**RELATÓRIO TÉCNICO**

<Sara Ferreira Fernandes - 835593>

**TEMA**

<Tema escolhido>

### 

### **Introdução**

Este relatório tem como objetivo apresentar o desenvolvimento de uma plataforma computacional distribuída para o cadastro, exibição e gerenciamento de informações sobre dinossauros. A aplicação, intitulada *DinoDex*, foi desenvolvida com tecnologias modernas e arquitetura orientada a serviços, utilizando FastAPI no backend e HTML/CSS/JavaScript no frontend.

### 

### **2. Objetivo**

O objetivo principal é construir um sistema funcional que permita:

* O cadastro e listagem de dinossauros com informações relevantes (nome, dieta, período, imagem, etc.);
* A criação, leitura, atualização e exclusão de dados (CRUD);
* A interação entre frontend e backend via API RESTful.

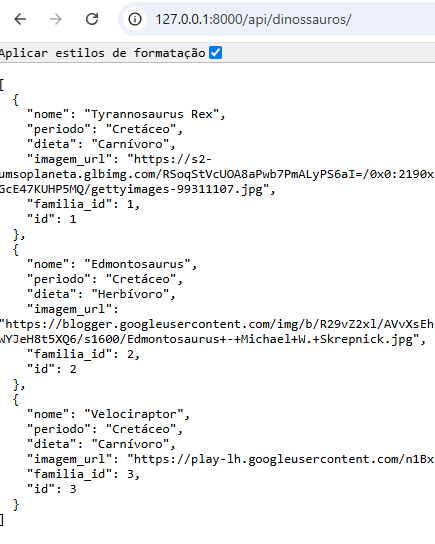
### **3. Metodologia**

**3.1 Arquitetura**

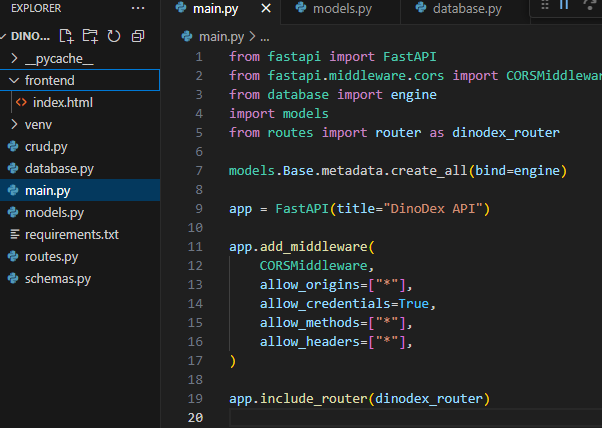
O projeto segue o modelo cliente-servidor:

* Backend: Desenvolvido com a linguagem de programação Python, utilizando o framework FastAPI para construção da API RESTful. A lógica de acesso ao banco de dados é implementada com SQLAlchemy, atuando como ORM, e os dados são persistidos em um banco relacional PostgreSQL. A documentação da API é gerada automaticamente por meio do Swagger UI, facilitando testes e integração.
* Frontend:Criado com HTML5, CSS3 e JavaScript puro. A interface consiste em uma página web responsiva que consome os dados da API utilizando requisições HTTP. A aplicação permite a exibição, cadastro e remoção de dinossauros com atualização dinâmica do conteúdo, sem a necessidade de recarregar a página.

**Imagem FastAPI:**



**Imagem demonstração Back-end:**



**3.1.1** **Script Banco de dados**:

CREATE DATABASE dinodex;

CREATE TABLE familias(

id SERIAL PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

descricao TEXT

);

CREATE TABLE dinossauros(

id SERIAL PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

periodo VARCHAR(50),

dieta VARCHAR(50),

tamanho\_metros NUMERIC(5,2),

familia\_id INTEGER REFERENCES familias(id) ON DELETE CASCADE,

imagem\_url TEXT

);

INSERT INTO familias(nome, descricao) VALUES

('Tyrannosauridae', 'Família de grandes dinossauros carnívoros, como o Tyrannosaurus.'),

