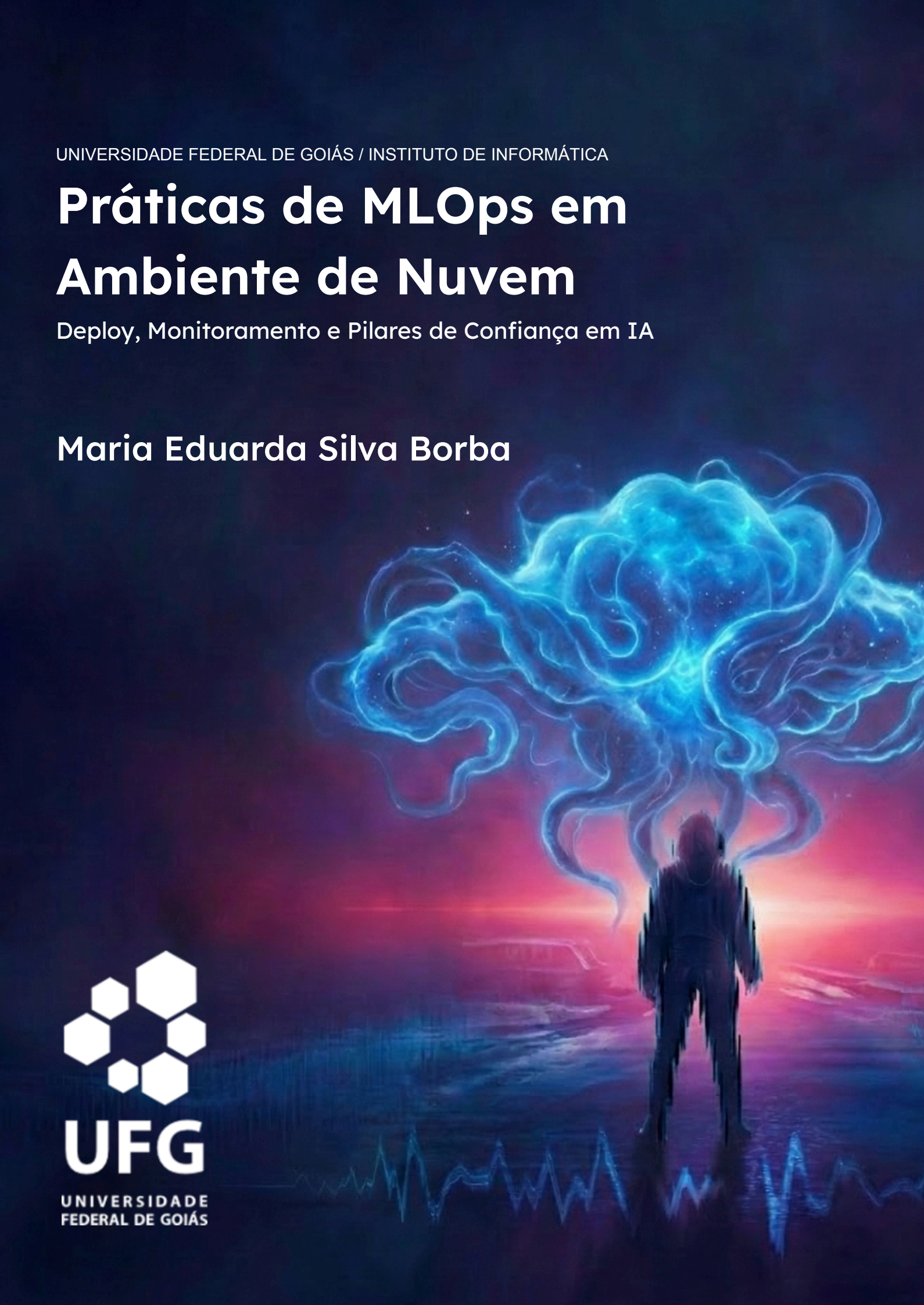
UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Práticas de MLOps em Ambiente de Nuvem

Deploy, Monitoramento e Pilares de Confiança em IA

Maria Eduarda Silva Borba



UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

MARIA EDUARDA SILVA BORBA

Práticas de MLOps em Ambiente de Nuvem

Deploy, Monitoramento e Pilares de Confiança em IA

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): MARIA EDUARDA SILVA BORBA

Título do trabalho: Práticas de MLOps em Ambiente de Nuvem

Deploy, Monitoramento e Pilares de Confiança em IA

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Maria Eduarda Silva Borba**, **Usuário Externo**, em 04/02/2026, às 19:01, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 13/03/2026, às 11:41, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5956785** e o código CRC **B03052D1**.

Referência: Processo nº 23070.005527/2026-40 SEI nº 5956785

MARIA EDUARDA SILVA BORBA

Práticas de MLOps em Ambiente de Nuvem

Deploy, Monitoramento e Pilares de Confiança em IA

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

BORBA, MARIA EDUARDA SILVA
Práticas de MLOps em Ambiente de Nuvem [manuscrito]: Deploy, Monitoramento e Pilares de Confiança em IA / MARIA EDUARDA SILVA BORBA. - 2025.
93 f.: 2025

Orientador: Prof. Dr. Fernando Marques Federson
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. Inteligência Artificial. 2. Mlops. 3. Monitoramento.

I. Federson, Fernando Marques , orient. II. Título.

CDU 004

MARIA EDUARDA SILVA BORBA

Práticas de MLOps em Ambiente de Nuvem
Deploy, Monitoramento e Pilares de Confiança em IA

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 09 de dezembro de 2025.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Prof. Dr. Sávio Salvarino Teles de Oliveira
(INF-UFG)

MARIA EDUARDA SILVA BORBA

Práticas de MLOps em Ambiente de Nuvem

Deploy, Monitoramento e Pilares de Confiança em IA

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **MLOps**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: Inteligência artificial; MLOps; Monitoramento.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **MLOps**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: Artificial intelligence; MLOps; Monitoring.

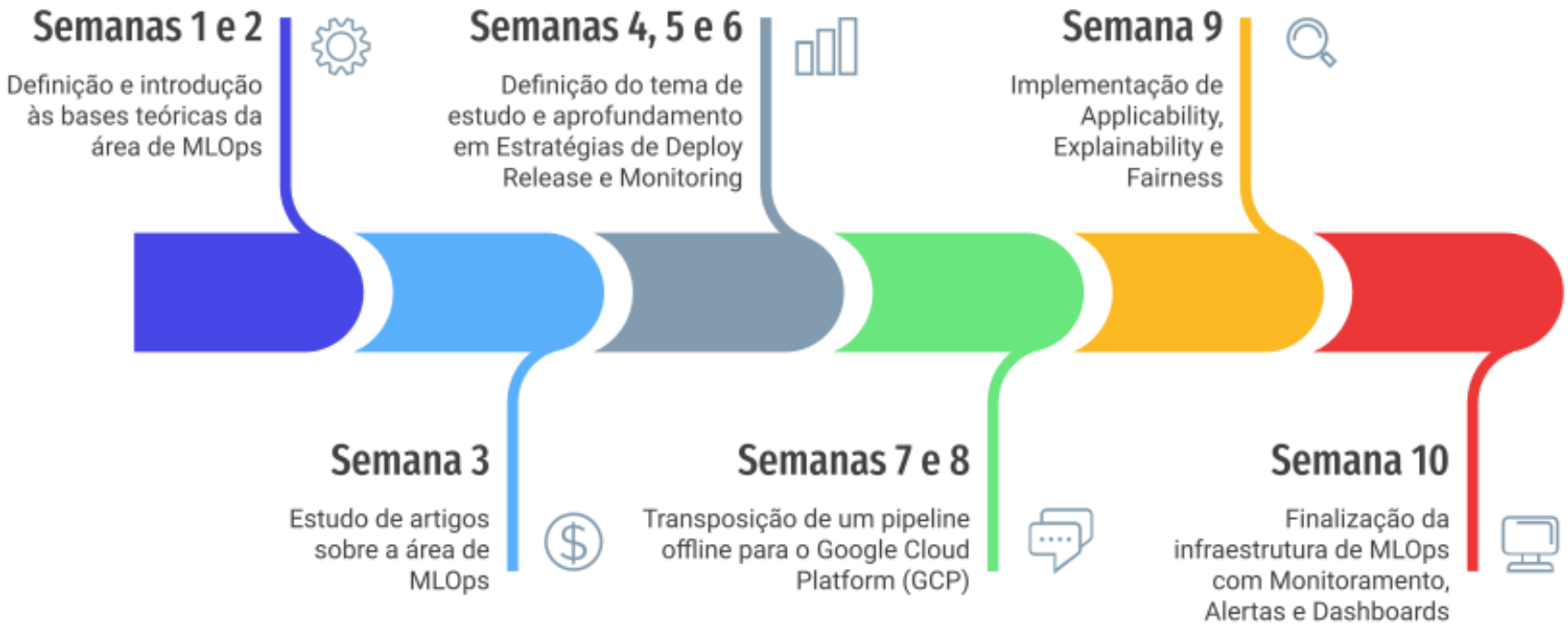
Goiânia

2025

Minha Jornada

Maria Eduarda Silva Borba

Especialista em: MLOps



MINHA JORNADA

Nome: **Maria Eduarda Silva Borba**

Especialidade: **MLOps**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha Jornada começou na busca pela área de estudo: afinal, o que me motivaria pelas próximas **Semanas**? Para responder a isso, busquei na memória e no coração as disciplinas/assuntos que mais me despertaram interesse durante o Bacharelado. Foi um processo difícil, mas, entre as opções, destaquei a disciplina de Processamento de Dados Massivos. Foi ali que surgiu o interesse que se consolidou na **Semana 1** com o tema de MLOps (*Machine Learning Operations*).

Na **Semana 1**, realizei um esforço concentrado em estudar os fundamentos e conceitos-chave de MLOps com o objetivo de construir uma base sólida. Concluí que a área vai além de um *framework* técnico: trata-se de um conjunto de práticas que viabilizam a operação de modelos em produção. Ficou claro que, embora aproveite princípios do DevOps (como automação e CI/CD), MLOps não é uma evolução direta dele, mas uma resposta aos desafios específicos do Ciclo de Vida de modelos, como a necessidade de

[1] Dez Semanas, entre setembro de 2025 e dezembro de 2025.

reprodutibilidade e escalabilidade. Como entrega desta fase, estabeleci os princípios e componentes encontrados nos materiais estudados e os organizei em duas tabelas comparativas detalhadas (Princípios e Componentes Técnicos) que podem ser consultadas no **Apêndice 1**.

Na sequência, durante a **Semana 2**, aprofundei a investigação histórica e teórica. Utilizei ferramentas como *Research Rabbit*, *Parsifal* e *Connected Papers* para explorar grafos de citação e identificar a evolução histórica, desde o artigo seminal de Sculley (2015)[2] sobre "Dívida Técnica Oculta" até a consolidação do MLOps em 2020. Além da investigação histórica, realizei um estudo de caso prático analisando a arquitetura da plataforma *Michelangelo*, da Uber, mapeando seus elementos aos princípios de Kreuzberger et al. (2022)[3]. O diagrama elaborado deste fluxo, bem como a linha do tempo da evolução da área, compõem o material do **Apêndice 2**.

Com a base teórica estabelecida, a **Semana 3** foi dedicada à curadoria e absorção de conhecimento. Refinei a seleção bibliográfica e a leitura crítica dos artigos, agora consolidados na tabela de "Artigos Finais". Para otimizar a extração de conhecimento, desenvolvi um processo de criação de resumos que aliou leitura ativa (com marcações manuais) ao uso de um *prompt* estruturado no ChatGPT, garantindo fidelidade e padronização. Essa metodologia permitiu mapear rapidamente as áreas de aplicação prática de MLOps, identificando tendências em *Edge AI* e *IoT* (focados em latência e atualização na borda), Mobilidade (como as barreiras inteligentes em Santander, na Espanha), além de Finanças e Saúde, onde o monitoramento e a explicabilidade são críticos. Todo o processo de engenharia de *prompt* e os resumos estão integrados ao material do **Apêndice 3**.

Durante as **Semanas 4, 5 e 6**, as atividades focaram no aprofundamento das Estratégias de *Deploy Release* e *Monitoring*. Na **Semana 4**, finalizei a leitura dos artigos de referência e elaborei a versão inicial do Glossário de termos técnicos (em colaboração com duas colegas), concluindo que, diferentemente do DevOps, o MLOps exige monitorar a lógica do modelo (como *drift* e *performance* preditiva) para validar um *deploy*. A **Semana 5**

[2] SCULLEY, D. et al. *Hidden technical debt in Machine learning systems*. 2015.
[3] KREUZBERGER, D.; KÜHL, N.; HIRSCHL, S. *Machine Learning Operations (MLOps): Overview, Definition, and Architecture*. 2022.

expandiu o estudo para *AutoML* e *Kaizen ML*, além de comparar estratégias clássicas (*Blue-Green*, *Canary*) com base na leitura do Capítulo 5 do livro *Practical MLOps*. Iniciei também a construção de um mapa mental dos subtemas relacionados à área. Já na **Semana 6**, dediquei-me ao Capítulo 6 (*Monitoring and Logging*), do mesmo livro, e à triagem de trabalhos e *datasets*. Selecionei o *dataset Heart Disease Health Indicators*[4], da Kaggle, para experimentos e um artigo sobre ML em laboratórios clínicos como referência de aplicação[5]. O glossário incremental, o mapa mental e a análise das estratégias compõem, respectivamente, o **Apêndice 4,** o **Apêndice 5** e o **Apêndice 6**.

