
<code>nivelId</code>	integer	Não	ID do nivel
<code>quantidade</code>	integer	Não	Quantidade (0 quando não existir)
<code>quantidadeAproximada</code>	boolean	Não	Indica se a quantidade informada e aproximada
<code>produto</code>	ProdutoResumoDTO	Não	Produto (null quando não existir)

ItemEstoqueUpsertRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/ItemEstoqueUpsertRequestDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>itemEstoqueVersion</code>	integer	Não	Payload para criar/atualizar item de estoque em um nível
<code>nivelVersion</code>	integer	Não	Versao atual do nivel para controle de concorrencia otimista
<code>quantidade</code>	integer	Sim	Quantidade do item no nível
<code>quantidadeAproximada</code>	boolean	Não	Indica se a quantidade informada e aproximada
<code>observacao</code>	string	Não	Observação específica do item nesta posição do estoque
<code>produto</code>	ProdutoDTO	Sim	Dados do produto

LoginDTO

Fonte: `src/main/java/com/wester/storage/dto/LoginDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>login</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>senha</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.

MoldeAlocadoResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/MoldeAlocadoResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>slotId</code>	integer	Não	
<code>moldeId</code>	integer	Não	
<code>moldeCodigo</code>	string	Não	
<code>moldeDescricao</code>	string	Não	
<code>moldeCategoria</code>	string	Não	
<code>ordem</code>	integer	Não	

MoldeRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/MoldeRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>codigo</code>	string	Sim	DTO para criação ou atualização de molde Não pode ser vazio ou conter apenas espaços.
<code>descricao</code>	string	Não	Descrição do molde
<code>categoria</code>	string	Não	Categoria do molde
<code>observacoes</code>	string	Não	Observações adicionais

MovimentacaoEstoqueAreaOpcaoDTO

Fonte: `src/main/java/com/wester/storage/dto/MovimentacaoEstoqueAreaOpcaoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Resumo de área disponível para a tela de movimentações
<code>nome</code>	string	Não	Nome curto da area

MovimentacaoEstoqueContextoInicialDTO

Fonte: `src/main/java/com/wester/storage/dto/MovimentacaoEstoqueContextoInicialDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>produtos</code>	array de <code>MovimentacaoEstoqueProdutoOpcaoDTO</code>	Não	Dados iniciais agregados para carregar a tela de movimentações
<code>areas</code>	array de <code>MovimentacaoEstoqueAreaOpcaoDTO</code>	Não	Areas ativas disponiveis para selecao
<code>movimentacoesRecentes</code>	array de <code>MovimentacaoEstoqueRecenteDTO</code>	Não	Ultimas movimentacoes operacionais exibidas no resumo da tela

MovimentacaoEstoqueProdutoOpcaoDTO

Fonte: `src/main/java/com/wester/storage/dto/MovimentacaoEstoqueProdutoOpcaoDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Resumo de produto ativo disponível para movimentações
<code>codigoSistemaWester</code>	string	Não	Codigo Wester do produto
<code>nomeModelo</code>	string	Não	Nome/modelo do produto

MovimentacaoEstoqueRecenteDTO

Fonte: `src/main/java/com/wester/storage/dto/MovimentacaoEstoqueRecenteDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>timestamp</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>tipoMovimentacao</code>	string	Não	
<code>produtoNomeModelo</code>	string	Não	
<code>quantidadeMovimentada</code>	integer	Não	

MovimentacaoEstoqueRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/MovimentacaoEstoqueRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>tipoMovimentacao</code>	string	Não	Payload para registrar uma movimentação operacional de estoque
<code>produtoId</code>	integer	Não	Produto informado manualmente quando aplicavel
<code>nivelId</code>	integer	Não	Nível alvo para entrada, saída ou ajuste
<code>nivelVersion</code>	integer	Não	Versão atual do nível alvo
<code>nivelOrigemId</code>	integer	Não	Nível de origem da transferência
<code>nivelOrigemVersion</code> <code>n</code>	integer	Não	Versão atual do nível de origem
<code>nivelDestinoId</code>	integer	Não	Nível de destino da transferência
<code>nivelDestinoVersion</code> <code>n</code>	integer	Não	Versão atual do nível de destino
<code>itemEstoqueVersion</code> <code>n</code>	integer	Não	Versão atual do item do nível alvo
<code>itemOrigemVersion</code>	integer	Não	Versão atual do item da origem
<code>itemDestinoVersion</code> <code>n</code>	integer	Não	Versão atual do item do destino
<code>quantidade</code>	integer	Não	Quantidade movimentada para entrada, saída e transferência
<code>quantidadeAjustada</code> <code>a</code>	integer	Não	Quantidade final desejada no ajuste
<code>quantidadeDiferenca</code> <code>a</code>	integer	Não	Diferença aplicada no ajuste
<code>motivo</code>	string	Não	Motivo da operação, obrigatório no ajuste
<code>observacao</code>	string	Não	Observação complementar

MovimentacaoEstoqueResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/MovimentacaoEstoqueResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>historicoMovimentacaoId</code>	integer	Não	
<code>registradoEm</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>tipoMovimentacao</code>	string	Não	
<code>produtoId</code>	integer	Não	
<code>produtoNomeModelo</code>	string	Não	
<code>produtoCodigoSistemaWester</code>	string	Não	
<code>itemEstoqueId</code>	integer	Não	
<code>nivelId</code>	integer	Não	
<code>nivelOrigemId</code>	integer	Não	
<code>nivelDestinoId</code>	integer	Não	
<code>nivelLabel</code>	string	Não	
<code>nivelOrigemLabel</code>	string	Não	
<code>nivelDestinoLabel</code>	string	Não	
<code>quantidadeMovimentada</code>	integer	Não	
<code>quantidadeAnterior</code>	integer	Não	
<code>quantidadeNova</code>	integer	Não	
<code>motivo</code>	string	Não	
<code>observacao</code>	string	Não	
<code>detalhesAlteracao</code>	string	Não	

NivelCreateRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/NivelCreateRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>identificador</code>	string	Sim	DTO para criação de um nível dentro de uma grade Não pode ser vazio ou conter apenas espaços.
<code>ordem</code>	integer	Sim	Posição alvo do nível dentro da grade
<code>item</code>	ItemEstoqueUpsertRequestDTO	Não	Produto a vincular à nova gaiola. Quando informado (setor dinâmico), a gaiola e o item de estoque

NivelDTO

Fonte: `src/main/java/com/wester/storage/dto/NivelDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>version</code>	integer	Não	
<code>identificador</code>	string	Não	
<code>ordem</code>	integer	Não	

NivelMaterialDashboardDTO

Fonte: `src/main/java/com/wester/storage/dto/NivelMaterialDashboardDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>nivelMaterialId</code>	integer	Não	
<code>materialId</code>	integer	Não	
<code>materialType</code>	string	Não	
<code>ordem</code>	integer	Não	
<code>quantidade</code>	integer	Não	
<code>quantidadeAproximada</code>	boolean	Não	
<code>areaId</code>	integer	Não	
<code>fileiraId</code>	integer	Não	
<code>fileiraIdentificador</code>	string	Não	
<code>gradeId</code>	integer	Não	
<code>gradeIdentificador</code>	string	Não	
<code>nivelId</code>	integer	Não	
<code>nivelIdentificador</code>	string	Não	
<code>nivelOrdem</code>	integer	Não	

NivelMaterialResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/NivelMaterialResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>nívelId</code>	integer	Não	
<code>materialId</code>	integer	Não	
<code>materialType</code>	string	Não	
<code>materialCodigo</code>	string	Não	
<code>materialDescricao</code>	string	Não	
<code>materialCategoria</code>	string	Não	
<code>produtoDescricao</code>	string	Não	
<code>ordem</code>	integer	Não	
<code>quantidade</code>	integer	Não	
<code>quantidadeAproximada</code>	boolean	Não	

NívelMoveRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/NívelMoveRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>sourceGradeId</code>	integer	Sim	DTO para mover um nível entre grades e posicoes
<code>targetGradeId</code>	integer	Sim	Grade de destino
<code>targetPosition</code>	integer	Sim	Posição alvo dentro da grade de destino
<code>nívelVersion</code>	integer	Não	Versao esperada do nível para controle otimista

NívelMoveResultDTO

Fonte: `src/main/java/com/wester/storage/dto/NívelMoveResultDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>nívelId</code>	integer	Não	Resultado da movimentação estrutural de um nível
<code>sourceGradeId</code>	integer	Não	ID da grade de origem
<code>targetGradeId</code>	integer	Não	ID da grade de destino
<code>targetPosition</code>	integer	Não	Posição final ocupada pelo nível no destino
<code>impactedGrades</code>	array de GradeResequenceResultDTO	Não	Grades impactadas pela movimentação

NívelResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/NívelResponseDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	DTO de resposta após criação ou consulta de um nível
<code>version</code>	integer	Não	Versão da entidade para concorrência otimista
<code>identificador</code>	string	Não	Identificador do nível (ex: Ordem 01, Ordem 02)
<code>ordem</code>	integer	Não	Ordem do nível dentro da grade
<code>gradeId</code>	integer	Não	ID da grade à qual o nível pertence
<code>itemEstoqueId</code>	integer	Não	ID do item de estoque criado junto com o nível (somente quando a criação recebeu <code>item</code>). Permite ao cliente confirmar que o produto foi vinculado.

PerfilDisponívelDTO

Fonte: `src/main/java/com/wester/storage/dto/PerfilDisponívelDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	DTO com os perfis disponíveis para vinculação em usuários
<code>code</code>	string	Não	Código persistido do perfil
<code>description</code>	string	Não	Descrição legível do perfil
<code>type</code>	string	Não	Tipo de acesso do perfil
<code>active</code>	boolean	Não	Indica se o perfil esta ativo

PerfilResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/PerfilResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	DTO completo de perfil para gerenciamento
<code>code</code>	string	Não	Codigo estavel do perfil
<code>description</code>	string	Não	Descricao de negocio do perfil
<code>type</code>	string	Não	Tipo de acesso do perfil
<code>allowedScreens</code>	array de string	Não	Telas permitidas para o perfil
<code>active</code>	boolean	Não	Indica se o perfil esta ativo
<code>createdAt</code>	string	Não	Data/hora de criacao
<code>updatedAt</code>	string	Não	Data/hora da ultima atualizacao

PerfilUpsertRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/PerfilUpsertRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>description</code>	string	Sim	DTO de criacao/atualizacao de perfil Não pode ser vazio ou conter apenas espaços.
<code>type</code>	string	Sim	Tipo de acesso do perfil
<code>allowedScreens</code>	array de string	Sim	Lista de telas permitidas para o perfil
<code>active</code>	boolean	Não	Status ativo do perfil

ProdutoDTO

Fonte: `src/main/java/com/wester/storage/dto/ProdutoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>codigoSistemaWester</code>	string	Não	Dados do produto para criação/atualização no item de estoque
<code>nomeModelo</code>	string	Não	Nome/modelo do produto
<code>cor</code>	string	Não	Cor do produto
<code>descricao</code>	string	Não	Descrição do produto

ProdutoResumoDTO

Fonte: `src/main/java/com/wester/storage/dto/ProdutoResumoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Resumo do produto associado ao item de estoque
<code>codigoSistemaWester</code>	string	Não	Código Wester (pode ser null)
<code>nomeModelo</code>	string	Não	Nome/Modelo
<code>cor</code>	string	Não	Cor
<code>descricao</code>	string	Não	Descrição

ProdutoUpsertRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/ProdutoUpsertRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>codigoSistemaWester</code>	string	Não	Dados do produto para criação e atualização Aliases de entrada: "codigo"
<code>nomeModelo</code>	string	Não	Nome/modelo do produto. Obrigatório apenas se o código não for informado.
<code>cor</code>	string	Não	Cor do produto (opcional)
<code>descricao</code>	string	Não	Descrição do produto
<code>ativo</code>	boolean	Não	Status ativo do produto. Quando omitido na atualização, mantém o valor atual.
<code>estoqueMinimo</code>	integer	Não	Quantidade minima recomendada em estoque para o produto
<code>estoqueMaximo</code>	integer	Não	Quantidade maxima recomendada em estoque para o produto
<code>categoriaId</code>	integer	Não	
<code>removerCategoria</code>	boolean	Não	Quando true, remove a categoria do produto na atualização

RelatorioEstoqueItemDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioEstoqueItemDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>areaId</code>	integer	Não	
<code>areaNome</code>	string	Não	
<code>areaDescricao</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>totalFileiras</code>	integer	Não	
<code>totalGrades</code>	integer	Não	
<code>totalNiveis</code>	integer	Não	
<code>niveisOcupados</code>	integer	Não	
<code>totalItensEstoque</code>	integer	Não	
<code>quantidadeTotal</code>	integer	Não	
<code>quantidadeTotalAproximada</code>	boolean	Não	
<code>ocupacaoPercentual</code>	number	Não	Percentual de ocupacao calculado sobre niveis ocupados

RelatorioEstoqueFiltroDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioEstoqueFiltroDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>area</code>	string	Não	Filtros do relatorio de estoques
<code>fileira</code>	string	Não	Filtro por identificador da fileira
<code>grade</code>	string	Não	Filtro por identificador da grade
<code>nivel</code>	string	Não	Filtro por identificador do nivel
<code>ativo</code>	boolean	Não	Filtro por status ativo da area
<code>sortBy</code>	string	Não	Campo de ordenacao
<code>sortDirection</code>	string	Não	Direcao da ordenacao
<code>page</code>	integer	Não	Pagina desejada
<code>size</code>	integer	Não	Quantidade por pagina

RelatorioEstoqueResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioEstoqueResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>generatedAt</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>summary</code>	RelatorioEstoqueResumoDTO	Não	
<code>pagination</code>	RelatorioPaginacaoDTO	Não	
<code>items</code>	array de RelatorioEstoqueItemDTO	Não	
<code>quantityByAreaChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>occupancyByAreaChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>quantityByFileiraChart</code>	array de RelatorioGraficoDadoDTO	Não	

RelatorioEstoqueResumoDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioEstoqueResumoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>totalAreas</code>	integer	Não	
<code>totalAreasAtivas</code>	integer	Não	
<code>totalFileiras</code>	integer	Não	
<code>totalGrades</code>	integer	Não	
<code>totalNiveis</code>	integer	Não	
<code>totalItensEstoque</code>	integer	Não	
<code>quantidadeTotalArmazenada</code>	integer	Não	Soma total das quantidades armazenadas
<code>quantidadeTotalArmazenadaAproximada</code>	boolean	Não	Indica se a quantidade total armazenada inclui valores aproximados

RelatorioGraficoDadoDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioGraficoDadoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>label</code>	string	Não	Ponto de dados de um grafico de relatorio
<code>value</code>	number	Não	Valor numerico do ponto
<code>formattedValue</code>	string	Não	Valor formatado para exibicao
<code>aproximado</code>	boolean	Não	Indica se o valor exibido representa uma quantidade aproximada
<code>color</code>	string	Não	Cor opcional do ponto

RelatorioHistoricoMovimentacaoFiltroDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioHistoricoMovimentacaoFiltroDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>produto</code>	string	Não	Filtros do relatório de histórico de movimentação
<code>codigoSistemaWester</code>	string	Não	Filtro por codigo Wester
<code>area</code>	string	Não	Filtro por area ou setor
<code>usuario</code>	string	Não	Filtro por nome ou login do usuário
<code>tipoOperacao</code>	string	Não	Filtro por tipo de movimentação
<code>dataInicio</code>	string	Não	Data inicial recomendada (yyyy-MM-dd)
<code>dataFim</code>	string	Não	Data final recomendada (yyyy-MM-dd)
<code>sortDirection</code>	string	Não	Direcao da ordenacao pelo timestamp
<code>page</code>	integer	Não	Pagina desejada
<code>size</code>	integer	Não	Quantidade por pagina

RelatorioHistoricoMovimentacaoItemDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioHistoricoMovimentacaoItemDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>timestamp</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>tipoOperacao</code>	string	Não	
<code>usuarioNome</code>	string	Não	
<code>usuarioLogin</code>	string	Não	
<code>produtoNomeModelo</code>	string	Não	
<code>codigoSistemaWester</code>	string	Não	
<code>areaNome</code>	string	Não	
<code>fileira</code>	string	Não	
<code>grade</code>	string	Não	
<code>nivel</code>	string	Não	
<code>quantidadeAlterada</code>	integer	Não	
<code>quantidadeAnterior</code>	integer	Não	
<code>quantidadeNova</code>	integer	Não	
<code>detalhesAlteracao</code>	string	Não	

RelatorioHistoricoMovimentacaoResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioHistoricoMovimentacaoResponseDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>generatedAt</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>summary</code>	RelatorioHistoricoMovimentacaoResumoDTO	Não	
<code>pagination</code>	RelatorioPaginacaoDTO	Não	
<code>items</code>	array de RelatorioHistoricoMovimentacaoItemDTO	Não	
<code>typeChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>periodChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>productChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>areaChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>userChart</code>	array de RelatorioGraficoDadoDTO	Não	

RelatorioHistoricoMovimentacaoResumoDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioHistoricoMovimentacaoResumoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>totalMovimentacoes</code>	integer	Não	
<code>totalEntradas</code>	integer	Não	
<code>totalSaidas</code>	integer	Não	
<code>totalAjustes</code>	integer	Não	
<code>totalMovimentacoesInternas</code>	integer	Não	
<code>totalResequenciamentos</code>	integer	Não	
<code>quantidadeMovimentadaTotal</code>	integer	Não	Soma absoluta das quantidades movimentadas

RelatorioItemEstoqueItemDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioItemEstoqueItemDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>itemEstoqueId</code>	integer	Não	
<code>produtoId</code>	integer	Não	
<code>produtoNomeModelo</code>	string	Não	
<code>codigoSistemaWester</code>	string	Não	
<code>areaNome</code>	string	Não	
<code>tipoEstrutura</code>	string	Não	
<code>localizacaoFormatada</code>	string	Não	
<code>fileira</code>	string	Não	
<code>grade</code>	string	Não	
<code>nivel</code>	string	Não	
<code>quantidade</code>	integer	Não	
<code>quantidadeAproximada</code>	boolean	Não	
<code>statusItem</code>	string	Não	
<code>dataAtualizacao</code>	string	Não	Data/hora local sem offset; não assumir UTC.

RelatorioItensEstoqueFiltroDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioItensEstoqueFiltroDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>produto</code>	string	Não	Filtros do relatorio de itens de estoque
<code>codigoSistemaWester</code>	string	Não	Filtro por codigo Wester
<code>area</code>	string	Não	Filtro por nome da area
<code>fileira</code>	string	Não	Filtro por fileira
<code>grade</code>	string	Não	Filtro por grade
<code>nivel</code>	string	Não	Filtro por nivel
<code>statusItem</code>	string	Não	Filtro por status do item: DISPONIVEL, BAIXO ou SEM_ESTOQUE
<code>dataInicio</code>	string	Não	Data inicial de atualizacao (yyyy-MM-dd)
<code>dataFim</code>	string	Não	Data final de atualizacao (yyyy-MM-dd)
<code>sortBy</code>	string	Não	Campo de ordenacao
<code>sortDirection</code>	string	Não	Direcao da ordenacao
<code>page</code>	integer	Não	Pagina desejada
<code>size</code>	integer	Não	Quantidade por pagina

RelatorioItensEstoqueResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioItensEstoqueResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>generatedAt</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>summary</code>	RelatorioItensEstoqueResumoDTO	Não	
<code>pagination</code>	RelatorioPaginacaoDTO	Não	
<code>items</code>	array de RelatorioItemEstoqueItemDTO	Não	
<code>topProductsChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>quantityByAreaChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>quantityByLocationChart</code>	array de RelatorioGraficoDadoDTO	Não	

RelatorioItensEstoqueResumoDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioItensEstoqueResumoDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>totalItensEstoque</code>	integer	Não	
<code>quantidadeTotalArmazenada</code>	integer	Não	Soma de todas as quantidades armazenadas
<code>quantidadeTotalArmazenadaAproximada</code>	boolean	Não	Indica se a quantidade total armazenada inclui valores aproximados
<code>totalProdutosUnicos</code>	integer	Não	
<code>totalAreasComItens</code>	integer	Não	

RelatorioPaginacaoDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioPaginacaoDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>page</code>	integer	Não	Metadados de paginacao de um relatorio
<code>size</code>	integer	Não	Tamanho da pagina
<code>totalItems</code>	integer	Não	Total de registros filtrados
<code>totalPages</code>	integer	Não	Total de paginas

RelatorioProdutoItemDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioProdutoItemDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>codigoSistemaWester</code>	string	Não	
<code>nomeModelo</code>	string	Não	
<code>cor</code>	string	Não	
<code>descricao</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>categoriaId</code>	integer	Não	Id da categoria do produto (null quando sem categoria)
<code>categoriaDescricao</code>	string	Não	Descricao da categoria do produto
<code>quantidadeTotalEmEstoque</code>	integer	Não	Quantidade total deste produto atualmente em estoque
<code>quantidadeTotalEmEstoqueAproximada</code>	boolean	Não	Indica se a quantidade total consolidada deste produto e aproximada

RelatorioProdutosFiltroDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioProdutosFiltroDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>nome</code>	string	Não	Filtros do relatorio de produtos
<code>codigoSistemaWester</code>	string	Não	Filtro por codigo Wester
<code>descricao</code>	string	Não	Filtro por descricao
<code>cor</code>	string	Não	Filtro por cor
<code>ativo</code>	boolean	Não	Filtro por status ativo/inativo
<code>categoria</code>	string	Não	Filtro por categoria: id da categoria ou SEM_CATEGORIA
<code>sortBy</code>	string	Não	Campo de ordenacao
<code>sortDirection</code>	string	Não	Direcao da ordenacao
<code>page</code>	integer	Não	Pagina desejada
<code>size</code>	integer	Não	Quantidade por pagina

RelatorioProdutosResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioProdutosResponseDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>generatedAt</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>summary</code>	RelatorioProdutosResumoDTO	Não	
<code>pagination</code>	RelatorioPaginacaoDTO	Não	
<code>items</code>	array de RelatorioProdutoItemDTO	Não	
<code>statusChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>colorChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>usageChart</code>	array de RelatorioGraficoDadoDTO	Não	
<code>categoryTotal</code>	array de RelatorioGraficoDadoDTO	Não	Quantidade total em estoque somada por categoria (todos os produtos filtrados)

RelatorioProdutosResumoDTO

Fonte: `src/main/java/com/wester/storage/dto/RelatorioProdutosResumoDTO.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>totalProdutos</code>	integer	Não	
<code>totalProdutosAtivos</code>	integer	Não	
<code>totalProdutosInativos</code>	integer	Não	
<code>quantidadeTotalEmEstoque</code>	integer	Não	Soma das quantidades atualmente armazenadas para os produtos filtrados
<code>quantidadeTotalEmEstoqueAproximada</code>	boolean	Não	Indica se a quantidade total em estoque inclui valores aproximados

SlotNivelRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/SlotNivelRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>posicao</code>	integer	Não	Payload para criar ou editar slot de nível fixo
<code>identificador</code>	string	Não	Identificador visual do slot

SlotNivelResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/SlotNivelResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>nivelId</code>	integer	Não	
<code>posicao</code>	integer	Não	
<code>identificador</code>	string	Não	
<code>moldesAlocados</code>	array de MoldeAlocadoResponseDTO	Não	

TabelaEncertoDTO

Fonte: `src/main/java/com/wester/storage/dto/TabelaEncertoDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>areaId</code>	integer	Não	
<code>titulo</code>	string	Não	
<code>colunas</code>	array de string	Não	
<code>itens</code>	array de TabelaEncertoItemDTO	Não	

TabelaEncertoItemDTO

Fonte: `src/main/java/com/wester/storage/dto/TabelaEncertoItemDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>produtoId</code>	integer	Não	
<code>codigo</code>	string	Não	
<code>nome</code>	string	Não	
<code>cor</code>	string	Não	
<code>descricao</code>	string	Não	
<code>observacao</code>	string	Não	
<code>quantidade</code>	integer	Não	
<code>ordem</code>	integer	Não	

TabelaEncertoSummaryDTO

Fonte: `src/main/java/com/wester/storage/dto/TabelaEncertoSummaryDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>titulo</code>	string	Não	
<code>colunas</code>	array de string	Não	
<code>totalItens</code>	integer	Não	

UsuarioChangePasswordRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/UsuarioChangePasswordRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>senhaAtual</code>	string	Sim	DTO de request para alteração de senha do usuário Não pode ser vazio ou conter apenas espaços.
<code>novaSenha</code>	string	Sim	Nova senha do usuário Não pode ser vazio ou conter apenas espaços.
<code>confirmarNovaSenha</code>	string	Sim	Confirmação da nova senha Não pode ser vazio ou conter apenas espaços.

UsuarioRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/UsuarioRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>login</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>nome</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>senha</code>	string	Não	
<code>perfil</code>	string	Não	

UsuarioResponseDTO

Fonte: `src/main/java/com/wester/storage/dto/UsuarioResponseDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>login</code>	string	Não	
<code>nome</code>	string	Não	
<code>perfil</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>token</code>	string	Não	
<code>perfilResumo</code>	PerfilResponseDTO	Não	

UsuarioStatusUpdateRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/UsuarioStatusUpdateRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>ativo</code>	boolean	Sim	DTO de request para atualizacao de status (ativo/inativo) do usuario

UsuarioUpdateRequestDTO

Fonte: `src/main/java/com/wester/storage/dto/UsuarioUpdateRequestDTO.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>login</code>	string	Sim	DTO de request para atualização de dados do usuário (sem alterar senha) Não pode ser vazio ou conter apenas espaços.
<code>nome</code>	string	Sim	Nome do usuário Não pode ser vazio ou conter apenas espaços.
<code>perfil</code>	string	Sim	Codigo do perfil do usuário Não pode ser vazio ou conter apenas espaços.

AreaEstoque

Fonte: `src/main/java/com/wester/storage/model/AreaEstoque.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Area fisica/logica de estoque
<code>descricao</code>	string	Sim	Descricao da area Não pode ser vazio ou conter apenas espaços.
<code>nome</code>	string	Não	Nome curto da area
<code>ativo</code>	boolean	Não	Status ativo da area
<code>tipoEstrutura</code>	string	Não	Tipo de estrutura do setor: DINAMICO ou FIXO
<code>labelEstruturaPrimaria</code>	string	Não	Label da estrutura primaria do setor Aliases de entrada: {"labelFileira", "label_fileira", "nomenclaturaFileira"}
<code>labelEstruturaSecundaria</code>	string	Não	Label da estrutura secundaria do setor Aliases de entrada: {"labelGrade", "label_grade", "nomenclaturaGrade"}
<code>labelNivel</code>	string	Não	Label dos niveis do setor Aliases de entrada: {"nomenclaturaNivel"}
<code>labelSlot</code>	string	Não	Label dos slots do setor Aliases de entrada: {"labelPosicao", "label_posicao", "nomenclaturaPosicao"}
<code>labelFileira</code>	string	Não	Alias de saída calculado pelo getter.
<code>labelGrade</code>	string	Não	Alias de saída calculado pelo getter.
<code>labelPosicao</code>	string	Não	Alias de saída calculado pelo getter.

AreaMaterialTag

Fonte: `src/main/java/com/wester/storage/model/AreaMaterialTag.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>areaEstoque</code>	AreaEstoque	Não	
<code>materialType</code>	string	Não	

CategoriaProduto

Fonte: `src/main/java/com/wester/storage/model/CategoriaProduto.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>codigo</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>descricao</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.

Fileira

Fonte: `src/main/java/com/wester/storage/model/Fileira.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Representa uma fileira do armazém vinculada a uma área de estoque
<code>identificador</code>	string	Sim	Identificador da fileira Não pode ser vazio ou conter apenas espaços.
<code>ordem</code>	integer	Sim	Ordem da fileira
<code>areaEstoque</code>	AreaEstoque	Não	Área de estoque vinculada
<code>ativo</code>	boolean	Não	Inativação lógica — não há reativação

Grade

Fonte: `src/main/java/com/wester/storage/model/Grade.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>fileira</code>	Fileira	Sim	
<code>identificador</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>ordem</code>	integer	Não	
<code>ativo</code>	boolean	Não	

HistoricoMovimentacao

Fonte: `src/main/java/com/wester/storage/model/HistoricoMovimentacao.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>timestamp</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>usuario</code>	Usuario	Sim	
<code>tipoOperacao</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>itemEstoque</code>	ItemEstoque	Não	
<code>produto</code>	Produto	Não	
<code>nivel</code>	Nivel	Não	
<code>nivelOrigem</code>	Nivel	Não	
<code>nivelDestino</code>	Nivel	Não	
<code>nivelIdentificador</code>	string	Não	
<code>nivelOrigemIdentificador</code>	string	Não	
<code>nivelDestinoIdentificador</code>	string	Não	
<code>quantidadeAlterada</code>	integer	Não	
<code>quantidadeAnterior</code>	integer	Não	
<code>quantidadeNova</code>	integer	Não	
<code>detalhesAlteracao</code>	string	Não	
<code>areaEstoqueId</code>	integer	Não	
<code>areaNome</code>	string	Não	
<code>tipoEstruturaArea</code>	string	Não	
<code>labelEstruturaPrimaria</code>	string	Não	
<code>labelEstruturaSecundaria</code>	string	Não	
<code>labelNivel</code>	string	Não	
<code>labelSlot</code>	string	Não	
<code>fileiraId</code>	integer	Não	
<code>fileiraIdentificador</code>	string	Não	
<code>gradeId</code>	integer	Não	
<code>gradeIdentificador</code>	string	Não	
<code>slotId</code>	integer	Não	
<code>slotIdentificador</code>	string	Não	
<code>slotPosicao</code>	integer	Não	
<code>moldeId</code>	integer	Não	

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>moldeCodigo</code>	string	Não	
<code>entidade</code>	string	Não	
<code>entidadeId</code>	integer	Não	
<code>campo</code>	string	Não	
<code>valorAnterior</code>	string	Não	
<code>valorNovo</code>	string	Não	
<code>payloadJson</code>	string	Não	

ItemEstoque

Fonte: `src/main/java/com/wester/storage/model/ItemEstoque.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>version</code>	integer	Não	Versão para concorrência otimista; obtida na consulta.
<code>nivel</code>	Nível	Sim	
<code>produto</code>	Produto	Sim	
<code>quantidade</code>	integer	Sim	
<code>quantidadeAproximada</code>	boolean	Não	
<code>observacao</code>	string	Não	
<code>dataAtualizacao</code>	string	Não	Data/hora local sem offset; não assumir UTC.

Molde

Fonte: `src/main/java/com/wester/storage/model/Molde.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	Molde físico da ferramentaria
<code>codigo</code>	string	Sim	Código único do molde Não pode ser vazio ou conter apenas espaços.
<code>descricao</code>	string	Não	Descrição do molde
<code>categoria</code>	string	Não	Categoria do molde (prefixo do código)
<code>observacoes</code>	string	Não	Observações adicionais sobre o molde
<code>ativo</code>	boolean	Não	Indica se o molde está ativo no catálogo

MoldeAlocado

Fonte: `src/main/java/com/wester/storage/model/MoldeAlocado.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>slot</code>	SlotNivel	Sim	
<code>molde</code>	Molde	Sim	
<code>ordem</code>	integer	Não	Ordem do molde dentro do slot (para múltiplos moldes no mesmo slot)

Nivel

Fonte: `src/main/java/com/wester/storage/model/Nivel.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>version</code>	integer	Não	Versão para concorrência otimista; obtida na consulta.
<code>grade</code>	Grade	Sim	
<code>identificador</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>ordem</code>	integer	Não	
<code>identificadorCompletoLegivel</code>	string	Não	
<code>ativo</code>	boolean	Não	

NivelMaterial

Fonte: `src/main/java/com/wester/storage/model/NivelMaterial.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>nivel</code>	Nivel	Não	
<code>materialId</code>	integer	Não	
<code>materialType</code>	string	Não	
<code>ordem</code>	integer	Não	
<code>quantidade</code>	integer	Não	
<code>quantidadeAproximada</code>	boolean	Não	

Perfil

Fonte: `src/main/java/com/wester/storage/model/Perfil.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>codigo</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>descricao</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>tipoAcesso</code>	string	Não	
<code>telasPermitidas</code>	array de string	Não	
<code>ativo</code>	boolean	Não	
<code>createdAt</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>updatedAt</code>	string	Não	Data/hora local sem offset; não assumir UTC.

Produto

Fonte: `src/main/java/com/wester/storage/model/Produto.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>codigoSistemaWester</code>	string	Não	Aliases de entrada: "codigo"
<code>nomeModelo</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>cor</code>	string	Não	
<code>descricao</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>estoqueMinimo</code>	integer	Não	Quantidade minima recomendada em estoque
<code>estoqueMaximo</code>	integer	Não	Quantidade maxima recomendada em estoque
<code>categoria</code>	CategoriaProduto	Não	

SlotNivel

Fonte: `src/main/java/com/wester/storage/model/SlotNivel.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>nivel</code>	Nivel	Sim	
<code>posicao</code>	integer	Sim	Posição do slot dentro do nível (1=esquerda, 2=direita, ...)
<code>identificador</code>	string	Sim	Identificador editavel do slot dentro do nivel Não pode ser vazio ou conter apenas espaços.
<code>moldesAlocados</code>	array de MoldeAlocado	Não	

TabelaEncerto

Fonte: `src/main/java/com/wester/storage/model/TabelaEncerto.java`.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>area</code>	AreaEstoque	Sim	
<code>titulo</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>colunas</code>	array de string	Não	
<code>ativo</code>	boolean	Não	
<code>criadoEm</code>	string	Não	Data/hora local sem offset; não assumir UTC.
<code>itens</code>	array de TabelaEncertoItem	Não	

TabelaEncertoItem

Fonte: `src/main/java/com/wester/storage/model/TabelaEncertoItem.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>tabela</code>	TabelaEncerto	Sim	
<code>produto</code>	Produto	Sim	
<code>quantidade</code>	integer	Não	
<code>ordem</code>	integer	Não	
<code>observacao</code>	string	Não	

Usuario

Fonte: `src/main/java/com/wester/storage/model/Usuario.java` .

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>id</code>	integer	Não	
<code>nome</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>login</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>perfil</code>	string	Sim	Não pode ser vazio ou conter apenas espaços.
<code>ativo</code>	boolean	Não	

MesToolSettings

Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>lot</code>	string	Não	
<code>moldName</code>	string	Não	
<code>nozzleMode</code>	string	Não	
<code>pieceWeightGrams</code>	number	Não	
<code>netWeightGrams</code>	number	Não	
<code>grossWeightGrams</code>	number	Não	
<code>sprueWeightGrams</code>	number	Não	
<code>cyclesPerKit</code>	integer	Não	

MesMachineInput

Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Sim	
<code>name</code>	string	Sim	
<code>numero</code>	string	Sim	
<code>marca</code>	string	Sim	
<code>modelo</code>	string	Sim	
<code>edgeId</code>	string	Não	
<code>nozzleMode</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>testYellow</code>	boolean	Não	
<code>testRed</code>	boolean	Não	

MesOrderInput

Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Não	
<code>orderCode</code>	string	Sim	
<code>productCode</code>	string	Sim	
<code>productName</code>	string	Sim	
<code>toolCode</code>	string	Não	
<code>material</code>	string	Não	
<code>notes</code>	string	Não	
<code>quantityPieces</code>	integer	Sim	
<code>piecesPerCycle</code>	integer	Sim	
<code>idealCycleMs</code>	integer	Sim	
<code>toolSettings</code>	MesToolSettings	Não	

MesPositionInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>queuePosition</code>	integer	Sim	
<code>machineId</code>	string	Não	

MesAssignInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Sim	

MesReasonInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>reason</code>	string	Sim	

MesQualityInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>reject</code>	integer	Sim	
<code>secondQuality</code>	integer	Não	
<code>residueGrams</code>	number	Sim	
<code>notes</code>	string	Sim	

MesDowntimeInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>kind</code>	string	Sim	
<code>reason</code>	string	Sim	
<code>notes</code>	string	Não	

MesReleaseInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>notes</code>	string	Não	

MesRejection

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>code</code>	string	Sim	
<code>status</code>	integer	Sim	
<code>at</code>	string	Sim	

MesHeartbeatInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Sim	
<code>adamReadAt</code>	string	Não	
<code>outboxDepth</code>	integer	Sim	
<code>quarantineDepth</code>	integer	Sim	
<code>lastAcknowledgedSequence</code>	integer	Não	
<code>collectorFailureCount</code>	integer	Não	
<code>collectorFailureAt</code>	string	Não	
<code>collectorLastError</code>	string	Não	
<code>lastRejection</code>	MesRejection	Não	

MesEdgeEventInput

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Sim	
<code>edgeId</code>	string	Não	
<code>eventId</code>	string	Sim	
<code>sequence</code>	integer	Sim	
<code>timestamp</code>	string	Sim	
<code>type</code>	string	Sim	
<code>classification</code>	string	Não	
<code>cycleState</code>	string	Não	
<code>previousCycleState</code>	string	Não	
<code>interruptionReason</code>	string	Não	
<code>currentCycleStartedAt</code>	string	Não	
<code>lamp</code>	string	Sim	
<code>outputStatus</code>	string	Sim	
<code>reason</code>	string	Não	
<code>signals</code>	object	Sim	
<code>schemaVersion</code>	integer	Não	
<code>namedSignals</code>	object	Não	
<code>executionId</code>	string	Não	
<code>cycle</code>	object	Não	

MesOrder

Campos comprovados no mapeamento JDBC e migrações; propriedades adicionais permitidas para projeções e contadores. Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>orderId</code>	string	Não	
<code>machineId</code>	string	Não	
<code>orderCode</code>	string	Não	
<code>productCode</code>	string	Não	
<code>productName</code>	string	Não	
<code>toolCode</code>	string	Não	
<code>material</code>	string	Não	
<code>quantityPieces</code>	integer	Não	
<code>piecesPerCycle</code>	integer	Não	
<code>idealCycleMs</code>	integer	Não	
<code>notes</code>	string	Não	
<code>toolSettings</code>	MesToolSettings	Não	
<code>status</code>	string	Não	
<code>queuePosition</code>	integer	Não	
<code>createdAt</code>	string	Não	
<code>author</code>	string	Não	
<code>insumos</code>	string	Não	
<code>executionId</code>	string	Não	
<code>goodPieces</code>	integer	Não	
<code>targetReached</code>	boolean	Não	

MesExecution

Campos comprovados no mapeamento JDBC e migrações; propriedades adicionais permitidas para projeções e contadores. Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>executionId</code>	string	Não	
<code>orderId</code>	string	Não	
<code>machineId</code>	string	Não	
<code>contextRevision</code>	integer	Não	
<code>status</code>	string	Não	
<code>preparedAt</code>	string	Não	
<code>startedAt</code>	string	Não	
<code>finishedAt</code>	string	Não	
<code>author</code>	string	Não	
<code>cancelReason</code>	string	Não	
<code>orderCode</code>	string	Não	
<code>productCode</code>	string	Não	
<code>productName</code>	string	Não	
<code>toolCode</code>	string	Não	
<code>material</code>	string	Não	
<code>quantityPieces</code>	integer	Não	
<code>piecesPerCycle</code>	integer	Não	
<code>idealCycleMs</code>	integer	Não	
<code>notes</code>	string	Não	
<code>toolSettings</code>	MesToolSettings	Não	
<code>insumos</code>	string	Não	
<code>goodPieces</code>	integer	Não	
<code>targetReached</code>	boolean	Não	

MesQuality

Campos comprovados no mapeamento JDBC e migrações; propriedades adicionais permitidas para projeções e contadores. Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>qualityId</code>	string	Não	
<code>executionId</code>	string	Não	
<code>secondQuality</code>	integer	Não	
<code>reject</code>	integer	Não	
<code>residueGrams</code>	number	Não	
<code>notes</code>	string	Não	
<code>createdAt</code>	string	Não	
<code>author</code>	string	Não	

MesDowntime

Campos comprovados no mapeamento JDBC e migrações; propriedades adicionais permitidas para projeções e contadores. Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>downtimeId</code>	string	Não	
<code>machineId</code>	string	Não	
<code>executionId</code>	string	Não	
<code>kind</code>	string	Não	
<code>reason</code>	string	Não	
<code>notes</code>	string	Não	
<code>startedAt</code>	string	Não	
<code>releasedAt</code>	string	Não	
<code>recoveredAt</code>	string	Não	
<code>author</code>	string	Não	
<code>releasedBy</code>	string	Não	
<code>releaseNotes</code>	string	Não	
<code>category</code>	string	Não	
<code>source</code>	string	Não	
<code>mergedInto</code>	string	Não	

MesCycleOutcome

Campos comprovados no mapeamento JDBC e migrações; propriedades adicionais permitidas para projeções e contadores. Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>eventId</code>	string	Não	
<code>executionId</code>	string	Não	
<code>machineId</code>	string	Não	
<code>timestamp</code>	string	Não	
<code>classification</code>	string	Não	
<code>cycleState</code>	string	Não	
<code>previousCycleState</code>	string	Não	
<code>reason</code>	string	Não	
<code>startedAt</code>	string	Não	
<code>endedAt</code>	string	Não	
<code>pieces</code>	integer	Não	
<code>durationMs</code>	integer	Não	
<code>body</code>	string	Não	

MesMachine

Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Não	
<code>name</code>	string	Não	
<code>numero</code>	string	Não	
<code>marca</code>	string	Não	
<code>modelo</code>	string	Não	
<code>nozzleMode</code>	string	Não	
<code>ativo</code>	boolean	Não	
<code>testYellow</code>	boolean	Não	
<code>testRed</code>	boolean	Não	
<code>revision</code>	integer	Não	
<code>activeExecutionId</code>	string	Não	
<code>mechanicalFault</code>	boolean	Não	

MesMetrics

Fonte: `MesService` , `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>grossPieces</code>	number	Não	
<code>cycles</code>	number	Não	
<code>validCycles</code>	number	Não	
<code>averageCycleMs</code>	number	Não	
<code>averageInjectionMs</code>	number	Não	
<code>averageMoldClosedMs</code>	number	Não	
<code>averageMoldNotClosedMs</code>	number	Não	
<code>averageNozzleContactMs</code>	number	Não	
<code>averageNozzleAwayMs</code>	number	Não	
<code>averageDosingMs</code>	number	Não	
<code>secondQuality</code>	number	Não	
<code>reject</code>	number	Não	
<code>residueGrams</code>	number	Não	
<code>goodPieces</code>	number	Não	
<code>qualityBasis</code>	string	Não	
<code>projection</code>	string	Não	

MesReport

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>grossPieces</code>	number	Não	
<code>cycles</code>	number	Não	
<code>validCycles</code>	number	Não	
<code>averageCycleMs</code>	number	Não	
<code>averageInjectionMs</code>	number	Não	
<code>averageMoldClosedMs</code>	number	Não	
<code>averageMoldNotClosedMs</code>	number	Não	
<code>averageNozzleContactMs</code>	number	Não	
<code>averageNozzleAwayMs</code>	number	Não	
<code>averageDosingMs</code>	number	Não	
<code>secondQuality</code>	number	Não	
<code>reject</code>	number	Não	
<code>residueGrams</code>	number	Não	
<code>goodPieces</code>	number	Não	
<code>qualityBasis</code>	string	Não	
<code>projection</code>	string	Não	
<code>executionTotals</code>	MesMetrics	Não	

MesStatus

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machine</code>	MesMachine	Não	
<code>execution</code>	MesExecution	Não	
<code>telemetry</code>	object	Não	
<code>telemetryAt</code>	string	Não	
<code>hasCompletedCycle</code>	boolean	Não	
<code>cycleState</code>	string	Não	
<code>cycleClassification</code>	string	Não	
<code>validationRequired</code>	boolean	Não	
<code>validationReason</code>	string	Não	
<code>lastSequence</code>	integer	Não	
<code>currentCycleElapsedMs</code>	number	Não	
<code>lastCycleMs</code>	number	Não	
<code>desiredLamp</code>	string	Não	
<code>productionBlocked</code>	boolean	Não	
<code>communication</code>	string	Não	
<code>metrics</code>	MesMetrics	Não	
<code>edgeHealth</code>	object	Não	
<code>collectorHealth</code>	object	Não	
<code>ingestionHealth</code>	object	Não	
<code>nozzleModeInferred</code>	object	Não	

MesConfig

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>machineId</code>	string	Não	
<code>revision</code>	integer	Não	
<code>executionId</code>	string	Não	
<code>idealCycleMs</code>	integer	Não	
<code>piecesPerCycle</code>	integer	Não	
<code>mechanicalFault</code>	boolean	Não	
<code>recoveryPending</code>	boolean	Não	
<code>hasCompletedCycle</code>	boolean	Não	
<code>productionBlocked</code>	boolean	Não	
<code>manualLamp</code>	string	Não	
<code>desiredLamp</code>	string	Não	

MesPermissions

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>read</code>	boolean	Não	
<code>operate</code>	boolean	Não	
<code>maintain</code>	boolean	Não	
<code>plan</code>	boolean	Não	
<code>admin</code>	boolean	Não	

MesActiveDowntime

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>active</code>	MesDowntime	Não	
<code>linked</code>	array de MesDowntime	Não	

MesEventAck

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>eventId</code>	string	Não	
<code>duplicate</code>	boolean	Não	

MesHeartbeatAck

Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
<code>accepted</code>	boolean	Não	
<code>receivedAt</code>	string	Não	

MesImportPreview

Prévia variável; confira campos, source (NATIVE/OCR) e informações não resolvidas. Não cria OS. Campos do parser: bags, boxes, field, idealCycleSeconds, lot, machineHint, multiple, orderCode, orderDate, piecesPerCycle, productCode, productName, quantityPieces, reason, toolCode, toolDescription. Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
-------	----------	--------------------	-------------

ProdutoListaVariavel

Sem filtros/paginação: array; com filtros/paginação: página Spring Data. Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
-------	----------	--------------------	-------------

MoldeListaVariavel

Formato depende de `shouldUsePaginatedListing`; ver regras do serviço. Fonte: `MesService`, `MesRules` e migrações MES, ou composição do tipo de retorno.

