

Documentação da API G.R.I.M.O.R.I.A.

Visão Geral

A API do G.R.I.M.O.R.I.A. (Gerador de Respostas Inteligentes e Multimodais Orquestradas por Recuperação de Informações e Análises) fornece uma interface RESTful para interagir com o sistema RAG e seus componentes de IA. Ela permite o gerenciamento de coleções de documentos, a realização de consultas contextuais (RAG) e a chamada direta aos modelos de linguagem generativa.

URL Base: `http://<IP_DO_SERVIDOR>:5000`

Formato dos Dados: Todas as requisições e respostas utilizam o formato JSON, exceto pelo endpoint de upload de arquivos, que usa multipart/form-data.

Autenticação: A versão atual da API não requer autenticação.

Referência dos Endpoints

Gerenciamento de Coleções

1. Criar Coleções via Upload de Arquivos

Cria coleções no banco de dados a partir de um ou mais arquivos enviados diretamente para a API, associando-os a um tema.

- **Endpoint:** POST `/upload_e_criar_colecao`
- **Tipo de Requisição:** multipart/form-data
- **Parâmetros (Form Data):**
 - **tema** (string, **obrigatório**): O tema a ser associado a todos os arquivos enviados.
 - **files** (arquivo, **obrigatório**): Um ou mais arquivos para upload.
-

Exemplo de Sucesso (200 OK):

Generated json

```
{
  "mensagem": "2 arquivos do tema 'capivaras' foram processados e adicionados com sucesso."
}
```

-

Exemplo de Erro (400 Bad Request):

Generated json

```
{
  "erro": "O parâmetro 'tema' é obrigatório no formulário."
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

2. Criar Coleções a partir de uma Pasta no Servidor

Processa todos os arquivos de uma pasta localizada na máquina onde a API está rodando. O nome da pasta é usado como o "tema" para as coleções.

- **Endpoint:** POST /criar_colecao
- **Tipo de Requisição:** application/json

Corpo da Requisição (JSON):

Generated json

```
{
  "diretorio": "C:/caminho/para/pasta/capivaras"
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END
- **Parâmetros:**
 - diretorio (string, **obrigatório**): Caminho completo para a pasta no servidor da API.
-

Exemplo de Sucesso (200 OK):

Generated json

```
{
  "mensagem": "Coleções criadas a partir do diretório 'C:/caminho/para/pasta/capivaras'."
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

3. Listar Coleções

Retorna uma lista de todas as coleções existentes no banco de dados, ordenadas por tema, nome do arquivo e ID.

- **Endpoint:** GET /listar_colecoes
- **Parâmetros:** Nenhum.

Exemplo de Sucesso (200 OK):

Generated json

```
[
  {
    "id": "0001",
    "nome_arquivo": "fatos_sobre_capivaras.pdf",
    "tema": "capivara"
  },
  {
    "id": "0002",
    "nome_arquivo": "habitat_do_ornitorrinco.docx",
    "tema": "ornitorrinco"
  }
]
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

4. Deletar uma Coleção

Remove uma coleção específica do banco de dados pelo seu ID.

- **Endpoint:** DELETE /deletar_colecao
- **Tipo de Requisição:** application/json

Corpo da Requisição (JSON):

Generated json

```
{
  "id_colecao": "0001"
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END
- **Parâmetros:**
 - id_colecao (string, **obrigatório**): O ID da coleção a ser deletada.
-

Exemplo de Sucesso (200 OK):

Generated json

```
{
  "mensagem": "Coleção '0001' deletada com sucesso."
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

Exemplo de Erro (404 Not Found):

Generated json

```
{
  "erro": "Coleção '0001' não encontrada."
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

5. Zerar Todas as Coleções

Remove **TODAS** as coleções do banco de dados. Esta ação é irreversível.

- **Endpoint:** DELETE /zerar_todas_colecoes
- **Parâmetros:** Nenhum.

Exemplo de Sucesso (200 OK):

Generated json

```
{
  "mensagem": "Todas as coleções foram removidas com sucesso."
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

Endpoints de Consulta

6. Consulta RAG (Contextual)

Realiza uma busca semântica nos documentos do banco de dados (filtrando por tema) e usa os resultados como contexto para a IA responder a uma pergunta.

