

1. Visão Geral e Arquitetura

Esta API RESTful unificada gerencia jogos de mídia física (Cartucho e Disco) através de um único endpoint (/api/jogos).

Endpoint Base: `http://[host]:[port]/api/jogos`

2. Guia de Operações CRUD (Postman)

2.1. INCLUIR NOVO JOGO (POST) +

Inclui um novo item. O ID será gerado automaticamente.

- **Método:** `POST`
- **Endpoint:** `/api/jogos`
- **Obrigatório:** Incluir o campo `"tipoMidia"`. Não inclua o `"id"`.

Exemplo 1: Cartucho (POST)

```
{
  "tipoMidia": "cartucho",
  "console": "Nintendo 64",
  "titulo": "Super Mario 64",
  "desenvolvedora": "Nintendo EAD",
  "genero": "Plataforma",
  "anoLancamento": "1996",
  "quantidadeEmEstoque": 2,
  "precoCusto": 80.0,
  "precoVenda": 250.0,
  "observacoes": "Mídia e manual em excelente estado.",
  "cartuchoConservacao": "OTIMO",
  "possuiCaixaOriginal": true,
  "cartuchoRegiao": "NTSC_US"
}
```

Exemplo 2: Disco (POST)

```
{
  "tipoMidia": "disco",
  "console": "Playstation 5",
  "titulo": "Elden Ring",
  "desenvolvedora": "FromSoftware",
  "genero": "RPG de Ação",
  "anoLancamento": "2022",
  "quantidadeEmEstoque": 1,
  "precoCusto": 150.0,
}
```

```
"precoVenda": 220.0,  
"observacoes": "",  
"discoEstado": "RISCO_BAIXO",  
"discoCapa": "ORIGINAL",  
"discoConteudoExtra": "NAO_POSSUI"  
}
```

2.2. ALTERAR JOGO (PUT)

Atualiza um item existente.

- **Método:** PUT
- **Endpoint:** `/api/jogos/{id}`
- **Obrigatório:** O ID deve estar no Path (`{id}`). Não inclua o campo `"id"` no corpo do JSON.

Exemplo: Alterar o Cartucho de ID 10

URL: PUT `/api/jogos/10`

```
{  
  "tipoMidia": "cartucho",  
  "console": "Nintendo 64",  
  "titulo": "Super Mario 64 (Revisado)",  
  "desenvolvedora": "Nintendo EAD",  
  "genero": "Plataforma",  
  "anoLancamento": "1996",  
  "quantidadeEmEstoque": 1,  
  "precoCusto": 80.0,  
  "precoVenda": 200.0,  
  "observacoes": "Preço de venda reduzido para R$200.",  
  "cartuchoConservacao": "BOM",  
  "possuiCaixaOriginal": false,  
  "cartuchoRegiao": "REGION_FREE"  
}
```

2.3. CONSULTAR JOGO (GET)

2.3.1. Obter por ID

Busca um item específico.

- **Método:** GET
- **Endpoint:** `/api/jogos/{id}`

- Exemplo: **GET** `/api/jogos/10`

2.3.2. Listar Todos ou Filtrar

- Método: **GET**
- Endpoint: `/api/jogos`
- Filtro Opcional: Use o parâmetro de query `?tipo={cartucho|disco}`
- Listar apenas discos: **GET** `/api/jogos?tipo=disco`
- Listar apenas cartuchos: **GET** `/api/jogos?tipo=cartucho`
- Listar todos os jogos (discos e cartuchos): **GET** `/api/jogos`

2.3.3. Pesquisar por Título Parcial (Query Method)

Busca jogos no catálogo (Cartucho e Disco) cujo título **contém** um termo, ignorando a caixa (maiúsculas/minúsculas).

- Método: **GET**
- Endpoint: `/api/jogos/busca`
- Parâmetro de Query (Obrigatório): `?termo={termoBusca}`
- Exemplo: **GET** `/api/jogos/busca?termo=zelda`
- Resposta: Retorna uma lista de jogos que correspondem ao critério.

2.4. EXCLUIR JOGO (DELETE)

Remove um item da coleção pelo ID.

- Método: **DELETE**
- Endpoint: `/api/jogos/{id}`
- Exemplo: **DELETE** `/api/jogos/10`
- Resposta de Sucesso: Status **204 No Content**.

3. Guia de Operações de Vendas (Postman)

3.1. REALIZAR NOVO PEDIDO (POST)

Processa uma lista de itens, verifica o estoque, calcula o valor total, registra o pedido e decrementa o estoque dos jogos envolvidos.

- **Método:** **POST**
- **Endpoint:** **/api/pedidos**
- **Resposta de Sucesso:** Status **201 Created** com o detalhe do pedido.
- **Regras:**
 - O ID do jogo deve existir (**JogoNaoEncontradoException** se falhar).
 - A quantidade pedida não pode exceder o **quantidadeEmEstoque** (**IllegalArgumentException**).