As **Semanas 7 e 8** marcaram a fase de experimentação utilizando o Google Cloud Platform (GCP) e foi caracterizada por trazer à tona os desafios das aplicações em nuvem. Na **Semana 7**, implementei o *pipeline* de ingestão e preparação de dados, baixando arquivos brutos e carregando-os no *BigQuery* para criar um *ground truth* reprodutível baseado em regras de tempo real e com retrospectivas. Os *scripts* de ingestão, logs de erros e detalhes dessa arquitetura constam no **Apêndice 7**. Um modelo inicial foi treinado via *AutoML Tabular* no *Vertex AI*, permitindo o primeiro contato com ferramentas de explicabilidade. Contudo, devido aos custos elevados e para garantir a fidelidade ao trabalho-base (originalmente implementado em linguagem R), realizei na **Semana 8** uma transposição de domínio: substitui a etapa de retreino pelo empacotamento do modelo original (.rds) em um container com *Plumber* no *Cloud Run*. O resultado foi a criação de um ambiente gerenciado que oferece governança e observabilidade ao artefato original, sem os custos de um treino recorrente (**Apêndice 8**).

A **Semana 9** foi dedicada à implementação técnica dos pilares de Confiança em IA: *Applicability*, *Explainability* (XAI) e *Fairness*. Migrei o modelo validado para a GCP sem retreinar, servindo-o via *Plumber* e orquestrando releases pelo *Vertex AI*. Implementei a segurança de uso (*Applicability*) com um filtro de distância de Mahalanobis que força o sistema a se abster (HTTP 422) caso a entrada divirja da distribuição de treino. Para *Explainability*, habilitei rotas de explicação local e *jobs batch* para análises globais (SHAP e PDP). Em *Fairness*, configurei a exportação de *logs* estruturados para o *BigQuery*,

[4] TEBOUL, A. *Heart Disease Health Indicators Dataset*. Kaggle, 2020.
[5] SPIES, N. C. et al. Validating, Implementing, and Monitoring Machine Learning Solutions in the Clinical Laboratory Safely and Effectively. *Clinical Chemistry*, v. 70, n. 11, p. 1334-1343, 2024.

permitindo o cálculo cruzado de métricas (como paridade demográfica e igualdade de oportunidades) para detecção de viés. Também iniciei a construção de *dashboards* de monitoramento no *Looker Studio* (**Apêndice 9**).

Na **Semana 10**, finalizei o desenvolvimento consolidando os três pilares apresentados. Concluí o *deploy* do modelo preditivo utilizando o serviço *Cloud Run*, empacotando-o em um container *Docker* — um ambiente padronizado que garante a reprodutibilidade. Esse contêiner foi publicado no *Artifact Registry* e integrado a uma estrutura de observabilidade completa (*BigQuery*, *Cloud Logging* e *Cloud Monitoring*). Implementei políticas automáticas de monitoramento para latência, disponibilidade e erros do tipo 422 (indicadores de desvio de *Applicability*). Além disso, configurei a geração automática de métricas de explicabilidade global (*XAI Global*), aumentando a transparência das variáveis mais influentes. Com essa infraestrutura, estabeleci a base sólida para os painéis visuais no *Looker Studio*. Todos os detalhes técnicos da infraestrutura e os resultados finais de monitoramento encontram-se no **Apêndice 10**.

Desde o início, defini que meu foco seria compreender e aplicar os princípios de MLOps. Ao longo das **Semanas**, aprofundei meus estudos no ecossistema do Google Cloud Platform (GCP), consolidando uma visão prática sobre *pipelines* de *machine learning* escaláveis, interpretáveis e monitoráveis. Essa experiência me permitiu compreender como os pilares de reprodutibilidade, automação e observabilidade se articulam em um ambiente moderno, reforçando a importância de garantir explicabilidade e responsabilidade no Ciclo de Vida dos modelos.

Olhando para trás, percebo que este projeto foi muito mais que um exercício acadêmico. Ele consolidou uma base sólida para atuar na interseção entre engenharia de machine learning e operações em nuvem, permitindo-me navegar por novas áreas de conhecimento com autonomia.

Finalmente, gostaria de registrar meu agradecimento primeiramente à minha família , em especial à minha mãe, ao meu irmão e ao meu namorado, que heroicamente suportaram minhas noites mal dormidas e meu mau humor nas semanas finais. Agradeço também aos meus amigos, pelo apoio moral . Estendo minha gratidão aos professores, em especial

Fernando Federson, Sávio Teles, Leonardo Alves e Telma Woerle, e aos colegas que tornaram este caminho possível e, sobretudo, mais leve.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 4 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, concentrei meus esforços em estudar os fundamentos e os conceitos-chave relacionados a MLOps, com o objetivo de construir uma base sólida para a especialização futura. A partir disto, concluí que:

- Concluí que MLOps vai além de um framework técnico: trata-se de um **conjunto de práticas e princípios** que viabilizam a operação de modelos de ML em produção de forma confiável e escalável.
- Embora aproveite princípios do **DevOps** (como automação, versionamento e CI/CD), **MLOps não é uma evolução direta dele**: surgiu como **resposta a desafios específicos do ciclo de vida de modelos de machine learning**, especialmente diante da dificuldade das empresas em escalar modelos com sucesso.
- Estabeleci os princípios e componentes encontrados nos materiais estudados e organizei em duas tabelas de fácil acesso para quando precisar.

Além disso, iniciei a busca por **artigos e referências da área**, incluindo materiais que discutem **a origem e evolução histórica do termo MLOps**, a fim de complementar e aprofundar as descobertas desta semana.

O link para o material complementar pode ser encontrado aqui.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Aprofundar a pesquisa sobre a origem do termo "MLOps"

- Explorar artigos e white papers anteriores a 2020 (ex: Google, Microsoft, Uber, ThoughtWorks) e compará-los com publicações mais recentes, observando a **evolução dos conceitos e práticas de MLOps** ao longo do tempo.
- Identificar os principais autores e fontes de referência.

Explorar arquiteturas reais de MLOps

- Estudar cases de arquitetura de empresas (ex: ML pipelines da Google, Netflix, Spotify ou Nubank).

- Identificar quais princípios/componentes estão presentes na prática e quais ferramentas são utilizadas.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Agradeço ao Artur pelo envio do artigo que orientou parte da fundamentação teórica, e à Luísa pelo compartilhamento do documento gerado com o auxílio do Gemini, que contribuiu para enriquecer a compreensão dos conceitos abordados.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Documento Semana 1

O que é MLOps?

MLOps (Machine Learning Operations) é um paradigma e prática de engenharia que combina cultura, melhores práticas e conceitos das áreas de machine learning, engenharia de software (principalmente DevOps) e engenharia de dados, com o objetivo de produzir, operacionalizar, monitorar, implantar e escalar produtos de ML de forma contínua. Visa automatizar todo o ciclo de vida dos modelos – da experimentação à produção – utilizando princípios como CI/CD, versionamento, orquestração, monitoramento e feedback loops [1].

Noah Gift [2], por outro lado, descreve MLOps como o processo de automação de machine learning usando as metodologias DevOps.

E o que é DevOps, então?

DevOps é uma cultura e conjunto de práticas que unificam o desenvolvimento de software (Dev) e a operação de sistemas (Ops), com foco em automação, entrega contínua, integração frequente e monitoramento constante. O objetivo é tornar o ciclo de entrega de software mais rápido, confiável e colaborativo entre equipes de desenvolvimento e infraestrutura.

Embora MLOps compartilhe vários princípios do DevOps, como automar processos, CI/CD e monitoramento, ele não é uma evolução direta de DevOps. MLOps surge como uma resposta aos desafios específicos de colocar modelos de machine learning em produção, envolvendo dados dinâmicos, pipelines não determinísticos, reprodutibilidade e integração entre cientistas de dados e engenheiros de software.

Fundamentação e Organização Inicial

Nesta semana, estabeleci os principais conceitos encontrados nos materiais estudados e organizei as duas tabelas que estão no apêndice. Também iniciei um levantamento de artigos e white papers da área, tanto anteriores a 2020 quanto mais recentes, com o objetivo de entender a evolução histórica do termo MLOps e as práticas consolidadas ao longo do tempo.

Usei as ferramentas **Research Rabbit** [3], **Parsifal** [4] e **Connected Papers** [5] para iniciar o levantamento bibliográfico. O link para o documento gerado está aqui.

TABELAS

Tabela 1. Princípios MLOpsTabela 2. Componentes MLOps

Princípios de MLOps	Componentes Técnicos Correspondentes
P1. CI/CD automation: integração, entrega e implantação contínuas para acelerar feedback	**C1. CI/CD Component** – automatiza build, testes, entrega e deployment (e.g., Jenkins, GitHub Actions)
P2. Workflow orchestration: coordenação de tarefas via DAGs	**C3. Workflow Orchestration Component** – orquestra pipelines com DAGs (e.g., Apache Airflow, Kubeflow Pipelines)
P3. Reproducibility: capacidade de reproduzir experimentos e obter os mesmos resultados	Embutido em diversos componentes, especialmente C3, C4, C6, C7 (ex.: rastreamento completo do pipeline)
P4. Versioning: versionamento de dados, modelos e código para rastreabilidade	**C2. Source Code Repository**, **C4. Feature Store System**, **C6. Model Registry**, **C7. ML Metadata Stores**
P5. Collaboration: favorecer trabalho colaborativo e quebrar silos	Facilitado por **C2. Source Code Repository** (ex.: GitHub, GitLab
P6. Continuous ML training & evaluation: re-treinamento periódico com avaliação automática	Suportado por **C1. CI/CD Component**, **C3**, **C5. Model Training Infrastructure**
P7. ML metadata tracking/logging: registro de metadados de cada etapa (parâmetros, métricas, lineage)	**C7. ML Metadata Stores** – rastreia dados, código, métricas, lineage

P8. Continuous monitoring: monitoramento contínuo do desempenho do modelo e da infraestrutura	**C9. Monitoring Component** (ex.: Prometheus, ELK, TensorBoard)
P9. Feedback loops: ciclos de realimentação para melhorar continuamente o sistema	Implementado especialmente via **C9 + C1/C3** integrados com o agendador e o pipeline, permitindo re-treinamento automático

Tabela 2. Componentes MLOps

Componente	Descrição	Exemplos
C1. CI/CD Component	Automatiza o processo de integração, teste, entrega e implantação de código e modelos.	Jenkins, GitHub Actions, GitLab CI
C2. Source Code Repository	Armazena e versiona o código-fonte e scripts de ML, possibilitando rastreabilidade e colaboração.	Git, GitHub, GitLab
C3. Workflow Orchestration	Coordena e agenda tarefas em pipelines ML via DAGs (fluxos de trabalho dependentes).	Apache Airflow, Kubeflow Pipelines, Prefect
C4. Feature Store System	Armazena, versiona e serve **features reutilizáveis** para treinamento e inferência. Garante consistência entre treino e produção.	Feast, Tecton
C5. Model Training Infrastructure	Ambiente (on-premise ou cloud) para executar o treinamento de modelos em escala. Pode incluir suporte a GPUs, escalonamento e gerenciamento de jobs.	Vertex AI, SageMaker, Databricks, clusters com Kubernetes
C6. Model Registry	Repositório central para armazenar, versionar e promover modelos treinados (com metadata e estado de aprovação).	MLflow Model Registry, SageMaker Model Registry
C7. ML Metadata Store	Sistema que registra e rastreia todos os metadados do ciclo de vida do ML: datasets, parâmetros, métricas, artefatos, lineage.	ML Metadata (TFX), MLflow Tracking, Neptune.ai

C8. Model Serving Component	Gerencia a **exposição dos modelos treinados** para uso em produção. Pode lidar com deploy via APIs REST, gRPC, servidores batch ou streaming, e gerenciar versões de modelos ativamente rodando. Também envolve latência, escalabilidade e rollback.	TensorFlow Serving, TorchServe, KFServing, BentoML, FastAPI, Flask, Triton Inference Server
C9. Monitoring Component	Monitora a performance do modelo em produção (ex: drift de dados, métricas de erro, latência) e da infraestrutura.	Prometheus, ELK Stack, TensorBoard, WhyLabs

REFERÊNCIAS

[1] **KREUZBERGER, Dominik; KÜHL, Niklas; HIRSCHL, Sebastian**. *Machine Learning Operations (MLOps): Overview, Definition, and Architecture.* *arXiv preprint arXiv:2205.02302*, 14 maio 2022. Disponível em: <https://arxiv.org/pdf/2205.02302>. Acesso em: 26 de agosto de 2025.
[2] **GIFT, Noah; DEZA, Alfredo.** *Practical MLOps: Operationalizing Machine Learning Models*. Sebastopol: O'Reilly Media, 2021.
[3] **RESEARCH RABBIT.** ResearchRabbit – AI Tool for smarter, faster literature reviews. Seattle: ResearchRabbit Inc., [s.d.]. Disponível em: <https://researchrabbitapp.com/>. Acesso em: 30 de agosto de 2025.
[4] **PARSIFAL.** Parsifal – Perform Systematic Literature Reviews. [s.l.]: Simple Complex, 2021. Disponível em: <https://parsif.al/>. Acesso em: 25 de agosto de 2025.
[5] **CONNECTED PAPERS.** Connected Papers – visual tool to explore and find relevant academic papers. [s.l.]: Connected Papers, [s.d.]. Disponível em: <https://www.connectedpapers.com/>. Acesso em: 30 de agosto de 2025.

