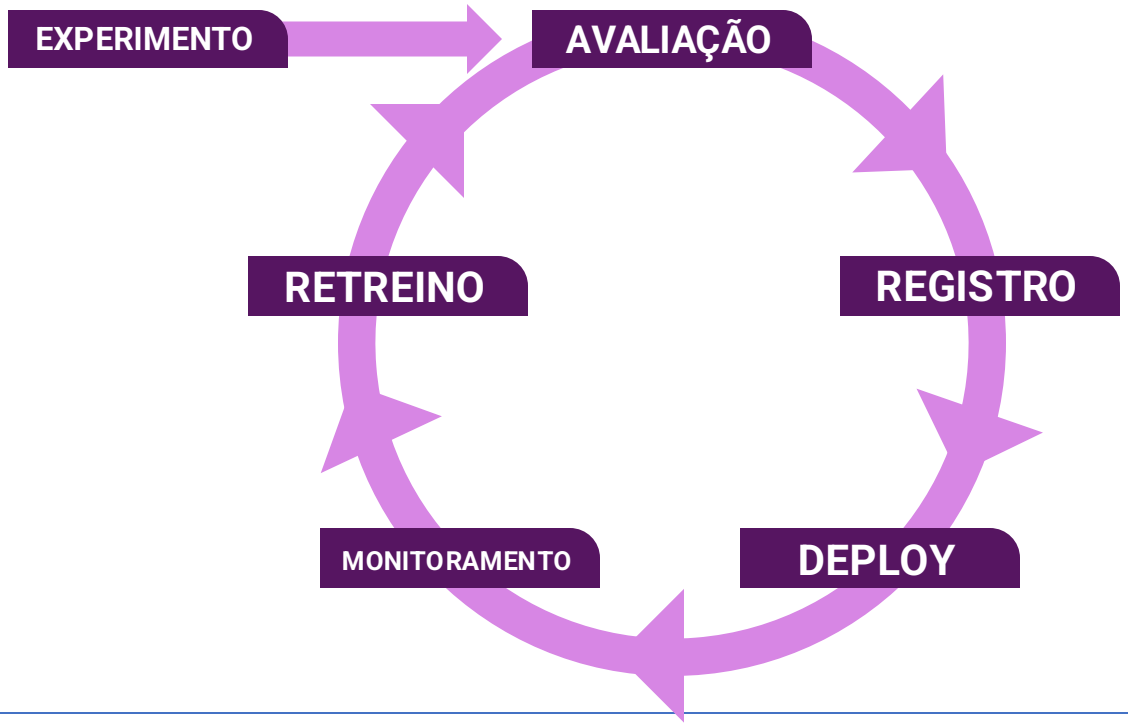
MLOPs: DA CIÊNCIA DE DADOS À OPERAÇÃO

EXPERT ACADEMY

O QUE É MLOPS?

- Disciplina que une Ciência de Dados, Engenharia de Software e Operações para colocar ML em produção com qualidade.
- Trata modelos e dados como artefatos: versionar, testar, empacotar, implantar, monitorar, melhorar.
- Benefícios: desenvolvimento em times, rastreabilidade, reprodutibilidade e governança.

O QUE É MLOPS?



EXPERT ACADEMY

CONTEXTO HISTÓRICO

- Era “artesanal”: notebooks, dados locais, ambientes frágeis; “na minha máquina funciona”.
- Escala e negócio exigiram previsibilidade, governança e escala.
- Herança do DevOps: automação, CI/CD, observabilidade; em ML, soma-se dados e pipelines de treino/inferência.

CONTEXTO HISTÓRICO

- Do manual ao industrial: versionar dados/código/modelos; rastrear runs; registrar, empacotar e promover versões, e orquestrar a execução da pipeline.
- Pilares práticos: tracking remoto de experimentos/modelos, contêineres padronizados, orquestração e agendamento confiáveis.
- Resultado: ciclos mais rápidos, menor risco, custo reduzido e modelos auditáveis em produção.