('Hadrosauridae', 'Dinossauros herbívoros conhecidos como “dinossauros de bico de pato”.'),

('Dromaeosauridae', 'Família dos raptors, pequenos carnívoros rápidos e inteligentes.');

INSERT INTO dinossauros (nome, periodo, dieta, tamanho\_metros, familia\_id, imagem\_url) VALUES

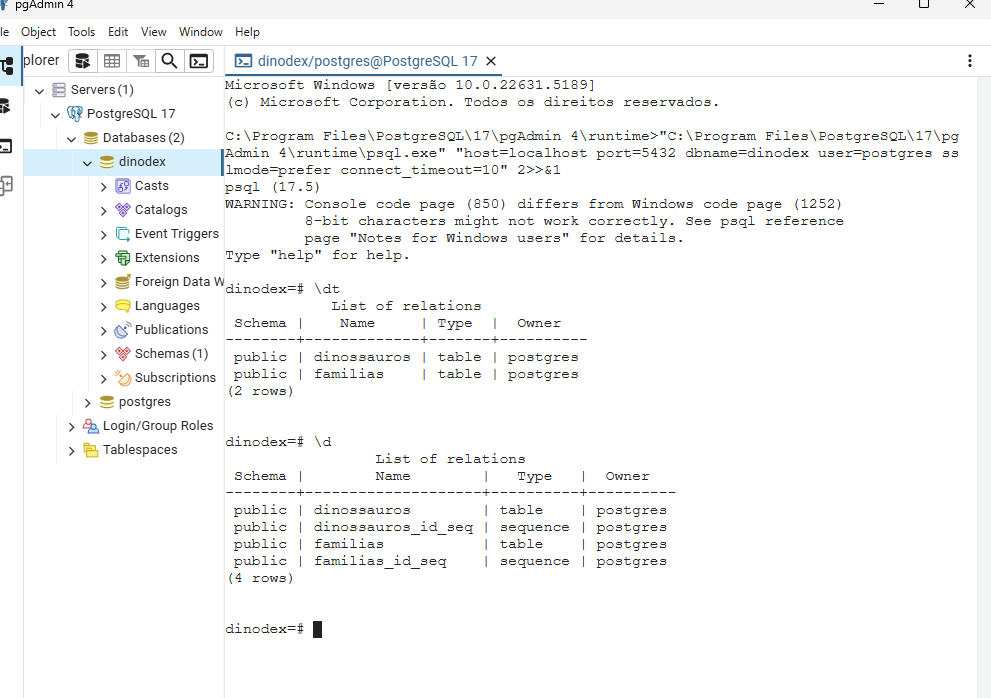
('Tyrannosaurus Rex', 'Cretáceo', 'Carnívoro', 12.3, 1, 'https://s2-umsoplaneta.glbimg.com/RSoqStVcUOA8aPwb7PmALyPS6aI=/0x0:2190x1369/984x0/smart/filters:strip\_icc()/i.s3.glbimg.com/v1/AUTH\_7d5b9b5029304d27b7ef8a7f28b4d70f/internal\_photos/bs/2022/h/7/YlKvgERrGcE47KUHP5MQ/gettyimages-99311107.jpg'),

('Edmontosaurus', 'Cretáceo', 'Herbívoro', 10.0, 2, 'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhlbYPAv5yT6hcxvKBngjX60x4o6SWfrQW8qUrPZ2FPCRaaGxzG6BRO6VKJdFmGI9p0nou9O7yTDeKehoDX3soRgxffnqGelJm\_kGHv5zoXOOYsqxgwJ8UhaqFSVNgQIKnb-WYJeH8t5XQ6/s1600/Edmontosaurus+-+Michael+W.+Skrepnick.jpg'),

('Velociraptor', 'Cretáceo', 'Carnívoro', 2.0, 3, 'https://play-lh.googleusercontent.com/n1Bxkn-afxfStE90s5iukBja6HRzqU8PEP70pATjL7DLd28ddHETSgnDbaDXT-Q8q1M');

