

Machine Learning Engineer

# Python para ML e IA

Por que a pessoa engenheira de machine learning precisa saber APIs?

Leonardo Pena

# Objetivo dessa terceira parte



Passo 1

Aprender  
boas práticas  
de  
desenvolvime  
nto de APIs



Passo 2

Entender  
autenticação e  
autorização



Passo  
3

Conhecer  
ferramentas  
para  
documentaç  
ão e  
versionamen  
to



Passo 4

Discutir  
escalabilidad  
e, custo e  
segurança



Passo  
5

Olhar para o  
futuro das  
APIs no  
MLOps



Boas práticas para desenvolvimento

## / O que podemos fazer



- Código limpo e organizado por rotas
- Padrões claros de nomenclatura
- Testes unitários e de integração
- Manter a lógica de negócio separada do código de rota
- Evitar “tudo em um único arquivo”



## / O que podemos fazer

- Uso de **ambientes virtuais** e **gerenciadores de dependência**
- Lidar com **exceções** e **erros** de forma padronizada
- **Logar** eventos críticos e respostas do servidor
- Garantir retornos consistentes em JSON
- Reduzir latência com otimizações simples





# Autenticação e Autorização

## / O que é



**Autenticação** verifica a identidade do usuário ou sistema que acessa a API, garantindo que é quem afirma ser, geralmente por meio de tokens, chaves de API ou credenciais.

**Autorização**, por outro lado, define os privilégios e permissões do usuário autenticado, controlando quais recursos ele pode acessar ou quais ações pode realizar.

# / Autenticação e Autorização

- Token baseado (JWT) ou OAuth2 como práticas comuns
- Proteção de endpoints sensíveis (ex.: /train, /delete)
- Controle de permissões para diferentes perfis de usuário
- HTTPS para criptografar o tráfego
- Prevenir acesso indevido a dados e modelos





# Versionamento e Documentação

# / Versionamento da API

- Evitar “quebrar” clientes antigos
- Adicionar versões em endpoints (ex.: /v1, /v2)
- Planejar ciclo de depreciação de versões obsoletas
- Comunicar mudanças via documentação
- Migrar gradualmente clientes para a nova versão

## / Documentação com Swagger/OpenAPI

- Facilita a vida de quem consome a API
- Gera documentação interativa e atualizada
- Possibilita testes de requisições direto na interface
- Ajuda a padronizar a descrição dos endpoints
- Reduz a curva de aprendizagem de novos desenvolvedores



Swagger™



# Ferramentas e Arquiteturas

# / Opções de framework

**Fast API:** Framework moderno e rápido para APIs em Python, com validação automática de dados, suporte assíncrono, e documentação interativa integrada.



**Flask:** Framework leve e flexível que permite criação rápida de APIs RESTful, ideal para prototipagem e projetos simples de machine learning.



**Django REST Framework (DRF):** Extensão robusta do Django para APIs, oferecendo segurança, autenticação, e uma infraestrutura escalável para aplicações mais complexas.







# Desafios em produção

## / Latência

Ocorre quando o tempo de resposta é elevado, impactando a experiência do usuário e a eficiência do sistema.

Isso pode ser causado por modelos complexos, alta demanda computacional ou gargalos em rede e infraestrutura.

Para mitigar, é essencial otimizar modelos, utilizar cache, balanceadores de carga e técnicas como quantização, além de monitorar continuamente o desempenho.

## / Escalabilidade

Envolve a capacidade de lidar com aumentos na demanda por requisições sem comprometer o desempenho.

Soluções incluem o uso de auto-scaling em nuvem, balanceadores de carga, containerização e microsserviços, permitindo distribuir cargas e adaptar recursos dinamicamente para manter a eficiência e disponibilidade.

## / Segurança

Está relacionado à proteção contra acessos não autorizados, vazamento de dados sensíveis e ataques como injeção de código e negação de serviço (DoS). Soluções incluem autenticação robusta com tokens (JWT, OAuth2), criptografia de dados em trânsito (HTTPS), rate limiting, firewalls de aplicação (WAF) e conformidade com regulamentações como LGPD e GDPR.

## / Custo

Está relacionado ao uso ineficiente de recursos computacionais, como instâncias de nuvem e GPUs, que podem gerar despesas elevadas. Soluções incluem o uso de recursos sob demanda, auto-scaling, otimização de modelos para reduzir complexidade, monitoramento de métricas para análise de custo-benefício e evitar o superdimensionamento ou o uso excessivo de modelos grandes e pesados.



# / Conclusão

- Importância das APIs para engenheiros de ML
- Pronto para aprofundar e praticar com frameworks e exemplos reais
- Lembre-se de documentar, testar e monitorar sempre
- APIs permitem que seu modelo impacte o mundo real
- Próximos passos: aplicações com flask

POSTECH

FIAP + alura

# Conclusão do Vídeo 3

Final da Aula 1: importância das APIs para engenheiros de ML

Pronto para aprofundar e praticar com frameworks e exemplos reais

Lembre-se de documentar, testar e monitorar sempre

APIs permitem que seu modelo impacte o mundo real

Próximos passos: mãos na massa e prática contínua