CAMPO	CONTRATO	OBRIGATÓRIO NO DTO	OBSERVAÇÕES
-------	----------	--------------------	-------------

Referência OpenAPI (Swagger)

Especificação **OpenAPI 3.0.3** de todas as rotas documentadas (211 combinações método + rota, 145 operações), com esquemas de dados, autenticação (`bearerAuth` para usuários e `edgeKey` para o coletor industrial) e exemplos.

Como usar

- **Referência interativa:** lista as rotas por módulo, com parâmetros, corpos e respostas. O botão "Try it out" fica **desativado**: esta página não envia chamadas a nenhum servidor.
- **Importar em ferramentas:** baixe o `openapi.yaml` e importe no **Postman** (*Import* → *arquivo*), no **Insomnia** ou no **Swagger Editor**. Defina o servidor (`BASE_URL`) do seu ambiente e o token de login (`Authorization: Bearer ...`).
- **Exemplos são sintéticos:** valores como `"Exemplo"` , `1` ou `false` apenas ilustram o formato; troque por dados reais do seu ambiente.

Para entender autenticação, permissões e erros antes de testar, leia **Autenticação e autorização** e **Tratamento de erros**.

Acesso, usuários e perfis

[Índice de endpoints](#) · [Modelos](#)

API-013 — Login

POST [/api/auth/login](#)

POST [/api/usuarios/login](#)

Finalidade: Login.

Fonte: `src/main/java/com/wester/storage/controller/AuthController.java:40` — `AuthController.login`.

Autenticação, autorização e perfis: Acesso público. Login não exige token. Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: nenhum de autenticação obrigatório. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `LoginDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "login": "operador@example.com",
  "senha": "SenhaExemplo123"
}
```

json

Resposta de sucesso: `UsuarioResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

Códigos HTTP: 200, 400, 401, 403, 429. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Usuário inativo. Procure um administrador.
- Informe login e senha para continuar.

Regras de negócio e dependências: `AuthenticationService.authenticate`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/auth/login" -H 'Content-Type: application/json' --data '{"login":"operador@examp
```

API-014 — Me

GET `/api/auth/me`

Finalidade: Me.

Fonte: `src/main/java/com/wester/storage/controller/AuthController.java:50` — `AuthController.me`.

Autenticação, autorização e perfis: Usuário autenticado; sem autoridade adicional declarada no método. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `UsuarioResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Informe login e senha para continuar.

Regras de negócio e dependências: `AuthenticationService.getAuthenticatedUser`, `CurrentUserService.getCurrentLogin`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/auth/me" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-069 — Listar perfis

GET `/api/perfis`

GET `/api/profiles`

Finalidade: Listar perfis.

Fonte: `src/main/java/com/wester/storage/controller/PerfilController.java:57` — `PerfilController.listarPerfis`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PROFILES')` or `hasAuthority('SCREEN_USERS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	10
<code>search</code>	query	<code>String</code>	Não	—
<code>sort</code>	query	<code>String</code>	Não	—
<code>direction</code>	query	<code>String</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Page<PerfilResponseDTO>` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "content": [],
  "totalElements": 1,
  "totalPages": 0,
  "size": 1,
  "number": 1,
  "first": false,
  "last": false,
  "empty": false,
  "numberOfElements": 1,
  "pageable": {},
  "sort": {}
}
```

Códigos HTTP: `200` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `PerfilService.listarPerfis` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/perfis" -H "Authorization: Bearer ${TOKEN}"
```

API-070 — Listar perfis ativos para vínculo em usuários

GET `/api/perfis/ativos`

GET `/api/profiles/ativos`

Finalidade: Listar perfis ativos para vínculo em usuários.

Fonte: `src/main/java/com/wester/storage/controller/PerfilController.java:73` — `PerfilController.listarPerfisAtivos`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PROFILES')` or `hasAuthority('SCREEN_USERS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<PerfilDisponivelDTO>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `PerfilService.listarPerfisAtivos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/perfis/ativos" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-071 — Criar perfil

POST `/api/perfis`

POST `/api/profiles`

Finalidade: Criar perfil.

Fonte: `src/main/java/com/wester/storage/controller/PerfilController.java:84` — `PerfilController.criarPerfil`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PROFILES')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `PerfilUpsertRequestDTO` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "description": "Consulta de estoque",
  "type": "READ_ONLY",
  "allowedScreens": [
    "DASHBOARD",
    "WAREHOUSE",
    "PRODUCTS"
  ],
  "active": true
}
```

json

Resposta de sucesso: `PerfilResponseDTO` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "code": "CONSULTOR",
  "description": "Exemplo",
  "type": "READ_ONLY",
  "allowedScreens": [],
  "active": false,
  "createdAt": "2026-10-15T09:00:00",
  "updatedAt": "2026-10-15T09:00:00"
}
```

json

Códigos HTTP: `201` , `400` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Já existe um perfil com o código: [mensagem pode receber complemento dinâmico]
- Descrição do perfil é obrigatória.
- Não foi possível gerar o código do perfil.
- Selecione pelo menos uma tela permitida.
- Telas inválidas informadas: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `PerfilService.criarPerfil`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/perfis" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-072 — Atualizar perfil

PUT `/api/perfis/{id}`

PUT `/api/profiles/{id}`

Finalidade: Atualizar perfil.

Fonte: `src/main/java/com/wester/storage/controller/PerfilController.java:100` — `PerfilController.atualizarPerfil`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PROFILES')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `PerfilUpsertRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "description": "Consulta de estoque",
  "type": "READ_ONLY",
  "allowedScreens": [
    "DASHBOARD",
    "WAREHOUSE",
    "PRODUCTS"
  ],
  "active": true
}
```

Resposta de sucesso: `PerfilResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "code": "CONSULTOR",
  "description": "Exemplo",
  "type": "READ_ONLY",
  "allowedScreens": [],
  "active": false,
  "createdAt": "2026-10-15T09:00:00",
  "updatedAt": "2026-10-15T09:00:00"
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Descrição do perfil é obrigatória.
- Selecione pelo menos uma tela permitida.
- Telas inválidas informadas: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `PerfilService.atualizarPerfil`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/perfis/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-073 — Ativar perfil

PATCH `/api/perfis/{id}/ativar`

PATCH `/api/perfis/{id}/activate`

PATCH `/api/profiles/{id}/ativar`

PATCH `/api/profiles/{id}/activate`

Finalidade: Ativar perfil.

Fonte: `src/main/java/com/wester/storage/controller/PerfilController.java:123` — `PerfilController.ativarPerfil`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PROFILES')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `PerfilResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "code": "CONSULTOR",
  "description": "Exemplo",
  "type": "READ_ONLY",
  "allowedScreens": [],
  "active": false,
  "createdAt": "2026-10-15T09:00:00",
  "updatedAt": "2026-10-15T09:00:00"
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: Implementação no próprio controller. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/perfis/1/ativar" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-074 — Inativar perfil

PATCH `/api/perfis/{id}/inativar`

PATCH `/api/perfis/{id}/inactivate`

PATCH `/api/profiles/{id}/inativar`

PATCH `/api/profiles/{id}/inactivate`

Finalidade: Inativar perfil.

Fonte: `src/main/java/com/wester/storage/controller/PerfilController.java:134` — `PerfilController.inativarPerfil`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PROFILES')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `PerfilResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "code": "CONSULTOR",
  "description": "Exemplo",
  "type": "READ_ONLY",
  "allowedScreens": [],
  "active": false,
  "createdAt": "2026-10-15T09:00:00",
  "updatedAt": "2026-10-15T09:00:00"
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: Implementação no próprio controller. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/perfis/1/inativar" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-109 — Listar Todos Usuarios

GET `/api/usuarios`

Finalidade: Listar Todos Usuarios.

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:53` — `UsuarioController.listarTodosUsuarios`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_USERS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<UsuarioResponseDTO>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `UsuarioService.listarTodosUsuarios`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/usuarios" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-110 — Buscar Usuario Por Id

GET `/api/usuarios/id/{id}`

Finalidade: Buscar Usuario Por Id.

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:65` — `UsuarioController.buscarUsuarioPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_USERS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `UsuarioResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `UsuarioService.buscarUsuarioPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/usuarios/id/1" -H "Authorization: Bearer ${TOKEN}"
```

API-111 — Buscar Por Login

GET `/api/usuarios/by-login/{login}`

Finalidade: Buscar Por Login.

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:76` — `UsuarioController.buscarPorLogin`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_USERS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>login</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `UsuarioResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `UsuarioService.buscarPorLogin`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/usuarios/by-login/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-112 — Criar Usuario

POST `/api/usuarios`

Finalidade: Criar Usuario. Público. Somente SCREEN_USERS + FULL_ACCESS podem escolher perfil; nos demais casos é forçado CONSULTOR.

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:86` — `UsuarioController.criarUsuario`.

Autenticação, autorização e perfis: Acesso público. Público. Somente SCREEN_USERS + FULL_ACCESS podem escolher perfil; nos demais casos é forçado CONSULTOR. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: nenhum de autenticação obrigatório. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `UsuarioRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "login": "novo.usuario@example.com",
  "nome": "Usuário de exemplo",
  "senha": "SenhaExemplo123",
  "perfil": "CONSULTOR"
}
```

json

Resposta de sucesso: `UsuarioResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`, `429`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Login já existe: [mensagem pode receber complemento dinâmico]
- Nenhum perfil ativo cadastrado para validar usuários.

Regras de negócio e dependências: `UsuarioService.salvarUsuario`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/usuarios" -H 'Content-Type: application/json' --data '{"login":"novo.usuario@exa
```

bash

API-113 — Atualizar usuário (nome/login/perfil)

PUT /api/usuarios/id/{id}

Finalidade: Atualizar usuário (nome/login/perfil).

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:117` —

`UsuarioController.atualizarUsuario`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_USERS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `UsuarioUpdateRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "login": "operador@example.com",
  "nome": "Usuário de exemplo",
  "perfil": "CONSULTOR"
}
```

json

Resposta de sucesso: `UsuarioResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Login já existe: [mensagem pode receber complemento dinâmico]
- Nenhum perfil ativo cadastrado para validar usuários.

Regras de negócio e dependências: `UsuarioService.atualizarUsuario`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/usuarios/id/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/
```

API-114 — Alterar senha do usuário

POST `/api/usuarios/id/{id}/alterar-senha`

Finalidade: Alterar senha do usuário. Próprio usuário OU SCREEN_USERS + FULL_ACCESS. A senha atual é validada nos dois casos; não é um fluxo de recuperação administrativa sem senha.

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:160` — `UsuarioController.alterarSenha`.

Autenticação, autorização e perfis: Usuário autenticado; sem autoridade adicional declarada no método. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo da requisição: `UsuarioChangePasswordRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "senhaAtual": "SenhaExemplo123",
  "novaSenha": "NovaSenhaExemplo123",
  "confirmarNovaSenha": "NovaSenhaExemplo123"
}
```

json

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `400`, `401`, `403`, `404`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Confirmação de senha não confere.
- Nova senha deve ter no mínimo 6 caracteres.
- Senha atual inválida.

Regras de negócio e dependências: `UsuarioService.alterarSenha`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/usuarios/id/1/alterar-senha" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

bash

API-115 — Atualizar status do usuário (ativo/inativo)

PATCH `/api/usuarios/id/{id}/status`

Finalidade: Atualizar status do usuário (ativo/inativo).

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:215` — `UsuarioController.atualizarStatusUsuario`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_USERS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo da requisição: `UsuarioStatusUpdateRequestDTO` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "ativo": false
}
```

json

Resposta de sucesso: `UsuarioResponseDTO` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "login": "operador@example.com",
  "nome": "Exemplo",
  "perfil": "CONSULTOR",
  "ativo": false,
  "token": "<TOKEN_OBTIDO_NO_LOGIN>",
  "perfilResumo": {
    "id": 1,
    "code": "CONSULTOR",
    "description": "Exemplo",
    "type": "READ_ONLY",
    "allowedScreens": [],
    "active": false,
    "createdAt": "2026-10-15T09:00:00",
    "updatedAt": "2026-10-15T09:00:00"
  }
}
```

json

Códigos HTTP: `200` , `400` , `401` , `403` , `404` , `500` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `UsuarioService.atualizarStatus` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/usuarios/id/1/status" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: app
```

bash

API-116 — Excluir usuário por ID

DELETE `/api/usuarios/id/{id}`

Finalidade: Excluir usuário por ID.

Fonte: `src/main/java/com/wester/storage/controller/UsuarioController.java:238` —
`UsuarioController.deletarUsuario`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_USERS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`, `404`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `UsuarioService.deletarUsuario`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/usuarios/id/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

Produtos

Índice de endpoints · Modelos

API-075 — Listar Todos Produtos

GET /api/products

GET /api/produtos

Finalidade: Listar Todos Produtos.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:41` — `ProdutoController.listarTodosProdutos`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>Integer</code>	Não	—
<code>size</code>	query	<code>Integer</code>	Não	—
<code>search</code>	query	<code>String</code>	Não	—
<code>status</code>	query	<code>String</code>	Não	—
<code>categoria</code>	query	<code>String</code>	Não	—
<code>nome</code>	query	<code>String</code>	Não	—
<code>name</code>	query	<code>String</code>	Não	—
<code>codigo</code>	query	<code>String</code>	Não	—
<code>codigoSistemaWester</code>	query	<code>String</code>	Não	—
<code>cor</code>	query	<code>String</code>	Não	—
<code>descricao</code>	query	<code>String</code>	Não	—
<code>sort</code>	query	<code>String</code>	Não	—
<code>direction</code>	query	<code>String</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `ProdutoListaVariavel`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{}
```

json

Códigos HTTP: 200, 401, 403. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Categoria inválida: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `ProdutoService.shouldUsePaginatedListing`, `ProdutoService.listarTodosProdutos`, `ProdutoService.listarProdutosPaginados`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/products" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-076 — Buscar Produto Por Id

GET `/api/products/{id}`

GET `/api/produtos/{id}`

Finalidade: Buscar Produto Por Id.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:79` — `ProdutoController.buscarProdutoPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Produto`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigoSistemaWester": "Exemplo",
  "nomeModelo": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "ativo": false,
  "estoqueMinimo": 1,
  "estoqueMaximo": 1,
  "categoria": {
    "id": 1,
    "codigo": "Exemplo",
    "descricao": "Exemplo"
  }
}
```

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `ProdutoService.buscarProdutoPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/products/1" -H "Authorization: Bearer ${TOKEN}"
```

API-077 — Buscar Por Codigo Wester

GET `/api/products/codigo/{codigoWester}`

GET `/api/produtos/codigo/{codigoWester}`

Finalidade: Buscar Por Codigo Wester.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:88` — `ProdutoController.buscarPorCodigoWester`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>codigoWester</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Produto`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigoSistemaWester": "Exemplo",
  "nomeModelo": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "ativo": false,
  "estoqueMinimo": 1,
  "estoqueMaximo": 1,
  "categoria": {
    "id": 1,
    "codigo": "Exemplo",
    "descricao": "Exemplo"
  }
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `ProdutoService.buscarPorCodigoWester`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/products/codigo/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-078 — Criar Produto

POST `/api/products`

POST `/api/produtos`

Finalidade: Criar Produto.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:98` — `ProdutoController.criarProduto`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `ProdutoUpsertRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigoSistemaWester": "DOC-001",
  "nomeModelo": "Produto de exemplo",
  "cor": "Preto",
  "descricao": "Exemplo documental",
  "ativo": true,
  "estoqueMinimo": 10,
  "estoqueMaximo": 100
}
```

json

Resposta de sucesso: `Produto`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigoSistemaWester": "Exemplo",
  "nomeModelo": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "ativo": false,
  "estoqueMinimo": 1,
  "estoqueMaximo": 1,
  "categoria": {
    "id": 1,
    "codigo": "Exemplo",
    "descricao": "Exemplo"
  }
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Código Wester já existe: [mensagem pode receber complemento dinâmico]
- Informe ao menos o código ou o nome do produto.
- Estoque máximo não pode ser menor que o estoque mínimo.

Regras de negócio e dependências: `ProdutoService.salvarProduto`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/products" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-079 — Atualizar Produto

PUT /api/products/{id}

PUT /api/produtos/{id}

Finalidade: Atualizar Produto.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:113` —
`ProdutoController.atualizarProduto`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `ProdutoUpsertRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigoSistemaWester": "DOC-001",
  "nomeModelo": "Produto de exemplo",
  "cor": "Preto",
  "descricao": "Exemplo documental",
  "ativo": true,
  "estoqueMinimo": 10,
  "estoqueMaximo": 100
}
```

json

Resposta de sucesso: `Produto`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigoSistemaWester": "Exemplo",
  "nomeModelo": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "ativo": false,
  "estoqueMinimo": 1,
  "estoqueMaximo": 1,
  "categoria": {
    "id": 1,
    "codigo": "Exemplo",
    "descricao": "Exemplo"
  }
}
```

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Código Wester já existe: [mensagem pode receber complemento dinâmico]
- Informe ao menos o código ou o nome do produto.
- Estoque máximo não pode ser menor que o estoque mínimo.

Regras de negócio e dependências: `ProdutoService.atualizarProduto`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/products/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-080 — Deletar Produto

DELETE `/api/products/{id}`

DELETE `/api/produtos/{id}`

Finalidade: Deletar Produto.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:133` — `ProdutoController.deletarProduto`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: 204, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `ProdutoService.deletarProduto`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/products/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-081 — Ativar Produto

PATCH /api/products/{id}/activate ; **POST** /api/products/{id}/activate

PATCH /api/products/{id}/ativar ; **POST** /api/products/{id}/ativar

PATCH /api/produtos/{id}/activate ; **POST** /api/produtos/{id}/activate

PATCH /api/produtos/{id}/ativar ; **POST** /api/produtos/{id}/ativar

Finalidade: Ativar Produto.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:151` — `ProdutoController.ativarProduto`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Produto`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigoSistemaWester": "Exemplo",
  "nomeModelo": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "ativo": false,
  "estoqueMinimo": 1,
  "estoqueMaximo": 1,
  "categoria": {
    "id": 1,
    "codigo": "Exemplo",
    "descricao": "Exemplo"
  }
}
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `ProdutoService.ativarProduto`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PATCH "${BASE_URL}/api/products/1/activate" -H "Authorization: Bearer ${TOKEN}"
```

API-082 — Inativar Produto

PATCH `/api/products/{id}/inactivate` ; **POST** `/api/products/{id}/inactivate`

PATCH `/api/products/{id}/inativar` ; **POST** `/api/products/{id}/inativar`

PATCH `/api/produtos/{id}/inactivate` ; **POST** `/api/produtos/{id}/inactivate`

PATCH `/api/produtos/{id}/inativar` ; **POST** `/api/produtos/{id}/inativar`

Finalidade: Inativar Produto.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:162` — `ProdutoController.inativarProduto`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Produto`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigoSistemaWester": "Exemplo",
  "nomeModelo": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "ativo": false,
  "estoqueMinimo": 1,
  "estoqueMaximo": 1,
  "categoria": {
    "id": 1,
    "codigo": "Exemplo",
    "descricao": "Exemplo"
  }
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Não é possível inativar o produto porque ele possui itens em estoque.
- Não é possível inativar o produto porque ele está alocado em uma ou mais posições.

Regras de negócio e dependências: `ProdutoService.inativarProduto`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/products/1/inactivate" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-083 — Listar produtos ativos com busca opcional por código ou nome

GET `/api/products/ativos`

GET `/api/produtos/ativos`

Finalidade: Listar produtos ativos com busca opcional por código ou nome.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:168` — `ProdutoController.listarAtivos`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>search</code>	query	<code>String</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<Produto>` . Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `ProdutoService.buscarAtivos` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/products/ativos" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-084 — Buscar Produtos

`GET` `/api/products/buscar`

`GET` `/api/produtos/buscar`

Finalidade: Buscar Produtos.

Fonte: `src/main/java/com/wester/storage/controller/ProdutoController.java:177` — `ProdutoController.buscarProdutos` .

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
nomeModelo	query	String	Não	—
cor	query	String	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<Produto>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `ProdutoService.buscarPorNomeModeloContendo`, `ProdutoService.buscarPorCor`, `ProdutoService.listarTodosProdutos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/products/buscar" -H "Authorization: Bearer ${TOKEN}"
```

bash

Categorias de produtos

[Índice de endpoints](#) · [Modelos](#)

API-015 — Listar

GET `/api/categorias-produto`

GET `/api/product-categories`

Finalidade: Listar.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:39` — `CategoriaProdutoController.listar`.