SELECT \* FROM dinossauros

**Imagem Banco de dados tabela:**

****

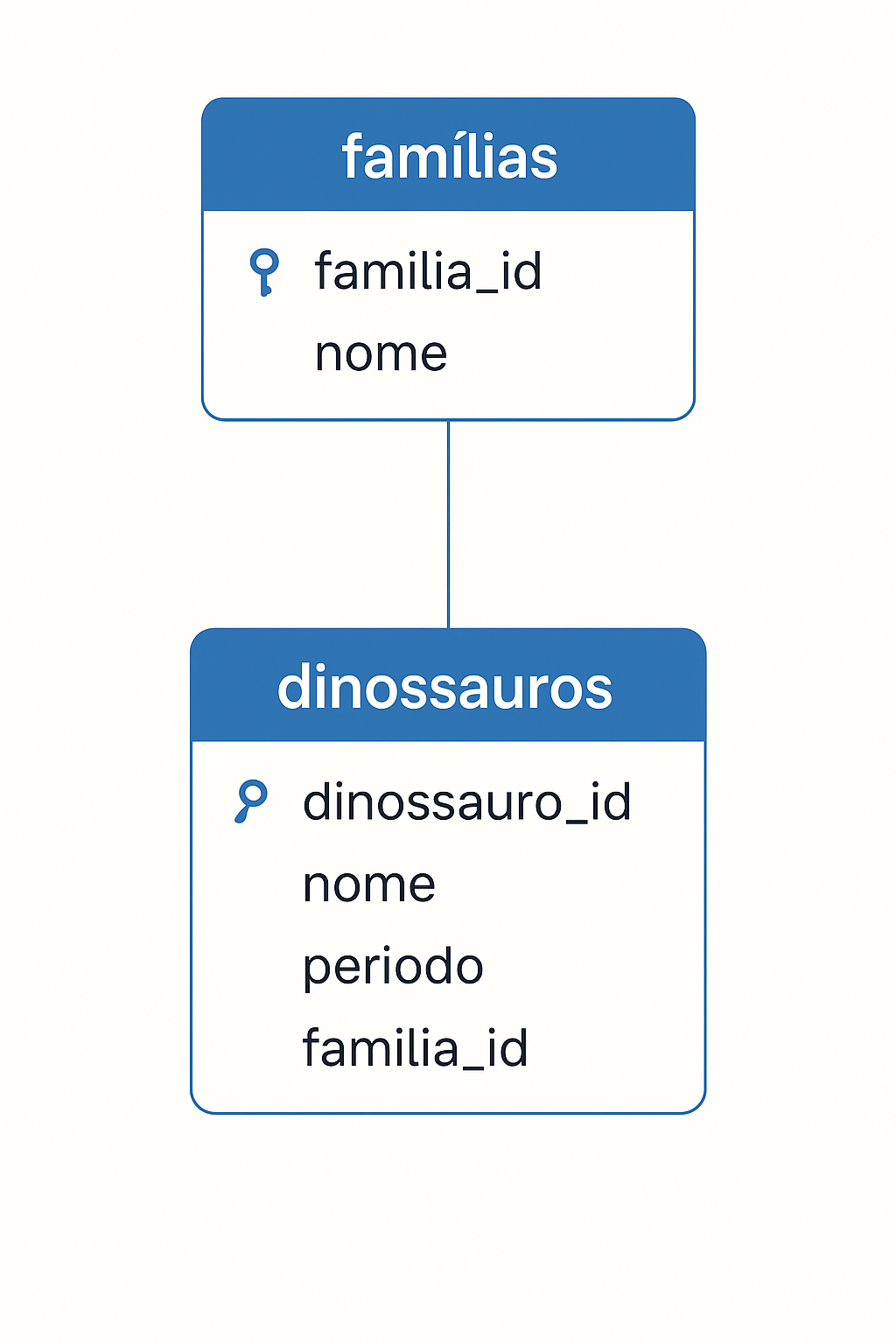
**Imagem Banco de Dados Script**:



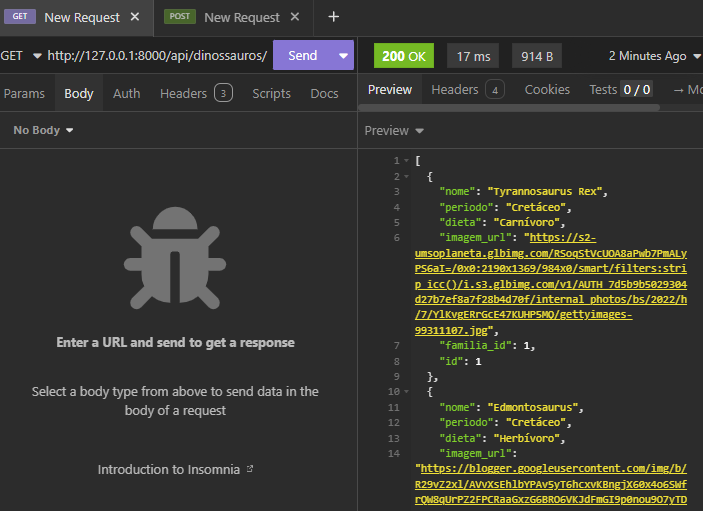
#### **3.2 Estrutura do Sistema**

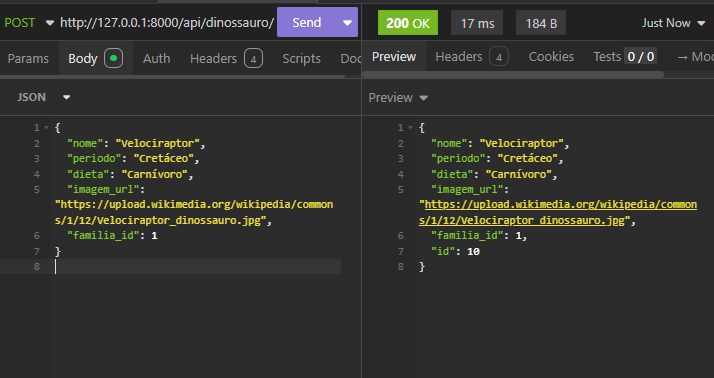
* Modelos: Dinossauro, Familia (com relacionamento entre si);
* Rotas: CRUD completo para ambos os modelos;
* API REST: endpoints definidos sob “/api” para facilitar a integração com sistemas externos.