- **Endpoint:** POST /consultar
- **Tipo de Requisição:** application/json
- **Corpo da Requisição (JSON):**
 - pergunta (string, **obrigatório**): A pergunta do usuário.
 - instrucao (string, opcional): Instruções para a IA (ex: "Aja como um biólogo").
 - usar_ia_local (boolean, opcional, padrão: false): Se true, usa a IA local (Ollama).
 - modelo_de_pensamento (string, opcional): Nome do modelo de IA a ser usado (ex: gemini-1.5-pro-latest ou gemma3:latest).
 - n_results_per_colecao (integer, opcional, padrão: 10): N° de chunks a buscar por documento.
 - max_distance_threshold (float, opcional, padrão: 0.8): Limiar de similaridade da busca (menor = mais similar).
 - imagem_path (string, opcional): Caminho de uma imagem no servidor da API para análise multimodal.
 - pdf_path (string, opcional): (Uso futuro) Caminho de um PDF no servidor.
-

Exemplo de Sucesso (200 OK):

Generated json

```
{  
  "resposta": "A capivara é o maior roedor do mundo, conhecida por seu temperamento dócil e hábitos semiaquáticos."  
}
```

- IGNORE_WHEN_COPYING_START
content_copy download
Use code [with caution](#). Json
IGNORE_WHEN_COPYING_END

7. Chamada Direta à IA

Envia uma solicitação diretamente para um modelo de linguagem (local ou via API), sem usar o contexto do banco de dados.

- **Endpoint:** POST /chamar_ia_direto
- **Tipo de Requisição:** application/json
- **Corpo da Requisição (JSON):**
 - pergunta (string, **obrigatório**): A pergunta ou comando para a IA.
 - instrucao (string, opcional): Instruções para a IA.
 - contexto (string, opcional): Um texto base para a IA usar na resposta.
 - usar_ia_local (boolean, opcional, padrão: false): Se true, usa a IA local (Ollama).
 - modelo_ia (string, opcional): Nome do modelo de IA a ser usado.

- imagem_path (string, opcional): Caminho de uma imagem no servidor da API.

•

Exemplo de Sucesso (200 OK):

Generated json

```
{
  "resposta": "Claro, aqui está uma breve história sobre uma capivara que viaja no tempo..."
}
```

•

```
IGNORE_WHEN_COPYING_START
content_copy download
Use code with caution. Json
IGNORE_WHEN_COPYING_END
```

Endpoints de Utilidade

8. Listar Modelos de IA Disponíveis

Retorna uma lista dos modelos da API do Google (Gemini/Gemma) que estão disponíveis para uso.

- **Endpoint:** GET /listar_modelos_ia
- **Parâmetros:** Nenhum.

Exemplo de Sucesso (200 OK):

Generated json

```
{
  "modelos": [
    "models/gemini-1.5-pro-latest",
    "models/gemini-1.5-flash-latest",
    "models/gemma-3-27b-it",
    "models/gemma-3-4b-it"
  ]
}
```

•

```
IGNORE_WHEN_COPYING_START
content_copy download
Use code with caution. Json
IGNORE_WHEN_COPYING_END
```

Exemplos de Uso

Exemplo 1: Consultar RAG com curl

Generated bash

```
curl -X POST http://127.0.0.1:5000/consultar \
-H "Content-Type: application/json" \
-d '{
  "pergunta": "Quais são os predadores da capivara?",
  "instrucao": "Responda em uma lista com no máximo 3 itens.",
  "modelo_de_pensamento": "gemini-1.5-flash-latest"
}'
```

IGNORE_WHEN_COPYING_START

content_copy download

Use code [with caution](#). Bash

IGNORE_WHEN_COPYING_END

Exemplo 2: Upload de Arquivo com curl

Generated bash

```
curl -X POST http://127.0.0.1:5000/upload_e_criar_colecao \
-F "tema=biologia_roedores" \
-F "files=@/caminho/local/para/meu_arquivo.pdf" \
-F "files=@/caminho/local/para/outra_imagem.jpg"
```

IGNORE_WHEN_COPYING_START

content_copy download

Use code [with caution](#). Bash

IGNORE_WHEN_COPYING_END

Exemplo 3: Chamada Direta à IA com Python requests

Generated python

```
import requests
import json
```

```
api_url = "http://127.0.0.1:5000/chamar_ia_direto"
```

```
payload = {
  "pergunta": "Crie um slogan para uma cafeteria cujo mascote é uma capivara.",
  "instrucao": "O slogan deve ser curto e amigável.",
  "usar_ia_local": False,
  "modelo_ia": "gemini-1.5-flash-latest"
}
```

```
try:
```

```
    response = requests.post(api_url, json=payload)
    response.raise_for_status() # Lança exceção para códigos de erro
```

```
    data = response.json()
```

```
print("Resposta da IA:", data.get("resposta"))
```

```
except requests.exceptions.RequestException as e:  
    print(f"Erro ao chamar a API: {e}")
```

```
IGNORE_WHEN_COPYING_START
```

```
content_copy download
```

```
Use code with caution. Python
```

```
IGNORE_WHEN_COPYING_END
```