Autenticação, autorização e perfis: `hasAnyAuthority('SCREEN_PRODUCTS', 'SCREEN_REPORTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<CategoriaProdutoDTO>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `CategoriaProdutoService.listar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/categorias-produto" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-016 — Criar

POST `/api/categorias-produto`

POST `/api/product-categories`

Finalidade: Criar.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:45` — `CategoriaProdutoController.criar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `CategoriaProdutoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigo": "Exemplo",
  "descricao": "Exemplo"
}
```

json

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Código de categoria já existe: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `CategoriaProdutoService.criar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/categorias-produto" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applic
```

bash

API-017 — Atualizar

PUT /api/categorias-produto/{id}

PUT /api/product-categories/{id}

Finalidade: Atualizar.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:55` — `CategoriaProdutoController.atualizar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `CategoriaProdutoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigo": "Exemplo",
  "descricao": "Exemplo"
}
```

json

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Código de categoria já existe: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `CategoriaProdutoService.atualizar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/categorias-produto/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: appli
```

bash

API-018 — Excluir

DELETE `/api/categorias-produto/{id}`

DELETE `/api/product-categories/{id}`

Finalidade: Excluir.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:61` — `CategoriaProdutoController.excluir`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Não é possível excluir: a categoria possui

Regras de negócio e dependências: `CategoriaProdutoService.excluir`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/categorias-produto/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-019 — Vincular

PUT `/api/categorias-produto/{id}/produtos`

PUT `/api/product-categories/{id}/produtos`

Finalidade: Vincular.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:70` — `CategoriaProdutoController.vincular`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `CategoriaProdutoVinculoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produtoIds": [
    1
  ]
}
```

json

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Um ou mais produtos informados não foram encontrados.

Regras de negócio e dependências: `CategoriaProdutoService.vincularProdutos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/categorias-produto/1/produtos" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-020 — Desvincular

DELETE `/api/categorias-produto/{id}/produtos`

DELETE `/api/product-categories/{id}/produtos`

Finalidade: Desvincular.

Fonte: `src/main/java/com/wester/storage/controller/CategoriaProdutoController.java:79` — `CategoriaProdutoController.desvincular`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `CategoriaProdutoVinculoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produtoIds": [
    1
  ]
}
```

Resposta de sucesso: `CategoriaProdutoDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "totalProdutos": 1
}
```

Códigos HTTP: 200, 400, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Um ou mais produtos informados não foram encontrados.

Regras de negócio e dependências: `CategoriaProdutoService.desvincularProdutos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X DELETE "${BASE_URL}/api/categorias-produto/1/produtos" -H "Authorization: Bearer ${TOKEN}" -H 'Content
```

Setores e tipos de material

[Índice de endpoints](#) · [Modelos](#)

API-002 — Listar Todas Areas

GET `/api/areas`

GET `/api/setores`

Finalidade: Listar Todas Areas.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:46` —
`AreaEstoqueController.listarTodasAreas`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<AreaEstoque>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `AreaEstoqueService.listarTodasAreas`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/areas" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-003 — Buscar Area Por Id

GET `/api/areas/{id}`

GET `/api/setores/{id}`

Finalidade: Buscar Area Por Id.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:53` — `AreaEstoqueController.buscarAreaPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `AreaEstoque`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `AreaEstoqueService.buscarAreaPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/areas/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-004 — Buscar Por Nome

GET `/api/areas/nome/{nome}`

GET `/api/setores/nome/{nome}`

Finalidade: Buscar Por Nome.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:62` —
`AreaEstoqueController.buscarPorNome`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nome</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `AreaEstoque`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `AreaEstoqueService.buscarPorNome`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/areas/nome/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-005 — Criar Area

POST /api/areas

POST /api/setores

Finalidade: Criar Area.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:72` —
`AreaEstoqueController.criarArea`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `AreaEstoque`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

json

```
{
  "nome": "Setor de exemplo",
  "descricao": "Área de exemplo",
  "ativo": true,
  "tipoEstrutura": "DINAMICO"
}
```

Resposta de sucesso: `AreaEstoque`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}
```

Códigos HTTP: 201, 400, 401, 403, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Nome da área de estoque já existe: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `AreaEstoqueService.salvarArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/areas" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json' -
```

bash

API-006 — Atualizar Area

PUT `/api/areas/{id}`

PUT `/api/setores/{id}`

Finalidade: Atualizar Area.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:86` — `AreaEstoqueController.atualizarArea`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `AreaEstoque`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "nome": "Setor de exemplo",
  "descricao": "Área de exemplo",
  "ativo": true,
  "tipoEstrutura": "DINAMICO"
}
```

json

Resposta de sucesso: `AreaEstoque`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}
```

json

Códigos HTTP: 200, 400, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Nome da área de estoque já existe: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `AreaEstoqueService.atualizarArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/areas/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

bash

API-007 — Atualizar Labels

PATCH `/api/areas/{id}/labels`

PATCH `/api/setores/{id}/labels`

Finalidade: Atualizar Labels.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:107` — `AreaEstoqueController.atualizarLabels`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo da requisição: `AreaLabelsUpdateRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```

{
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo"
}

```

Resposta de sucesso: `AreaEstoque`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```

{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}

```

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Labels configuraveis estao disponiveis para setores FIXO.

Regras de negócio e dependências: `AreaEstoqueService.atualizarLabels`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```

curl -X PATCH "${BASE_URL}/api/areas/1/labels" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicati

```

API-008 — Ativar Area

PATCH `/api/areas/{id}/activate` ; **POST** `/api/areas/{id}/activate`

PATCH `/api/areas/{id}/ativar` ; **POST** `/api/areas/{id}/ativar`

PATCH `/api/setores/{id}/activate` ; **POST** `/api/setores/{id}/activate`

PATCH `/api/setores/{id}/ativar` ; **POST** `/api/setores/{id}/ativar`

Finalidade: Ativar Area.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:129` — `AreaEstoqueController.ativarArea`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `AreaEstoque` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}
```

json

Códigos HTTP: `200` , `401` , `403` , `404` , `500` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `AreaEstoqueService.ativarArea` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/areas/1/activate" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-009 — Inativar Area

PATCH /api/areas/{id}/inactivate ; **POST** /api/areas/{id}/inactivate

PATCH /api/areas/{id}/inativar ; **POST** /api/areas/{id}/inativar

PATCH /api/setores/{id}/inactivate ; **POST** /api/setores/{id}/inactivate

PATCH /api/setores/{id}/inativar ; **POST** /api/setores/{id}/inativar

Finalidade: Inativar Area.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:145` —
`AreaEstoqueController.inativarArea`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `AreaEstoque` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "descricao": "Exemplo",
  "nome": "Exemplo",
  "ativo": false,
  "tipoEstrutura": "DINAMICO",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelPosicao": "Exemplo"
}
```

json

Códigos HTTP: 200, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `AreaEstoqueService.inativarArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/areas/1/inactivate" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-010 — Deletar Area

DELETE `/api/areas/{id}`

DELETE `/api/setores/{id}`

Finalidade: Deletar Area.

Fonte: `src/main/java/com/wester/storage/controller/AreaEstoqueController.java:162` — `AreaEstoqueController.deletarArea`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: 204, 401, 403, 404, 409, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Não é possível deletar a área de estoque pois ela contém fileiras associadas.

Regras de negócio e dependências: `AreaEstoqueService.deletarArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/areas/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-011 — Listar

GET `/api/area-material-tags/area/{areaId}`

Finalidade: Listar.

Fonte: `src/main/java/com/wester/storage/controller/AreaMaterialTagController.java:26` — `AreaMaterialTagController.listar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<MaterialType>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `AreaMaterialTagService.listarTags`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/area-material-tags/area/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-012 — Configurar

PUT `/api/area-material-tags/area/{areaId}`

Finalidade: Configurar.

Fonte: `src/main/java/com/wester/storage/controller/AreaMaterialTagController.java:33` — `AreaMaterialTagController.configurar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `AreaMaterialTagsRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "materialTypes": [
    "MOLDE",
    "PRODUTO"
  ]
}
```

json

Resposta de sucesso: `List<MaterialType>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `AreaMaterialTagService.configurarTags`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/area-material-tags/area/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type:
```

bash

Corredores, blocos e níveis

[Índice de endpoints](#) · [Modelos](#)

API-024 — Listar fileiras de uma área de estoque

GET `/api/fileiras/area/{areaId}`**Finalidade:** Listar fileiras de uma área de estoque.**Fonte:** `src/main/java/com/wester/storage/controller/FileiraController.java:38` — `FileiraController.listarFileirasPorArea`.**Autenticação, autorização e perfis:** `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).**Headers:** `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.**Parâmetros:**

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.**Resposta de sucesso:** `List<FileiraResponseDTO>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).**Regras de negócio e dependências:** `FileiraService.listarFileirasPorArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.**cURL (Bash):**

```
curl -X GET "${BASE_URL}/api/fileiras/area/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-025 — Buscar Fileira Por Id

GET `/api/fileiras/{id}`**Finalidade:** Buscar Fileira Por Id.

Fonte: `src/main/java/com/wester/storage/controller/FileiraController.java:45` — `FileiraController.buscarFileiraPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Fileira`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "identificador": "Exemplo",
  "ordem": 1,
  "areaEstoque": {
    "id": 1,
    "descricao": "Exemplo",
    "nome": "Exemplo",
    "ativo": false,
    "tipoEstrutura": "DINAMICO",
    "labelEstruturaPrimaria": "Exemplo",
    "labelEstruturaSecundaria": "Exemplo",
    "labelNivel": "Exemplo",
    "labelSlot": "Exemplo",
    "labelFileira": "Exemplo",
    "labelGrade": "Exemplo",
    "labelPosicao": "Exemplo"
  },
  "ativo": false
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `FileiraService.buscarFileiraPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/fileiras/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-026 — Criar nova fileira para uma área de estoque

POST `/api/fileiras/area/{areaId}`

Finalidade: Criar nova fileira para uma área de estoque.

Fonte: `src/main/java/com/wester/storage/controller/FileiraController.java:68` — `FileiraController.criarFileira`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `CriarFileiraRequestDTO`; opcional. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "identificador": "Exemplo",
  "ordem": 1
}
```

json

Resposta de sucesso: `FileiraResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "identificador": "Exemplo",
  "ordem": 1,
  "areaEstoqueId": 1
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- N\u00e3o foi poss\u00edvel criar o corredor porque a sequ\u00eancia da \u00e1rea ficou desatualizada. Atualize a estrutura e tente novamente.
- Identificador \u00e9 obrigat\u00f3rio.

- Identificador de Fileira já existe nesta Área: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `FileiraService.criar` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/fileiras/area/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicati
```

API-027 — Atualizar Fileira

`PUT` `/api/fileiras/{id}`

Finalidade: Atualizar Fileira.

Fonte: `src/main/java/com/wester/storage/controller/FileiraController.java:82` — `FileiraController.atualizarFileira` .

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `Fileira` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "identificador": "Exemplo",
  "ordem": 1,
  "areaEstoque": {
    "nome": "Setor de exemplo",
    "descricao": "Área de exemplo",
    "ativo": true,
    "tipoEstrutura": "DINAMICO"
  },
  "ativo": false
}
```

Resposta de sucesso: `Fileira` . Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "identificador": "Exemplo",
  "ordem": 1,
  "areaEstoque": {
    "id": 1,
    "descricao": "Exemplo",
    "nome": "Exemplo",
    "ativo": false,
    "tipoEstrutura": "DINAMICO",
    "labelEstruturaPrimaria": "Exemplo",
    "labelEstruturaSecundaria": "Exemplo",
    "labelNivel": "Exemplo",
    "labelSlot": "Exemplo",
    "labelFileira": "Exemplo",
    "labelGrade": "Exemplo",
    "labelPosicao": "Exemplo"
  },
  "ativo": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Não é permitido alterar a área de Estoque de uma Fileira existente.
- Identificador de Fileira já existe nesta área: [mensagem pode receber complemento dinâmico]
- Ordem de Fileira já existe nesta área: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `FileiraService.atualizarFileira`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/fileiras/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-028 — Atualizar identificador de uma estrutura primaria fixa

PATCH `/api/fileiras/{id}/identificador`

Finalidade: Atualizar identificador de uma estrutura primaria fixa.

Fonte: `src/main/java/com/wester/storage/controller/FileiraController.java:105` — `FileiraController.atualizarIdentificador`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `IdentifierUpdateRequestDTO` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "identificador": "Exemplo",
  "version": 0
}
```

json

Resposta de sucesso: `Fileira` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "identificador": "Exemplo",
  "ordem": 1,
  "areaEstoque": {
    "id": 1,
    "descricao": "Exemplo",
    "nome": "Exemplo",
    "ativo": false,
    "tipoEstrutura": "DINAMICO",
    "labelEstruturaPrimaria": "Exemplo",
    "labelEstruturaSecundaria": "Exemplo",
    "labelNivel": "Exemplo",
    "labelSlot": "Exemplo",
    "labelFileira": "Exemplo",
    "labelGrade": "Exemplo",
    "labelPosicao": "Exemplo"
  },
  "ativo": false
}
```

json

Códigos HTTP: `200` , `400` , `401` , `403` , `404` , `409` , `500` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Identificador manual é permitido apenas para setores FIXO.
- Identificador é obrigatório.

- Identificador de Fileira já existe nesta Área: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `FileiraService.atualizarIdentificador`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/fileiras/1/identificador" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

bash

API-029 — Desativa uma Estante (inativação lógica, permanente — esvazia tudo dentro antes)

DELETE `/api/fileiras/{id}`

Finalidade: Desativa uma Estante (inativação lógica, permanente — esvazia tudo dentro antes).

Fonte: `src/main/java/com/wester/storage/controller/FileiraController.java:127` — `FileiraController.deletarFileira`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`, `404`, `409`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Estante já está desativada.

Regras de negócio e dependências: `FileiraService.desativarFileira`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/fileiras/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-030 — Listar Grades Por Fileira

GET `/api/grades/fileira/{fileiraId}`

Finalidade: Listar Grades Por Fileira.

Fonte: `src/main/java/com/wester/storage/controller/GradeController.java:40` — `GradeController.listarGradesPorFileira`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>fileiraId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<Grade>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `GradeService.listarGradesPorFileira`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/grades/fileira/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-031 — Buscar Grade Por Id

GET `/api/grades/{id}`

Finalidade: Buscar Grade Por Id.

Fonte: `src/main/java/com/wester/storage/controller/GradeController.java:48` — `GradeController.buscarGradePorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Grade`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "fileira": {
    "id": 1,
    "identificador": "Exemplo",
    "ordem": 1,
    "areaEstoque": {
      "id": 1,
      "descricao": "Exemplo",
      "nome": "Exemplo",
      "ativo": false,
      "tipoEstrutura": "DINAMICO",
      "labelEstruturaPrimaria": "Exemplo",
      "labelEstruturaSecundaria": "Exemplo",
      "labelNivel": "Exemplo",
      "labelSlot": "Exemplo",
      "labelFileira": "Exemplo",
      "labelGrade": "Exemplo",
      "labelPosicao": "Exemplo"
    },
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "ativo": false
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `GradeService.buscarGradePorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/grades/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-032 — Criar grade em uma fileira

POST `/api/grades/fileira/{fileiraId}`

Finalidade: Criar grade em uma fileira.

Fonte: `src/main/java/com/wester/storage/controller/GradeController.java:63` — `GradeController.criarGrade`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>fileiraId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `GradeCreateRequestDTO`; opcional. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "identificador": "Exemplo",
  "ordem": 1
}
```

json

Resposta de sucesso: `GradeResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "identificador": "Exemplo",
  "ordem": 1,
  "fileiraId": 1
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- `N\u00e3o foi poss\u00edvel criar o bloco porque a sequ\u00eancia da fileira ficou desatualizada. Atualize a estrutura e tente novamente.`
- Identificador é obrigatório.

- Identificador de Grade já existe nesta Fileira: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `GradeService.criarGrade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/grades/fileira/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicat
```

bash

API-033 — Atualizar Grade

PUT `/api/grades/{id}`

Finalidade: Atualizar Grade.

Fonte: `src/main/java/com/wester/storage/controller/GradeController.java:78` — `GradeController.atualizarGrade`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `Grade`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "fileira": {
    "identificador": "Exemplo",
    "ordem": 1,
    "areaEstoque": {
      "nome": "Setor de exemplo",
      "descricao": "Área de exemplo",
      "ativo": true,
      "tipoEstrutura": "DINAMICO"
    },
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "ativo": false
}
```

json

Resposta de sucesso: `Grade`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "fileira": {
    "id": 1,
    "identificador": "Exemplo",
    "ordem": 1,
    "areaEstoque": {
      "id": 1,
      "descricao": "Exemplo",
      "nome": "Exemplo",
      "ativo": false,
      "tipoEstrutura": "DINAMICO",
      "labelEstruturaPrimaria": "Exemplo",
      "labelEstruturaSecundaria": "Exemplo",
      "labelNivel": "Exemplo",
      "labelSlot": "Exemplo",
      "labelFileira": "Exemplo",
      "labelGrade": "Exemplo",
      "labelPosicao": "Exemplo"
    },
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "ativo": false
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- `N\u00e3o \u00e9` permitido alterar a Fileira de uma Grade existente.
- Identificador de Grade `j\u00e1` existe nesta Fileira: [mensagem pode receber complemento dinâmico]
- Ordem de Grade `j\u00e1` existe nesta Fileira: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `GradeService.atualizarGrade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/grades/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

bash

API-034 — Atualizar identificador de uma estrutura secundaria fixa

PATCH `/api/grades/{id}/identificador`

Finalidade: Atualizar identificador de uma estrutura secundaria fixa.

Fonte: `src/main/java/com/wester/storage/controller/GradeController.java:101` — `GradeController.atualizarIdentificador`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `IdentifierUpdateRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "identificador": "Exemplo",
  "version": 0
}
```

json

Resposta de sucesso: `Grade`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "fileira": {
    "id": 1,
    "identificador": "Exemplo",
    "ordem": 1,
    "areaEstoque": {
      "id": 1,
      "descricao": "Exemplo",
      "nome": "Exemplo",
      "ativo": false,
      "tipoEstrutura": "DINAMICO",
      "labelEstruturaPrimaria": "Exemplo",
      "labelEstruturaSecundaria": "Exemplo",
      "labelNivel": "Exemplo",
      "labelSlot": "Exemplo",
      "labelFileira": "Exemplo",
      "labelGrade": "Exemplo",
      "labelPosicao": "Exemplo"
    },
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "ativo": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Identificador manual é permitido apenas para setores FIXO.
- Identificador é obrigatório.
- Identificador de Grade já existe nesta Fileira: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `GradeService.atualizarIdentificador`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PATCH "${BASE_URL}/api/grades/1/identificador" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: a
```

API-035 — Desativa um Bloco (inativação lógica, permanente — esvazia tudo dentro antes)

DELETE `/api/grades/{id}`

Finalidade: Desativa um Bloco (inativação lógica, permanente — esvazia tudo dentro antes).

Fonte: `src/main/java/com/wester/storage/controller/GradeController.java:123` — `GradeController.deletarGrade`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`, `404`, `409`, `500`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Bloco `j\u00e9 est\u00e9` desativado.

Regras de negócio e dependências: `GradeService.desativarGrade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/grades/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-055 — Listar Níveis Por Grade

GET `/api/niveis/grade/{gradeId}`

Finalidade: Listar Níveis Por Grade.

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:44` — `NivelController.listarNiveisPorGrade`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>gradeId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<Nivel>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `NivelService.listarNiveisPorGrade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/niveis/grade/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-056 — Buscar Nivel Por Id

GET `/api/niveis/{id}`

Finalidade: Buscar Nivel Por Id.

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:52` — `NivelController.buscarNivelPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Nivel`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "version": 0,
  "grade": {
    "fileira": {
      "id": 1,
      "identificador": "Exemplo",
      "ordem": 1,
      "areaEstoque": {
        "id": 1,
        "descricao": "Exemplo",
        "nome": "Exemplo",
        "ativo": false,
        "tipoEstrutura": "DINAMICO",
        "labelEstruturaPrimaria": "Exemplo",
        "labelEstruturaSecundaria": "Exemplo",
        "labelNivel": "Exemplo",
        "labelSlot": "Exemplo",
        "labelFileira": "Exemplo",
        "labelGrade": "Exemplo",
        "labelPosicao": "Exemplo"
      },
      "ativo": false
    },
    "identificador": "Exemplo",
    "ordem": 1,
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "identificadorCompletoLegivel": "Exemplo",
  "ativo": false
}
```

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `NivelService.buscarNivelPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/niveis/1" -H "Authorization: Bearer ${TOKEN}"
```

API-057 — Criar nível em uma grade

POST `/api/niveis/grade/{gradeId}`

Finalidade: Criar nível em uma grade.