**Imagem Diagrama de classe:**

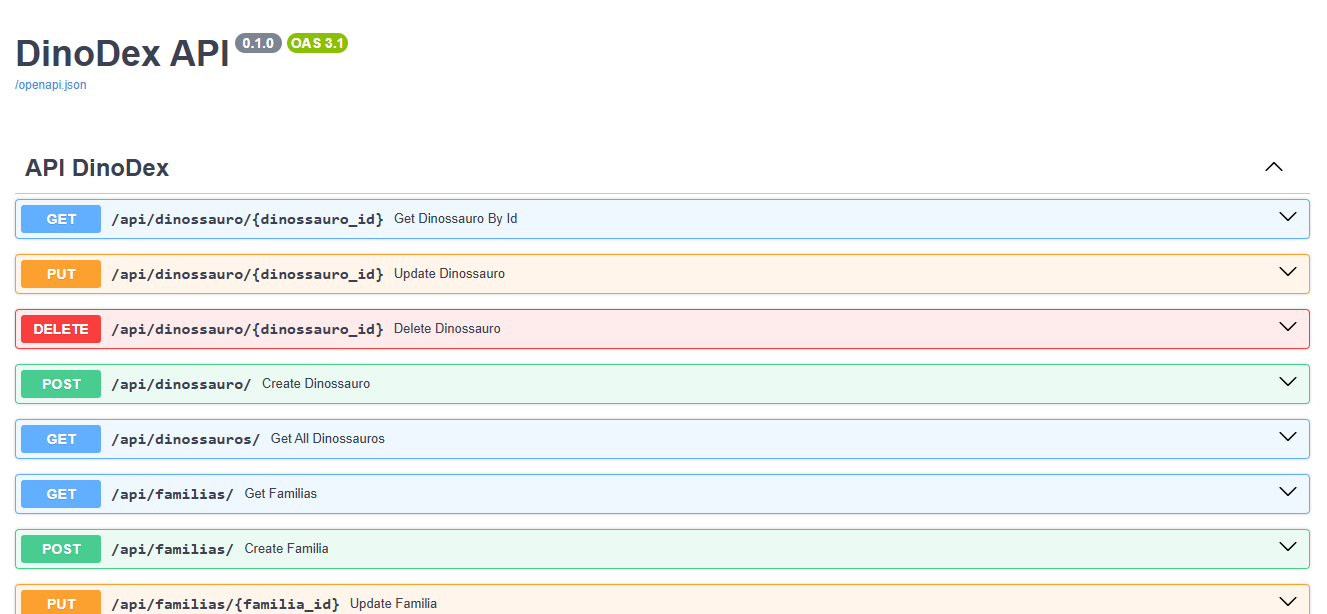


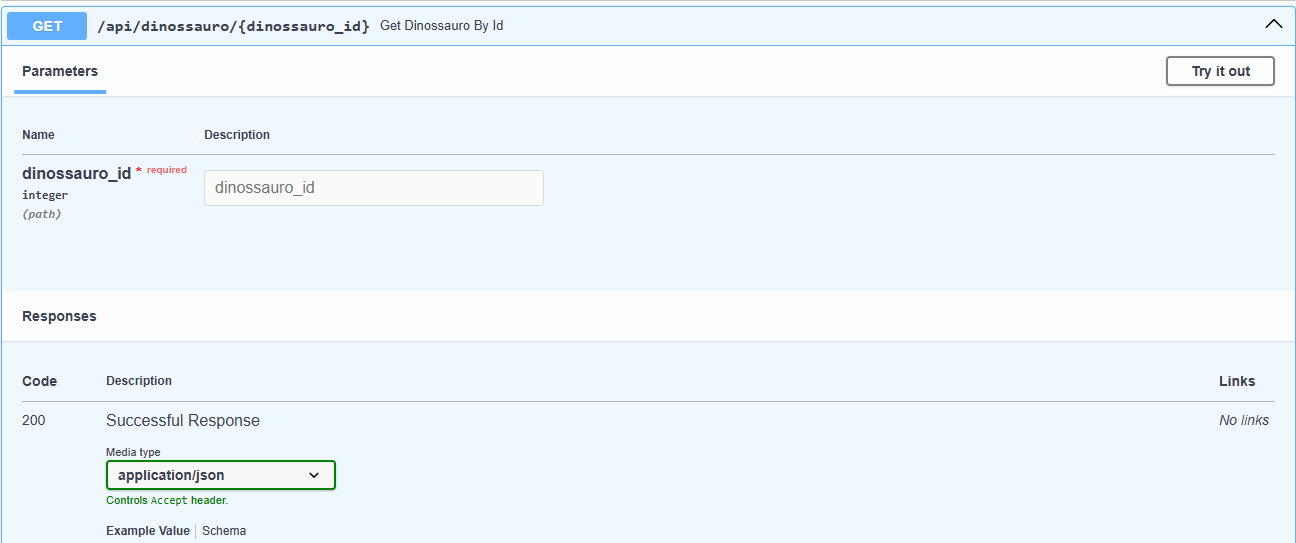
**3.2.1 Teste do Back-end com Insomnia:**

****

****

**3.2.2 Teste do Back-end com Swagger:**

****



### **4. Resultados**

A aplicação foi testada localmente com sucesso. O frontend exibe os dinossauros cadastrados no banco de dados em cartões organizados, com imagens e detalhes descritivos. O usuário também pode adicionar e deletar registros diretamente da interface web, com comunicação fluida com a API backend.

**Imagem Front-end(Interface Web):**