Exemplo: Realizar Pedido (POST)

```
{
  "itens": [
    {
      "jogold": 1,
      "quantidadeComprada": 1
    },
    {
      "jogold": 10,
      "quantidadeComprada": 2
    }
  ]
}
```

Resposta de Sucesso Esperada (Status 201 Created)

```
{
  "id": 101,
  "dataPedido": "2025-11-09T23:59:59.000",
  "valorTotal": 590.00,
  "itens": [
    {
      "jogold": 1,
      "titulo": "The Legend of Zelda: Tears of the Kingdom",
      "console": "NINTENDO_SWITCH",
      "tipoMidia": "DISCO",
      "precoUnitario": 350.00,
      "quantidadeComprada": 1
    },
    {
      "jogold": 10,
      "titulo": "The Legend of Zelda: Ocarina of Time",
      "console": "NINTENDO_64",
      "tipoMidia": "CARTUCHO",
      "precoUnitario": 120.00,
      "quantidadeComprada": 2
    }
  ]
}
```

3.2. CONSULTAR PEDIDOS (GET)

Lista todos os pedidos já realizados no sistema.

- **Método:** GET
- **Endpoint:** /api/pedidos
- **Exemplo:** GET /api/pedidos
- **Resposta:** Retorna uma lista simplificada dos objetos Pedido persistidos.

4. Dicionário de Dados e Validações da API Jogo

Esta seção detalha os campos, tipos de dados e regras de validação aplicadas no backend da API.

4.1. Campos de Entrada e Persistência (Classe Jogo)

Estes campos são comuns a Cartucho e Disco e são enviados pelo usuário.

Campo JSON (Input)	Tipo de Dado	Validação de Código (@NotNull / Service)	Descrição
id	Integer	Gerado automaticamente (JPA Sequence).	Identificador único global do jogo. Não deve ser enviado no POST.
tipoMidia	String	Obrigatório para o roteamento do Controller.	Usado para distinguir a subclasse (disco ou cartucho).
console	String	Obrigatório (@NotNull e Service).	O console ao qual o jogo pertence (Ex: "PS5", "N64").
titulo	String	Obrigatório (@NotNull e Service).	Título completo do jogo.
desenvolvedora	String	Obrigatório (@NotNull e Service).	Estúdio que desenvolveu o jogo.
genero	String	Obrigatório (@NotNull e Service).	Gênero principal do jogo.
anoLancamento	String	Obrigatório (@NotNull e Service).	Ano em que o jogo foi lançado (Ex: "1997").
quantidadeEm	Integer	Obrigatório (@Min(0) e	Quantidade de cópias deste item na

Estoque		Service). Deve ser ≥0.	coleção/estoque.
precoCusto	Double	Obrigatório (@Min(0) e Service). Deve ser ≥0.	Preço pago pelo item (custo).
precoVenda	Double	Obrigatório (@Min(0) e Service). Deve ser ≥0.	Preço sugerido para venda do item.
observacoes	String	Opcional (o Service converte null/vazio para "").	Notas adicionais sobre o item.

4.2. Campos Específicos de **Cartucho**

Campo JSON (Input)	Tipo de Dado	Validação de Código	Descrição
cartuchoConservacao	Enum (String)	Obrigatório no backend (caso contrário, erro de Enum).	Estado do disco (mídia). Valores: LACRADO_COLECIONADOR, RISCO_BAIXO, RISCO_MODERADO.
possuiCaixaOriginal	Boolean	Obrigatório no Backend.	Tipo de capa/embalagem. Valores: ORIGINAL, REPRODUCAO.
cartuchoRegiao	Enum (String)	Obrigatório no backend (caso contrário, erro de Enum).	Conteúdo adicional. Valores: NAO_POSSUI, DLC, SOUNDTRACK_CD, ARTBOOK_MINI.

4.3. Campos Específicos de **Disco**

Campo JSON (Input)	Tipo de Dado	Validação de Código	Descrição
discoEstado	Enum (String)	Obrigatório no backend (caso contrário, erro de Enum).	Estado do disco (mídia). Valores: LACRADO_COLECIONADOR, RISCO_BAIXO, RISCO_MODERADO.
discoCapa	Enum (String)	Obrigatório no backend (caso contrário, erro de Enum).	Tipo de capa/embalagem. Valores: ORIGINAL, REPRODUCAO.
discoConteudoExtra	Enum (String)	Obrigatório no backend (caso contrário, erro de Enum).	Conteúdo adicional. Valores: NAO_POSSUI, DLC, SOUNDTRACK_CD, ARTBOOK_MINI.

4.4 Campos de Scores e Sincronização (API Externa - Somente Output)

Estes campos são preenchidos automaticamente pelo sistema durante a inclusão ou alteração, buscando dados da API RAWG e Metacritic. Eles são serializados com *getters* formatados e **não devem ser enviados no Input**.

Campo JSON (Output)	Propriedade Original	Tipo (Output Formatado)	Descrição
metacriticScore	metacriticScore (Integer)	String (ou "N/A")	Pontuação oficial do jogo no Metacritic.
RAWG_userScore	userScore (Double)	String formatada (Ex: "4.2")	Pontuação de usuários do jogo na plataforma RAWG.
RAWG_userClassification	userRatingText (String)	String (Ex: "Recomendado")	Classificação de aceitação do jogo pelos usuários do RAWG (Ex: Excepcional, Mediano, Não recomendado).
lastApiSync	lastApiSync (LocalDateTime)	String formatada	Data e hora da última sincronização bem-sucedida com a API externa.