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:67` — `NivelController.criarNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>gradeId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `NivelCreateRequestDTO` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```

{
  "identificador": "Exemplo",
  "ordem": 1,
  "item": {
    "quantidade": 10,
    "quantidadeAproximada": false,
    "observacao": "Contagem conferida",
    "produto": {
      "codigoSistemaWester": "DOC-001",
      "nomeModelo": "Produto de exemplo",
      "cor": "Preto",
      "descricao": "Exemplo documental"
    }
  }
}
```

Resposta de sucesso: `NivelResponseDTO` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```

{
  "id": 1,
  "version": 0,
  "identificador": "Exemplo",
  "ordem": 1,
  "gradeId": 1,
  "itemEstoqueId": 1
}
```

Códigos HTTP: `201` , `400` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Posição inválida para inserção. Informe um valor entre 1 e
- Informe uma quantidade maior que zero para o produto da gaiola.
- Identificador de nível já existe nesta grade: [mensagem pode receber complemento dinâmico]
- Ordem de nível já existe nesta grade: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `NivelService.criarNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/niveis/grade/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicatio
```

bash

API-058 — Mover nível entre grades e posicoes

POST `/api/niveis/{nivelId}/move`

Finalidade: Mover nível entre grades e posicoes.

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:87` — `NivelController.moverNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `NivelMoveRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "sourceGradeId": 1,
  "targetGradeId": 2,
  "targetPosition": 1,
  "nivelVersion": 0
}
```

json

Resposta de sucesso: `NivelMoveResultDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "nivelId": 1,
  "sourceGradeId": 1,
  "targetGradeId": 1,
  "targetPosition": 1,
  "impactedGrades": []
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- A grade de origem informada não corresponde ao nível selecionado.
- Grade de origem ou destino não encontrada.
- Posição inválida para movimentação. Informe um valor entre 1 e
- Nível não encontrado na sequência atual: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `NivelService.moverNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/niveis/1/move" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-059 — Atualizar Nível

PUT `/api/niveis/{id}`

Finalidade: Atualizar Nível.

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:107` — `NivelController.atualizarNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `Nivel`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "version": 0,
  "grade": {
    "fileira": {
      "identificador": "Exemplo",
      "ordem": 1,
      "areaEstoque": {
        "nome": "Setor de exemplo",
        "descricao": "Área de exemplo",
        "ativo": true,
        "tipoEstrutura": "DINAMICO"
      },
      "ativo": false
    },
    "identificador": "Exemplo",
    "ordem": 1,
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "identificadorCompletoLegivel": "Exemplo",
  "ativo": false
}
```

Resposta de sucesso: **Nível** . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "version": 0,
  "grade": {
    "fileira": {
      "id": 1,
      "identificador": "Exemplo",
      "ordem": 1,
      "areaEstoque": {
        "id": 1,
        "descricao": "Exemplo",
        "nome": "Exemplo",
        "ativo": false,
        "tipoEstrutura": "DINAMICO",
        "labelEstruturaPrimaria": "Exemplo",
        "labelEstruturaSecundaria": "Exemplo",
        "labelNivel": "Exemplo",
        "labelSlot": "Exemplo",
        "labelFileira": "Exemplo",
        "labelGrade": "Exemplo",
        "labelPosicao": "Exemplo"
      },
      "ativo": false
    },
    "identificador": "Exemplo",
    "ordem": 1,
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "identificadorCompletoLegivel": "Exemplo",
  "ativo": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 500. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Não é permitido alterar a grade de um nível existente.
- A ordem do nível deve ser maior ou igual a 1.
- Identificador de nível já existe nesta grade: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `NivelService.atualizarNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/niveis/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-060 — Atualizar identificador de um nível fixo

PATCH /api/niveis/{id}/identificador

Finalidade: Atualizar identificador de um nível fixo.

Fonte: src/main/java/com/wester/storage/controller/NivelController.java:138 —
NivelController.atualizarIdentificador .

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: Authorization: Bearer <TOKEN> . Accept: application/json recomendado. Content-Type: application/json para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo da requisição: IdentifierUpdateRequestDTO ; obrigatório. Validação Jakarta ativada por @Valid . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "identificador": "Exemplo",
  "version": 0
}
```

json

Resposta de sucesso: Nivel . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "version": 0,
  "grade": {
    "fileira": {
      "id": 1,
      "identificador": "Exemplo",
      "ordem": 1,
      "areaEstoque": {
        "id": 1,
        "descricao": "Exemplo",
        "nome": "Exemplo",
        "ativo": false,
        "tipoEstrutura": "DINAMICO",
        "labelEstruturaPrimaria": "Exemplo",
        "labelEstruturaSecundaria": "Exemplo",
        "labelNivel": "Exemplo",
        "labelSlot": "Exemplo",
        "labelFileira": "Exemplo",
        "labelGrade": "Exemplo",
        "labelPosicao": "Exemplo"
      },
      "ativo": false
    },
    "identificador": "Exemplo",
    "ordem": 1,
    "ativo": false
  },
  "identificador": "Exemplo",
  "ordem": 1,
  "identificadorCompletoLegivel": "Exemplo",
  "ativo": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Identificador manual é permitido apenas para setores FIXO.
- Identificador é obrigatório.
- Identificador de nível já existe nesta grade: [mensagem pode receber complemento dinâmico]
- Ordem de nível já existe nesta grade: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `NivelService.atualizarIdentificador`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/niveis/1/identificador" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: a
```

API-061 — Remove um nível e resequencia níveis da grade

DELETE `/api/niveis/{nivelId}/resequence`

Finalidade: Remove um nível e resequencia níveis da grade. Remove e renumera os níveis restantes; devolve GradeResequeneceResultDTO. Não confundir com DELETE simples.

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:162` — `NivelController.deleteAndResequenece`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—
<code>version</code>	query	<code>Long</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `GradeResequeneceResultDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "gradeId": 1,
  "niveis": []
}
```

json

Códigos HTTP: `200`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Nível não pode ser removido pela regra estrutural atual.
- Erro de integridade ao resequenciar níveis. Verifique constraints de ordem e identificador.

Regras de negócio e dependências: `NivelService.deleteAndResequenece`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/niveis/1/resequence" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-062 — Remove um nível (sem resequenciar)

DELETE /api/niveis/{nivelId}

Finalidade: Remove um nível (sem resequenciar).

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:184` — `NivelController.deleteNivelOnly`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—
<code>version</code>	query	<code>Long</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Não foi possível remover o nível porque existem registros vinculados a ele.

Regras de negócio e dependências: `NivelService.deleteNivelOnly`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/niveis/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-063 — Desativa um Nível de setor fixo (inativação lógica, permanente — esvazia materiais antes)

DELETE /api/niveis/{nivelId}/desativar

Finalidade: Desativa um Nível de setor fixo (inativação lógica, permanente — esvazia materiais antes).

Fonte: `src/main/java/com/wester/storage/controller/NivelController.java:206` — `NivelController.desativarNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
nivelId	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204` , `401` , `403` , `404` , `409` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Nível já está desativado.

Regras de negócio e dependências: `NivelService.desativarNivel` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/niveis/1/desativar" -H "Authorization: Bearer ${TOKEN}"
```

bash

Itens e transferências de estoque

Índice de endpoints · Modelos

API-040 — Buscar Por Nível

GET /api/itens-estoque/nivel/{nivelId}

Finalidade: Buscar Por Nível. Nível existente sem item retorna 204. Nível inexistente lança IllegalArgumentException no serviço, tratado como 400; a anotação declara 404, mas não representa esse ramo real.

Fonte: src/main/java/com/wester/storage/controller/ItemEstoqueController.java:50 — ItemEstoqueController.buscarPorNivel .

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: Authorization: Bearer <TOKEN> . Accept: application/json recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
nivelId	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: ItemEstoqueDTO . Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "version": 0,
  "nivelId": 1,
  "nivelIdentificador": "Exemplo",
  "nivelCompletoIdentificador": "Exemplo",
  "produtoId": 1,
  "produtoCodigoWester": "Exemplo",
  "produtoNomeModelo": "Exemplo",
  "produtoCor": "Exemplo",
  "produtoDescricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "quantidadeAproximada": false,
  "dataAtualizacao": "2026-10-15T09:00:00"
}
```

Códigos HTTP: 200, 204, 401, 403. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- nivelId não pode ser nulo.
- Nível não encontrado: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: ItemEstoqueService.buscarPorNivel. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/itens-estoque/nivel/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-041 — Create or update ItemEstoque by nivelId (UPSERT)

PUT /api/itens-estoque/nivel/{nivelId}

Finalidade: Create or update ItemEstoque by nivelId (UPSERT).

Fonte: src/main/java/com/wester/storage/controller/ItemEstoqueController.java:67 — ItemEstoqueController.upsertPorNivel.

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: Authorization: Bearer <TOKEN>. Accept: application/json recomendado. Content-Type: application/json para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
nivelId	path	Long	Sim	—

Corpo da requisição: ItemEstoqueUpsertRequestDTO; obrigatório. Validação Jakarta ativada por @Valid. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

json

```
{
  "quantidade": 10,
  "quantidadeAproximada": false,
  "observacao": "Contagem conferida",
  "produto": {
    "codigoSistemaWester": "DOC-001",
    "nomeModelo": "Produto de exemplo",
    "cor": "Preto",
    "descricao": "Exemplo documental"
  }
}
```

Resposta de sucesso: `ItemEstoqueDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "version": 0,
  "nivelId": 1,
  "nivelIdentificador": "Exemplo",
  "nivelCompletoIdentificador": "Exemplo",
  "produtoId": 1,
  "produtoCodigoWester": "Exemplo",
  "produtoNomeModelo": "Exemplo",
  "produtoCor": "Exemplo",
  "produtoDescricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "quantidadeAproximada": false,
  "dataAtualizacao": "2026-10-15T09:00:00"
}
```

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- nivelId não pode ser nulo.
- request não pode ser nulo.
- Este nível pertence a um setor fixo — use a alocação de materiais (moldes/produtos),
- Dados do produto são obrigatórios.

Regras de negócio e dependências: `CurrentUserService.getCurrentUserIdOrFallback`, `ItemEstoqueService.salvarPorNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/itens-estoque/nivel/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: appl
```

API-042 — Criar Por Nivel

POST `/api/itens-estoque/nivel/{nivelId}`

Finalidade: Criar Por Nivel. Embora use POST e retorne 201, chama o mesmo upsert do PUT: pode atualizar o item existente.

Fonte: `src/main/java/com/wester/storage/controller/ItemEstoqueController.java:84` — `ItemEstoqueController.criarPorNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `ItemEstoqueUpsertRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "quantidade": 10,
  "quantidadeAproximada": false,
  "observacao": "Contagem conferida",
  "produto": {
    "codigoSistemaWester": "DOC-001",
    "nomeModelo": "Produto de exemplo",
    "cor": "Preto",
    "descricao": "Exemplo documental"
  }
}
```

json

Resposta de sucesso: `ItemEstoqueDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "version": 0,
  "nivelId": 1,
  "nivelIdentificador": "Exemplo",
  "nivelCompletoIdentificador": "Exemplo",
  "produtoId": 1,
  "produtoCodigoWester": "Exemplo",
  "produtoNomeModelo": "Exemplo",
  "produtoCor": "Exemplo",
  "produtoDescricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "quantidadeAproximada": false,
  "dataAtualizacao": "2026-10-15T09:00:00"
}
```

Códigos HTTP: 201, 400, 401, 403. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- nivelId não pode ser nulo.
- request não pode ser nulo.
- Este nível pertence a um setor fixo — use a alocação de materiais (moldes/produtos),
- Dados do produto são obrigatórios.

Regras de negócio e dependências: `currentUserService.getCurrentUserIdOrFallback`, `ItemEstoqueService.salvarPorNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/itens-estoque/nivel/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: app
```

API-043 — Movimenta item entre níveis

POST `/api/itens-estoque/movimentacao`

Finalidade: Movimenta item entre níveis.

Fonte: `src/main/java/com/wester/storage/controller/ItemEstoqueController.java:101` — `ItemEstoqueController.movimentar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `ItemEstoqueMovimentacaoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "nivelOrigemId": 1,
  "nivelDestinoId": 2,
  "quantidade": 2
}
```

json

Resposta de sucesso: `ItemEstoqueDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "version": 0,
  "nivelId": 1,
  "nivelIdentificador": "Exemplo",
  "nivelCompletoIdentificador": "Exemplo",
  "produtoId": 1,
  "produtoCodigoWester": "Exemplo",
  "produtoNomeModelo": "Exemplo",
  "produtoCor": "Exemplo",
  "produtoDescricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "quantidadeAproximada": false,
  "dataAtualizacao": "2026-10-15T09:00:00"
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- `nivelOrigemId` e `nivelDestinoId` são obrigatórios.
- Origem e destino devem ser níveis diferentes.
- Quantidade de movimentação deve ser maior que zero.
- Quantidade solicitada maior que o estoque disponível no nível de origem.
- Nível destino já contém outro produto.

Regras de negócio e dependências: `CurrentUserService.getCurrentUserIdOrFallback`, `ItemEstoqueService.movimentarEntreNiveis`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/itens-estoque/movimentacao" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-044 — Remove item de estoque do nível

DELETE `/api/itens-estoque/nivel/{nivelId}`

Finalidade: Remove item de estoque do nível.

Fonte: `src/main/java/com/wester/storage/controller/ItemEstoqueController.java:125` — `ItemEstoqueController.deletePorNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—
<code>itemVersion</code>	query	<code>Long</code>	Não	—
<code>nivelVersion</code>	query	<code>Long</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- `nivelId` não pode ser nulo.

Regras de negócio e dependências: `CurrentUserService.getCurrentUserIdOrFallback`, `ItemEstoqueService.removerPorNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/itens-estoque/nivel/1" -H "Authorization: Bearer ${TOKEN}"
```

Moldes

[Índice de endpoints](#) · [Modelos](#)

API-045 — Listar moldes

GET `/api/moldes`**Finalidade:** Listar moldes.**Fonte:** `src/main/java/com/wester/storage/controller/MoldeController.java:39` — `MoldeController.listar`.**Autenticação, autorização e perfis:** `hasAuthority('SCREEN_PRODUCTS')` or `hasAuthority('SCREEN_MOLDS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).**Headers:** `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.**Parâmetros:**

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>Integer</code>	Não	—
<code>size</code>	query	<code>Integer</code>	Não	—
<code>search</code>	query	<code>String</code>	Não	—
<code>status</code>	query	<code>String</code>	Não	—
<code>sort</code>	query	<code>String</code>	Não	—
<code>direction</code>	query	<code>String</code>	Não	—

Corpo: não declarado; não enviar corpo.**Resposta de sucesso:** `MoldeListaVariavel` . Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{}
```

json

Códigos HTTP: `200` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).**Regras de negócio e dependências:** `MoldeService.shouldUsePaginatedListing` , `MoldeService.listarMoldesPaginados` , `MoldeService.buscarAtivos` , `MoldeService.buscar` , `MoldeService.listarTodos` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.**cURL (Bash):**

```
curl -X GET "${BASE_URL}/api/moldes" -H "Authorization: Bearer ${TOKEN}"
```

API-046 — Buscar molde por ID

GET `/api/moldes/{id}`

Finalidade: Buscar molde por ID.

Fonte: `src/main/java/com/wester/storage/controller/MoldeController.java:61` — `MoldeController.buscarPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_PRODUCTS')` or `hasAuthority('SCREEN_MOLDS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Molde`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo",
  "ativo": false
}
```

Códigos HTTP: `200`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `MoldeService.buscarPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/moldes/1" -H "Authorization: Bearer ${TOKEN}"
```

API-047 — Criar molde

POST /api/moldes

Finalidade: Criar molde.

Fonte: `src/main/java/com/wester/storage/controller/MoldeController.java:71` — `MoldeController.criar`.

Autenticação, autorização e perfis: (hasAuthority('SCREEN_PRODUCTS') or hasAuthority('SCREEN_MOLDS')) and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `MoldeRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo"
}
```

json

Resposta de sucesso: `Molde`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo",
  "ativo": false
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Já existe um molde com o código: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `MoldeService.criar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/moldes" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-048 — Atualizar molde

PUT `/api/moldes/{id}`

Finalidade: Atualizar molde.

Fonte: `src/main/java/com/wester/storage/controller/MoldeController.java:83` — `MoldeController.atualizar`.

Autenticação, autorização e perfis: (hasAuthority('SCREEN_PRODUCTS') or hasAuthority('SCREEN_MOLDS')) and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `MoldeRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo"
}
```

Resposta de sucesso: `Molde`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo",
  "ativo": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Já existe um molde com o código: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `MoldeService.atualizar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/molde/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-049 — Ativar molde

PATCH /api/molde/{id}/ativar ; **POST** /api/molde/{id}/ativar

PATCH /api/molde/{id}/activate ; **POST** /api/molde/{id}/activate

Finalidade: Ativar molde.

Fonte: `src/main/java/com/wester/storage/controller/MoldeController.java:96` — `MoldeController.ativar`.

Autenticação, autorização e perfis: (hasAuthority('SCREEN_PRODUCTS') or hasAuthority('SCREEN_MOLDS')) and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Molde`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo",
  "ativo": false
}
```

json

Códigos HTTP: 200, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `MoldeService.ativar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/molde/1/ativar" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-050 — Inativar molde

PATCH `/api/molde/{id}/inativar` ; **POST** `/api/molde/{id}/inativar`

PATCH `/api/molde/{id}/inactivate` ; **POST** `/api/molde/{id}/inactivate`

Finalidade: Inativar molde.

Fonte: `src/main/java/com/wester/storage/controller/MoldeController.java:107` — `MoldeController.inativar`.

Autenticação, autorização e perfis: (hasAuthority('SCREEN_PRODUCTS') or hasAuthority('SCREEN_MOLDS')) and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `Molde`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "codigo": "Exemplo",
  "descricao": "Exemplo",
  "categoria": "Exemplo",
  "observacoes": "Exemplo",
  "ativo": false
}
```

Códigos HTTP: 200, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Molde está alocado em uma ou mais posições. Remova-o antes de inativar.

Regras de negócio e dependências: `MoldeService.inativar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PATCH "${BASE_URL}/api/molde/1/inativar" -H "Authorization: Bearer ${TOKEN}"
```

API-051 — Excluir molde

DELETE `/api/molde/{id}`

Finalidade: Excluir molde.

Fonte: `src/main/java/com/wester/storage/controller/MoldeController.java:118` — `MoldeController.deletar`.

Autenticação, autorização e perfis: (hasAuthority('SCREEN_PRODUCTS') or hasAuthority('SCREEN_MOLDS')) and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: 204, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Molde está alocado em uma ou mais posições. Remova-o antes de excluir.

Regras de negócio e dependências: `MoldeService.deletar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/moldes/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

Materiais fixos e slots legados

[Índice de endpoints](#) · [Modelos](#)

API-064 — Listar

GET `/api/nivel-materiais/nivel/{nivelId}`**Finalidade:** Listar.**Fonte:** `src/main/java/com/wester/storage/controller/NivelMaterialController.java:27` — `NivelMaterialController.listar`.**Autenticação, autorização e perfis:** `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).**Headers:** `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.**Parâmetros:**

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.**Resposta de sucesso:** `List<NivelMaterialResponseDTO>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).**Regras de negócio e dependências:** `NivelMaterialService.listarMateriais`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.**cURL (Bash):**

```
curl -X GET "${BASE_URL}/api/nivel-materiais/nivel/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-065 — Alocar Batch

POST `/api/nivel-materiais/nivel/{nivelId}/alocar-batch`

Finalidade: Alocar Batch. Ignora duplicatas por nível/material/tipo. Produto exige quantidade > 0; molde ativo é validado. Os tipos configurados na área não são validados neste método; materialId de produto não recebe a mesma validação de existência aplicada a molde.

Fonte: `src/main/java/com/wester/storage/controller/NivelMaterialController.java:34` — `NivelMaterialController.alocarBatch`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `AlocarMateriaisRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "materiais": [
    {
      "materialId": 1,
      "materialType": "PRODUTO",
      "quantidade": 10,
      "quantidadeAproximada": false
    }
  ]
}
```

json

Resposta de sucesso: `List<NivelMaterialResponseDTO>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Nível desativado não pode receber alocações.
- Molde não encontrado: [mensagem pode receber complemento dinâmico]
- Molde inativo não pode ser alocado: [mensagem pode receber complemento dinâmico]

- Informe a quantidade do produto id=

Regras de negócio e dependências: `NivelMaterialService.alocarMateriais`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/nivel-materiais/nivel/1/alocar-batch" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-066 — Remove

DELETE `/api/nivel-materiais/{nivelMaterialId}`

Finalidade: Remove.

Fonte: `src/main/java/com/wester/storage/controller/NivelMaterialController.java:43` — `NivelMaterialController.remover`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelMaterialId</code>	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `NivelMaterialService.removerMaterial`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/nivel-materiais/1" -H "Authorization: Bearer ${TOKEN}"
```

API-067 — Atualizar Quantidade

PATCH `/api/nivel-materiais/{nivelMaterialId}/quantidade`

Finalidade: Atualizar Quantidade.

Fonte: `src/main/java/com/wester/storage/controller/NivelMaterialController.java:51` —

`NivelMaterialController.atualizarQuantidade`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelMaterialId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `AtualizarQuantidadeNivelMaterialRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "quantidade": 1,
  "quantidadeAproximada": false
}
```

json

Resposta de sucesso: `NivelMaterialResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "nivelId": 1,
  "materialId": 1,
  "materialType": "MOLDE",
  "materialCodigo": "Exemplo",
  "materialDescricao": "Exemplo",
  "materialCategoria": "Exemplo",
  "produtoDescricao": "Exemplo",
  "ordem": 1,
  "quantidade": 1,
  "quantidadeAproximada": false
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Ajuste de quantidade disponível apenas para produtos.
- Quantidade não pode ser negativa.

Regras de negócio e dependências: `NivelMaterialService.atualizarQuantidadeProduto`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/nivel-materiais/1/quantidade" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-068 — Limpar

DELETE `/api/nivel-materiais/nivel/{nivelId}/limpar`

Finalidade: Limpar.

Fonte: `src/main/java/com/wester/storage/controller/NivelMaterialController.java:62` — `NivelMaterialController.limpar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nivelId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `NivelMaterialService.limparNivel`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/nivel-materiais/nivel/1/limpar" -H "Authorization: Bearer ${TOKEN}"
```

API-089 — Listar slots de um nível com seus moldes alocados

GET `/api/slots/nivel/{nivelId}`

Finalidade: Listar slots de um nível com seus moldes alocados.