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

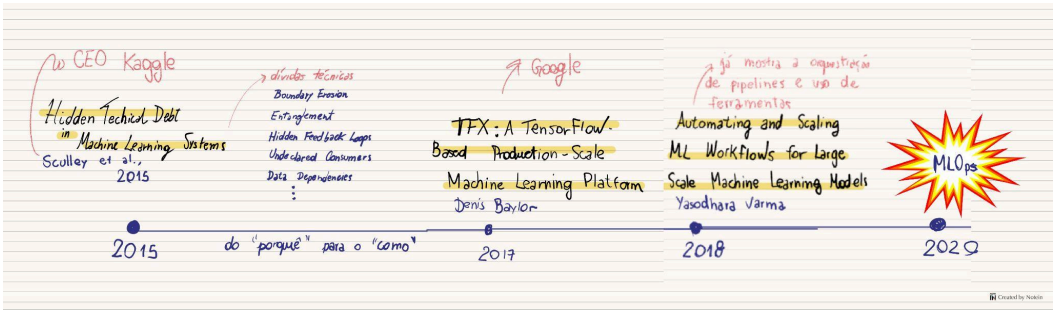
Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

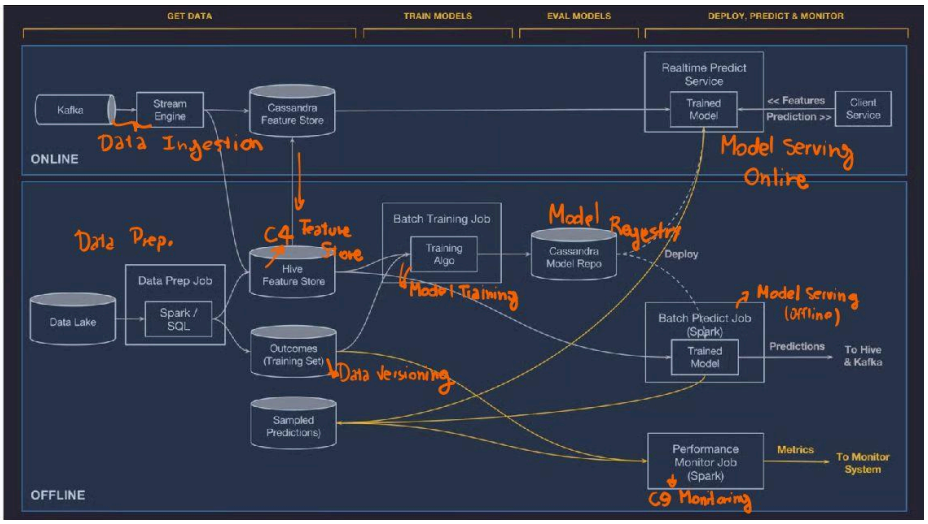
Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Resumo da Semana 2

- Estruturei a revisão sobre o tema **"MLOps"**.
- Organizei o processo no **Parsifal** e fiz screening no **Google Scholar** para levantar sinônimos/termos frequentes.
- Usei o **Research Rabbit** para explorar grafos de citação/co-citação e identificar artigos **"pré-MLOps"** (precursores conceituais e de plataforma), ampliando o conjunto de artigos.
- Defini e padronizei uma string de busca ampliada.
- Iniciei buscas no **Google Scholar** e **IEEE Xplore** e **mantive/atualizei** a tabela iniciada anteriormente.
- Fiz a **pré-seleção** e **pré-classificação** de artigos a serem lidos com maior prioridade.
- Desenhei uma ideia de como o termo MLOps foi surgindo e se modificando



- Análise da arquitetura MLOps da Uber com base nos princípios propostos por Kreuzberger et al. (2022).



- Documento Complementar 2

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Dar continuidade à leitura crítica dos artigos já pré-selecionados, aprofundando a compreensão dos fundamentos, práticas e desafios do MLOps.
- Explorar as áreas de aplicação de MLOps, identificando contextos variados como: computação de borda (Edge AI), IoT, saúde, finanças, mobilidade, entre outros.
- Observar como os princípios de MLOps se adaptam a diferentes domínios e ambientes operacionais.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

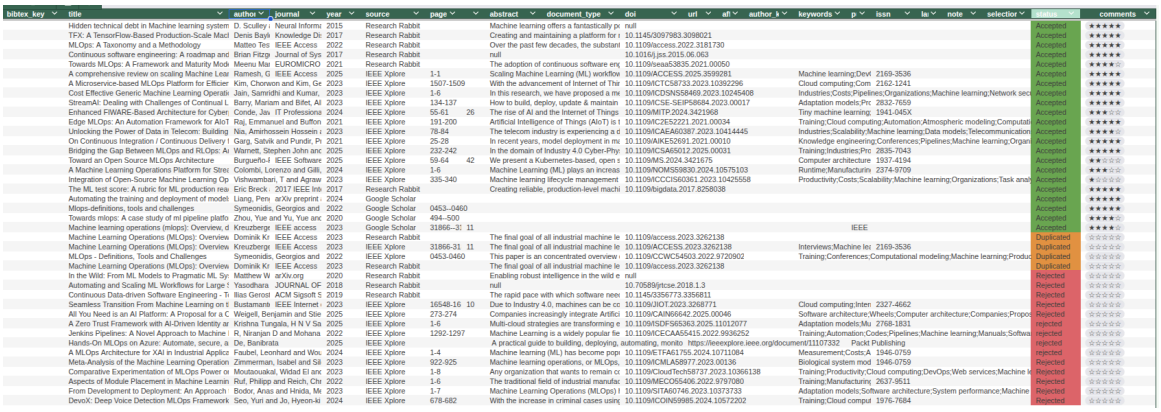
Documento Semana 2

- Definição do passo a passo da Revisão da Literatura:

- Estruturei a revisão bibliográfica sobre a **origem e evolução do termo "MLOps"** com foco em materiais **pré-2020** e contraste com publicações **mais recentes**.
  Para isso, a minha busca iniciou com os seguintes passos:
  - Utilização do site Parsifal a fim de organizar a busca por esses artigos.
  - Um **screening no Google Scholar** para extrair termos frequentes e sinônimos de busca.
  - Definição de uma string search mais ampla:

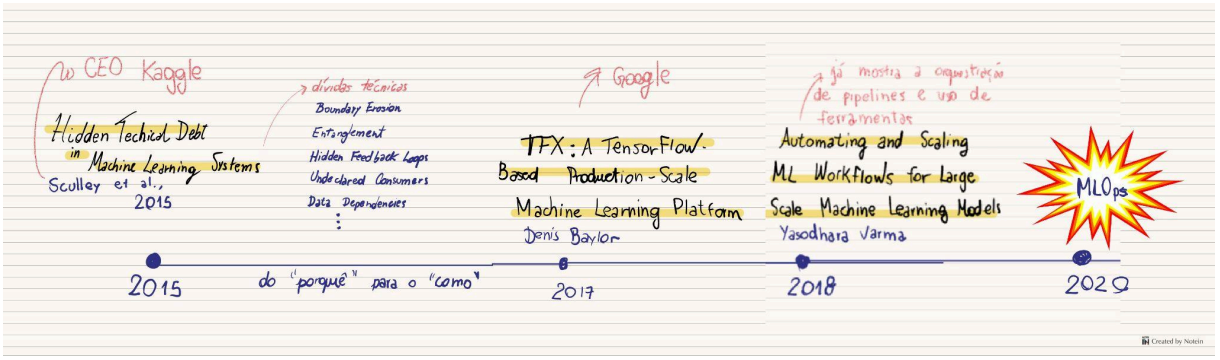
```
("MLOps" OR "Machine-Learning Operations" OR "Machine Learning Operations" OR
"ML Operations"
OR "rapid deployment of machine learning" OR "ML workflows"
OR "deployment of machine learning" OR "ML model deployments")
AND
("framework" OR "frameworks" OR "tool" OR "tools" OR "platform" OR
"platforms"
OR "pipeline" OR "pipelines" OR "architecture" OR "architectures" OR
"workflow" OR "workflows" OR "system" OR "systems")
AND
("efficiency" OR "reliability" OR "scalability" OR "reproducibility"
OR "automation" OR "monitoring" OR "deployment" OR "maintainability" OR
"observability")
AND
("production" OR "production environment" OR "production environments"
OR "cloud" OR "cloud computing" OR "industry" OR "data industry" OR "data
science" OR "enterprise")
```

  - Busca de artigos nas bases Google Scholar e IEEE e manutenção da tabela iniciada anteriormente.
  - Fiz a pré-seleção e pré-classificação de artigos a serem lidos com maior prioridade.



bibtex_key	title	author	journal	year	source	page	abstract	document_type	doi	url	aff	author_k	keywords	p	issn	la	note	selectior	status	comments
	Hidden technical debt in Machine learning system	D. Sculley	Neural Informa	2015	Research Rabbit		Machine learning offers a fantastically px	null											Accepted	★★★★★
	TFX: A TensorFlow-Based Production-Scale Mac	Denis Bayk	Knowledge Di	2017	Research Rabbit		Creating and maintaining a platform for r	10.1145/3097983.3098021											Accepted	★★★★★
	MLOps: A Taxonomy and a Methodology	Matteo Tes	IEEE Access	2022	Research Rabbit		Over the past few decades, the substani	10.1109/access.2022.3181730											Accepted	★★★★★
	Continuous software engineering: A roadmap and	Brian Fitzg	Journal of Sys	2017	Research Rabbit		null	10.1016/j.jss.2015.06.063											Accepted	★★★★★
	Towards MLOps: A Framework and Maturity Mod	Meenu Mar	EUROMICRO	2021	Research Rabbit		The adoption of continuous software eng	10.1109/seaa53835.2021.00050											Accepted	★★★★★
	A comprehensive review on scaling Machine Lear	Ramesh, G	IEEE Access	2025	IEEE Xplore	1-1	Scaling Machine Learning (ML) workflow	10.1109/ACCESS.2025.3599281					Machine learning;Devl		2169-3536				Accepted	★★★★★
	A Microservice-based MLOps Platform for Efficier	Kim, Chorwon and Kim, Ge	2023	IEEE Xplore	1507-1509	With the advancement of Internet of Thir	10.1109/ICTC58733.2023.10392296						Cloud computing;Com		2162-1241				Accepted	★★★★★
	Cost Effective Generic Machine Learning Operati	Jain, Samridhi and Kumar,	2023	IEEE Xplore	1-6	In this research, we have proposed a me	10.1109/ICDSNS58469.2023.10245408						Industries;Costs;Pipelines;Organizations;Machine learning;Network sec						Accepted	★★★★★
	StreamAI: Dealing with Challenges of Continual L	Barry, Mariam and Billet, Al	2023	IEEE Xplore	134-137	How to build, deploy, update & maintain	10.1109/ICSE-SEIP58684.2023.00017						Adaptation models;Pre		2832-7659				Accepted	★★★★★
	Enhanced FIWARE-Based Architecture for Cyber	Conde, Jav	IT Professiona	2024	IEEE Xplore	55-61 26	The rise of AI and the Internet of Things	10.1109/MITP.2024.3421968					Tiny machine learning;		1941-045X				Accepted	★★★☆☆
	Edge MLOps: An Automation Framework for AIoT	Raj, Emmanuel and Buffon	2021	IEEE Xplore	191-200	Artificial Intelligence of Things (AIoT) is t	10.1109/IC2E52221.2021.00034						Training;Cloud computing;Automation;Atmospheric modeling;Computati						Accepted	★★★★★
	Unlocking the Power of Data in Telecom: Building	Nia, Amirhossein Hossein	2023	IEEE Xplore	78-84	The telecom industry is experiencing a d	10.1109/ICAEA60387.2023.10414445						Industries;Scalability;Machine learning;Data models;Telecommunication						Accepted	★★★★☆
	On Continuous Integration / Continuous Delivery f	Garg, Satvik and Pundir, Pi	2021	IEEE Xplore	25-28	In recent years, model deployment in mc	10.1109/AIKE52691.2021.00010						Knowledge engineering;Conferences;Pipelines;Machine learning;Organ						Accepted	★★★★★
	Bridging the Gap Between MLOps and RLOps: Ai	Warnett, Stephen John and	2025	IEEE Xplore	232-242	In the domain of Industry 4.0 Cyber-Phy	10.1109/ICSA65012.2025.00031						Training;Industries;Pre		2835-7043				Accepted	★★★★★
	Toward an Open Source MLOps Architecture	Burgueño-I	IEEE Software	2025	IEEE Xplore	59-64 42	We present a Kubernetes-based, open s	10.1109/MS.2024.3421675					Computer architecture		1937-4194				Accepted	★★☆☆☆
	A Machine Learning Operations Platform for Stre	Colombi, Lorenzo and Gilli,	2024	IEEE Xplore	1-6	Machine Learning (ML) plays an increas	10.1109/NOMS59830.2024.10575103						Runtime;Manufacturin		2374-9709				Accepted	★★★☆☆
	Integration of Open-Source Machine Learning Op	Vishwambari, T and Agraw	2023	IEEE Xplore	335-340	Machine learning lifecycle management	10.1109/ICCICS60361.2023.10425558						Productivity;Costs;Scalability;Machine learning;Organizations;Task anal						Accepted	★☆☆☆☆
	The ML test score: A rubric for ML production rea	Eric Breck	2017 IEEE Int	2017	Research Rabbit		Creating reliable, production-level machi	10.1109/bigdata.2017.8258038											Accepted	★★★★★
	Automating the training and deployment of models	Liang, Pers	arXiv preprint	2024	Google Scholar														Accepted	★★★★★
	Mlops-definitions, tools and challenges	Symeonidis, Georgios and	2022	Google Scholar	0453--0460														Accepted	★★★★★
	Towards mlops: A case study of ml pipeline platfo	Zhou, Yue and Yu, Yue and	2020	Google Scholar	494--500														Accepted	★★★★☆
	Machine learning operations (mlops): Overview, d	Kreuzberger	IEEE access	2023	Google Scholar	31866--31 11													Accepted	★★★★☆
	Machine Learning Operations (MLOps): Overview	Dominik Kr	IEEE Access	2023	Research Rabbit		The final goal of all industrial machine le	10.1109/access.2023.3262138											Duplicated	☆☆☆☆☆
	Machine Learning Operations (MLOps): Overview	Kreuzberger	IEEE Access	2023	IEEE Xplore	31866-31 11	The final goal of all industrial machine le	10.1109/ACCESS.2023.3262138					Interviews;Machine lei		2169-3536				Duplicated	☆☆☆☆☆
	MLOps - Definitions, Tools and Challenges	Symeonidis, Georgios and	2022	IEEE Xplore	0453-0460	This paper is an concentrated overview	10.1109/CCWC54503.2022.9720902						Training;Conferences;Computational modeling;Machine learning;Produc						Duplicated	☆☆☆☆☆
	Machine Learning Operations (MLOps): Overview	Dominik Kr	IEEE Access	2023	Research Rabbit		The final goal of all industrial machine le	10.1109/access.2023.3262138											Duplicated	☆☆☆☆☆
	In the Wild: From ML Models to Pragmatic ML Sy	Matthew W	arXiv.org	2020	Research Rabbit		Enabling robust intelligence in the wild e	null											Rejected	☆☆☆☆☆
	Automating and Scaling ML Workflows for Large !	Yasodhara	JOURNAL OF	2018	Research Rabbit		null	10.70589/jrtcse.2018.1.3											Rejected	☆☆☆☆☆
	Continuous Data-driven Software Engineering - Ti	Ilias Gerost	ACM Sigsoft S	2019	Research Rabbit		The rapid pace with which software nee	10.1145/3356773.3356811											Rejected	☆☆☆☆☆
	Seamless Transition From Machine Learning on t	Bustamanti	IEEE Internet	2023	IEEE Xplore	16548-16 10	Due to Industry 4.0, machines can be co	10.1109/JIOT.2023.3268771					Cloud computing;Inter		2327-4662				Rejected	☆☆☆☆☆
	All You Need is an AI Platform: A Proposal for a C	Weigelt, Benjamin and Ste	2025	IEEE Xplore	273-274	Companies increasingly integrate Artifici	10.1109/CAIN66642.2025.00046						Software architecture;Wheels;Computer architecture;Companies;Propos						Rejected	☆☆☆☆☆
	A Zero Trust Framework with AI-Driven Identity ar	Krishna Tungala, H N V Sa	2025	IEEE Xplore	1-6	Multi-cloud strategies are transforming e	10.1109/ISDFS65363.2025.11012077						Adaptation models;Mu		2768-1831				rejected	☆☆☆☆☆
	Jenkins Pipelines: A Novel Approach to Machine	R, Niranjan D and Mohana	2022	IEEE Xplore	1292-1297	Machine Learning is a widely popular tie	10.1109/ICECAA55415.2022.9936252						Training;Automation;Codes;Pipelines;Machine learning;Manuals;Softwa						rejected	☆☆☆☆☆
	Hands-On MLOps on Azure: Automate, secure, a	De, Banibrata		2025	IEEE Xplore		A practical guide to building, deploying, automating, monito	https://ieeexplore.ieee.org/document/11107332					Packt Publishing						rejected	☆☆☆☆☆
	A MLOps Architecture for XAI in Industrial Applica	Faubel, Leonhard and Woc	2024	IEEE Xplore	1-4	Machine learning (ML) has become popu	10.1109/ETFA61755.2024.10711084						Measurement;Costs;A		1946-0759				rejected	☆☆☆☆☆
	Meta-Analysis of the Machine Learning Operation	Zimmerman, Isabel and Sil	2023	IEEE Xplore	922-925	Machine learning operations, or MLOps,	10.1109/ICMLA58977.2023.00136						Biological system mod		1946-0759				Rejected	☆☆☆☆☆
	Comparative Experimentation of MLOps Power o	Moubarak, Widad El and	2023	IEEE Xplore	1-8	Any organization that wants to remain c	10.1109/CloudTech58737.2023.10366138						Training;Productivity;Cloud computing;DevOps;Web services;Machine l						Rejected	☆☆☆☆☆
	Aspects of Module Placement in Machine Learnin	Ruf, Philipp and Reich, Chr	2022	IEEE Xplore	1-6	The traditional field of industrial manufac	10.1109/MECO55406.2022.9797080						Training;Manufacturin		2637-9511				Rejected	☆☆☆☆☆
	From Development to Deployment: An Approach	Bodor, Anas and Hnida, Mc	2023	IEEE Xplore	1-7	Machine Learning Operations (MLOps) i	10.1109/SITA60746.2023.10373733						Adaptation models;Software architecture;System performance;Machine						Rejected	☆☆☆☆☆
	DevoX: Deep Voice Detection MLOps Framework	Seo, Yuri and Jo, Hyeon-ki	2024	IEEE Xplore	678-682	With the increase in criminal cases using	10.1109/ICOIN59985.2024.10572202						Training;Cloud computi		1976-7684				Rejected	☆☆☆☆☆

