

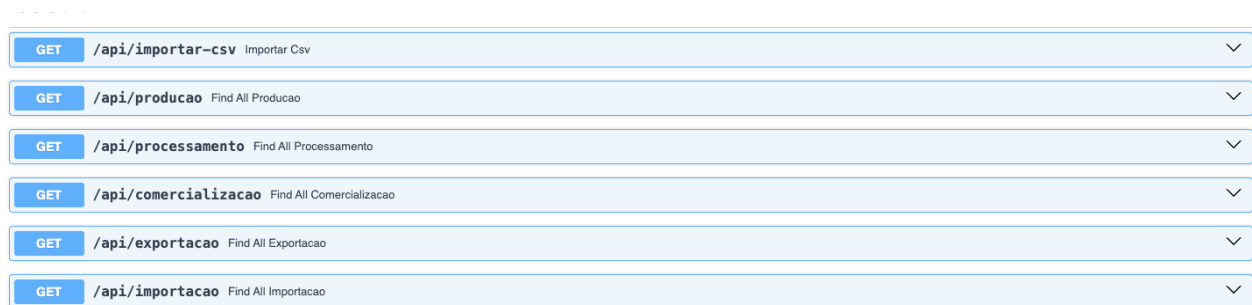
# Tech Challenge

Criação de uma API pública para consumo dos dados fornecidos pela Embrapa.

Desenvolvido com Python e FastAPI, é possível encontrar o código fonte no link: [https://github.com/muhlucas/embrapa\\_tech\\_challenge\\_fiap](https://github.com/muhlucas/embrapa_tech_challenge_fiap)

A aplicação encontra-se hospedada na AWS no link: <https://utq9zejz3b.us-east-1.awsapprunner.com/docs>

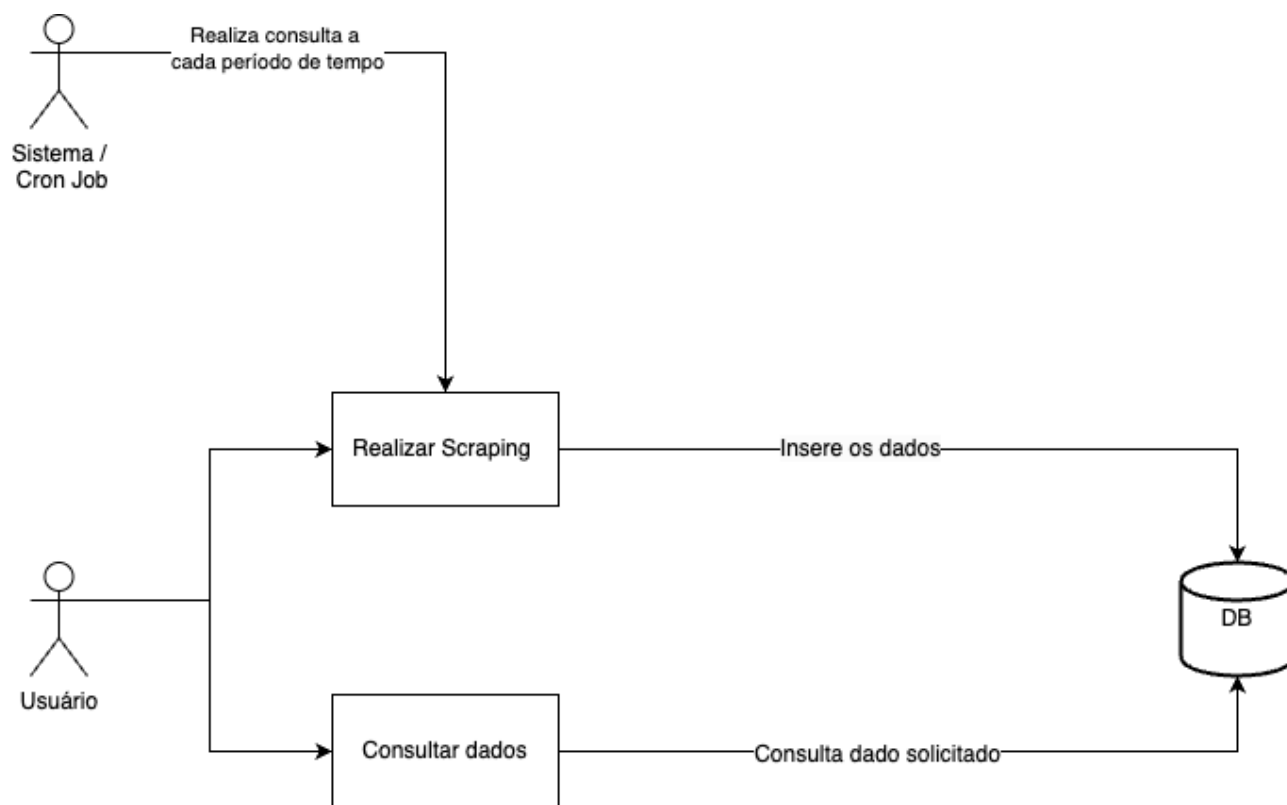
Entre no link <https://utq9zejz3b.us-east-1.awsapprunner.com/docs> para acesso à documentação da API.