Fonte: `src/main/java/com/wester/storage/controller/SlotNivelController.java:35` — `SlotNivelController.listarPorNivel`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>nívelId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<SlotNivelResponseDTO>` . Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: `200` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `SlotNivelService.listarSlotsPorNivel` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/slots/nivel/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-090 — Buscar todas as estantes de uma área fixa (batch — substitui N chamadas por fileira)

GET `/api/slots/area/{areaId}`

Finalidade: Buscar todas as estantes de uma área fixa (batch — substitui N chamadas por fileira).

Fonte: `src/main/java/com/wester/storage/controller/SlotNivelController.java:41` — `SlotNivelController.buscarPorArea` .

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<EstanteFixaResponseDTO>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `SlotNivelService.buscarEstantesPorArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/slots/area/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-091 — Buscar estrutura completa de uma estante (fileira fixa) com níveis e slots

GET `/api/slots/estante/{fileiraId}`

Finalidade: Buscar estrutura completa de uma estante (fileira fixa) com níveis e slots.

Fonte: `src/main/java/com/wester/storage/controller/SlotNivelController.java:47` — `SlotNivelController.buscarEstante`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>fileiraId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `EstanteFixaResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "fileiraId": 1,
  "fileiraIdentificador": "Exemplo",
  "fileiraOrdem": 1,
  "grades": []
}
```

json

Códigos HTTP: 200, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: SlotNivelService.buscarEstanteComSlots. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/slots/estante/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-092 — Inicializar slots físicos de um nível (cria N posições; idempotente)

POST /api/slots/nivel/{nivelId}/inicializar

Finalidade: Inicializar slots físicos de um nível (cria N posições; idempotente).

Fonte: src/main/java/com/wester/storage/controller/SlotNivelController.java:59 — SlotNivelController.inicializarSlots.

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: Authorization: Bearer <TOKEN>. Accept: application/json recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
nivelId	path	Long	Sim	—
quantidade	query	int	Não	2

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: List<SlotNivelResponseDTO>. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: 201, 400, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Validações e erros identificados na cadeia local do serviço:

- A quantidade de slots deve ser entre 1 e 10.

Regras de negócio e dependências: SlotNivelService.inicializarSlots. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/slots/nivel/1/inicializar" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-093 — Adicionar um slot individual em um nível fixo

POST /api/slots/nivel/{nivelId}

Finalidade: Adicionar um slot individual em um nível fixo.

Fonte: src/main/java/com/wester/storage/controller/SlotNivelController.java:76 — SlotNivelController.adicionarSlot.

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: Authorization: Bearer <TOKEN>. Accept: application/json recomendado. Content-Type: application/json para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
nivelId	path	Long	Sim	—

Corpo da requisição: SlotNivelRequestDTO; opcional. Validação Jakarta ativada por @Valid. Consulte os campos e restrições em Modelos.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "posicao": 1,
  "identificador": "Exemplo"
}
```

json

Resposta de sucesso: SlotNivelResponseDTO. Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "nivelId": 1,
  "posicao": 1,
  "identificador": "Exemplo",
  "moldesAlocados": []
}
```

Códigos HTTP: 201, 400, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Identificador do slot é obrigatório.
- Posição de slot já existe neste nível: [mensagem pode receber complemento dinâmico]
- Identificador de slot já existe neste nível: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: SlotNivelService.adicionarSlot. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/slots/nivel/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-094 — Editar um slot fixo

PATCH /api/slots/{slotId}

Finalidade: Editar um slot fixo.

Fonte: src/main/java/com/wester/storage/controller/SlotNivelController.java:96 — SlotNivelController.atualizarSlot.

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: Authorization: Bearer <TOKEN>. Accept: application/json recomendado. Content-Type: application/json para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
slotId	path	Long	Sim	—

Corpo da requisição: SlotNivelRequestDTO; obrigatório. Validação Jakarta ativada por @Valid. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

json

```
{
  "posicao": 1,
  "identificador": "Exemplo"
}
```

Resposta de sucesso: `SlotNivelResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "id": 1,
  "nivelId": 1,
  "posicao": 1,
  "identificador": "Exemplo",
  "moldesAlocados": []
}
```

Códigos HTTP: `200`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Identificador do slot é obrigatório.
- Posição de slot já existe neste nível: [mensagem pode receber complemento dinâmico]
- Identificador de slot já existe neste nível: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `SlotNivelService.atualizarSlot`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PATCH "${BASE_URL}/api/slots/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-095 — Remover um slot fixo vazio

DELETE `/api/slots/{slotId}`

Finalidade: Remover um slot fixo vazio.

Fonte: `src/main/java/com/wester/storage/controller/SlotNivelController.java:112` — `SlotNivelController.removerSlot`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>slotId</code>	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Não é possível remover um slot ocupado.

Regras de negócio e dependências: `SlotNivelService.removerSlot`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/slots/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-096 — Alocar um molde em um slot

POST `/api/slots/{slotId}/alocar`

Finalidade: Alocar um molde em um slot.

Fonte: `src/main/java/com/wester/storage/controller/SlotNivelController.java:126` — `SlotNivelController.alocar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>slotId</code>	path	Long	Sim	—

Corpo da requisição: `AlocarMoldeRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "moldeId": 1,
  "ordem": 1
}
```

json

Resposta de sucesso: `SlotNivelResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "nivelId": 1,
  "posicao": 1,
  "identificador": "Exemplo",
  "moldesAlocados": []
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`, `404`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- O molde está inativo e não pode ser alocado.
- Este molde já está alocado neste slot.

Regras de negócio e dependências: `SlotNivelService.alocarMolde`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/slots/1/alocar" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicatio
```

bash

API-097 — Remover um molde específico de um slot

DELETE `/api/slots/{slotId}/moldes/{moldeAlocadoId}`

Finalidade: Remover um molde específico de um slot.

Fonte: `src/main/java/com/wester/storage/controller/SlotNivelController.java:143` — `SlotNivelController.removerMolde`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
slotId	path	Long	Sim	—
moldeAlocadoId	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: 204, 400, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- A alocação não pertence ao slot informado.

Regras de negócio e dependências: SlotNivelService.removerMolde. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/slots/1/molde/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-098 — Limpar todos os moldes de um slot

DELETE /api/slots/{slotId}/limpar

Finalidade: Limpar todos os moldes de um slot.

Fonte: src/main/java/com/wester/storage/controller/SlotNivelController.java:160 — SlotNivelController.limparSlot.

Autenticação, autorização e perfis: hasAuthority('SCREEN_WAREHOUSE') and hasRole('FULL_ACCESS') Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: Authorization: Bearer <TOKEN>. Accept: application/json recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
slotId	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: 204, 401, 403, 404. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `SlotNivelService.limparSlot`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/slots/1/limpar" -H "Authorization: Bearer ${TOKEN}"
```

bash

Listas de materiais

[Índice de endpoints](#) · [Modelos](#)

API-099 — Listar tabelas por área

GET `/api/encerto`**Finalidade:** Listar tabelas por área.**Fonte:** `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:45` — `TabelaEncertoController.listarPorArea`.**Autenticação, autorização e perfis:** `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).**Headers:** `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.**Parâmetros:**

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	query	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.**Resposta de sucesso:** `List<TabelaEncertoSummaryDTO>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).**Regras de negócio e dependências:** `TabelaEncertoService.listarPorArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.**cURL (Bash):**

```
curl -X GET "${BASE_URL}/api/encerto?areaId=1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-100 — Buscar tabela com itens

GET `/api/encerto/{id}`**Finalidade:** Buscar tabela com itens.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:53` —
`TabelaEncertoController.buscarPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `TabelaEncertoDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "areaId": 1,
  "titulo": "Exemplo",
  "colunas": [],
  "itens": []
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `TabelaEncertoService.buscarPorId`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/encerto/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-101 — Criar nova tabela

POST `/api/encerto`

Finalidade: Criar nova tabela.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:60` —
`TabelaEncertoController.criar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `CriarTabelaEncertoRequestDTO` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "areaId": 1,
  "titulo": "Lista de exemplo",
  "colunas": [
    "codigo",
    "produto",
    "quantidade",
    "observacao"
  ]
}
```

json

Resposta de sucesso: `TabelaEncertoDTO` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "areaId": 1,
  "titulo": "Exemplo",
  "colunas": [],
  "itens": []
}
```

json

Códigos HTTP: `201` , `400` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `TabelaEncertoService.criar` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/encerto" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

bash

API-102 — Atualizar título ou colunas de uma tabela

PATCH `/api/encerto/{id}`

Finalidade: Atualizar título ou colunas de uma tabela.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:67` — `TabelaEncertoController.atualizar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	<code>path</code>	<code>Long</code>	Sim	—

Corpo da requisição: `AtualizarTabelaEncertoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "titulo": "Lista revisada"
}
```

json

Resposta de sucesso: `TabelaEncertoDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "areaId": 1,
  "titulo": "Exemplo",
  "colunas": [],
  "itens": []
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `TabelaEncertoService.atualizar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/encerto/1" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/js
```

bash

API-103 — Excluir tabela

DELETE /api/encerto/{id}

Finalidade: Excluir tabela.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:76` — `TabelaEncertoController.excluir`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	Long	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `TabelaEncertoService.excluir`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/encerto/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-104 — Adicionar item à tabela

POST /api/encerto/{tabelaId}/itens

Finalidade: Adicionar item à tabela.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:86` — `TabelaEncertoController.adicionarItem`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
tabelaId	path	Long	Sim	—

Corpo da requisição: `AdicionarItemEncertoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produtoId": 1,
  "quantidade": 1,
  "observacao": "Exemplo"
}
```

json

Resposta de sucesso: `TabelaEncertoItemDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "produtoId": 1,
  "codigo": "Exemplo",
  "nome": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "ordem": 1
}
```

json

Códigos HTTP: `201`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Este produto já está nesta lista.

Regras de negócio e dependências: `TabelaEncertoService.adicionarItem`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/encerto/1/itens" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicati
```

bash

API-105 — Remover item da tabela

DELETE `/api/encerto/itens/{itemId}`

Finalidade: Remover item da tabela.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:95` — `TabelaEncertoController.removerItem`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>itemId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `TabelaEncertoService.removerItem`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/encerto/itens/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-106 — Atualizar quantidade de um item (sincroniza com item_estoque)

PATCH `/api/encerto/itens/{itemId}/quantidade`

Finalidade: Atualizar quantidade de um item (sincroniza com item_estoque). Atualiza a quantidade no item da lista. A anotação menciona sincronização com item_estoque, mas a implementação atual não altera esse estoque: ver lacunas.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:103` — `TabelaEncertoController.atualizarQuantidade`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
itemId	path	Long	Sim	—

Corpo da requisição: `AtualizarQuantidadeEncertoRequestDTO`; obrigatório. Validação Jakarta ativada por `@Valid`. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "quantidade": 1
}
```

json

Resposta de sucesso: `TabelaEncertoItemDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "produtoId": 1,
  "codigo": "Exemplo",
  "nome": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "ordem": 1
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `TabelaEncertoService.atualizarQuantidade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/encerto/itens/1/quantidade" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

bash

API-107 — Atualizar observação de um item (texto livre, até 100 chars)

PATCH `/api/encerto/itens/{itemId}/observacao`

Finalidade: Atualizar observação de um item (texto livre, até 100 chars).

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:112` — `TabelaEncertoController.atualizarObservacao`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>itemId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `AtualizarObservacaoEncertoRequestDTO` ; obrigatório. Validação Jakarta ativada por `@Valid` . Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "observacao": "Exemplo"
}
```

Resposta de sucesso: `TabelaEncertoItemDTO` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "produtoId": 1,
  "codigo": "Exemplo",
  "nome": "Exemplo",
  "cor": "Exemplo",
  "descricao": "Exemplo",
  "observacao": "Exemplo",
  "quantidade": 1,
  "ordem": 1
}
```

Códigos HTTP: `200` , `400` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `TabelaEncertoService.atualizarObservacao` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/encerto/itens/1/observacao" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-108 — Reordenar item na tabela

PATCH `/api/encerto/itens/{itemId}/ordem`

Finalidade: Reordenar item na tabela.

Fonte: `src/main/java/com/wester/storage/controller/TabelaEncertoController.java:121` —

`TabelaEncertoController.reordenar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>itemId</code>	path	<code>Long</code>	Sim	—

Corpo da requisição: `int`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. O corpo é um número inteiro simples (a nova ordem), não um objeto.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

1

json

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `TabelaEncertoService.reordenar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PATCH "${BASE_URL}/api/encerto/itens/1/ordem" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: ap
```

bash

Consultas de estoque e mapa

Índice de endpoints · Modelos

API-021 — Listar Itens Dashboard

GET /api/estoque/itens

Finalidade: Listar Itens Dashboard.

Fonte: `src/main/java/com/wester/storage/controller/EstoqueController.java:51` — `EstoqueController.listarItensDashboard`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	10
<code>areaId</code>	query	<code>Long</code>	Não	—
<code>filtro</code>	query	<code>String</code>	Não	—
<code>sort</code>	query	<code>String</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `EstoqueItemDashboardPageResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "content": [],
  "page": 0,
  "size": 1,
  "totalElements": 1,
  "totalPages": 0
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `EstoqueService.listarItensDashboard`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/estoque/itens" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-022 — Listar Itens Consolidado

GET `/api/estoque/itens-consolidado`

Finalidade: Listar Itens Consolidado.

Fonte: `src/main/java/com/wester/storage/controller/EstoqueController.java:81` — `EstoqueController.listarItensConsolidado`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_DASHBOARD')` or `hasAuthority('SCREEN_WAREHOUSE')`
Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	10
<code>areaId</code>	query	<code>Long</code>	Não	—
<code>filtro</code>	query	<code>String</code>	Não	—
<code>sort</code>	query	<code>String</code>	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `EstoqueItemConsolidadoPageResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "content": [],
  "page": 0,
  "size": 1,
  "totalElements": 1,
  "totalPages": 0
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `EstoqueConsolidadoService.listarItensConsolidado`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/estoque/itens-consolidado" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-023 — Retorna o mapa completo do estoque por área

GET `/api/estoque/posicoes/area/{areaId}`

Finalidade: Retorna o mapa completo do estoque por área.

Fonte: `src/main/java/com/wester/storage/controller/EstoqueController.java:102` — `EstoqueController.buscarMapaPorArea`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_WAREHOUSE')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>areaId</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<EstoquePosicaoDTO>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `EstoqueService.buscarMapaPorArea`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/estoque/posicoes/area/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

Histórico e auditoria

[Índice de endpoints](#) · [Modelos](#)

API-036 — Listar histórico paginado

GET `/api/historico-movimentacao`**Finalidade:** Listar histórico paginado.**Fonte:** `src/main/java/com/wester/storage/controller/HistoricoMovimentacaoController.java:43` — `HistoricoMovimentacaoController.listar`.**Autenticação, autorização e perfis:** `hasAuthority('SCREEN_HISTORY')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).**Headers:** `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.**Parâmetros:**

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	20

Corpo: não declarado; não enviar corpo.**Resposta de sucesso:** `Page<HistoricoMovimentacaoResponseDTO>` . Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "content": [],
  "totalElements": 1,
  "totalPages": 0,
  "size": 1,
  "number": 1,
  "first": false,
  "last": false,
  "empty": false,
  "numberOfElements": 1,
  "pageable": {},
  "sort": {}
}
```

json

Códigos HTTP: `200` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `HistoricoMovimentacaoService.listar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/historico-movimentacao" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-037 — Filtrar histórico

POST `/api/historico-movimentacao/filter`

Finalidade: Filtrar histórico.

Fonte: `src/main/java/com/wester/storage/controller/HistoricoMovimentacaoController.java:68` — `HistoricoMovimentacaoController.filtrar`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_HISTORY')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	20

Corpo da requisição: `HistoricoMovimentacaoFilterRequestDTO`; opcional. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "dataInicio": "2026-10-01",
  "dataFim": "2026-10-15",
  "tipoOperacao": "ENTRADA"
}
```

json

Resposta de sucesso: `Page<HistoricoMovimentacaoResponseDTO>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "content": [],
  "totalElements": 1,
  "totalPages": 0,
  "size": 1,
  "number": 1,
  "first": false,
  "last": false,
  "empty": false,
  "numberOfElements": 1,
  "pageable": {},
  "sort": {}
}
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Data inválida: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `HistoricoMovimentacaoService.filtrar`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/historico-movimentacao/filter" -H "Authorization: Bearer ${TOKEN}" -H 'Content-T
```

API-038 — Buscar evento de histórico por ID

GET `/api/historico-movimentacao/{id}`

Finalidade: Buscar evento de histórico por ID.

Fonte: `src/main/java/com/wester/storage/controller/HistoricoMovimentacaoController.java:85` — `HistoricoMovimentacaoController.buscarPorId`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_HISTORY')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>Long</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `HistoricoMovimentacaoResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "id": 1,
  "timestamp": "2026-10-15T09:00:00",
  "tipoOperacao": "Exemplo",
  "usuarioId": 1,
  "usuarioLogin": "Exemplo",
  "usuarioNome": "Exemplo",
  "produtoId": 1,
  "produtoNomeModelo": "Exemplo",
  "produtoDescricao": "Exemplo",
  "produtoCor": "Exemplo",
  "produtoCodigoSistemaWester": "Exemplo",
  "itemEstoqueId": 1,
  "areaId": 1,
  "areaNome": "Exemplo",
  "setorId": 1,
  "setorNome": "Exemplo",
  "tipoEstruturaArea": "Exemplo",
  "labelEstruturaPrimaria": "Exemplo",
  "labelEstruturaSecundaria": "Exemplo",
  "labelFileira": "Exemplo",
  "labelGrade": "Exemplo",
  "labelNivel": "Exemplo",
  "labelSlot": "Exemplo",
  "labelPosicao": "Exemplo",
  "fileiraId": 1,
  "fileiraIdentificador": "Exemplo",
  "gradeId": 1,
  "gradeIdentificador": "Exemplo",
  "nivelId": 1,
  "nivelIdentificador": "Exemplo",
  "nivelOrigemId": 1,
  "nivelOrigemIdentificador": "Exemplo",
  "nivelDestinoId": 1,
  "nivelDestinoIdentificador": "Exemplo",
  "slotId": 1,
  "slotIdentificador": "Exemplo",
  "slotPosicao": 1,
  "posicao": "Exemplo",
  "moldeId": 1,
  "moldeCodigo": "Exemplo",
  "entidade": "Exemplo",
  "entidadeId": 1,
  "campo": "Exemplo",
  "valorAnterior": "Exemplo",
  "valorNovo": "Exemplo",
  "payloadJson": "Exemplo",
  "quantidadeAlterada": 1,
  "quantidadeAnterior": 1,
  "quantidadeNova": 1,
  "detalhesAlteracao": "Exemplo"
}
```

Códigos HTTP: 200 , 401 , 403 . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em Erros.

Regras de negócio e dependências: HistoricoMovimentacaoService.buscarPorId . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/historico-movimentacao/1" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-039 — Histórico de uma entidade específica

GET /api/historico-movimentacao/entidade/{entidade}/{entidadeId}

Finalidade: Histórico de uma entidade específica.

Fonte: src/main/java/com/wester/storage/controller/HistoricoMovimentacaoController.java:106 — HistoricoMovimentacaoController.listarPorEntidade .

Autenticação, autorização e perfis: hasAuthority('SCREEN_HISTORY') or hasAuthority('SCREEN_USERS') or hasAuthority('SCREEN_PROFILES') or hasAuthority('SCREEN_PRODUCTS') Consulte a correspondência entre autoridades e perfis em Autenticação.

Headers: Authorization: Bearer <TOKEN> . Accept: application/json recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
entidade	path	String	Sim	—
entidadeId	path	Long	Sim	—
page	query	int	Não	0
size	query	int	Não	20

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: Page<HistoricoMovimentacaoResponseDTO> . Campos em Modelos.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "content": [],
  "totalElements": 1,
  "totalPages": 0,
  "size": 1,
  "number": 1,
  "first": false,
  "last": false,
  "empty": false,
  "numberOfElements": 1,
  "pageable": {},
  "sort": {}
}
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `HistoricoMovimentacaoService.listarPorEntidade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/historico-movimentacao/entidade/1/1" -H "Authorization: Bearer ${TOKEN}"
```

Movimentações operacionais

[Índice de endpoints](#) · [Modelos](#)

API-052 — Carregar contexto inicial da tela de movimentacoes

GET `/api/movimentacoes/contexto-inicial`

Finalidade: Carregar contexto inicial da tela de movimentacoes.

Fonte: `src/main/java/com/wester/storage/controller/MovimentacaoEstoqueController.java:44` — `MovimentacaoEstoqueController.buscarContextoInicial`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_MOVEMENTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`, `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MovimentacaoEstoqueContextoInicialDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "produtos": [],
  "areas": [],
  "movimentacoesRecentes": []
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `MovimentacaoEstoqueService.buscarContextoInicial`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/movimentacoes/contexto-inicial" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-053 — Registrar movimentação operacional de estoque

POST `/api/movimentacoes`

Finalidade: Registrar movimentação operacional de estoque.

