



Tech Challenge 1 - Books API

Pipeline Completo de Dados



TECH CHALLENGE - FASE 1

MACHINE LEARNING ENGINEERING

Books API - Pipeline Completo de Dados

Aluno: **Guilherme Favaron** ([github @guifav](#))

Uma API pública escalável para consulta de livros

Visão Geral do Projeto

Objetivo Principal

Pipeline completo de dados para Machine Learning

Características

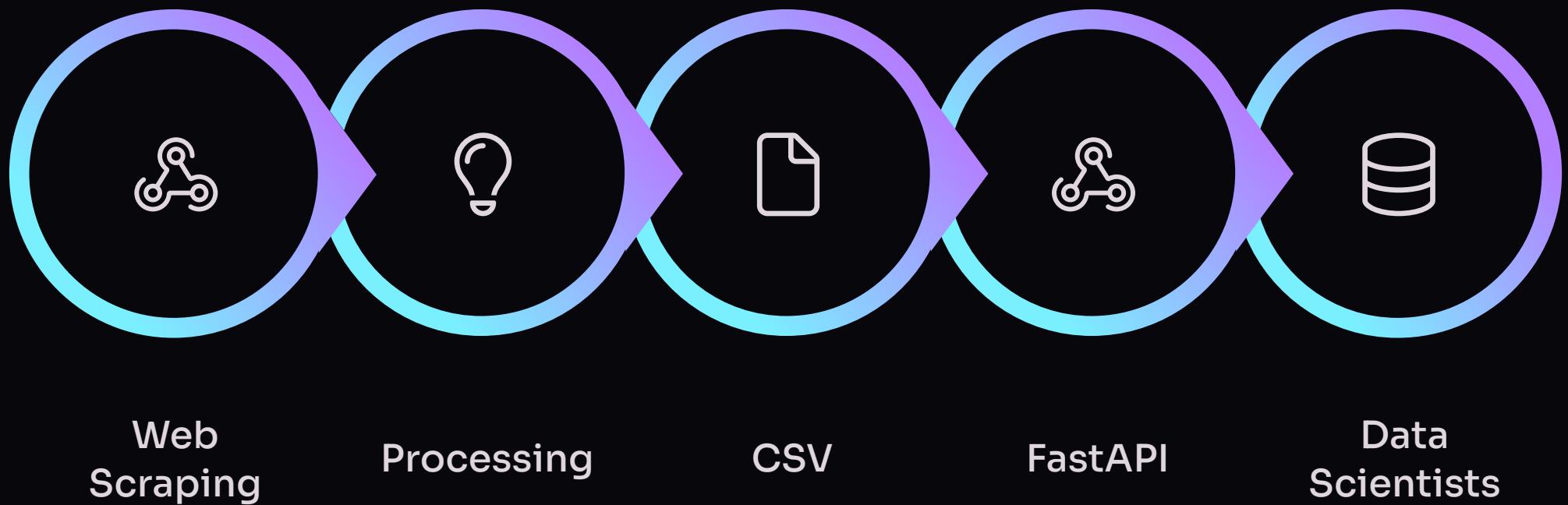
- Web scraping automatizado
- API RESTful em produção
- Documentação completa
- Arquitetura escalável