Da dívida técnica ao MLOps: uma evolução da engenharia de machine learning



A linha do tempo acima não tem a intenção de apresentar marcos pioneiros ou os primeiros trabalhos em cada área, mas sim destacar alguns **trabalhos e iniciativas relevantes que ajudaram a construir o caminho até o surgimento e consolidação do conceito de MLOps**.

2015 – Identificação dos problemas

O artigo *"Hidden Technical Debt in Machine Learning Systems"* (Sculley et al., 2015), escrito por engenheiros do Google, foi um dos primeiros a chamar atenção para os riscos e desafios únicos de se manter sistemas de machine learning em produção. O texto aponta que, além da complexidade algorítmica, esses sistemas acumulam **dívidas técnicas ocultas** ligadas à dependência de dados, acoplamento excessivo, ciclos de feedback não controlados, entre outros.

Essa publicação teve grande repercussão e ajudou **a iniciar uma reflexão mais profunda sobre os custos invisíveis da engenharia de ML**.

2017 – Primeiras soluções práticas

Dois anos depois, o Google publica *"TFX: A TensorFlow-Based Production-Scale Machine Learning Platform"* (Denis Baylor et al.), apresentando uma arquitetura voltada para escalar ML de

forma mais segura e eficiente. O foco aqui já é sair do “por quê” (os problemas) e ir para o **“como” resolver**, por meio de ferramentas e boas práticas.

TFX mostrou a importância de estruturar o ciclo de vida do ML com componentes reutilizáveis, **versionamento e validação de dados**, promovendo maior confiabilidade na produção.

2018 – Orquestração e workflows

O trabalho *“Automating and Scaling ML Workflows for Large Scale Machine Learning Models”* (Yasodhara Varma et al.) reforça a necessidade de **automatizar e orquestrar pipelines de ML**, algo essencial para lidar com a complexidade de modelos em larga escala. Já não se trata apenas de treinar modelos, mas de integrá-los com outras partes do sistema, monitorá-los e mantê-los.

Essa fase marca a transição para a **engenharia de ML como disciplina própria**, conectada à infraestrutura e à operação contínua.

2020 – MLOps ganha nome e força

A partir de 2020, o termo **MLOps** se consolida como o conjunto de práticas que une Machine Learning, DevOps e Engenharia de Dados para operacionalizar modelos em produção com confiabilidade, escalabilidade e rastreabilidade.

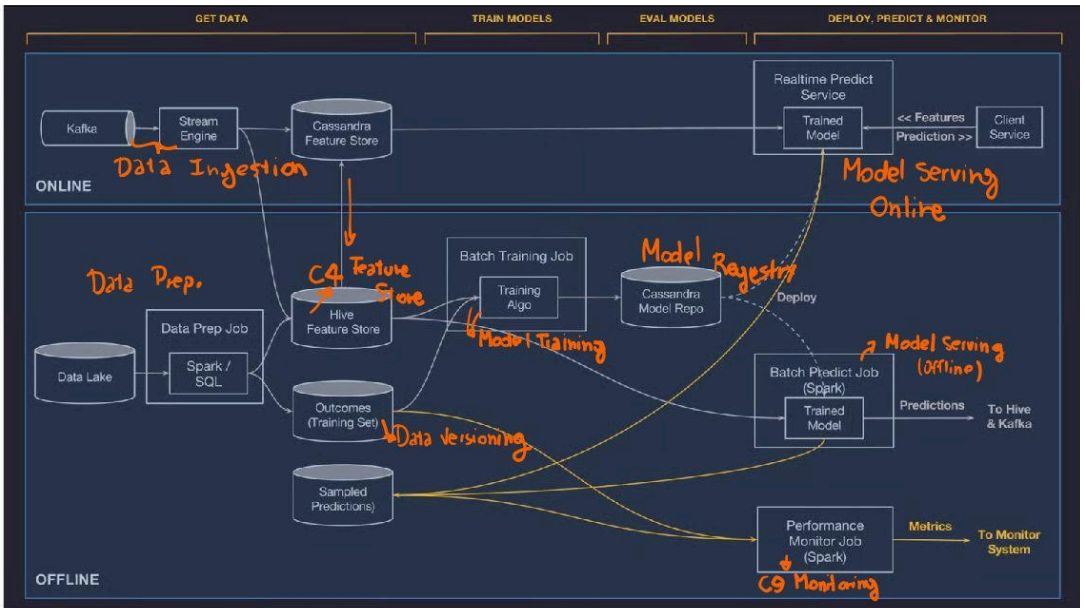
O que começou com alertas sobre dívidas técnicas, evoluiu para ferramentas e frameworks, e se transformou em uma cultura de engenharia voltada para todo o ciclo de vida do ML.

Conclusão

Essa linha do tempo ilustra como a evolução do MLOps **não aconteceu de forma instantânea**, mas sim como uma resposta crescente a desafios técnicos enfrentados na prática. Cada trabalho citado representa **uma peça do quebra-cabeça** que ajudou a estruturar o que hoje chamamos de MLOps.

Estudo de arquitetura MLOps na prática

Nesta etapa da atividade, busquei atender à proposta de **analisar casos reais de arquitetura de MLOps utilizados por empresas**. Escolhi como estudo de caso a plataforma **Michelangelo**, desenvolvida pela **Uber**, que é amplamente reconhecida por sua maturidade e escalabilidade em machine learning em produção.



A partir do estudo da arquitetura, elaborei um diagrama que representa o pipeline completo utilizado pela empresa, destacando os principais componentes operacionais envolvidos no ciclo de vida de um modelo — desde a ingestão e preparação de dados, até o treinamento, deploy e monitoramento.

Em seguida, associei esses elementos práticos aos **princípios e componentes de MLOps** descritos por **Kreuzberger et al. (2022)**, como:

- **Automação**
- **Reprodutibilidade**
- **Ciclo de vida completo**
- **Monitoramento**
- **Escalabilidade**

Também identifiquei na arquitetura da Uber o uso de ferramentas e tecnologias específicas como **Kafka, Spark, Cassandra, Hive** e serviços internos de model serving e experiment tracking, que concretizam esses princípios em um ambiente corporativo real.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 17 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nessa Semana 3, foram realizadas as seguintes atividades:

- Refinamento da *tabela* de artigos selecionados e início da leitura. Definido na página intitulada ¨Artigos Finais¨
- Criação de resumos a partir de anotações e marcações em PDFs, com auxílio do ChatGPT.
- Prompt definido para padronizar os resumos, garantindo fidelidade ao texto grifado, clareza e síntese final.

> Quero que você resuma o artigo científico que enviarei com base em minhas anotações e marcações no PDF.
> As instruções são:
>
> 1. **Obrigatório:** todo conteúdo que está **grifado** no PDF deve aparecer no resumo, de forma fiel, mas podendo ser reescrito para maior clareza.
> 2. **Anotações pessoais:** use minhas anotações/comentários como guia para destacar os pontos mais relevantes ou explicar termos.
> 3. **Organização:** apresente o resumo de forma estruturada, em tópicos ou seções, preservando a ordem lógica do texto.
> 4. **Fidelidade:** mantenha a linguagem acadêmica e não omita conceitos importantes.
> 5. **Clareza:** elimine repetições e torne as ideias mais objetivas, sem alterar o sentido original.
> 6. **Resumo final:** após os tópicos, traga também um **parágrafo de síntese geral** do artigo.
>
> Não invente nada além do que está no texto, nas marcações ou nas minhas notas.

Áreas de aplicação observadas até o momento:

- **Edge AI e IoT** – atualização automática de modelos na borda, redução de latência/consumo, uso de IoT Agents e padronização de dados (artigos *Edge MLOps* e *FIWARE + tinyML*).
- **Mobilidade** – barreiras de tráfego inteligentes em Santander, com re-treinamento automático e compressão de modelos.
- **Finanças** – citadas em artigos de definições como setor sensível, com foco em monitoramento e compliance.

- **Saúde** – destacado em *MLOps – Definitions, Tools and Challenges*, com ênfase em qualidade de dados, explicabilidade e monitoramento rigoroso.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

- Finalizar a leitura dos artigos selecionados.
- Construir um dicionário de termos de MLOps, a partir da leitura e dos resumos já elaborados.