Fonte: `src/main/java/com/wester/storage/controller/MovimentacaoEstoqueController.java:56` —
`MovimentacaoEstoqueController.registrarMovimentacao`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_MOVEMENTS')` and `hasRole('FULL_ACCESS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `MovimentacaoEstoqueRequestDTO`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "tipoMovimentacao": "ENTRADA",
  "produtoId": 1,
  "nivelId": 1,
  "quantidade": 10,
  "motivo": "Recebimento",
  "observacao": "Exemplo documental"
}
```

json

Resposta de sucesso: `MovimentacaoEstoqueResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "historicoMovimentacaoId": 1,
  "registradoEm": "Exemplo",
  "tipoMovimentacao": "Exemplo",
  "produtoId": 1,
  "produtoNomeModelo": "Exemplo",
  "produtoCodigoSistemaWester": "Exemplo",
  "itemEstoqueId": 1,
  "nivelId": 1,
  "nivelOrigemId": 1,
  "nivelDestinoId": 1,
  "nivelLabel": "Exemplo",
  "nivelOrigemLabel": "Exemplo",
  "nivelDestinoLabel": "Exemplo",
  "quantidadeMovimentada": 1,
  "quantidadeAnterior": 1,
  "quantidadeNova": 1,
  "motivo": "Exemplo",
  "observacao": "Exemplo",
  "detalhesAlteracao": "Exemplo"
}
```

Códigos HTTP: 201, 401, 403. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Payload da movimentação é obrigatório.
- Tipo de movimentação não suportado.
- Tipo de movimentação é obrigatório.
- Tipo de movimentação não suportado: [mensagem pode receber complemento dinâmico]
- O nível de destino já possui outro produto.
- Produto é obrigatório para esta movimentação.
- O produto informado não corresponde ao item existente no nível.
- A quantidade solicitada é maior que o saldo disponível.
- Origem e destino devem ser níveis diferentes.
- O produto informado não corresponde ao item existente na origem.
- A quantidade solicitada é maior que o saldo disponível na origem.
- O ajuste não pode resultar em quantidade negativa.
- Não é permitido trocar o produto durante um ajuste.
- Motivo do ajuste é obrigatório.
- Informe somente quantidade ajustada ou quantidade diferença para o ajuste.
- Não é permitido informar produto diferente do item já existente.
- Não existe item no nível informado para zerar via ajuste.

Regras de negócio e dependências: `MovimentacaoEstoqueService.registrarMovimentacao`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/movimentacoes" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-054 — Listar movimentacoes recentes

GET `/api/movimentacoes/recentes`

Finalidade: Listar movimentacoes recentes.

Fonte: `src/main/java/com/wester/storage/controller/MovimentacaoEstoqueController.java:70` — `MovimentacaoEstoqueController.listarRecentes`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_MOVEMENTS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>limite</code>	query	<code>Integer</code>	Não	10

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<MovimentacaoEstoqueRecenteDTO>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[]
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `MovimentacaoEstoqueService.listarRecentes`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/movimentacoes/recentes" -H "Authorization: Bearer ${TOKEN}"
```

Alertas de estoque

[Índice de endpoints](#) · [Modelos](#)

API-001 — Consultar alertas de estoque

POST `/api/alertas-estoque`

Finalidade: Consultar alertas de estoque.

Fonte: `src/main/java/com/wester/storage/controller/AlertaEstoqueController.java:37` —
`AlertaEstoqueController.consultarAlertas`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_ALERTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `AlertaEstoqueFiltroDTO`; opcional. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "tipoAlerta": "Exemplo",
  "produtoId": 1,
  "textoLivre": "Exemplo",
  "codigoWester": "Exemplo",
  "areaId": 1,
  "criticidade": "Exemplo",
  "produtoAtivo": false,
  "diasSemMovimentacao": 1,
  "ordenarPor": "Exemplo",
  "direcaoOrdenacao": "Exemplo",
  "pagina": 1,
  "tamanhoPagina": 1
}
```

json

Resposta de sucesso: `AlertaEstoqueResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "itens": [],
  "pagina": 1,
  "tamanhoPagina": 1,
  "totalItens": 1,
  "totalPaginas": 1,
  "resumo": {
    "totalAlertas": 1,
    "produtosSemEstoque": 1,
    "produtosAbaixoMinimo": 1,
    "produtosAcimaMaximo": 1,
    "produtosSemMovimentacao": 1,
    "criticidadeAlta": 1,
    "criticidadeMedia": 1,
    "criticidadeBaixa": 1
  }
}
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `AlertaEstoqueService.consultarAlertas`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/alertas-estoque" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applicati
```

Relatórios de estoque

[Índice de endpoints](#) · [Modelos](#)

API-085 — Gerar relatório de produtos

POST `/api/relatorios/produtos`

Finalidade: Gerar relatório de produtos.

Fonte: `src/main/java/com/wester/storage/controller/RelatorioController.java:47` — `RelatorioController.gerarRelatorioProdutos`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_REPORTS')` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `RelatorioProdutosFiltroDTO`; opcional. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "nome": "Exemplo",
  "codigoSistemaWester": "Exemplo",
  "descricao": "Exemplo",
  "cor": "Exemplo",
  "ativo": false,
  "categoria": "Exemplo",
  "sortBy": "Exemplo",
  "sortDirection": "Exemplo",
  "page": 0,
  "size": 1
}
```

json

Resposta de sucesso: `RelatorioProdutosResponseDTO`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "generatedAt": "Exemplo",
  "summary": {
    "totalProdutos": 1,
    "totalProdutosAtivos": 1,
    "totalProdutosInativos": 1,
    "quantidadeTotalEmEstoque": 1,
    "quantidadeTotalEmEstoqueAproximada": false
  },
  "pagination": {
    "page": 0,
    "size": 1,
    "totalItems": 1,
    "totalPages": 0
  },
  "items": [],
  "statusChart": [],
  "colorChart": [],
  "usageChart": [],
  "categoryTotals": []
}
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Categoria inválida: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `RelatorioService.gerarRelatorioProdutos`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/relatorios/produtos" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: appli
```

API-086 — Gerar relatório de estoques

POST `/api/relatorios/estoques`

Finalidade: Gerar relatório de estoques.

Fonte: `src/main/java/com/wester/storage/controller/RelatorioController.java:63` — `RelatorioController.gerarRelatorioEstoques`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_REPORTS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `RelatorioEstoquesFiltroDTO` ; opcional. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "area": "Exemplo",
  "fileira": "Exemplo",
  "grade": "Exemplo",
  "nivel": "Exemplo",
  "ativo": false,
  "sortBy": "Exemplo",
  "sortDirection": "Exemplo",
  "page": 0,
  "size": 1
}
```

json

Resposta de sucesso: `RelatorioEstoquesResponseDTO` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "generatedAt": "Exemplo",
  "summary": {
    "totalAreas": 1,
    "totalAreasAtivas": 1,
    "totalFileiras": 1,
    "totalGrades": 1,
    "totalNiveis": 1,
    "totalItensEstoque": 1,
    "quantidadeTotalArmazenada": 1,
    "quantidadeTotalArmazenadaAproximada": false
  },
  "pagination": {
    "page": 0,
    "size": 1,
    "totalItems": 1,
    "totalPages": 0
  },
  "items": [],
  "quantityByAreaChart": [],
  "occupancyByAreaChart": [],
  "quantityByFileiraChart": []
}
```

json

Códigos HTTP: `200` , `401` , `403` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Regras de negócio e dependências: `RelatorioService.gerarRelatorioEstoques`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/relatorios/estoques" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: appli
```

bash

API-087 — Gerar relatorio de itens de estoque

POST `/api/relatorios/itens-estoque`

Finalidade: Gerar relatorio de itens de estoque.

Fonte: `src/main/java/com/wester/storage/controller/RelatorioController.java:79` — `RelatorioController.gerarRelatorioItensEstoque`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_REPORTS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `RelatorioItensEstoqueFiltroDTO`; opcional. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produto": "Exemplo",
  "codigoSistemaWester": "Exemplo",
  "area": "Exemplo",
  "fileira": "Exemplo",
  "grade": "Exemplo",
  "nivel": "Exemplo",
  "statusItem": "Exemplo",
  "dataInicio": "2026-10-15T09:00:00",
  "dataFim": "2026-10-15T09:00:00",
  "sortBy": "Exemplo",
  "sortDirection": "Exemplo",
  "page": 0,
  "size": 1
}
```

json

Resposta de sucesso: `RelatorioItensEstoqueResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "generatedAt": "Exemplo",
  "summary": {
    "totalItensEstoque": 1,
    "quantidadeTotalArmazenada": 1,
    "quantidadeTotalArmazenadaAproximada": false,
    "totalProdutosUnicos": 1,
    "totalAreasComItens": 1
  },
  "pagination": {
    "page": 0,
    "size": 1,
    "totalItems": 1,
    "totalPages": 0
  },
  "items": [],
  "topProductsChart": [],
  "quantityByAreaChart": [],
  "quantityByLocationChart": []
}
```

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Data inválida: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `RelatorioService.gerarRelatorioItensEstoque`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/relatorios/itens-estoque" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type:
```

API-088 — Gerar relatório de histórico de movimentação

POST `/api/relatorios/historico-movimentacao`

Finalidade: Gerar relatório de histórico de movimentação.

Fonte: `src/main/java/com/wester/storage/controller/RelatorioController.java:95` — `RelatorioController.gerarRelatorioHistoricoMovimentacao`.

Autenticação, autorização e perfis: `hasAuthority('SCREEN_REPORTS')` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `RelatorioHistoricoMovimentacaoFiltroDTO`; opcional. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "produto": "Exemplo",
  "codigoSistemaWester": "Exemplo",
  "area": "Exemplo",
  "usuario": "Exemplo",
  "tipoOperacao": "Exemplo",
  "dataInicio": "2026-10-15T09:00:00",
  "dataFim": "2026-10-15T09:00:00",
  "sortDirection": "Exemplo",
  "page": 0,
  "size": 1
}
```

json

Resposta de sucesso: `RelatorioHistoricoMovimentacaoResponseDTO`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "generatedAt": "Exemplo",
  "summary": {
    "totalMovimentacoes": 1,
    "totalEntradas": 1,
    "totalSaidas": 1,
    "totalAjustes": 1,
    "totalMovimentacoesInternas": 1,
    "totalResequenciamentos": 1,
    "quantidadeMovimentadaTotal": 1
  },
  "pagination": {
    "page": 0,
    "size": 1,
    "totalItems": 1,
    "totalPages": 0
  },
  "items": [],
  "typeChart": [],
  "periodChart": [],
  "productChart": [],
  "areaChart": [],
  "userChart": []
}
```

json

Códigos HTTP: `200`, `401`, `403`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Data inválida: [mensagem pode receber complemento dinâmico]

Regras de negócio e dependências: `RelatorioService.gerarRelatorioHistoricoMovimentacao`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/relatorios/historico-movimentacao" -H "Authorization: Bearer ${TOKEN}" -H 'Conte bash
```

PCP e produção

[Índice de endpoints](#) · [Modelos](#)

API-117 — Consultar permissões de produção

GET `/api/mes/permissions`

Finalidade: Consultar permissões de produção.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:19` — `MesController.permissions`.

Autenticação, autorização e perfis: Usuário autenticado; sem autoridade adicional declarada no método. Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesPermissions`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "read": false,
  "operate": false,
  "maintain": false,
  "plan": false,
  "admin": false
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: `MesAccess.permissions`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/permissions" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-118 — Listar injetoras

GET /api/mes/machines

Finalidade: Listar injetoras.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:20` — `MesController.machines`.

Autenticação, autorização e perfis: @mesAccess.read(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `List<MesMachine>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
[ ]
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- JSON persistido invalido

Regras de negócio e dependências: `MesService.machines`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/machines" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-119 — Cadastrar injetora

POST /api/mes/machines

Finalidade: Cadastrar injetora.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:21` — `MesController.createMachine`.

Autenticação, autorização e perfis: @mesAccess.admin(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `MesMachineInput` ; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "machineId": "DOC.01",
  "name": "Injetora de exemplo",
  "numero": "01",
  "marca": "Exemplo",
  "modelo": "Exemplo",
  "ativo": true,
  "testYellow": false,
  "testRed": false
}
```

json

Resposta de sucesso: `MesMachine` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "machineId": "Exemplo",
  "name": "Exemplo",
  "numero": "Exemplo",
  "marca": "Exemplo",
  "modelo": "Exemplo",
  "nozzleMode": "ALWAYS_CONTACT",
  "ativo": false,
  "testYellow": false,
  "testRed": false,
  "revision": 1,
  "activeExecutionId": "00000000-0000-4000-8000-000000000001",
  "mechanicalFault": false
}
```

json

Códigos HTTP: `200` , `400` , `401` , `403` , `404` , `409` . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- machineId invalido
- Maquina existente
- machineId imutavel

- Finalize a execucao antes de alterar instrumentacao
- JSON persistido invalido

Regras de negócio e dependências: `MesService.saveMachine` . Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/machines" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/json'
```

API-120 — Atualizar injetora

PUT `/api/mes/machines/{id}`

Finalidade: Atualizar injetora.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:22` — `MesController.machine` .

Autenticação, autorização e perfis: @mesAccess.admin(authentication) Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesMachineInput` ; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "machineId": "DOC.01",
  "name": "Injetora de exemplo",
  "numero": "01",
  "marca": "Exemplo",
  "modelo": "Exemplo",
  "ativo": true,
  "testYellow": false,
  "testRed": false
}
```

Resposta de sucesso: `MesMachine` . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "machineId": "Exemplo",
  "name": "Exemplo",
  "numero": "Exemplo",
  "marca": "Exemplo",
  "modelo": "Exemplo",
  "nozzleMode": "ALWAYS_CONTACT",
  "ativo": false,
  "testYellow": false,
  "testRed": false,
  "revision": 1,
  "activeExecutionId": "00000000-0000-4000-8000-000000000001",
  "mechanicalFault": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- machineId inválido
- Máquina existente
- machineId imutável
- Finalize a execução antes de alterar instrumentação
- JSON persistido inválido

Regras de negócio e dependências: `MesService.saveMachine`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/mes/machines/DOC.01" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: applic
```

API-121 — Consultar estado da injetora

GET `/api/mes/machines/{id}/status`

Finalidade: Consultar estado da injetora.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:23` — `MesController.status`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	String	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesStatus`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```

{
  "machine": {
    "machineId": "Exemplo",
    "name": "Exemplo",
    "numero": "Exemplo",
    "marca": "Exemplo",
    "modelo": "Exemplo",
    "nozzleMode": "ALWAYS_CONTACT",
    "ativo": false,
    "testYellow": false,
    "testRed": false,
    "revision": 1,
    "activeExecutionId": "00000000-0000-4000-8000-000000000001",
    "mechanicalFault": false
  },
  "execution": {
    "executionId": "00000000-0000-4000-8000-000000000001",
    "orderId": "00000000-0000-4000-8000-000000000001",
    "machineId": "Exemplo",
    "contextRevision": 1,
    "status": "Exemplo",
    "preparedAt": "2026-10-15T12:00:00Z",
    "startedAt": "2026-10-15T12:00:00Z",
    "finishedAt": "2026-10-15T12:00:00Z",
    "author": "Exemplo",
    "cancelReason": "Exemplo",
    "orderCode": "Exemplo",
    "productCode": "Exemplo",
    "productName": "Exemplo",
    "toolCode": "Exemplo",
    "material": "Exemplo",
    "quantityPieces": 1,
    "piecesPerCycle": 1,
    "idealCycleMs": 1,
    "notes": "Exemplo",
    "toolSettings": {
      "lot": "Exemplo",
      "moldName": "Exemplo",
      "nozzleMode": "Exemplo",
      "pieceWeightGrams": 1,
      "netWeightGrams": 1,
      "grossWeightGrams": 1,
      "sprueWeightGrams": 1,
      "cyclesPerKit": 1
    },
    "insumos": "Exemplo",
    "goodPieces": 1,
    "targetReached": false
  },
  "telemetry": {},
  "telemetryAt": "2026-10-15T12:00:00Z",
  "hasCompletedCycle": false,
  "cycleState": "Exemplo",
  "cycleClassification": "Exemplo",

```

```

"validationRequired": false,
"validationReason": "Exemplo",
"lastSequence": 1,
"currentCycleElapsedMs": 1,
"lastCycleMs": 1,
"desiredLamp": "Exemplo",
"productionBlocked": false,
"communication": "NO_DATA",
"metrics": {
  "grossPieces": 1,
  "cycles": 1,
  "validCycles": 1,
  "averageCycleMs": 1,
  "averageInjectionMs": 1,
  "averageMoldClosedMs": 1,
  "averageMoldNotClosedMs": 1,
  "averageNozzleContactMs": 1,
  "averageNozzleAwayMs": 1,
  "averageDosingMs": 1,
  "secondQuality": 1,
  "reject": 1,
  "residueGrams": 1,
  "goodPieces": 1,
  "qualityBasis": "Exemplo",
  "projection": "Exemplo"
},
"edgeHealth": {},
"collectorHealth": {},
"ingestionHealth": {},
"nozzleModeInferred": {}
}

```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.status`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/machines/DOC.01/status" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-122 — Listar ordens de serviço

GET `/api/mes/orders`

Finalidade: Listar ordens de serviço.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:24` — `MesController.orders`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>machineId</code>	query	<code>String</code>	Não	—
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	25

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesPage<MesOrder>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "content": [],
  "page": 0,
  "size": 1,
  "totalElements": 1,
  "totalPages": 0
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- JSON persistido invalido
- UUID invalido

Regras de negócio e dependências: `MesService.orders`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/orders" -H "Authorization: Bearer ${TOKEN}"
```

bash

API-123 — Cadastrar ordem de serviço

POST `/api/mes/orders`

Finalidade: Cadastrar ordem de serviço.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:25` — `MesController.createOrder`.

Autenticação, autorização e perfis: `@mesAccess.plan(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `MesOrderInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "orderCode": "OS-DOC-001",
  "productCode": "DOC-001",
  "productName": "Produto de exemplo",
  "quantityPieces": 100,
  "piecesPerCycle": 2,
  "idealCycleMs": 30000,
  "toolCode": "M-DOC-001",
  "toolSettings": {
    "lot": "LOTE-DOC-001"
  },
  "notes": "Exemplo documental"
}
```

json

Resposta de sucesso: `MesOrder`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "status": "Exemplo",
  "queuePosition": 1,
  "createdAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "insumos": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- OS existente na maquina
- JSON persistido invalido
- toolSettings deve ser objeto
- Parametro de ferramenta desconhecido: [mensagem pode receber complemento dinâmico]
- JSON invalido

Regras de negócio e dependências: `MesService.createOrder`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/orders" -H "Authorization: Bearer ${TOKEN}" -H 'Content-Type: application/js
```

API-124 — Excluir OS ainda não programada

DELETE /api/mes/orders/{id}

Finalidade: Excluir OS ainda não programada.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:28` — `MesController.deleteOrder`.

Autenticação, autorização e perfis: @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	String	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: sem corpo quando 204.

Códigos HTTP: `204`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- Somente OS em A programar, ainda nao programada, pode ser excluida. Atualize a fila.
- JSON persistido invalido
- UUID invalido

Regras de negócio e dependências: `MesService.deleteOrder`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X DELETE "${BASE_URL}/api/mes/orders/00000000-0000-4000-8000-000000000001" -H "Authorization: Bearer ${T}  bash
```

API-125 — Editar OS sem injetora

PUT /api/mes/orders/{id}

Finalidade: Editar OS sem injetora. Somente QUEUED e machinelid nulo. Não há endpoint específico para remover a atribuição de máquina; a mensagem que pede desatribuir não garante que exista esse fluxo.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:29` — `MesController.updateOrder`.

Autenticação, autorização e perfis: @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesOrderInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "orderCode": "OS-DOC-001",
  "productCode": "DOC-001",
  "productName": "Produto de exemplo",
  "quantityPieces": 100,
  "piecesPerCycle": 2,
  "idealCycleMs": 30000,
  "toolCode": "M-DOC-001",
  "toolSettings": {
    "lot": "LOTE-DOC-001"
  },
  "notes": "Exemplo documental"
}
```

json

Resposta de sucesso: `MesOrder`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "status": "Exemplo",
  "queuePosition": 1,
  "createdAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "insumos": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Somente OS na fila pode ser editada
- OS já atribuída a uma injetora não pode ser editada; desatribua primeiro
- OS existente com este código
- JSON persistido inválido
- UUID inválido
- toolSettings deve ser objeto
- Parametro de ferramenta desconhecido: [mensagem pode receber complemento dinâmico]
- JSON inválido

Regras de negócio e dependências: `MesService.updateOrder`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/mes/orders/00000000-0000-4000-8000-000000000001" -H "Authorization: Bearer ${TOKE
```

API-126 — Extrair prévia de OS de PDF

POST `/api/mes/orders/import-preview`

Finalidade: Extrair prévia de OS de PDF. multipart/form-data, campo file. Primeiro extração de texto; fallback OCR da primeira página. Retorna prévia, não cria ordem. Sem orderCode reconhecido retorna erro.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:31` — `MesController.importPreview`.

Autenticação, autorização e perfis: @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo: arquivo PDF em multipart, campo obrigatório `file`; não enviar JSON.