Fonte dos Dados

books.toscrape.com - 1000+ livros

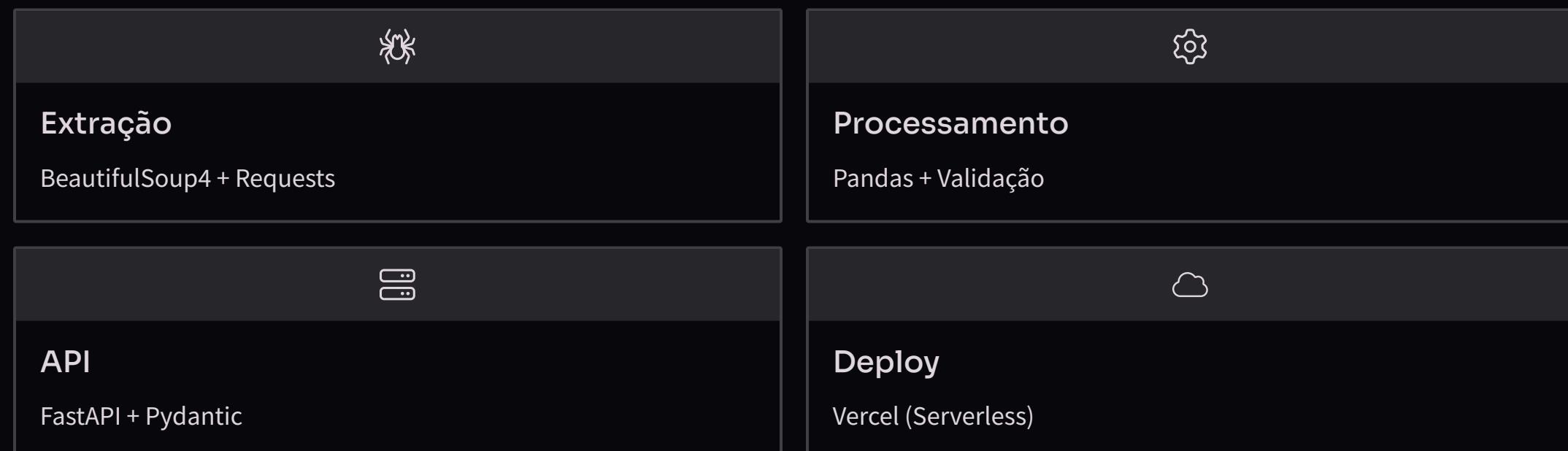


Arquitetura do Sistema



Web Scraping → Processamento → CSV → FastAPI → Cientistas de Dados

Pipeline: Ingestão → Processamento → API → Consumo



Web Scraping - Ingestão

Extração de Dados

Fonte: books.toscrape.com

Dados Coletados:

- Título do livro
- Preço (£)
- Rating (1-5 estrelas)
- Disponibilidade
- Categoria
- URL da imagem



Características Robustas:

- Tratamento de erros HTTP
- Rate limiting
- Validação de dados
- IDs únicos gerados