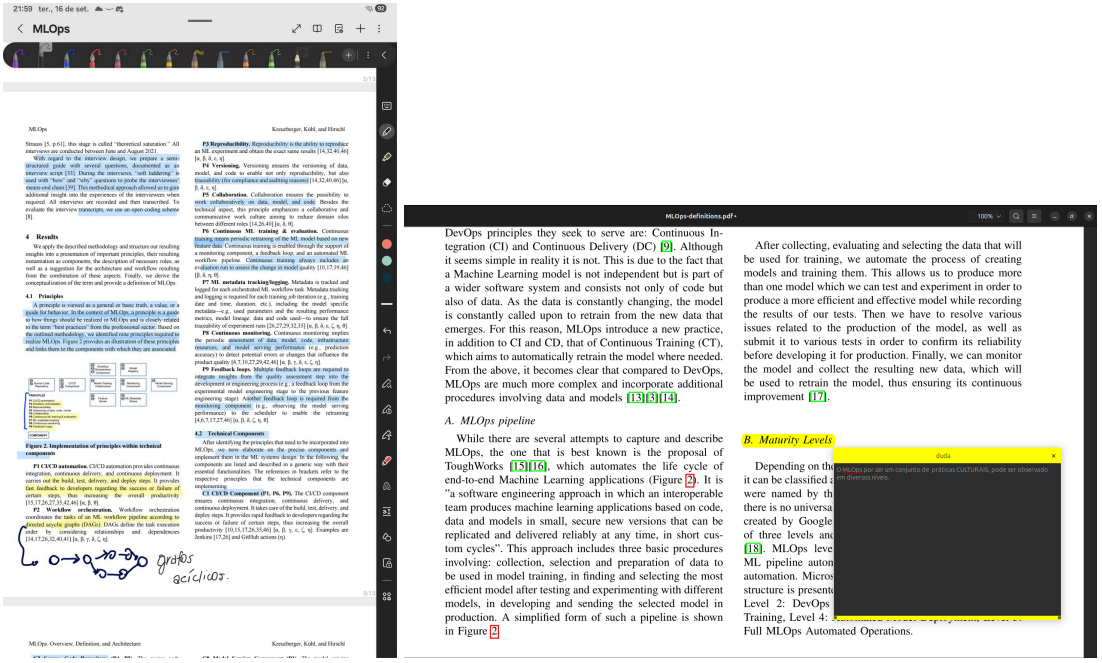
Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Processo de Criação dos Resumos

O processo de elaboração dos resumos foi organizado em etapas para garantir fidelidade ao conteúdo e clareza na síntese. Primeiro, houve a seleção e organização dos artigos, seguida da leitura com marcações nos PDFs e anotações pessoais. As anotações foram feitas no tablet, com escrita a próprio punho, ou no computador com um editor de pdfs.



Exemplos de anotações.

Com base nessas marcações, foi criado um prompt específico no ChatGPT, que passou por refinamentos até atender os objetivos. O prompt final foi:

> Quero que você resuma o artigo científico que enviarei com base em minhas anotações e marcações no PDF.
> As instruções são:
>
> 1. **Obrigatório:** todo conteúdo que está **grifado** no PDF deve aparecer no resumo, de forma fiel, mas podendo ser reescrito para maior clareza.
> 2. **Anotações pessoais:** use minhas anotações/comentários como guia para destacar os pontos mais relevantes ou explicar termos.
> 3. **Organização:** apresente o resumo de forma estruturada, em tópicos ou seções, preservando a ordem lógica do texto.
> 4. **Fidelidade:** mantenha a linguagem acadêmica e não omita conceitos importantes.
> 5. **Clareza:** elimine repetições e torne as ideias mais objetivas, sem alterar o sentido original.
> 6. **Resumo final:** após os tópicos, traga também um **parágrafo de síntese geral** do artigo.

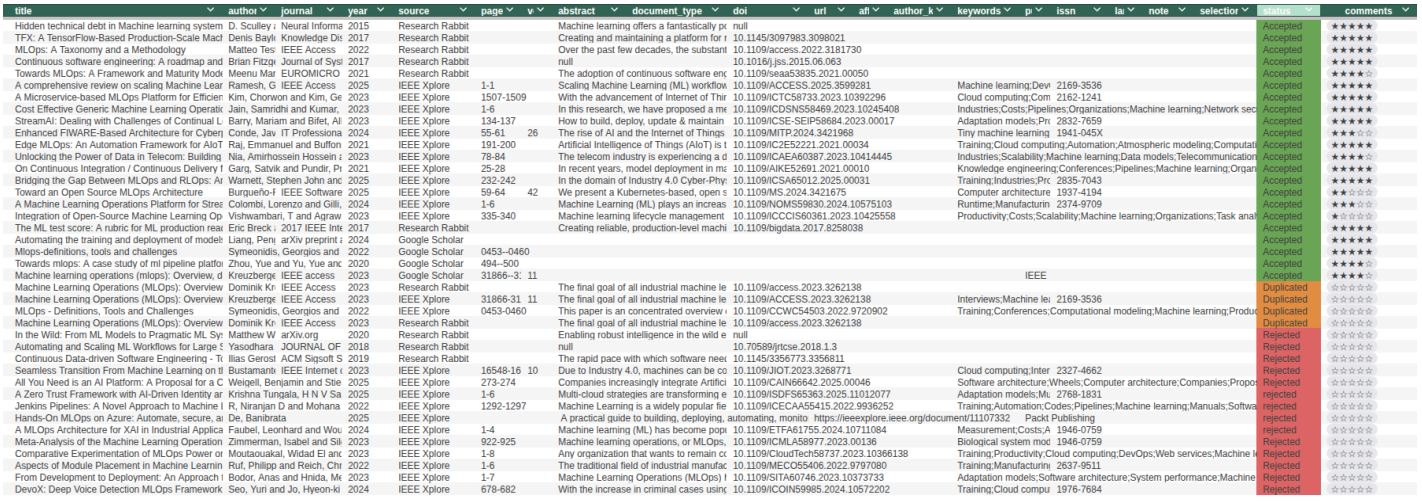
Não invente nada além do que está no texto, nas marcações ou nas minhas notas.

Esse refinamento foi essencial para que os resumos resultassem em materiais de estudo consistentes, permitindo não só a revisão rápida dos conteúdos, mas também a identificação das áreas de aplicação do MLOps em diferentes contextos, como Edge AI, IoT, mobilidade, finanças e saúde.

Assim, o processo adotado não apenas garante uniformidade entre os resumos, mas também estabelece uma base sólida para etapas futuras do trabalho, como a construção de um dicionário de termos de MLOps e a análise comparativa entre os artigos estudados.

Tabela de artigos

Podem ser encontradas no link para melhor visualização.



title	author	journal	year	source	page	v	abstract	document_type	doi	url	afl	author_k	keywords	p	issn	la	note	selection	status	comments
Hidden technical debt in Machine learning system	D. Sculley	Neural Informa	2015	Research Rabbit			Machine learning offers a fantastically p		null										Accepted	★★★★★
TFX: A TensorFlow-Based Production-Scale Mach	Denis Bayle	Knowledge Dis	2017	Research Rabbit			Creating and maintaining a platform for		10.1145/3097983.3098021										Accepted	★★★★★
MLOps: A Taxonomy and a Methodology	Matteo Tes	IEEE Access	2022	Research Rabbit			Over the past few decades, the substant		10.1109/access.2022.3181730										Accepted	★★★★★
Continuous software engineering: A roadmap and	Brian Fitzg	Journal of Syst	2017	Research Rabbit			null		10.1016/j.jss.2015.06.063										Accepted	★★★★★
Towards MLOps: A Framework and Maturity Mode	Meenu Mar	EUROMICRO	2021	Research Rabbit			The adoption of continuous software enc		10.1109/seaa53835.2021.00050										Accepted	★★★★☆
A comprehensive review on scaling Machine Lear	Ramesh, G	IEEE Access	2025	IEEE Xplore	1-1		Scaling Machine Learning (ML) workflow		10.1109/ACCESS.2025.3599281				Machine learning;Dev		2169-3536				Accepted	★★★★★
A Microservice-based MLOps Platform for Efficien	Kim, Chorwon and Kim, Ge		2023	IEEE Xplore	1507-1509		With the advancement of Internet of Thir		10.1109/ICTC58733.2023.10392296				Cloud computing;Com		2162-1241				Accepted	★★★★★
Cost Effective Generic Machine Learning Operatic	Jain, Samridhi and Kumar,		2023	IEEE Xplore	1-6		In this research, we have proposed a me		10.1109/ICDSNS58469.2023.10245408				Industries;Costs;Pipelines;Organizations;Machine learning;Network sec						Accepted	★★★★★
StreamAI: Dealing with Challenges of Continual L	Barry, Mariam and Bifet, Al		2023	IEEE Xplore	134-137		How to build, deploy, update & maintain		10.1109/ICSE-SEIP58684.2023.00017				Adaptation models;Pr		2832-7659				Accepted	★★★★★
Enhanced FIWARE-Based Architecture for Cyberp	Conde, Jav	IT Professiona	2024	IEEE Xplore	55-61	26	The rise of AI and the Internet of Things		10.1109/MITP.2024.3421968				Tiny machine learning		1941-045X				Accepted	★★★☆☆
Edge MLOps: An Automation Framework for AIoT	Raj, Emmanuel and Buffon		2021	IEEE Xplore	191-200		Artificial Intelligence of Things (AIoT) is t		10.1109/IC2E52221.2021.00034				Training;Cloud computing;Automation;Atmospheric modeling;Computati						Accepted	★★★★★
Unlocking the Power of Data in Telecom: Building	Nia, Amirhossein Hossein		2023	IEEE Xplore	78-84		The telecom industry is experiencing a d		10.1109/ICAEA60387.2023.10414445				Industries;Scalability;Machine learning;Data models;Telecommunication						Accepted	★★★★☆
On Continuous Integration / Continuous Delivery f	Garg, Satvik and Pundir, Pr		2021	IEEE Xplore	25-28		In recent years, model deployment in ma		10.1109/AIKE52691.2021.00010				Knowledge engineering;Conferences;Pipelines;Machine learning;Organ						Accepted	★★★★★
Bridging the Gap Between MLOps and RLOps: An	Warnett, Stephen John and		2025	IEEE Xplore	232-242		In the domain of Industry 4.0 Cyber-Phy		10.1109/ICSA65012.2025.00031				Training;Industries;Pro		2835-7043				Accepted	★★★★★
Toward an Open Source MLOps Architecture	Burgueño-F	IEEE Software	2025	IEEE Xplore	59-64	42	We present a Kubernetes-based, open s		10.1109/MS.2024.3421675				Computer architecture		1937-4194				Accepted	★★☆☆☆
A Machine Learning Operations Platform for Stres	Colombi, Lorenzo and Gilli,		2024	IEEE Xplore	1-6		Machine Learning (ML) plays an increas		10.1109/NOMS59830.2024.10575103				Runtime;Manufacturin		2374-9709				Accepted	★★★☆☆
Integration of Open-Source Machine Learning Op	Vishwambari, T and Agraw		2023	IEEE Xplore	335-340		Machine learning lifecycle management		10.1109/ICCCIS60361.2023.10425558				Productivity;Costs;Scalability;Machine learning;Organizations;Task ana						Accepted	★☆☆☆☆
The ML test score: A rubric for ML production rea	Eric Breck	2017 IEEE Inte	2017	Research Rabbit			Creating reliable, production-level machi		10.1109/bigdata.2017.8258038										Accepted	★★★★★
Automating the training and deployment of models	Liang, Peni	arXiv preprint	2024	Google Scholar															Accepted	★★★★★
Mlops-definitions, tools and challenges	Symeonidis, Georgios and		2022	Google Scholar	0453--0460														Accepted	★★★★★
Towards mlops: A case study of ml pipeline platfo	Zhou, Yue and Yu, Yue and		2020	Google Scholar	494--500														Accepted	★★★★☆
Machine learning operations (mlops): Overview, d	Kreuzberge	IEEE access	2023	Google Scholar	31866--3	11									IEEE				Accepted	★★★★☆
Machine Learning Operations (MLOps): Overview	Dominik Kr	IEEE Access	2023	Research Rabbit			The final goal of all industrial machine le		10.1109/access.2023.3262138										Duplicated	☆☆☆☆☆
Machine Learning Operations (MLOps): Overview	Kreuzberge	IEEE Access	2023	IEEE Xplore	31866-31	11	The final goal of all industrial machine le		10.1109/ACCESS.2023.3262138				Interviews;Machine le		2169-3536				Duplicated	☆☆☆☆☆
MLOps - Definitions, Tools and Challenges	Symeonidis, Georgios and		2022	IEEE Xplore	0453-0460		This paper is an concentrated overview		10.1109/CCWC54503.2022.9720902				Training;Conferences;Computational modeling;Machine learning;Produc						Duplicated	☆☆☆☆☆
Machine Learning Operations (MLOps): Overview	Dominik Kr	IEEE Access	2023	Research Rabbit			The final goal of all industrial machine le		10.1109/access.2023.3262138										Duplicated	☆☆☆☆☆
In the Wild: From ML Models to Pragmatic ML Sy	Matthew W	arXiv.org	2020	Research Rabbit			Enabling robust intelligence in the wild e		null										Rejected	☆☆☆☆☆
Automating and Scaling ML Workflows for Large S	Yasodhara	JOURNAL OF	2018	Research Rabbit			null		10.70589/jrtcse.2018.1.3										Rejected	☆☆☆☆☆
Continuous Data-driven Software Engineering - Tr	Ilias Gerost	ACM Sigsoft S	2019	Research Rabbit			The rapid pace with which software nee		10.1145/3356773.3356811										Rejected	☆☆☆☆☆
Seamless Transition From Machine Learning on tl	Bustamante	IEEE Internet	2023	IEEE Xplore	16548-16	10	Due to Industry 4.0, machines can be co		10.1109/JIOT.2023.3268771				Cloud computing;Inter		2327-4662				Rejected	☆☆☆☆☆
All You Need is an AI Platform: A Proposal for a C	Weigell, Benjamin and Stie		2025	IEEE Xplore	273-274		Companies increasingly integrate Artifici		10.1109/CAIN66642.2025.00046				Software architecture;Wheels;Computer architecture;Companies;Propo						Rejected	☆☆☆☆☆
A Zero Trust Framework with AI-Driven Identity ar	Krishna Tungala, H N V Sa		2025	IEEE Xplore	1-6		Multi-cloud strategies are transforming e		10.1109/ISDFS65363.2025.11012077				Adaptation models;Mu		2768-1831				rejected	☆☆☆☆☆
Jenkins Pipelines: A Novel Approach to Machine I	R, Niranjan D and Mohana		2022	IEEE Xplore	1292-1297		Machine Learning is a widely popular fie		10.1109/ICECAA55415.2022.9936252				Training;Automation;Codes;Pipelines;Machine learning;Manuals;Softwa						rejected	☆☆☆☆☆
Hands-On MLOps on Azure: Automate, secure, ai	De, Banibrata		2025	IEEE Xplore			A practical guide to building, deploying, automating, monito		https://ieeexplore.ieee.org/document/11107332				Packt Publishing						rejected	☆☆☆☆☆
A MLOps Architecture for XAI in Industrial Applica	Faubel, Leonhard and Wou		2024	IEEE Xplore	1-4		Machine learning (ML) has become popu		10.1109/ETFA61755.2024.10711084				Measurement;Costs;A		1946-0759				rejected	☆☆☆☆☆
Meta-Analysis of the Machine Learning Operation	Zimmerman, Isabel and Sik		2023	IEEE Xplore	922-925		Machine learning operations, or MLOps,		10.1109/ICMLA58977.2023.00136				Biological system mod		1946-0759				Rejected	☆☆☆☆☆
Comparative Experimentation of MLOps Power or	Moutaouakal, Widad El anc		2023	IEEE Xplore	1-8		Any organization that wants to remain cc		10.1109/CloudTech58737.2023.10366138				Training;Productivity;Cloud computing;DevOps;Web services;Machine l						Rejected	☆☆☆☆☆
Aspects of Module Placement in Machine Learnin	Ruf, Philipp and Reich, Chr		2022	IEEE Xplore	1-6		The traditional field of industrial manufac		10.1109/MECO55406.2022.9797080				Training;Manufacturin		2637-9511				Rejected	☆☆☆☆☆
From Development to Deployment: An Approach t	Bodor, Anas and Hnida, Me		2023	IEEE Xplore	1-7		Machine Learning Operations (MLOps) h		10.1109/SITA60746.2023.10373733				Adaptation models;Software architecture;System performance;Machine						Rejected	☆☆☆☆☆
DevoX: Deep Voice Detection MLOps Framework	Seo, Yuri and Jo, Hyeon-ki		2024	IEEE Xplore	678-682		With the increase in criminal cases using		10.1109/ICOIN59985.2024.10572202				Training;Cloud compu		1976-7684				Rejected	☆☆☆☆☆