MLOPS X DEVOPS

- Herança comum: automação, CI/CD, observabilidade.
- Desafios específicos de ML: artefatos, duas pipelines (treino e inferência), evolução dos dados.
- Qualidade e governança: métricas offline e online, testes de dados, validação de distribuição, fairness e robustez.
- Registro/versionamento de modelos; promoção baseada em evidências.
- Observabilidade ampliada: serviço e modelo.
- Conclusão: MLOps = DevOps + dados/modelos

CICLO DE VIDA ORIENTADO A MLOPS

- Ingestão e preparo
- Treino e avaliação
- Registro e empacotamento
- Deploy e operação
- Monitoramento
- Gatilhos, retreino e promoção
- **Cada etapa deixa evidências**

CARACTERÍSTICAS DE UM PROJETO MLOPS

- Reprodutibilidade: pipelines determinísticas, ambientes padronizados.
- Versionamento: código, dados, modelos no registro, imagens Docker com tags coerentes.
- Rastreabilidade: vínculo código → dataset → modelo → métricas.
- Auditoria e governança: histórico, justificativas, rollback e compliance.

CARACTERÍSTICAS DE UM PROJETO MLOPS

- Automação ponta a ponta: etapas do processo são tarefas, explicitar suas dependências, execuções automáticas.
- Orquestração: grafos de execução, agendamento, retries, logs.
- Empacotamento: ambientes imutáveis, imagens versionadas, portabilidade sem surpresa.
- Deploy: criar ou atualizar containers a partir de imagens versionadas.
- Operação contínua: gatilhos para retreino e rollback.

MLOPS

INTRODUÇÃO

2. FERRAMENTAS DO CURSO

FERRAMENTAS E PAPÉIS

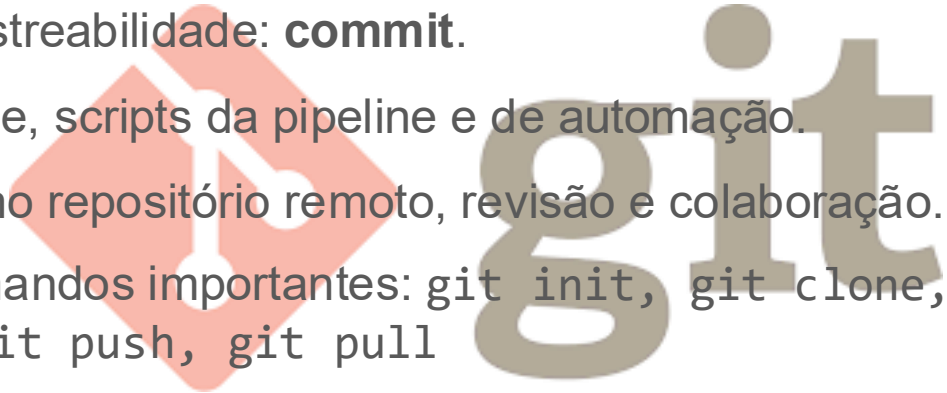
- **Git**: versionamento de código, colaboração e histórico auditável.
- GitHub: servidor remoto para hospedar repositórios versionados por Git.
- **DVC**: versionamento de dados, estágios de pipeline, execuções reprodutíveis.
- **MLflow**: rastreamento de runs, parâmetros, métricas e artefatos, registro/versionamento de modelos.
- DagsHub: servidor remoto que extende a funcionalidade do GitHub para também contemplar DVC e MLflow.

FERRAMENTAS E PAPÉIS

- **Apache Airflow**: orquestração por DAGs, agendamento, logs.
- Docker: empacotamento do ambiente (app + deps); portabilidade entre máquinas/ambientes.
- Docker Hub: servidor remoto para publicação e distribuição de imagens Docker versionadas, base para deploy e promoção.

