## **Documentação da**

## **API REST - Enciclopédia Botânica**

### **1. Visão Geral**

A **Enciclopédia Botânica API** é um serviço RESTful construído em **FastAPI (Python)**, utilizando **SQLAlchemy** como ORM e **MySQL 8.0** para persistência de dados. O projeto adota uma **Arquitetura em Pacotes Explícitos** (API, CRUD, Models, Schemas) para facilitar a manutenção.

* **URL Base:** http://127.0.0.1:8000
* **Documentação Interativa:** A documentação completa (Swagger UI) está disponível em http://127.0.0.1:8000/docs.

### **2. Modelo de Dados (Objeto: Planta)**

O objeto principal manipulado pela API é a **Planta**, com mapeamento direto para a tabela plantas\_enciclopedia.

#### **Estrutura do Objeto Planta:**

* **id:** integer. Chave primária. Identificador único da planta. **Restrição:** Somente Leitura.
* **nome\_popular:** string. Nome comum da planta. **Restrição:** Obrigatório.
* **nome\_cientifico:** string. Nome taxonômico. **Restrição:** **ÚNICO** (Não pode haver repetição) e Obrigatório.
* **familia:** string. Família botânica. **Restrição:** Obrigatório.
* **origem:** string. Local de origem. **Restrição:** Obrigatório.
* **cuidados:** string. Instruções detalhadas de cuidado. **Restrição:** Obrigatório.

### **3. Serviços Implementados (Endpoints CRUD)**

A API oferece um conjunto completo de endpoints para manipulação do acervo (CRUD):

* **Incluir (Create):** Método POST para /plantas/. Cria um novo registro de planta.
* **Listar (Read):** Método GET para /plantas/. Retorna todos os registros do acervo (suporta paginação via parâmetros de *query*).
* **Consultar (Read):** Método GET para /plantas/{planta\_id}. Retorna os detalhes de uma planta específica.
* **Alterar (Update):** Método PUT para /plantas/{planta\_id}. Atualiza completamente um registro existente.
* **Excluir (Delete):** Método DELETE para /plantas/{planta\_id}. Remove permanentemente um registro.

### **4. Detalhamento dos Serviços com Validações**

#### **A. Incluir (POST /plantas/)**

**Propósito:** Adicionar um novo registro de planta.

* **Requisição (Body JSON):** Espera o objeto PlantaCreate (sem o campo id).
* **Validação de Negócio:** Verifica se o **nome\_cientifico** já existe no acervo.
* **Códigos de Resposta:**
  + **201 Created:** Sucesso. Retorna o objeto Planta criado.
  + **400 Bad Request:** Falha na unicidade do nome científico. Exemplo de retorno: {"detail": "Nome científico já cadastrado na enciclopédia."}

#### **B. Alterar (PUT /plantas/{planta\_id})**

**Propósito:** Substituir um registro existente pelo ID.

* **Parâmetros de Path:** Requer o planta\_id (integer) no Path.
* **Requisição (Body JSON):** Espera o objeto PlantaUpdate (deve conter todos os campos para a atualização completa).
* **Validação de Negócio:**
  + Verifica se o **ID** existe.
  + Se o **nome\_cientifico** for alterado, verifica se o novo nome já está em uso por **outra** planta.
* **Códigos de Resposta:**
  + **200 OK:** Sucesso. Retorna o objeto Planta atualizado.
  + **404 Not Found:** ID da planta não encontrado.
  + **400 Bad Request:** Falha na unicidade. Exemplo de retorno: {"detail": "Nome científico já está em uso por outra planta."}

#### **C. Excluir (DELETE /plantas/{planta\_id})**

**Propósito:** Remover permanentemente um registro.

* **Parâmetros de Path:** Requer o planta\_id (integer) no Path.
* **Validação de Negócio:** Verifica se o **ID** existe.
* **Códigos de Resposta:**
  + **200 OK:** Sucesso. Retorna o objeto Planta que foi excluído.
  + **404 Not Found:** ID da planta não encontrado.