Resposta de sucesso: `MesImportPreview`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Regras de negócio e dependências: Implementação no próprio controller. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/mes/orders/import-preview" -H "Authorization: Bearer ${TOKEN}" -F 'file=@ordem.p
```

API-127 — Alterar prioridade e injetora da fila

PUT `/api/mes/orders/{id}/position`

Finalidade: Alterar prioridade e injetora da fila.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:67` — `MesController.position`.

Autenticação, autorização e perfis: @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesPositionInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "queuePosition": 0,
  "machineId": "DOC.01"
}
```

json

Resposta de sucesso: `MesOrder`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "status": "Exemplo",
  "queuePosition": 1,
  "createdAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "insumos": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Somente OS na fila pode mudar prioridade
- A injetora da OS mudou; atualize a fila
- OS existente na maquina
- JSON persistido invalido
- UUID invalido

Regras de negócio e dependências: `MesService.position`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X PUT "${BASE_URL}/api/mes/orders/00000000-0000-4000-8000-000000000001/position" -H "Authorization: Bear
```

API-128 — Atribuir injetora à OS

PUT /api/mes/orders/{id}/machine

Finalidade: Atribuir injetora à OS.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:68` — `MesController.assignMachine`.

Autenticação, autorização e perfis: @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
id	path	String	Sim	—

Corpo da requisição: `MesAssignInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "machineId": "DOC.01"
}
```

json

Resposta de sucesso: `MesOrder`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "status": "Exemplo",
  "queuePosition": 1,
  "createdAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "insumos": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Somente OS na fila pode trocar de injetora
- OS existente na maquina
- JSON persistido invalido
- UUID invalido

Regras de negócio e dependências: `MesService.assignMachine`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PUT "${BASE_URL}/api/mes/orders/00000000-0000-4000-8000-000000000001/machine" -H "Authorization: Bearer "
```

API-129 — Preparar execução

POST `/api/mes/orders/{id}/prepare`

Finalidade: Preparar execução.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:69` — `MesController.prepare`.

Autenticação, autorização e perfis: `@mesAccess.plan(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesExecution`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "executionId": "00000000-0000-4000-8000-000000000001",
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "contextRevision": 1,
  "status": "Exemplo",
  "preparedAt": "2026-10-15T12:00:00Z",
  "startedAt": "2026-10-15T12:00:00Z",
  "finishedAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "cancelReason": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "insumos": "Exemplo",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Atribua uma injetora antes de preparar a OS
- OS já preparada
- JSON persistido inválido
- UUID inválido
- JSON inválido

Regras de negócio e dependências: `MesService.prepare`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/orders/00000000-0000-4000-8000-000000000001/prepare" -H "Authorization: Bear
```

API-130 — Desfazer preparo

POST /api/mes/executions/{id}/unprepare

Finalidade: Desfazer preparo.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:70` — `MesController.unprepare`.

Autenticação, autorização e perfis: @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesOrder`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "status": "Exemplo",
  "queuePosition": 1,
  "createdAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "insumos": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Execução não está preparada
- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.unprepare`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001/unprepare" -H "Authorization"
```

API-131 — Consultar execução

GET `/api/mes/executions/{id}`

Finalidade: Consultar execução.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:71` — `MesController.execution`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesExecution`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "executionId": "00000000-0000-4000-8000-000000000001",
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "contextRevision": 1,
  "status": "Exemplo",
  "preparedAt": "2026-10-15T12:00:00Z",
  "startedAt": "2026-10-15T12:00:00Z",
  "finishedAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "cancelReason": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "insumos": "Exemplo",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.execution`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001" -H "Authorization: Bearer ${{
```

API-132 — Iniciar produção

POST `/api/mes/executions/{id}/start`

Finalidade: Iniciar produção.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:72` — `MesController.start` .

Autenticação, autorização e perfis: @mesAccess.operate(authentication) or @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>` . `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesExecution` . Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "executionId": "00000000-0000-4000-8000-000000000001",
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "contextRevision": 1,
  "status": "Exemplo",
  "preparedAt": "2026-10-15T12:00:00Z",
  "startedAt": "2026-10-15T12:00:00Z",
  "finishedAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "cancelReason": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "insumos": "Exemplo",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Execução não está preparada
- Máquina já tem execução ativa
- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.start`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001/start" -H "Authorization: Be
```

API-133 — Finalizar produção após atingir meta

POST `/api/mes/executions/{id}/finish`

Finalidade: Finalizar produção após atingir meta. Exige RUNNING e peças de primeira qualidade >= meta. Libera a injetora. Excedente continua contando até a finalização.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:73` — `MesController.finish`.

Autenticação, autorização e perfis: @mesAccess.operate(authentication) or @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesExecution`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "executionId": "00000000-0000-4000-8000-000000000001",
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "contextRevision": 1,
  "status": "Exemplo",
  "preparedAt": "2026-10-15T12:00:00Z",
  "startedAt": "2026-10-15T12:00:00Z",
  "finishedAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "cancelReason": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "insumos": "Exemplo",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Execucao nao esta em producao
- Producao ainda nao atingiu a meta de pecas de primeira qualidade
- JSON persistido invalido
- UUID invalido

Regras de negócio e dependências: `MesService.finish`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001/finish" -H "Authorization: E
```

API-134 — Retirar OS da injetora

POST `/api/mes/executions/{id}/reallocate`

Finalidade: Retirar OS da injetora. Exige RUNNING e motivo. Encerra ordem e execução em CANCELLED; preserva produção e não cria OS com saldo restante.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:74` — `MesController.reallocate`.

Autenticação, autorização e perfis: @mesAccess.operate(authentication) or @mesAccess.plan(authentication) Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesReasonInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "reason": "Mudança de programação"
}
```

json

Resposta de sucesso: `MesExecution`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "executionId": "00000000-0000-4000-8000-000000000001",
  "orderId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "contextRevision": 1,
  "status": "Exemplo",
  "preparedAt": "2026-10-15T12:00:00Z",
  "startedAt": "2026-10-15T12:00:00Z",
  "finishedAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "cancelReason": "Exemplo",
  "orderCode": "Exemplo",
  "productCode": "Exemplo",
  "productName": "Exemplo",
  "toolCode": "Exemplo",
  "material": "Exemplo",
  "quantityPieces": 1,
  "piecesPerCycle": 1,
  "idealCycleMs": 1,
  "notes": "Exemplo",
  "toolSettings": {
    "lot": "Exemplo",
    "moldName": "Exemplo",
    "nozzleMode": "Exemplo",
    "pieceWeightGrams": 1,
    "netWeightGrams": 1,
    "grossWeightGrams": 1,
    "sprueWeightGrams": 1,
    "cyclesPerKit": 1
  },
  "insumos": "Exemplo",
  "goodPieces": 1,
  "targetReached": false
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Execução não está em produção
- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.reallocate`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001/reallocate" -H "Authorization: Bearer ..."
```

API-135 — Registrar refugo e resíduo

POST `/api/mes/executions/{id}/quality`

Finalidade: Registrar refugo e resíduo. Refugo/resíduo são acréscimos. Motivo obrigatório; ao menos refugo ou resíduo deve ser positivo. Até 3 casas decimais para resíduo. O serviço não exige RUNNING: não equiparar restrição da tela à autorização da API.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:75` — `MesController.quality`.

Autenticação, autorização e perfis: `@mesAccess.operate(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesQualityInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "reject": 1,
  "residueGrams": 0,
  "notes": "Peça fora do padrão"
}
```

json

Resposta de sucesso: `MesQuality`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "qualityId": "00000000-0000-4000-8000-000000000001",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "secondQuality": 1,
  "reject": 1,
  "residueGrams": 1,
  "notes": "Exemplo",
  "createdAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo"
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- residueGrams deve ser numerico
- Residuo invalido; use ate 3 casas decimais
- Motivo/observacao e obrigatorio
- Informe refugo ou residuo
- JSON persistido invalido
- UUID invalido

Regras de negócio e dependências: `MesService.quality`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001/quality" -H "Authorization: bash
```

API-136 — Consultar apontamentos de qualidade

GET `/api/mes/executions/{id}/quality`

Finalidade: Consultar apontamentos de qualidade.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:76` — `MesController.qualities`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	25

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesPage<MesQuality>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "content": [],
  "page": 0,
  "size": 1,
  "totalElements": 1,
  "totalPages": 0
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.qualities`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/mes/executions/00000000-0000-4000-8000-000000000001/quality" -H "Authorization: Bearer <TOKEN>"
```

API-137 — Registrar ou complementar parada

POST `/api/mes/machines/{id}/downtimes`

Finalidade: Registrar ou complementar parada. Autorização no corpo: MECHANICAL exige maintain; demais valores passam por operate antes da validação do kind. Uma ocorrência existente pode receber nova causa ou ser escalada para mecânica.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:77` — `MesController.downtime`.

Autenticação, autorização e perfis: Usuário autenticado; sem autoridade adicional declarada no método. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesDowntimeInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

json

```
{
  "kind": "OPERATIONAL",
  "reason": "Ajuste operacional",
  "notes": "Exemplo documental"
}
```

Resposta de sucesso: `MesDowntime`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "downtimeId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "kind": "Exemplo",
  "reason": "Exemplo",
  "notes": "Exemplo",
  "startedAt": "2026-10-15T12:00:00Z",
  "releasedAt": "2026-10-15T12:00:00Z",
  "recoveredAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "releasedBy": "Exemplo",
  "releaseNotes": "Exemplo",
  "category": "Exemplo",
  "source": "Exemplo",
  "mergedInto": "00000000-0000-4000-8000-000000000001"
}
```

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- JSON persistido inválido
- UUID inválido
- Histórico de causas desta parada atingiu o limite; libere a ocorrência antes de registrar outra

Regras de negócio e dependências: `MesAccess.maintain`, `MesAccess.operate`, `MesService.downtime`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X POST "${BASE_URL}/api/mes/machines/DOC.01/downtimes" -H "Authorization: Bearer ${TOKEN}" -H 'Content-T
```

API-138 — Liberar parada

POST `/api/mes/downtimes/{id}/release`

Finalidade: Liberar parada. Exige read e maintain para parada mecânica, operate para operacional. Somente ocorrência principal e ainda não liberada. Mecânica aguarda ciclo válido para recuperação.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:81` — `MesController.release`.

Autenticação, autorização e perfis: Usuário autenticado; sem autoridade adicional declarada no método. Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo da requisição: `MesReleaseInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em [Modelos](#).

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "notes": "Intervenção concluída"
}
```

json

Resposta de sucesso: `MesDowntime`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "downtimeId": "00000000-0000-4000-8000-000000000001",
  "machineId": "Exemplo",
  "executionId": "00000000-0000-4000-8000-000000000001",
  "kind": "Exemplo",
  "reason": "Exemplo",
  "notes": "Exemplo",
  "startedAt": "2026-10-15T12:00:00Z",
  "releasedAt": "2026-10-15T12:00:00Z",
  "recoveredAt": "2026-10-15T12:00:00Z",
  "author": "Exemplo",
  "releasedBy": "Exemplo",
  "releaseNotes": "Exemplo",
  "category": "Exemplo",
  "source": "Exemplo",
  "mergedInto": "00000000-0000-4000-8000-000000000001"
}
```

json

Códigos HTTP: `200`, `400`, `401`, `403`, `404`, `409`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em [Erros](#).

Validações e erros identificados na cadeia local do serviço:

- JSON persistido invalido
- UUID invalido
- Libere a ocorrência principal desta parada
- Ocorrencia ja liberada

Regras de negócio e dependências: `MesAccess.read`, `MesService.downtimeById`, `MesAccess.maintain`, `MesAccess.operate`, `MesService.release`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/downtimes/00000000-0000-4000-8000-000000000001/release" -H "Authorization: E
```

bash

API-139 — Consultar paradas por período

GET `/api/mes/machines/{id}/downtimes`

Finalidade: Consultar paradas por período.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:87` — `MesController.downtimes`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em [Autenticação](#).

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—
<code>from</code>	query	<code>Instant</code>	Sim	—
<code>to</code>	query	<code>Instant</code>	Sim	—
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	25

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesPage<MesDowntime>`. Campos em [Modelos](#).

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "content": [],
  "page": 0,
  "size": 1,
  "totalElements": 1,
  "totalPages": 0
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- JSON persistido invalido

Regras de negócio e dependências: `MesService.downtimes`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/mes/machines/DOC.01/downtimes?from=2026-10-01T00%3A00%3A00Z&to=2026-10-15T00%3A00%3A00Z"
```

API-140 — Consultar parada aberta e ocorrências vinculadas

GET `/api/mes/machines/{id}/downtimes/active`

Finalidade: Consultar parada aberta e ocorrências vinculadas.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:88` — `MesController.activeDowntime`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>id</code>	path	<code>String</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesActiveDowntime`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "active": {
    "downtimeId": "00000000-0000-4000-8000-000000000001",
    "machineId": "Exemplo",
    "executionId": "00000000-0000-4000-8000-000000000001",
    "kind": "Exemplo",
    "reason": "Exemplo",
    "notes": "Exemplo",
    "startedAt": "2026-10-15T12:00:00Z",
    "releasedAt": "2026-10-15T12:00:00Z",
    "recoveredAt": "2026-10-15T12:00:00Z",
    "author": "Exemplo",
    "releasedBy": "Exemplo",
    "releaseNotes": "Exemplo",
    "category": "Exemplo",
    "source": "Exemplo",
    "mergedInto": "00000000-0000-4000-8000-000000000001"
  },
  "linked": []
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- JSON persistido inválido

Regras de negócio e dependências: `MesService.activeDowntime`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/mes/machines/DOC.01/downtimes/active" -H "Authorization: Bearer ${TOKEN}"
```

API-141 — Consolidar indicadores de produção

GET `/api/mes/reports`

Finalidade: Consolidar indicadores de produção. Período positivo de até 1826 dias. Totais de qualidade têm base EXECUTION; executionTotals, quando presente, representa a OS completa.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:89` — `MesController.reports`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>machineId</code>	query	<code>String</code>	Não	—
<code>executionId</code>	query	<code>String</code>	Não	—
<code>from</code>	query	<code>Instant</code>	Sim	—
<code>to</code>	query	<code>Instant</code>	Sim	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesReport`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "grossPieces": 1,
  "cycles": 1,
  "validCycles": 1,
  "averageCycleMs": 1,
  "averageInjectionMs": 1,
  "averageMoldClosedMs": 1,
  "averageMoldNotClosedMs": 1,
  "averageNozzleContactMs": 1,
  "averageNozzleAwayMs": 1,
  "averageDosingMs": 1,
  "secondQuality": 1,
  "reject": 1,
  "residueGrams": 1,
  "goodPieces": 1,
  "qualityBasis": "Exemplo",
  "projection": "Exemplo",
  "executionTotals": {
    "grossPieces": 1,
    "cycles": 1,
    "validCycles": 1,
    "averageCycleMs": 1,
    "averageInjectionMs": 1,
    "averageMoldClosedMs": 1,
    "averageMoldNotClosedMs": 1,
    "averageNozzleContactMs": 1,
    "averageNozzleAwayMs": 1,
    "averageDosingMs": 1,
    "secondQuality": 1,
    "reject": 1,
    "residueGrams": 1,
    "goodPieces": 1,
    "qualityBasis": "Exemplo",
    "projection": "Exemplo"
  }
}
```

json

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- UUID inválido
- JSON persistido inválido

Regras de negócio e dependências: `MesService.reports`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/reports?from=2026-10-01T00%3A00%3A00Z&to=2026-10-15T00%3A00%3A00Z" -H "Author bash
```

API-142 — Consultar resultados de ciclos

GET `/api/mes/history`

Finalidade: Consultar resultados de ciclos.

Fonte: `src/main/java/com/wester/storage/mes/MesController.java:90` — `MesController.history`.

Autenticação, autorização e perfis: `@mesAccess.read(authentication)` Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `Authorization: Bearer <TOKEN>`. `Accept: application/json` recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
<code>machineId</code>	query	<code>String</code>	Não	—
<code>executionId</code>	query	<code>String</code>	Não	—
<code>from</code>	query	<code>Instant</code>	Sim	—
<code>to</code>	query	<code>Instant</code>	Sim	—
<code>page</code>	query	<code>int</code>	Não	0
<code>size</code>	query	<code>int</code>	Não	25

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: `MesPage<MesCycleOutcome>`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "content": [],
  "page": 0,
  "size": 1,
  "totalElements": 1,
  "totalPages": 0
}
```

Códigos HTTP: 200, 400, 401, 403, 404, 409. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- UUID inválido
- JSON persistido inválido

Regras de negócio e dependências: `MesService.history`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X GET "${BASE_URL}/api/mes/history?from=2026-10-01T00%3A00%3A00Z&to=2026-10-15T00%3A00%3A00Z" -H "Author
```

Integração industrial Edge

Índice de endpoints · Modelos

API-143 — Obter configuração para o coletor

GET /api/mes/edge/config

Finalidade: Obter configuração para o coletor.

Fonte: src/main/java/com/wester/storage/mes/MesEdgeController.java:12 — MesEdgeController.config .

Autenticação, autorização e perfis: Chave X-Edge-Key válida (MES_EDGE_KEY com pelo menos 16 bytes). JWT não substitui a chave. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: X-Edge-Key: <CHAVE> . Accept: application/json recomendado.

Parâmetros:

NOME	LOCAL	TIPO	OBRIGATÓRIO	PADRÃO
machineId	query	String	Sim	—
edgeId	query	String	Não	—

Corpo: não declarado; não enviar corpo.

Resposta de sucesso: MesConfig . Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "machineId": "Exemplo",
  "revision": 1,
  "executionId": "00000000-0000-4000-8000-000000000001",
  "idealCycleMs": 1,
  "piecesPerCycle": 1,
  "mechanicalFault": false,
  "recoveryPending": false,
  "hasCompletedCycle": false,
  "productionBlocked": false,
  "manualLamp": "Exemplo",
  "desiredLamp": "Exemplo"
}
```

Códigos HTTP: 200 , 400 , 401 , 404 , 409 , 503 . Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Edge não vinculado a máquina
- JSON persistido inválido
- UUID inválido

Regras de negócio e dependências: `MesService.config`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X GET "${BASE_URL}/api/mes/edge/config?machineId=DOC.01" -H "X-Edge-Key: ${EDGE_KEY}"
```

bash

API-144 — Receber evento do coletor

POST `/api/mes/edge/events`

Finalidade: Receber evento do coletor. Idempotência por eventId com corpo idêntico. Conteúdo diferente ou reutilização da sequência gera conflito. Saltos na sequência exigem validação e não contabilizam peças daquele ciclo. Timestamp até 5 minutos no futuro. CYCLE exige execução original e timings; schemaVersion >= 3 exige namedSignals.

Fonte: `src/main/java/com/wester/storage/mes/MesEdgeController.java:13` — `MesEdgeController.events`.

Autenticação, autorização e perfis: Chave X-Edge-Key válida (MES_EDGE_KEY com pelo menos 16 bytes). JWT não substitui a chave. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `X-Edge-Key: <CHAVE>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `MesEdgeEventInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

json

```
{
  "machineId": "DOC.01",
  "eventId": "00000000-0000-4000-8000-000000000001",
  "sequence": 1,
  "timestamp": "2026-10-15T12:00:00Z",
  "type": "SNAPSHOT",
  "lamp": "OFF",
  "outputStatus": "DISABLED",
  "signals": {
    "DI0": 0,
    "DI1": 0,
    "DI2": 0,
    "DI3": 0,
    "DI4": 0
  },
  "schemaVersion": 3,
  "namedSignals": {
    "BOMBA": 0,
    "MOLDE_FECHADO": 0,
    "ENCOSTE_BICO": 0,
    "INJECAO": 0,
    "ALARME": 0
  }
}
```

Resposta de sucesso: `MesEventAck`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

json

```
{
  "eventId": "00000000-0000-4000-8000-000000000001",
  "duplicate": false
}
```

Códigos HTTP: `200`, `400`, `401`, `404`, `409`, `503`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- Edge não vinculado a máquina
- eventId reutilizado com conteúdo diferente
- Sequência do Edge reutilizada com outro eventId
- classification inválida
- CYCLE exige VALID_CYCLE
- Resultado válido deve usar CYCLE
- cycleState inválido
- Timestamp mais de cinco minutos no futuro
- Início do ciclo posterior ao evento

- signals obrigatorio
- namedSignals obrigatorio a partir de schemaVersion 3
- Execucao pertence a outra maquina
- Execucao ainda nao iniciada
- CYCLE exige executionId original
- cycle obrigatorio
- Pecas divergem do contexto imutavel
- timings obrigatorio
- CYCLE_RESULT exige cycle
- Ciclo interrompido nao pode produzir pecas
- Somente eventos de ciclo podem conter cycle
- JSON persistido invalido
- UUID invalido
- JSON invalido
- Histórico de causas desta parada atingiu o limite; libere a ocorrência antes de registrar outra

Regras de negócio e dependências: `MesService.ingest`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/edge/events" -H "X-Edge-Key: ${EDGE_KEY}" -H 'Content-Type: application/json' -d {}
```

API-145 — Receber saúde do coletor

POST `/api/mes/edge/heartbeat`

Finalidade: Receber saúde do coletor.

Fonte: `src/main/java/com/wester/storage/mes/MesEdgeController.java:14` — `MesEdgeController.heartbeat`.

Autenticação, autorização e perfis: Chave X-Edge-Key válida (MES_EDGE_KEY com pelo menos 16 bytes). JWT não substitui a chave. Consulte a correspondência entre autoridades e perfis em **Autenticação**.

Headers: `X-Edge-Key: <CHAVE>`. `Accept: application/json` recomendado. `Content-Type: application/json` para o corpo.

Parâmetros:

Sem query parameters, path variables ou headers de negócio declarados.

Corpo da requisição: `MesHeartbeatInput`; obrigatório. Sem `@Valid` no argumento; validações de negócio são executadas pelo serviço. Consulte os campos e restrições em **Modelos**.

Exemplo sintético (IDs devem existir e versões devem vir de consulta):

```
{
  "machineId": "DOC.01",
  "adamReadAt": "2026-10-15T12:00:00Z",
  "outboxDepth": 0,
  "quarantineDepth": 0
}
```

json

Resposta de sucesso: `MesHeartbeatAck`. Campos em **Modelos**.

Exemplo estrutural sintético, sem afirmar valores de produção:

```
{
  "accepted": false,
  "receivedAt": "2026-10-15T12:00:00Z"
}
```

json

Códigos HTTP: `200`, `400`, `401`, `404`, `409`, `503`. Códigos de segurança dependem da cadeia de filtros; falhas não capturadas podem chegar ao tratamento global descrito em **Erros**.

Validações e erros identificados na cadeia local do serviço:

- lastRejection invalida
- JSON persistido invalido

Regras de negócio e dependências: `MesService.heartbeat`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

```
curl -X POST "${BASE_URL}/api/mes/edge/heartbeat" -H "X-Edge-Key: ${EDGE_KEY}" -H 'Content-Type: application/j
```

bash