FERRAMENTAS E PAPÉIS

- Controle de versão: histórico documentado, rastreável e colaborativo.
- Base de rastreabilidade: **commit**.
- Código-fonte, scripts da pipeline e de automação.
- GitHub como repositório remoto, revisão e colaboração.
- Alguns comandos importantes: `git init`, `git clone`, `git add`, `git commit`, `git push`, `git pull`



FERRAMENTAS E PAPÉIS

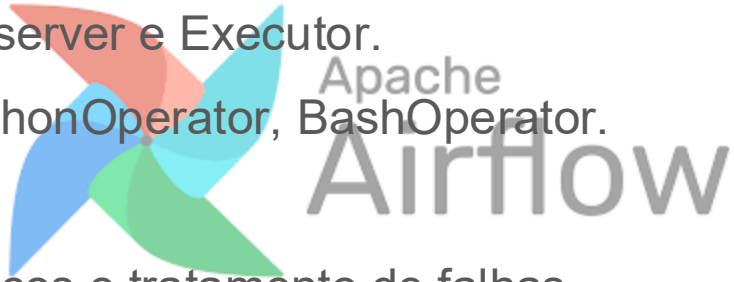
- Estende o versionamento a dados e pipelines.
- `dvc.yaml` define estágios com *deps*, *outs*, *cmd*, *params* e *metrics.*
- `params.yaml` define os valores dos parâmetros.
- `dvc.lock` registra versões e hashes para auditoria.
- Possível configurar vários serviços externos para repositório remoto de dados.
- Alguns comandos importantes:
  - gerenciamento: `dvc init`, `dvc add`, `dvc push`, `dvc pull`
  - execução: `dvc repro`, `dvc exp run`

FERRAMENTAS E PAPÉIS

- Rastreamento de experimentos.
- Registro automático de runs.
- Versionamento de modelos no registro.
- Log de métricas, parâmetros, artefatos.
- Deploy direto de modelos a partir do registro.
- UI web para comparação de resultados.
- Suporte a servidores remotos para time.

FERRAMENTAS E PAPÉIS

- Orquestração programática de workflows.
- DAG (Directed Acyclic Graph) como estrutura central.
- Scheduler, Webserver e Executor.
- Operadores: PythonOperator, BashOperator.
- Agendamento.
- Retries automáticos e tratamento de falhas.
- Interface web: gráficos, logs, status.



MLOPS

INTRODUÇÃO

3. APRESENTAÇÃO DO CURSO

EXPERT ACADEMY

ESCOPO

- Fundamentos práticos de MLOp aplicados a um projeto de classificação supervisionada usando redes neurais.
- Dataset didático: *breast cancer* do scikit-learn.
- Arquitetura do modelo: rede neural de duas camadas.

SEÇÕES PRÁTICAS

- Configuração do ambiente: WSL, Miniconda, Git, VSCode, repositórios remotos.
- Projeto Base
- DVC
- MLflow
- Airflow
- Primeiro ambiente local, depois ambiente remoto.

PROJETO BASE

- Rede neural para classificação de câncer de mama
- Estrutura do projeto modular, boas práticas de engenharia de software.
- Estrutura de pacotes/módulos Python especializados.
- Cinco camadas funcionais: carregamento dos dados, pré-processamento, engenharia de features, treinamento de modelo e avaliação.
- Aplicação via Flask com interface gráfica.
- Deploy com Docker.

PADRÕES E BOAS PRÁTICAS

- src-layout
- `pyproject.toml`
- Nomes autoexplicativos, convenção consistente.
- Logging centralizado.
- Docstrings.
- Type hints.
- Código e documentação em inglês.

NÃO SERÁ ABORDADO

- Pressupõe-se conhecimento prévio em ML.
- Não fazemos otimização de modelos.
- Não tratamos qualidade dos dados.
- Não implementamos testes automatizados.
- Não abordamos processamento paralelo, treinamento distribuído ou inferência em larga escala.
- Não abordamos infraestrutura cloud.
- Não abordamos monitoramento em produção.
- Não implementamos CI/CD.
- Não abordamos tópicos de segurança e governança avançados.

O QUE VOCÊ SAIRÁ SABENDO?

- Tópicos principais:
  - Versionamento e rastreabilidade
  - Automação
  - Orquestração
  - Integração do ciclo completo
  - Entregável completo
- Tópicos secundários:
  - Estruturação e organização de projetos
  - Serving de modelos
  - Conteinerização
  - Execução de serviços conteinerizados
  - Publicação de imagens em repositórios remotos
- Competências profissionais