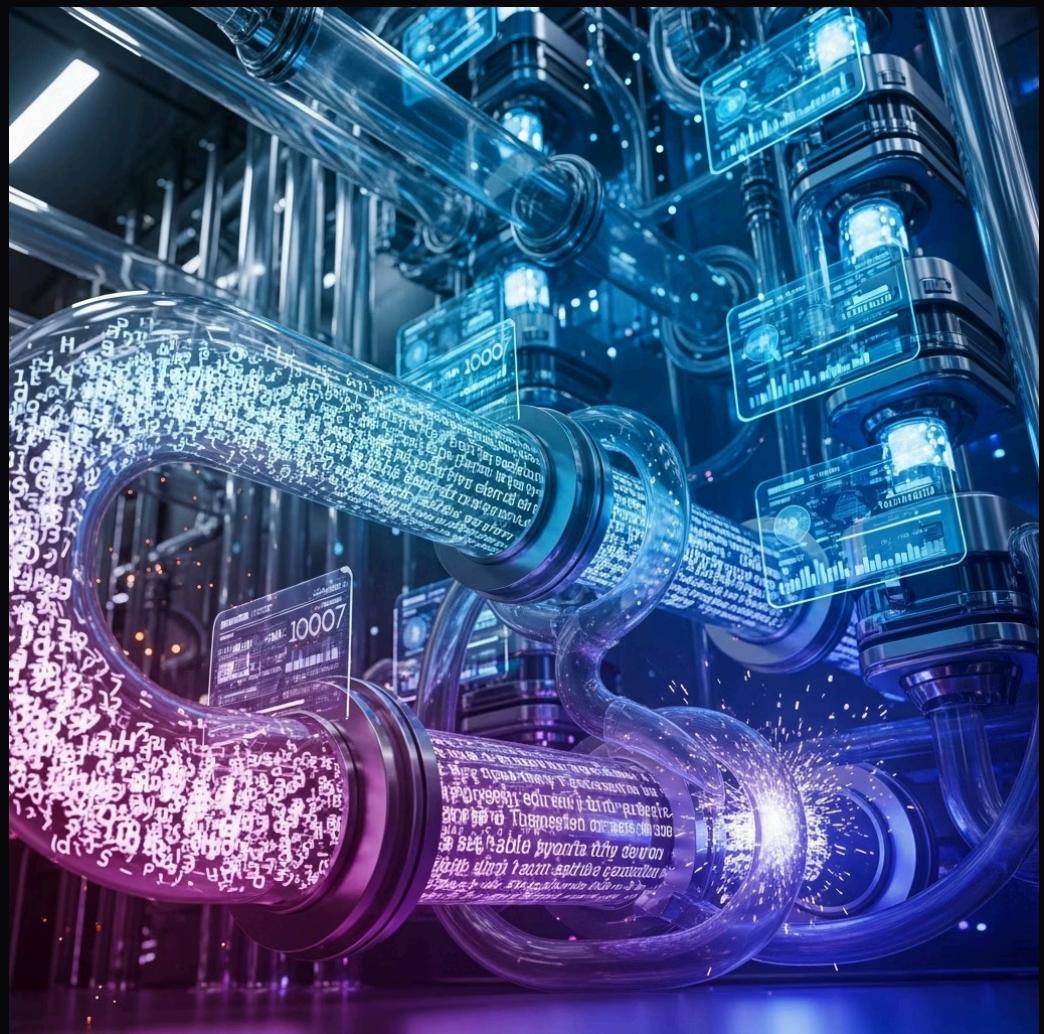
Processamento de Dados

Limpeza e Validação

- Normalização de preços
- Padronização de ratings
- Validação de URLs
- Detecção de duplicatas

Armazenamento

- **Formato:** CSV estruturado
- **Localização:** /data/books_data.csv
- **Qualidade:** Dados limpos e validados



Resultados

1000+

Livros

processados

50+

Categorias

identificadas

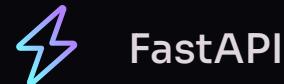
0%

Dados

corrompidos

API FastAPI

Tecnologias Utilizadas



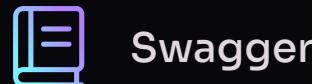
FastAPI

Framework moderno e rápido



Pydantic

Validação automática



Swagger

Documentação interativa



CORS

Acesso público configurado

Endpoints Principais

- GET /api/v1/health - Status da API
- GET /api/v1/books - Listagem paginada
- GET /api/v1/search - Busca inteligente
- GET /api/v1/stats/* - Dados para ML
- GET /api/v1/categories - Categorias disponíveis



Demonstração - Health Check

Endpoint: /api/v1/health

```
{  
  "status": "healthy",  
  "total_books": 1000,  
  "categories": 50,  
  "last_updated": "2024-08-03"  
}
```

Funcionalidades

- Verificação do status da API
- Contagem total de livros
- Informações do sistema



Demonstração - Listagem

Endpoint: /api/v1/books



Parâmetros:

- page: Número da página
- limit: Livros por página

Resposta:

```
{  
  "books": [  
    {  
      "id": "book_1",  
      "title": "A Light in the Attic",  
      "price": 51.77,  
      "rating": 3,  
      "category": "Poetry",  
      "availability": "In stock"  
    }  
  "pagination": {...}  
}
```

Demonstração - Busca

Endpoint: /api/v1/search

Filtros Disponíveis:

- title: Busca no título
- category: Filtro por categoria
- min_price / max_price: Faixa de preço
- rating: Rating mínimo

Exemplo:

/api/v1/search?title=light&category=poetry



Ideal para sistemas de recomendação

Estatísticas para ML

Endpoint: /api/v1/stats/overview

```
{  
  "total_books": 1000,  
  "avg_price": 35.68,  
  "rating_distribution": {  
    "1": 68,  
    "2": 151,  
    "3": 264,  
    "4": 292,  
    "5": 225  
  },  
  "price_ranges": {  
    "0-20": 394,  
    "20-40": 336,  
    "40-60": 270  
  }  
}
```



Dados agregados essenciais para Data Science

Casos de Uso - Machine Learning

Principais Aplicações

1

Análise Exploratória (EDA)

- Distribuições de preço e rating
- Análise por categoria
- Correlações entre variáveis

2

Sistemas de Recomendação

- Filtragem colaborativa
- Filtragem por conteúdo
- Modelos híbridos

3

Predição de Preços

- Baseada em categoria e rating
- Detecção de anomalias

4

NLP - Processamento de Linguagem

- Análise de sentimento em títulos
- Classificação automática

Exemplo de Uso - Código

Integração com Python

```
import requests  
import pandas as pd  
  
# Obter dados da API  
response = requests.get(  
    'https://tech-challenger-1.vercel.app/api/v1/books?limit=1000'  
)  
books_df = pd.DataFrame(response.json()['books'])  
  
# Análises possíveis  
books_df.groupby('category')['rating'].mean()  
books_df['price'].describe()  
  
# Visualizações  
books_df.hist(['price', 'rating'])
```