Tabela

title	author	journal	year	sourc	pages	abstr	doi	comments	Resumo
Hidden technical debt in Machine learning systems	D. Sculley and D. Sculley and Gary David Holt an	Neural Information	2015	Research Rabbit		Machine l	null	★★★★★	Hidde...
TFX: A TensorFlow-Based Production-Scale Machine Learning Platfor	Denis Baylor and Denis Baylor and Eric Breck and	Knowledge Discov	2017	Research Rabbit		Creating a	10.1145/3097983.3	★★★★★	TFX: ...
Continuous software engineering: A roadmap and agenda	Brian Fitzgerald and Brian Fitzgerald and Klaas-J	Journal of System	2017	Research Rabbit		null	10.1016/j.jss.2015.(	★★★★★	Conti...
Towards MLOps: A Framework and Maturity Model.	Meenu Mary John and Meenu Mary John and Hel	EUROMICRO Cor	2021	Research Rabbit		The adop	10.1109/seaa5383(	★★★★☆	Towar...
Edge MLOps: An Automation Framework for AIoT Applications	Raj, Emmanuel and Buffoni, David and Westerlund, Magnus and Aho		2021	IEEE Xplo	191-200	Artificial In	10.1109/IC2E2022(	★★★★★	Edge ...
On Continuous Integration / Continuous Delivery for Automated Deplo	Garg, Satvik and Pundir, Pradyumn and Rathee, Geetanjali and Gupt		2021	IEEE Xplo	25-28	In recent	10.1109/AIKE5269(	★★★★★	On Co...
MLOps: A Taxonomy and a Methodology	Matteo Testi and Matteo Testi and Matteo Ballabic	IEEE Access	2022	Research Rabbit		Over the p	10.1109/access.202	★★★★★	MLOps: A
Mlops-definitions, tools and challenges	Symeonidis, Georgios and Nerantzis, Evangelos and Kazakis, Apostc		2022	Google Sc	0453--0460			★★★★★	MLOp...
Machine learning operations (mlops): Overview, definition, and archite	Kreuzberger, Dominik and K{\"u}hl, Niklas and Hir	IEEE access	2023	Google Sc	31866--31879			★★★★☆	Machi...
Enhanced FIWARE-Based Architecture for Cyberphysical Systems Wi	Conde, Javier and Munoz-Arcentales, Andrés and	IT Professional	2024	IEEE Xplo	55-61	The rise c	10.1109/MITP.2024	★★★☆☆	Enha...
Automating the training and deployment of models in MLOps by integr	Liang, Penghao and Song, Bo and Zhan, Xiaoan	arXiv preprint arXi	2024	Google Scholar				★★★★★	
A comprehensive review on scaling Machine Learning workflows using	Ramesh, G and Vaikunta Pai, T and Birau, Ramo	IEEE Access	2025	IEEE Xplo	1-1	Scaling M	10.1109/ACCESS.2	★★★★★	A Co...
Bridging the Gap Between MLOps and RLOps: An Industry 4.0 Case S	Warnett, Stephen John and Zdun, Uwe		2025	IEEE Xplo	232-242	In the dor	10.1109/ICSA6501(	★★★★★	Bridgi...
Toward an Open Source MLOps Architecture	Burgueño-Romero, Antonio M. and Benítez-Hidal	IEEE Software	2025	IEEE Xplo	59-64	We prese	10.1109/MS.2024.3	★★☆☆☆	Towar...

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 24 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba, Luisa Francielle Oliveira Fagundes, Danielle Tavares da Silva

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a Semana 4, as seguintes atividades foram concluídas:

- **Finalização da leitura e resumo** dos artigos de referência.
- **Elaboração da versão inicial do Glossário de termos técnicos** (desenvolvida em colaboração com Luísa e Danielle, sendo esta a única atividade em grupo da Semana).
- **Definição e aprofundamento do tema de estudo:**

  Estratégias de Deploy Release e Monitoramento em MLOps.

MLOps adapta estratégias de deploy do DevOps, como *blue-green* e *canary*, mas as torna intrinsecamente dependentes de um monitoramento mais profundo. Diferente do DevOps, que foca na saúde da aplicação, o MLOps exige o monitoramento da lógica do modelo, como performance preditiva e *drift* de dados, para efetivamente validar o sucesso de um deploy e decidir sobre a promoção ou reversão da nova versão.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima Semana, iniciarei o estudo nos dois temas a seguir:

1. **Fundamentos de Deployment Release**

O objetivo é compreender as principais estratégias de lançamento de modelos em MLOps e seus usos ideais como Big Bang, Rolling Update, Blue-Green, Canary, Shadow, A/B testing, entre outros.

2. Fundamentos de Monitoring em MLOps

O objetivo é entender como o monitoramento garante a confiabilidade de modelos em produção, acompanhando métricas técnicas, detectando drifts, avaliando impacto no negócio e oferecendo suporte a decisões como retraining e rollback.

Observação: **[caso precise fazer alguma observação, de qualquer "natureza"]**

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Tabela resumo dos termos:

Termo	Definição
CI (Continuous Integration)	Processo de integração contínua de código e componentes de ML. Garante que alterações em scripts, pipelines ou configurações sejam testadas e validadas de forma automática, prevenindo falhas na colaboração entre equipes.
CD (Continuous Delivery/Deployment)	Prática de entregar e implantar modelos e pipelines de forma contínua e confiável em ambientes de produção. Permite atualizações rápidas e seguras, reduzindo o tempo
CT (Continuous Training)	Extensão das práticas de CI/CD para o aprendizado de máquina. Automatiza o reprocessamento e re-treinamento de modelos sempre que novos dados estão disponíveis ou quando ocorre drift, garantindo que o modelo permaneça atualizado e relevante.
CM (Continuous Monitoring)	Monitoramento constante de dados e modelos em produção, avaliando métricas técnicas e de negócio, além de detectar problemas como *drift* e vieses.
Orquestração	Gerenciamento e coordenação das várias tarefas e fluxos de trabalho envolvidos no ciclo de vida do aprendizado de máquina, desde a preparação de dados e treinamento de modelos até a implantação e monitoramento.
ML Observability	Conjunto de práticas, ferramentas e métricas voltadas para entender, monitorar e diagnosticar o comportamento de modelos de ML em produção.

Orquestração de Containers	É o processo de automatizar o gerenciamento, a implantação, a escala e a comunicação de aplicações empacotadas em containers.
Experiment Tracking	Processo de monitorar e registrar experimentos de ML, armazenando metadados como hiperparâmetros, métricas, modelos gerados e ambiente computacional. Garante reprodutibilidade, facilita comparações entre execuções e auxilia na escolha do melhor modelo.
Model Registry	Repositório central para versionar, organizar e gerenciar modelos de ML, permitindo controle de ciclo de vida, rastreabilidade e implantação estruturada em produção.
Data Lineage	Rastreamento da origem, transformação e uso dos dados ao longo do pipeline de ML. Permite entender de onde vêm os dados, como foram processados e onde são aplicados, garantindo transparência, auditoria e confiabilidade nos experimentos e modelos.
Feature Store	Repositório centralizado para armazenar, versionar e compartilhar *features* usadas em modelos de ML. Facilita a reutilização, mantém consistência entre treino e produção e acelera o desenvolvimento de novos modelos.
Explainability (XAI – Explainable Artificial Intelligence)	Conjunto de técnicas que tornam modelos de ML interpretáveis e transparentes, permitindo que especialistas entendam e confiem nas decisões dos algoritmos. Towards MLOps: A Framework and Maturity Model
GitOps	Extensão de DevOps que usa repositórios Git como "fonte de verdade" para infraestrutura e deployment, permitindo que alterações sejam rastreadas e aplicadas automaticamente.

Shadow Deployment	Estratégia de deployment em que um novo modelo roda em paralelo ao modelo atual, mas sem impactar usuários finais, servindo apenas para testes.
Canary Deployment	Implantação gradual de um novo modelo em uma fração controlada dos usuários, reduzindo riscos de falhas em larga escala. *Toward an Open Source MLOps Architecture.*
Technical Debt (Dívida Técnica)	Compromissos assumidos ao priorizar velocidade sobre qualidade na implementação, que podem aumentar custos de manutenção futura. Hidden Technical Debt in Machine Learning Systems
Maturity Model (Modelo de Maturidade)	Estrutura que descreve níveis de evolução na adoção de MLOps, desde fluxos manuais (nível inicial) até pipelines totalmente automatizados com monitoramento e re-treinamento. Towards MLOps: A Framework and Maturity Model, John, Olsson & Bosch, 2021
Compliance	Adequação de sistemas e pipelines a normas legais e regulatórias (ex.: GDPR, HIPAA), garantindo governança, privacidade e auditabilidade.
A/B Testing	Método experimental que compara duas ou mais versões de um modelo/sistema para avaliar impacto em métricas técnicas ou de negócio.
Blue-Green Deployment	Mantêm-se dois ambientes de produção idênticos: "Blue" (com a versão atual) e "Green" (com a nova versão). O tráfego é direcionado para o ambiente Blue. Quando a nova versão no Green é validada, o tráfego é todo redirecionado para ele. Se algo der errado, é fácil e rápido reverter para o ambiente Blue.
Rolling Deployment	A nova versão do modelo é gradualmente introduzida, substituindo as instâncias da versão antiga uma a uma, até que todas

	estejam atualizadas. Isso permite uma transição suave e sem tempo de inatividade (downtime).
Big Bang (ou Recreate)	A versão antiga é desligada e a nova é ligada de uma vez. É a abordagem mais simples, porém mais arriscada, pois qualquer problema na nova versão afeta todos os usuários.
Federated learning	É uma abordagem de aprendizado de máquina distribuído em que os dados permanecem nos dispositivos ou organizações que os geraram, sem necessidade de centralização; em vez disso, cada nó treina localmente um modelo e envia apenas atualizações de parâmetros (como gradientes ou pesos) para um servidor central, que agrega os resultados e atualiza o modelo global. Esse processo garante maior privacidade, já que os dados brutos não são compartilhados, melhora a eficiência ao reduzir a transferência de dados e permite personalização em contextos locais. No entanto, apresenta desafios como a heterogeneidade dos dados, altos custos de comunicação e riscos de segurança. É amplamente aplicado em áreas como teclados inteligentes de smartphones, saúde e sistemas financeiros.
Dataset Shift	Mudança na distribuição dos dados de entrada usados pelo modelo em produção em relação aos dados de treinamento. Isso faz com que o modelo enfrente padrões diferentes dos que aprendeu, podendo reduzir sua performance.
Concept Drift	Alteração na relação entre as variáveis de entrada (features) e o alvo (label) ao longo do tempo. Diferente do dataset drift, que é mudança só na distribuição dos dados, aqui muda o significado do que o modelo deve

	aprender. Mesmo que os dados de entrada não mudem muito, a regra que liga entrada e saída pode mudar, exigindo re-treino ou adaptação do modelo.
Ataques Adversariais	Técnicas usadas para enganar modelos de aprendizado de máquina, inserindo pequenas perturbações nos dados de entrada (muitas vezes imperceptíveis para humanos) que levam o modelo a tomar decisões erradas.
Interpretabilidade	Capacidade de compreender e explicar como um modelo de machine learning chega às suas previsões ou decisões. Permite identificar quais variáveis ou fatores influenciaram o resultado, sendo essencial para auditoria, transparência, confiança e uso ético de modelos.
Data Lake	É um repositório centralizado que armazena **grandes volumes de dados em seu formato bruto**, sem necessidade de estrutura pré-definida.
Data Warehouse	É um sistema usado para armazenar e analisar **dados já tratados, limpos e estruturados**, otimizados para consultas e relatórios.
Latência	Tempo entre pedido e resposta, medidos por percentis
AutoML	Automação de etapas críticas de ML (pré-processamento, seleção de features, tuning de hiperparâmetros, seleção de modelos, ensembling).

