**Documentação da API REST e Arquitetura**

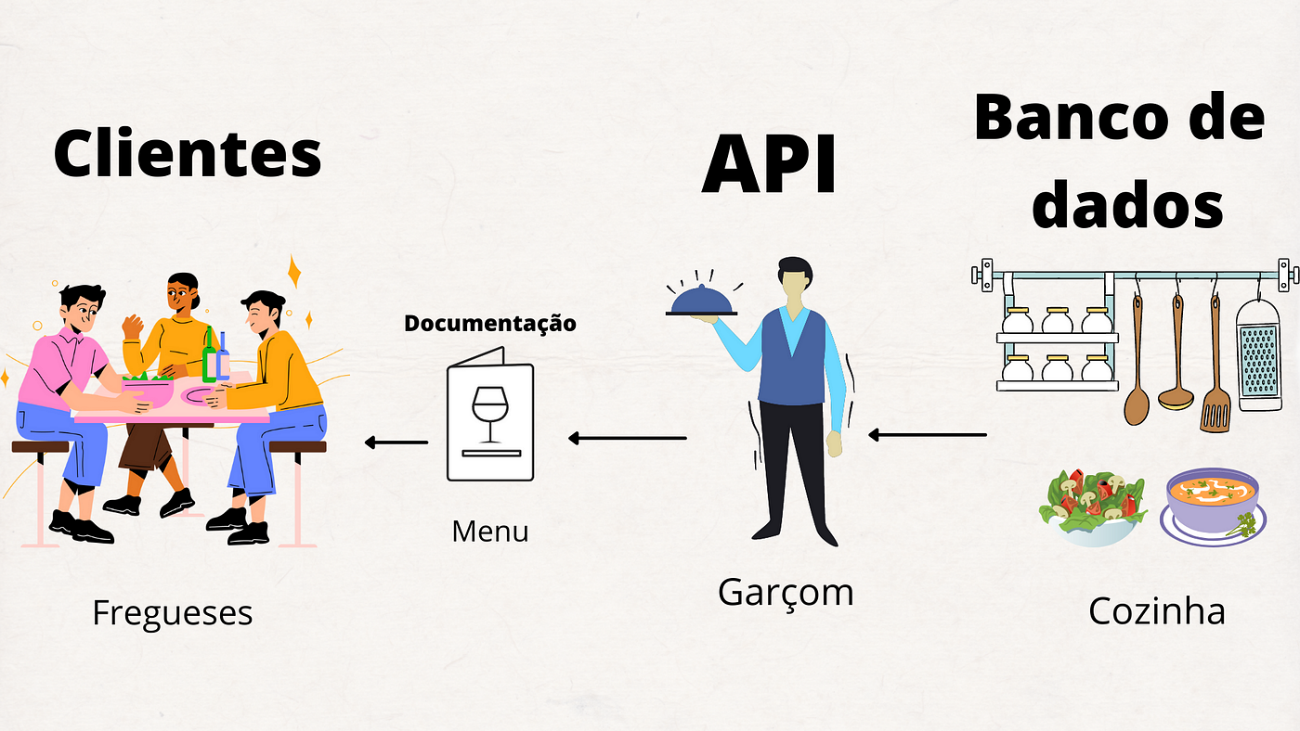
Github do Projeto: <https://github.com/biles83/api_flask>

Ler o README.md para mais instruções sobre o projeto.

**Documentação da API REST**

A API foi desenvolvida em Flask e utiliza o padrão CRUD para manipulação de dados do site da Embrapa nas seguintes rotas: PRODUCAO, COMERCIAL, PROCESSADA, EXPORTACAO e IMPORTACAO.

A API está configurada para funcionar localmente no endereço local [*http://127.0.0.1*](http://127.0.0.1) e também foi feito o deploy via Vercel no endereço [*https://api-flask-mifef5hbw-achiles-projects-02709c7f.vercel.app*](https://api-flask-mifef5hbw-achiles-projects-02709c7f.vercel.app).



A API foi construída para gerenciar dados relacionados a cinco áreas específicas (produção, comercialização, processamento, exportação e importação), cada uma representada por uma rota. Para cada rota, foram implementados o CRUD para criação dos métodos de inclusão (POST), leitura (GET), atualização (PUT) e exclusão (DELETE) de registros.

**Autenticação Básica HTTP**

Esta API utiliza autenticação básica HTTP. Para acessar qualquer um dos endpoints, você precisa fornecer um nome de usuário e senha válidos.

**Endpoints**

Cada endpoint é responsável por uma categoria de dados e possui as seguintes operações:

1. **GET**: Recupera informações de um ou mais registros.
2. **POST**: Cria um novo registro.
3. **PUT**: Atualiza um registro existente.
4. **DELETE**: Exclui um registro específico.

**Estrutura dos Endpoints**

* Base URL: http://127.0.0.1

Cada rota está estruturada da seguinte maneira:

* **PRODUCAO**: /producao
* **COMERCIAL**: /comercial
* **PROCESSADA**: /processada
* **EXPORTACAO**: /exportacao
* **IMPORTACAO**: /importacao

**Endpoints Detalhados**

A seguir, descrevemos as operações possíveis em cada uma das rotas.

**1. Rota: /producao**

Produção de vinhos, sucos e derivados do Rio Grande do Sul.

**GET /producao**

Retorna todos os registros de produção no formato JSON conforme exemplo abaixo.