Integração direta com ferramentas de Data Science

Arquitetura Futura - Escalabilidade

Evolução Planejada



Atual

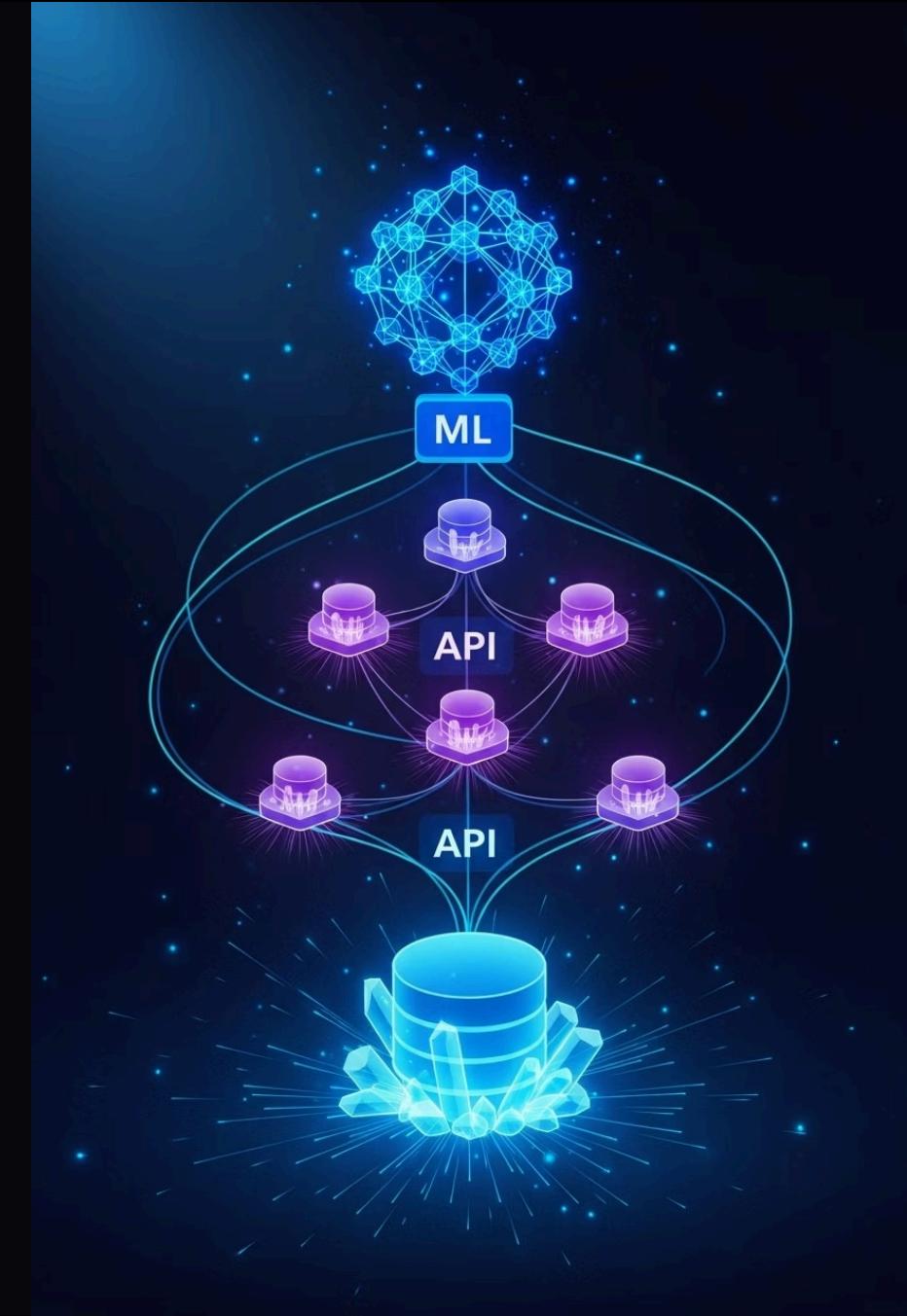
CSV → FastAPI → Vercel

Futuro

PostgreSQL + Redis → ML Pipeline → A/B Testing

Próximos Passos

- **Database:** PostgreSQL + Redis
- **Search:** Elasticsearch
- **ML:** Feature Store + Model Registry
- **Monitoring:** Métricas em tempo real



Integração ML - Endpoints Futuros

Endpoints de Machine Learning Planejados

/api/v1/ml/recommendations

```
{  
    "user_id": "user123",  
    "preferences": ["fiction",  
    "mystery"],  
    "recommendations": [...]  
}
```

/api/v1/ml/predict-prices

```
{  
    "title": "New Book",  
    "category": "fiction",  
    "predicted_price": 25.99  
}
```