KaizenML	Foco na melhoria contínua de todo o ciclo de vida do ML.
Drift Detection	Processo de identificar se ocorreu uma mudança significativa na distribuição dos dados ao longo do tempo. Tem como objetivo detectar quando um modelo pode perder desempenho devido a alterações no ambiente ou nos dados.
Drift Localization	Processo de identificar em quais regiões do espaço de dados a mudança ocorreu. Busca determinar onde exatamente o drift acontece, permitindo ações mais direcionadas, como re-treinar o modelo apenas para segmentos afetados.
Drift Explanations	Processo de tornar compreensível o fenômeno do drift, explicando como e por que a mudança ocorreu. Envolve indicar quais variáveis foram mais impactadas, quais padrões se alteraram e como isso afeta o comportamento do modelo.
Exponential backoff	Ao falhar uma chamada, você espera e tenta de novo com intervalos que dobram (1s, 2s, 4s…), com limite e jitter.
Jitter	Pequena aleatoriedade
Circuit breaker	Um “disjuntor” que abre quando a taxa de erro/latência explode; enquanto aberto, bloqueia novas chamadas e retorna rápido (fallback).
Autoscaling	Escala automaticamente a quantidade de instâncias conforme carga para manter

	SLOs.
Throughput	Taxa sustentada de processamento, é importante para definir a capacidade de aguentar pico sem estourar a latência.
Políticas de rollback	Regras para voltar rápido à versão anterior quando o novo deploy degrada métricas.
SLO (Service Level Objective)	lvo/limite da qualidade (ex.: p95 < 200 ms em 99% do mês).
SLI (Service Level Indicator)	Métrica medida (p95, AUC, PSI, 5xx) usada para verificar SLO.
SLA (Service Level Agreement)	Compromisso contratual externo derivado de SLOs.
Burn rate (de erro)	Quão rápido você "queima" o orçamento do SLO; usado para alertas rápido/lento.
Multi-armed bandits	Roteamento adaptativo de tráfego conforme performance online.
Batching vs. Streaming	Processamento em lotes vs. em fluxo/tempo real para deploy/monitoria.

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 1 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, dei continuidade aos estudos em **MLOps**, com ênfase em **estratégias de Deploy Release e Monitoramento**.

Durante a Semana 5, as seguintes atividades foram concluídas:

- **Leitura de artigos:**
  - Moving Faster and Reducing Risk: Using LLMs in Release Deployment
  - The MLOps Approach to Model Deployment: A Road Map to Seamless Scalability
- **Estudo sobre estratégias de Deployment Release em MLOps**, analisando diferenças entre abordagens clássicas (Big Bang, Rolling Update, Blue-Green, Canary, Shadow e A/B testing).
- **Leitura do capítulo 5 do livro *Practical MLOps, Operationalizing Machine Learning Models* (O'Reilly)**, totalizando 43 páginas.
- **Estudo sobre AutoML e Kaizen ML**, explorando como a automação de ponta a ponta e a melhoria contínua podem ser integradas ao fluxo de MLOps. Essa atividade não estava prevista originalmente, mas foi incluída por se mostrar relevante para a compreensão das práticas modernas de ML.
- **Produção inicial de um mapa mental** reunindo os subtemas de MLOps estudados até então.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega (Semana 6), pretendo:

- **Retomar o tema de Monitoring em MLOps**.

- **Dar continuidade ao mapa mental,** incorporando os novos conceitos de monitoring.
- **Iniciar o contato prático com ferramentas de MLOps**, explorando plataformas e soluções que apoiam estratégias de deploy e monitoramento, para alinhar a teoria estudada com experimentação prática.

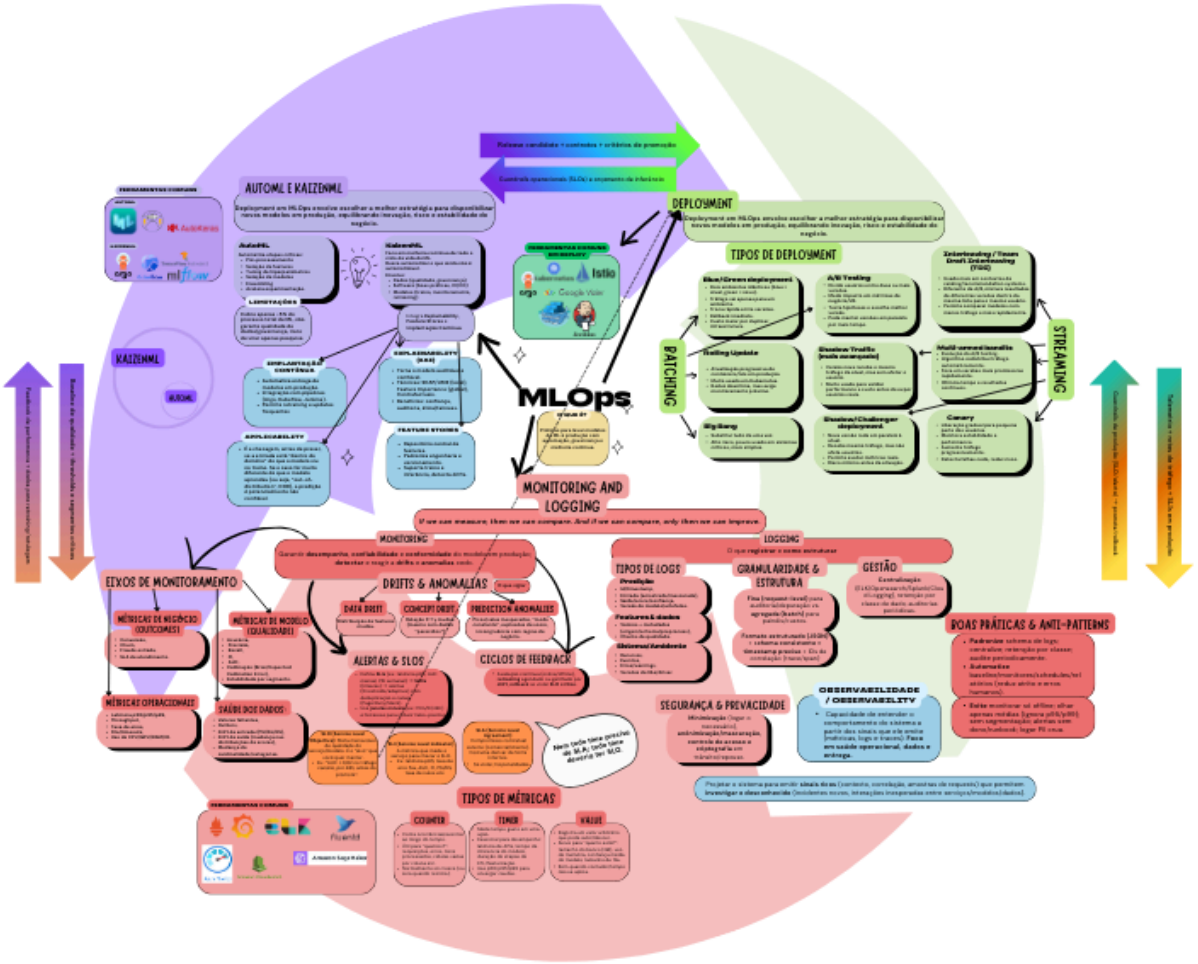
Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Mapa mental criado(versão final)

Para melhor visualização, acesse o link.



APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 8 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, dei continuidade aos estudos em **MLOps**, com ênfase em **estratégias de Deploy Release e Monitoramento**.

Durante a Semana 6, as seguintes atividades foram concluídas:

- **Leitura do capítulo 6 (Monitoring and Logging) do livro *Practical MLOps, Operationalizing Machine Learning Models* (O'Reilly)**, totalizando 24 páginas.
- **Continuação da produção de um mapa mental** reunindo os subtemas de MLOps estudados até então.
- Continuação incremental do dicionário de termos MLOps.
- **Levantamento de trabalhos e datasets (triagem):**
  - **Trabalho-base (pipeline conceitual):** *Practical Machine Learning in the Clinical Laboratory* — walkthrough didático para validação → implementação → monitoramento (site suplementar do mini-review).
  - **Artigo de referência:** Validating, Implementing, and Monitoring Machine Learning Solutions in the Clinical Laboratory Safely and Effectively
  - **Dataset explorado para experimentos paralelos:** *Heart Disease Health Indicators* (Kaggle) — candidato para testar variações de deploy/monitoring em ambiente controlado.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega (Semana 7), pretendo:

- Para começar a trabalhar com os frameworks, escolhi replicar um trabalho, priorizando **Deploy** e **Monitoramento**. O site *Practical Machine Learning in the Clinical Laboratory* oferece um **guia conceitual** e um **walk-through** didático, mas não entrega um **stack MLOps** nem um repositório de produção. Assim, implementarei a solução no **GCP** (aproveitando meus créditos existentes).

Observação: **[caso precise fazer alguma observação, de qualquer "natureza"]**

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

APÊNDICE 7

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 15 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana 7, dei continuidade aos estudos em **MLOps**, com ênfase em **estratégias de Deploy Release e Monitoramento**.

Implantei o Trabalho-base no GCP

Ao longo do projeto, combinei **scripts em shell** com a **interface do Google Cloud**, alternando conforme a tarefa (automação vs. configuração visual).

- **Ambiente:** Projeto **residencia-123**. Dados guardados nos EUA (**US, multi-região**). Treino e serviços de IA no **Vertex AI (us-central1)**.
- **Ingestão de dados:** Baixei os arquivos do **Figshare**, converti para **CSV**, subi para o **Cloud Storage (GCS)** e carreguei no **BigQuery** (tabelas `bmp_demo.train` e `bmp_demo.validation`)
- **Rotulagem (ground truth):** Apliquei duas regras simples:
  - **Tempo real (realtime):** cortes fixos em algumas variáveis.
  - **Revisão (retrospective):** cortes baseados nos **quantis** calculados só no conjunto de **treino** (pra não vazar informação).
    Combinei essas regras para criar o **label** final e salvei as saídas em:
  - **GCS:** `bmp/train_labeled.csv`, `bmp/validation_labeled.csv`
  - **BigQuery:** `bmp_demo.train_labeled`, `bmp_demo.validation_labeled`
- **Treinamento (AutoML Tabular – Vertex AI):** Usei o CSV do **GCS** como fonte, com **label** como alvo e objetivo de **maximizar AUC**. Divisão 80/10/10 e cerca de **3 horas de nó** (controle de custos). O modelo ficou registrado em **Models** → **Evaluation**.
- **Implantação (iniciada):** Criei o **endpoint** no Vertex (ainda **sem tráfego**, só para testes). A própria plataforma oferece **explicabilidade** (Feature Attributions) e **monitoramento** do modelo.
- **Teste rápido online:** Enviei **1 linha** do `validation_labeled.csv` pro endpoint (threshold provisório **THRESH=0,35**). Resultado do teste: **p_pos = 0,151921** → **label_hat = 0**.

Documento Complementar da Semana 7: inclui scripts, capturas de tela e os principais erros/correções.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próximos passos (Semana 8)

1. Treinar o modelo com mais node hours/converter o modelo treinado no artigo para ser utilizado no GCP.
2. Batch Predict no validation labeled.csv para obter probabilidades.
3. Deploy online com explicabilidade (attributions) e Model Monitoring no Vertex.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Foi muito legal perceber que eu não precisei "correr atrás" das ferramentas de MLOps: elas já estavam lá espalhadas por todo o sistema, prontas para eu usar!

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Residência – Semana 7 | Diário de execução (MLOps – BMP no GCP)

> **Meta da semana:** executar o plano da Semana 6 focando ingestão de dados, definição de *ground truth*, treino com Vertex AI AutoML, e preparar base para monitoramento/thresholds.

---

1) Visão geral do que foi realizado

- **Ambiente**: Projeto GCP `residencia-474423`.
- **Armazenamento**: bucket **US (multi-region)** `residencia-bucket` para dados.
- **BigQuery**: dataset **US** `bmp_demo` para tabelas.
- **Compute/ML**: Vertex AI em **us-central1** (bucket de *staging* regional criado para os jobs).
- **Fonte dos dados**: Figshare (arquivos feather `train` e `validation`).
- **Abordagem**: CSVs no GCS → tabelas no BigQuery → *ground truth* reprodutível → AutoML Tabular (Vertex) com `label` como alvo.

2) Ingestão: baixar Figshare → transformar → subir para GCS/BigQuery

Objetivo: padronizar dados do tutorial BMP em infraestrutura GCP para treino e análise.

Passos concluídos:

1. Criado ambiente Python (venv) e instaladas dependências (`pyarrow`, `pandas`, `google-cloud-*`, `requests`).
2. Script **`ingestao.py`** implementado para:
    - Baixar `train.feather` e `validation.feather` do Figshare.
    - Converter para **CSV** (`/tmp/train.csv`, `/tmp/validation.csv`).
    - **Enviar ao GCS**: `gs://residencia-bucket/bmp/train.csv` e `.../validation.csv`.

- **Criar/confirmar** dataset BigQuery `bmp_demo` **em US** e **carregar** tabelas `train` e `validation` via `load_table_from_uri`.