**Exemplo de resposta:**

[

{"id": 1, "Ano": "2020", "Item": “VINHO DE MESA”, "Sub\_item": "Tinto", "Quantidade": 1000 }

]

**POST /producao**

Cria um novo registro de produção.

**Requisição (JSON):**

    {

        "Id": 0,

        "Ano": "2020",

        "Item": "VINHO DE MESA",

        "Sub\_item": "Tinto",

        " Quantidade ": "1000"

    }

**PUT /producao/<id>**

Atualiza um registro específico de produção. Substituir o <id> pelo Id que deseja atualizar.

**Requisição (JSON):**

    {

        "Id": 0,

        "Ano": "2020",

        "Item": "VINHO DE MESA",

        "Sub\_item": "Tinto",

        " Quantidade ": "1000"

    }

**DELETE /producao/<id>**

Exclui um registro específico de produção. Substituir o <id> pelo Id que deseja deletar.

**2. Rota: /comercial**

Esta rota segue as mesmas operações (GET, POST, PUT, DELETE) que a rota /producao, com foco nos dados relacionados à Comercialização de vinhos e derivados no Rio Grande do Sul.

Exemplo dos dados:

    {

        "Id": 0,

        "Ano": "2020",

        "Item": "VINHO DE MESA",

        "Sub\_item": "Tinto",

        " Quantidade ": "1000"

    }

**3. Rota: /processada**

Esta rota segue as mesmas operações (GET, POST, PUT, DELETE) que a rota /producao, com foco nos dados relacionados à quantidade de uvas processadas no Rio Grande do Sul.

Exemplo dos dados:

    {

        "Id": 0,

        "Ano": "2020",

        "Tipo": "Viníferas",

        "Item": "TINTAS",

        "Sub\_item": "Ancelota",

        " Quantidade ": "1000"

    }

**4. Rota: /exportacao**

Esta rota segue as mesmas operações (GET, POST, PUT, DELETE) que a rota /producao, com foco nos dados relacionados à Exportação de derivados de uva.

Exemplo dos dados:

    {

        "Id": 0,

        "Ano": "2020",

        "Tipo": " Vinhos de mesa",

        "País": " Canadá",

        "Valor": 10000,

        " Quantidade ": "100"

    }

**5. Rota: /importacao**

Esta rota segue as mesmas operações (GET, POST, PUT, DELETE) que a rota /producao, com foco nos dados relacionados à Importação de derivados de uva.

Exemplo dos dados:

    {

        "Id": 0,

        "Ano": "2020",

        "Tipo": " Vinhos de mesa",

        "País": " Canadá",

        "Valor": 10000,

        " Quantidade ": "100"

    }

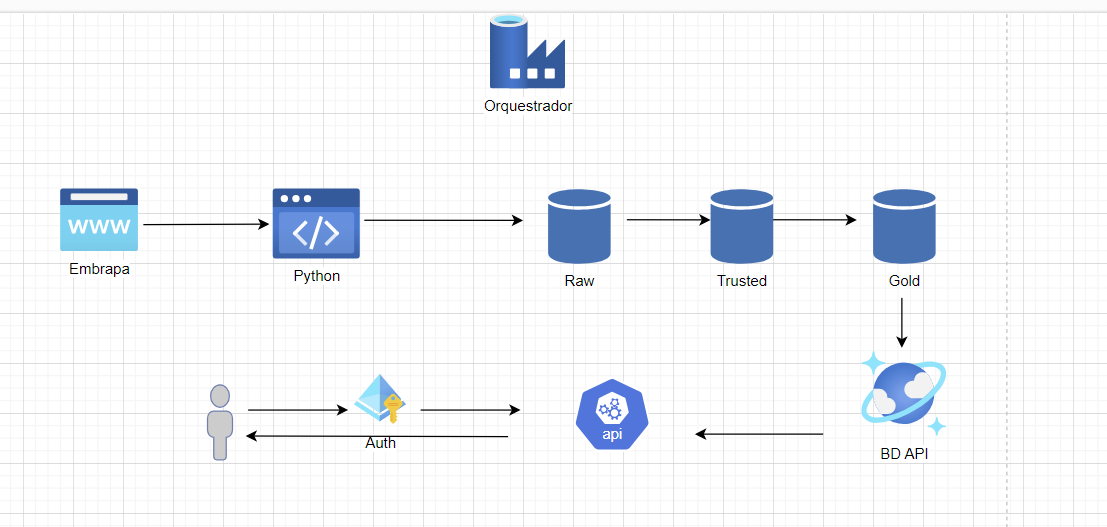
**Arquitetura**

Podemos usar uma arquitetura Medallion para organizar dados e fluxos de trabalho em projetos de machine learning e analytics. A estrutura em três camadas que organiza e prepara os dados em diferentes níveis de qualidade e processamento. Essas camadas são:

1. **Bronze (Dados Brutos)**
2. **Silver (Dados Filtrados e Processados)**
3. **Gold (Dados Prontos para Análise)**

O fluxo de dados segue uma sequência onde os dados migram da camada Bronze para a Silver e, finalmente, para a Gold. Em cada estágio, as transformações aumentam a qualidade, utilidade e especificidade dos dados.

Essa estrutura é comum em arquiteturas de dados em nuvem, especialmente com ferramentas como Delta Lake. A arquitetura Medallion facilita o pipeline de dados e permite o versionamento, tornando-a ideal para ambientes onde dados mudam constantemente e modelos de machine learning são refinados com frequência.

****

**REAMDE.md**