/api/v1/ml/sentiment

```
{  
    "text": "Amazing book!",  
    "sentiment": "positive",  
    "confidence": 0.89  
}
```



Stack Tecnológico

Tecnologias Utilizadas

Backend

- **FastAPI** - Framework web moderno
- **Pydantic** - Validação de dados
- **Pandas** - Processamento de dados



Scraping

- **BeautifulSoup4** - Parser HTML
- **Requests** - Cliente HTTP

Deploy & Infra

- **Vercel** - Serverless deployment
- **GitHub** - Controle de versão
- **Swagger** - Documentação

Boas Práticas Implementadas

Qualidade do Código

Validação

- Pydantic models para entrada/saída
- Tratamento robusto de erros
- Validação de tipos

Performance

- Paginação em todas as listagens
- Queries otimizadas
- Cache-friendly design

Documentação

- Swagger UI automático
- Exemplos de uso
- Código bem documentado

Segurança

- CORS configurado
- Rate limiting
- Validação de entrada



Resultados Alcançados

Métricas de Sucesso

Dados:

- **1000+ livros** extraídos e processados
- **50+ categorias** identificadas
- **100% dados válidos**

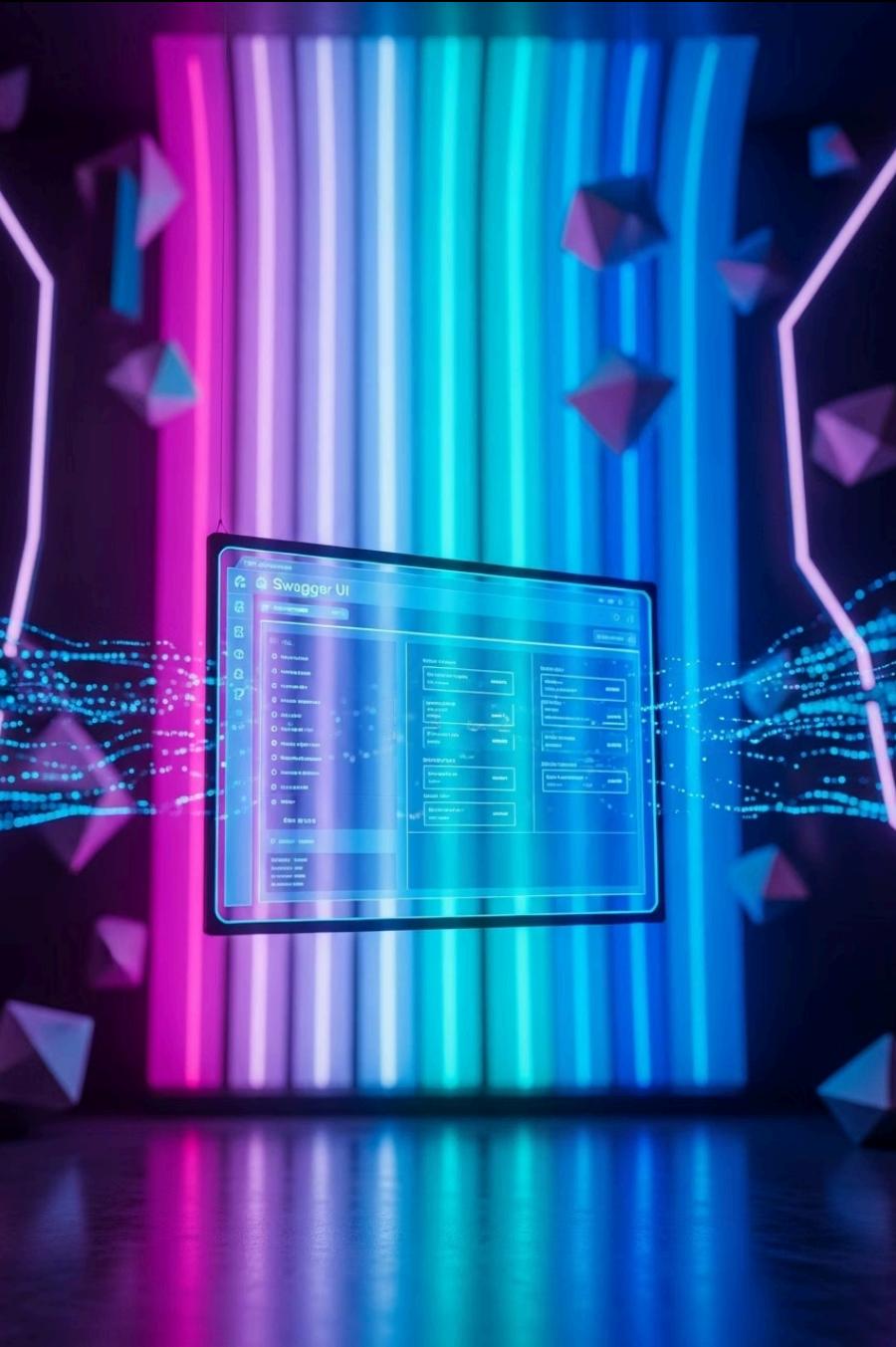
API:

- **8 endpoints** funcionais
- API **100% operacional** em produção
- **Documentação completa** com Swagger

Infraestrutura:

- **Deploy automatizado** no Vercel
- **Código versionado** no GitHub
- **Pipeline de dados** robusto





Demonstração Final

Links para Teste

API em Produção:

tech-challenger-1.vercel.app

Documentação Interativa:

</api/docs>

Repositório GitHub:

github.com/guifav/tech_challenger_1

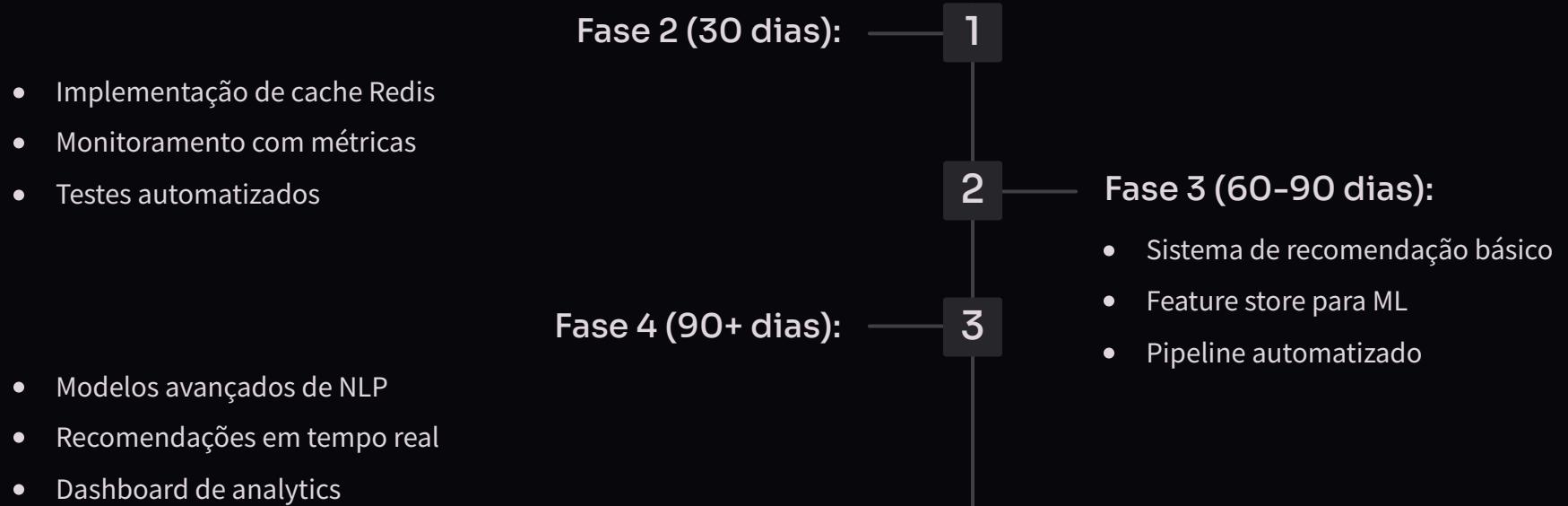
Dataset:

books_data.csv



Possíveis Próximos Passos

Roadmap de Desenvolvimento



Conclusão

Entrega Completa



Web scraping

funcional e robusto



API RESTful

completa em produção



Deploy operacional

no Vercel



Documentação técnica

detalhada



Arquitetura escalável

para ML

Ready for Machine Learning!

A API está pronta para ser consumida por cientistas de dados e pode evoluir facilmente para um sistema completo de recomendação.

Obrigado!

Contato

Guilherme Favaron

github @guifav

Links Importantes

- API: tech-challenger-1.vercel.app
- Docs: </api/docs>
- GitHub: [repositório](#)

Perguntas?