

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:

json

```
{
  "id": 1,
  "areaId": 1,
  "titulo": "Exemplo",
  "colunas": [],
  "itens": []
}
```

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):

bash

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

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>	path	<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/json'
```

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
<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`. 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 'Cont
```

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.removeItem`.

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.removeItem`. 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
<code>itemId</code>	path	<code>Long</code>	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:

json

```
{
  "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.atualizarQuantidade`. Consulte documento interno de regras de negócio para efeitos persistentes e validações compartilhadas.

cURL (Bash):

bash

```
curl -X PATCH "${BASE_URL}/api/encerto/itens/1/quantidade" -H "Authorization: Bearer ${TOKEN}"
```

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
itemId	path	Long	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"
}
```

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.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}"
```

bash

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}" -
```

bash