GET	/api/importar-csv	Importar Csv	▼
GET	/api/producao	Find All Producao	▼
GET	/api/processamento	Find All Processamento	▼
GET	/api/comercializacao	Find All Comercializacao	▼
GET	/api/exportacao	Find All Exportacao	▼
GET	/api/importacao	Find All Importacao	▼

1.0 imagem documentação da api rest

# Funcionamento da Aplicação



1.1 Diagrama de funciona

Para o funcionamento adequado da aplicação é necessário inserir as informações iniciais no banco de dados, e para isso é possível realizar chamando a rota da api rest ***api/importação-csv***.

Após realizado a inserção inicial o sistema pode deixar um cron sob realizando a tarefa de atualização das informações a cada Período de tempo pré-determinado. Ex: a cada mês.

Com os dados inseridos é possível realizar as consultas na API por meio das demais rotas. Ex: `api/producao`

É possível realizar diversos filtros em todas as rotas:

*Para mais informações consulte a documentação da ultima versão direto no link da api.*

Filtrar por ano que contém '2023': `/api/producao?q=ano__contains=2023`

Filtrar por quantidade que começa com '100': `/api/producao?q=qtd__startswith=100`

Filtrar por control que contém 'de': `/api/producao?q=control__contains=de`

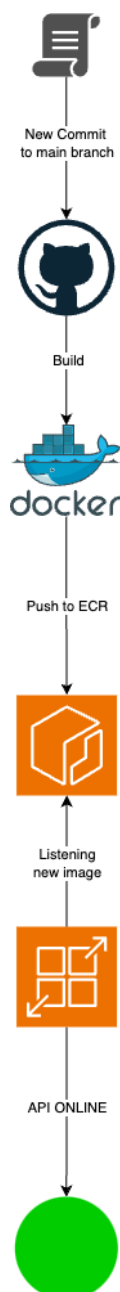
# CI/CD

Para atualização constante e funcionamento do deploy da aplicação, utilizamos o Github Actions onde a cada novo commit na **main** é iniciado o processo de integração e deploy da aplicação.

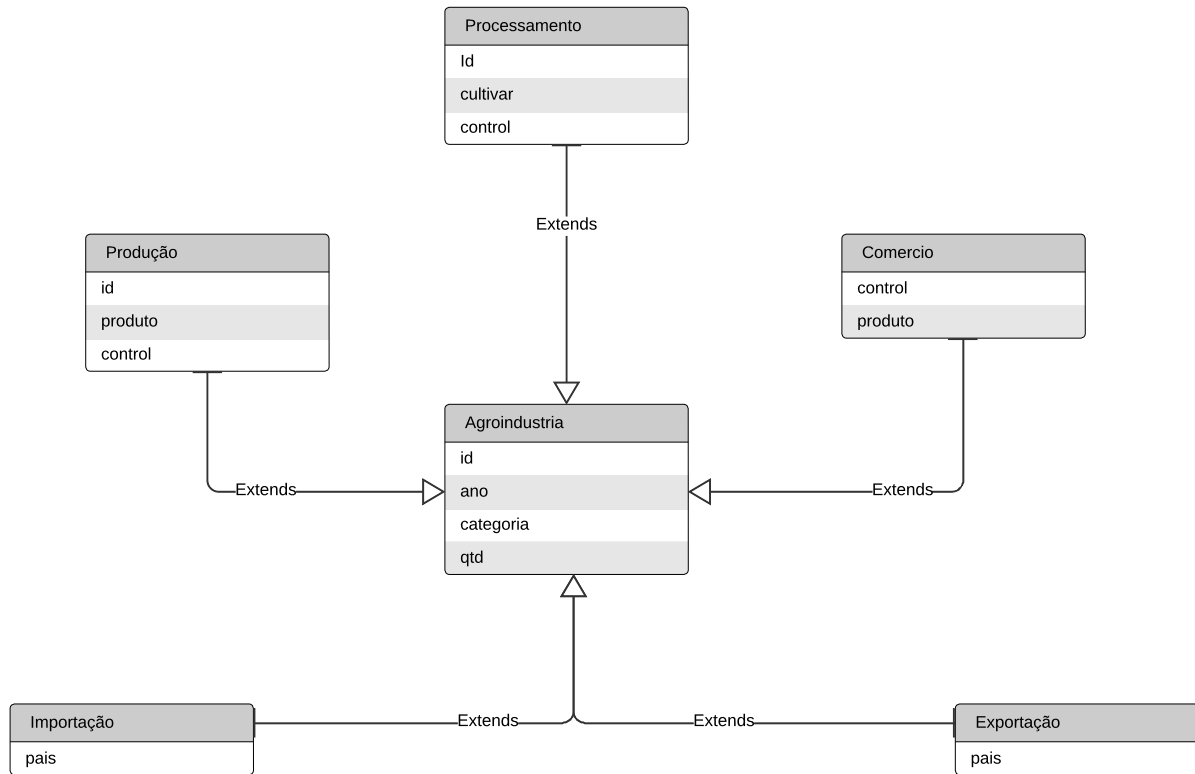
Após o envio de um novo commit para a branch main, é criado um Build utilizando o Docker.

Com o sucesso do Build é registrado um novo contêiner Docker no serviço Elastic Container Registry (ECR) com a imagem da ultima versão do código.

O serviço App Runner da AWS automaticamente é notificado de uma nova versão da imagem do ECR e atualiza a aplicação para a ultima versão.



# Diagrama de Banco de Dados



## 1.3 Diagrama de Banco de dados

O Mapeamento dos dados foram divididos em tabelas do mesmo domínio com normalização dos dados. Facilitando a forma de mapear a API e na consulta de dados com os filtros