Evidências geradas:

- GCS: `gs://residencia-bucket/bmp/train.csv`, `.../validation.csv`.
- BigQuery: `bmp_demo.train`, `bmp_demo.validation`.

```
# !pip install pyarrow pandas google-cloud-storage google-cloud-bigquery google-cloud-aiplatform
import pandas as pd, pyarrow.feather as feather, io, requests, os
from google.cloud import storage, bigquery
```

```
cat > ingestao.py << 'PY'
import os, io, requests
import pandas as pd
import pyarrow.feather as feather
from google.cloud import storage, bigquery

# ===== Config =====
PROJECT    = os.environ.get("PROJECT_ID") or "SEU_PROJETO"
BUCKET     = os.environ.get("BUCKET_NAME") or "seu-bucket-unico-global"
LOCATION   = os.environ.get("LOCATION")   or "us-central1"
BQ_DATASET = os.environ.get("BQ_DATASET") or "bmp_demo"

os.environ["GCLOUD_PROJECT"] = PROJECT

# Figshare (dados públicos do tutorial)
FIG_TRAIN = "https://figshare.com/ndownloader/files/45407401"
FIG_VALID = "https://figshare.com/ndownloader/files/45407398"

def read_feather_from_url(url: str) -> pd.DataFrame:
    r = requests.get(url, timeout=180)
    r.raise_for_status()
    buf = io.BytesIO(r.content)
    return feather.read_feather(buf)

def main():
    print(f"Projeto: {PROJECT} | Bucket: {BUCKET} | Dataset BQ: {BQ_DATASET} | Loc: {LOCATION}")

    # 1) Baixar dados do Figshare
    print("Baixando Figshare (train/validation)...")
    train_df = read_feather_from_url(FIG_TRAIN)
    valid_df = read_feather_from_url(FIG_VALID)

    print("Colunas (train):", train_df.columns.tolist()[:50])
    print("Colunas (valid):", valid_df.columns.tolist()[:50])

    # 2) Salvar CSVs temporários
    train_csv = "/tmp/train.csv"
```

```
    valid_csv = "/tmp/validation.csv"
    train_df.to_csv(train_csv, index=False)
    valid_df.to_csv(valid_csv, index=False)

    # 3) Enviar para GCS
    print("Subindo CSVs para GCS...")
    storage_client = storage.Client(project=PROJECT)
    bucket = storage_client.bucket(BUCKET)
    bucket.blob("bmp/train.csv").upload_from_filename(train_csv)
    bucket.blob("bmp/validation.csv").upload_from_filename(valid_csv)

    # 4) Carregar no BigQuery
    print("Criando dataset (se não existir) e carregando tabelas no BigQuery...")
    bq = bigquery.Client(project=PROJECT)
    bq.create_dataset(BQ_DATASET, exists_ok=True)

    def load_csv_to_bq(gcs_uri: str, table: str):
        job = bq.load_table_from_uri(
            gcs_uri,
            f"{PROJECT}.{BQ_DATASET}.{table}",
            job_config=bigquery.LoadJobConfig(
                source_format=bigquery.SourceFormat.CSV,
                autodetect=True,
                skip_leading_rows=1,
                allow_quoted_newlines=True,
            ),
            location=LOCATION,
        )
        job.result()
        print("Carregado no BigQuery:", table)

    load_csv_to_bq(f"gs://{BUCKET}/bmp/train.csv", "train")
    load_csv_to_bq(f"gs://{BUCKET}/bmp/validation.csv", "validation")

    print("✅ Concluído.")

if __name__ == "__main__":
    main()
PY

cat > ingestao.py << 'PY'
import os, io, requests
import pandas as pd
import pyarrow.feather as feather
from google.cloud import storage, bigquery

# ===== Config =====
PROJECT    = os.environ.get("PROJECT_ID") or "SEU_PROJETO"
BUCKET     = os.environ.get("BUCKET_NAME") or "seu-bucket-unico-global"
LOCATION   = os.environ.get("LOCATION")   or "us-central1"
BQ_DATASET = os.environ.get("BQ_DATASET") or "bmp_demo"

os.environ["GCLOUD_PROJECT"] = PROJECT
```

```python
# Figshare (dados públicos do tutorial)
FIG_TRAIN = "https://figshare.com/ndownloader/files/45407401"
FIG_VALID = "https://figshare.com/ndownloader/files/45407398"

def read_feather_from_url(url: str) -> pd.DataFrame:
    r = requests.get(url, timeout=180)
    r.raise_for_status()
    buf = io.BytesIO(r.content)
    return feather.read_feather(buf)

def main():
    print(f"Projeto: {PROJECT} | Bucket: {BUCKET} | Dataset BQ: {BQ_DATASET} | Loc: {LOCATION}")

    # 1) Baixar dados do Figshare
    print("Baixando Figshare (train/validation)...")
    train_df = read_feather_from_url(FIG_TRAIN)
    valid_df = read_feather_from_url(FIG_VALID)

    print("Colunas (train):", train_df.columns.tolist()[:50])
    print("Colunas (valid):", valid_df.columns.tolist()[:50])

    # 2) Salvar CSVs temporários
    train_csv = "/tmp/train.csv"
    valid_csv = "/tmp/validation.csv"
    train_df.to_csv(train_csv, index=False)
    valid_df.to_csv(valid_csv, index=False)

    # 3) Enviar para GCS
    print("Subindo CSVs para GCS...")
    storage_client = storage.Client(project=PROJECT)
    bucket = storage_client.bucket(BUCKET)
    bucket.blob("bmp/train.csv").upload_from_filename(train_csv)
    bucket.blob("bmp/validation.csv").upload_from_filename(valid_csv)

    # 4) Carregar no BigQuery
    print("Criando dataset (se não existir) e carregando tabelas no BigQuery...")
    bq = bigquery.Client(project=PROJECT)
    bq.create_dataset(BQ_DATASET, exists_ok=True)

    def load_csv_to_bq(gcs_uri: str, table: str):
        job = bq.load_table_from_uri(
            gcs_uri,
            f"{PROJECT}.{BQ_DATASET}.{table}",
            job_config=bigquery.LoadJobConfig(
                source_format=bigquery.SourceFormat.CSV,
                autodetect=True,
                skip_leading_rows=1,
                allow_quoted_newlines=True,
            ),
            location=LOCATION,
        )
        job.result()
        print("Carregado no BigQuery:", table)
```

```
load_csv_to_bq(f"gs://{BUCKET}/bmp/train.csv", "train")
load_csv_to_bq(f"gs://{BUCKET}/bmp/validation.csv", "validation")
```

3) Ground truth reprodutível

Objetivo: replicar as regras do tutorial com controle de distribuição (quantis calculados **apenas no `train`**) e aplicadas a `validation`.

Passos concluídos:

1. Script **`rotular_ground_truth.py`** lê `bmp_demo.train` e `bmp_demo.validation` do BigQuery.

2. **Regras aplicadas:**

    - **Realtime** (limiares fixos):

        - `chloride_delta_prior` > 7.7
        - `potassium_plas_delta_prior` < −0.7
        - `calcium_delta_prior` < −1.7

    - **Retrospective** (quantis **do `train`**):

        - `chloride_delta_prior` > q95
        - `chloride_delta_post` < q05
        - `calcium_delta_prior` < q05
        - `calcium_delta_post` > q95

    - **Rótulo final** `label` = `realtime_deltas` **OR** `retrospective_deltas`.

3. Escrita dos resultados:

    - **GCS**: `bmp/train_labeled.csv`, `bmp/validation_labeled.csv`.
    - **BigQuery**: tabelas `bmp_demo.train_labeled`, `bmp_demo.validation_labeled` (com colunas auxiliares + `label`).

4. Checagens rápidas adicionadas (contagem, *positive rate*, nulos nas colunas utilizadas).

Evidências geradas:

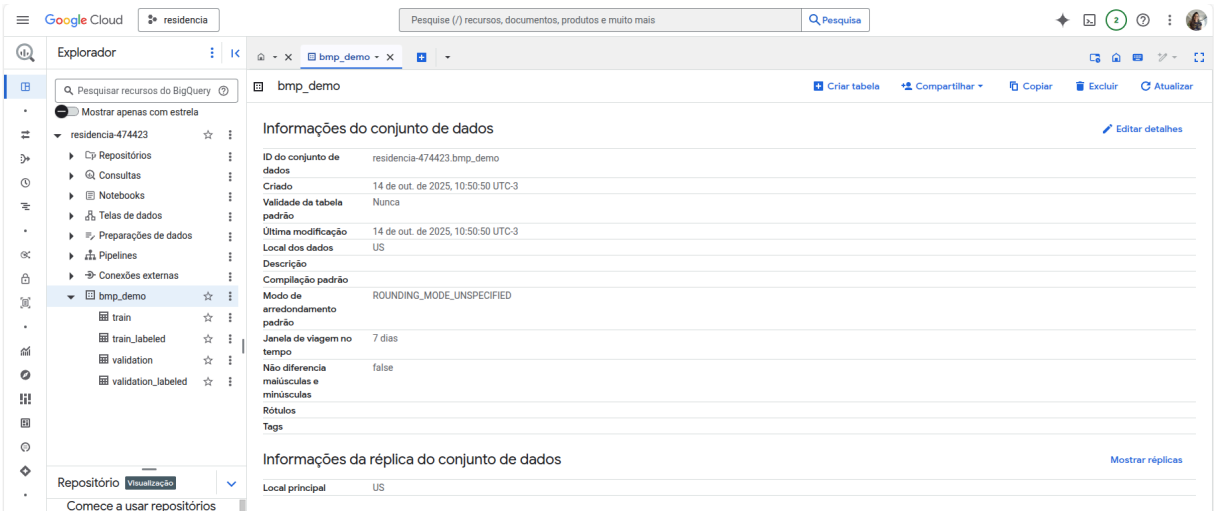
- GCS: `gs://residencia-bucket/bmp/train_labeled.csv`, `.../validation_labeled.csv`.

- BigQuery: `bmp_demo.train_labeled`, `bmp_demo.validation_labeled`.

```
import os, io
import pandas as pd
from google.cloud import bigquery, storage

PROJECT  = os.environ["PROJECT_ID"]
BUCKET   = os.environ["BUCKET_NAME"]
BQ_DATASET = os.environ.get("BQ_DATASET","bmp_demo")
LOCATION = os.environ.get("LOCATION","US")

# Colunas que usaremos(igual no artigo)
COLS = [
    "chloride_delta_prior", "potassium_plas_delta_prior", "calcium_delta_prior",
    "chloride_delta_post", "calcium_delta_post",
    # (opcionais para debug/checagem de NaN)
    "patient_id","specimen_id","drawn_dt_tm"
]

bq = bigquery.Client(project=PROJECT)
st = storage.Client(project=PROJECT)

def bq_to_df(table):
    table_id = f"{PROJECT}.{BQ_DATASET}.{table}"
    tbl = bq.get_table(table_id)
    df = bq.list_rows(tbl).to_dataframe(create_bqstorage_client=True)
    return df

def upload_csv_to_gcs(df: pd.DataFrame, gcs_path: str):
    bucket = st.bucket(BUCKET)
    tmp = "/tmp/_tmp.csv"
    df.to_csv(tmp, index=False)
    bucket.blob(gcs_path).upload_from_filename(tmp)

def load_df_to_bq(df: pd.DataFrame, table: str):
    table_id = f"{PROJECT}.{BQ_DATASET}.{table}"
    job = bq.load_table_from_dataframe(
        df, table_id, location=LOCATION,
job_config=bigquery.LoadJobConfig(write_disposition="WRITE_TRUNCATE")
    )
    job.result()
    print("-> Tabela gravada:", table_id)

# 1) Carregar train e validation
train = bq_to_df("train")
valid = bq_to_df("validation")
print("train shape:", train.shape, "valid shape:", valid.shape)

# 2) Garantir colunas necessárias e lidar com NaNs
for c in COLS:
    if c not in train.columns:
        train[c] = pd.NA
    if c not in valid.columns:
```

```
        valid[c] = pd.NA

# Converter numéricas onde possível
NUMS = [c for c in COLS if c not in ("patient_id","specimen_id","drawn_dt_tm")]
for df in (train, valid):
    for c in NUMS:
        df[c] = pd.to_numeric(df[c], errors="coerce")

# 3) Definir regras

# 3.1 Realtime (limiares fixos do tutorial)
def realtime_rule(df: pd.DataFrame):
    return (
        (df["chloride_delta_prior"] > 7.7) &
        (df["potassium_plas_delta_prior"] < -0.7) &
        (df["calcium_delta_prior"] < -1.7)
    )

# 3.2 Retrospective (limiares por quantil) -- IMPORTANTÍSSIMO: quantis do TRAIN
q95_chl_prior = train["chloride_delta_prior"].quantile(0.95)
q05_chl_post  = train["chloride_delta_post"].quantile(0.05)
q05_ca_prior  = train["calcium_delta_prior"].quantile(0.05)
q95_ca_post   = train["calcium_delta_post"].quantile(0.95)

def retrospective_rule(df: pd.DataFrame):
    return (
        (df["chloride_delta_prior"] > q95_chl_prior) &
        (df["chloride_delta_post"]  < q05_chl_post)  &
        (df["calcium_delta_prior"]  < q05_ca_prior)  &
        (df["calcium_delta_post"]   > q95_ca_post)
    )

# 4) Aplicar regras e criar labels nos dois splits
def add_labels(df: pd.DataFrame):
    df = df.copy()
    df["realtime_deltas"] = realtime_rule(df).fillna(False)
    df["retrospective_deltas"] = retrospective_rule(df).fillna(False)
    df["label"] = (df["realtime_deltas"] | df["retrospective_deltas"]).astype("int8")
    return df

train_lab = add_labels(train)
valid_lab = add_labels(valid)

# 5) Sanidade: proporção positivos
def pos_rate(df):
    return float(df["label"].mean()) if "label" in df else 0.0

print(f"Pos rate (train): {pos_rate(train_lab):.4f}")
print(f"Pos rate (valid): {pos_rate(valid_lab):.4f}")

# 6) Salvar em GCS e BigQuery (tabelas novas)
upload_csv_to_gcs(train_lab, "bmp/train_labeled.csv")
upload_csv_to_gcs(valid_lab, "bmp/validation_labeled.csv")
print("CSVs enviados para GCS: gs://%s/bmp/train_labeled.csv e validation_labeled.csv" % BUCKET)
```

```
load_df_to_bq(train_lab, "train_labeled")
load_df_to_bq(valid_lab, "validation_labeled")

print(" Ground truth gerado e tabelas *_labeled criadas.")
```



4) Treinamento com Vertex AI AutoML (Tabular)

Objetivo: treinar um classificador com alvo `label`, maximizando ROC AUC.

Passos concluídos:

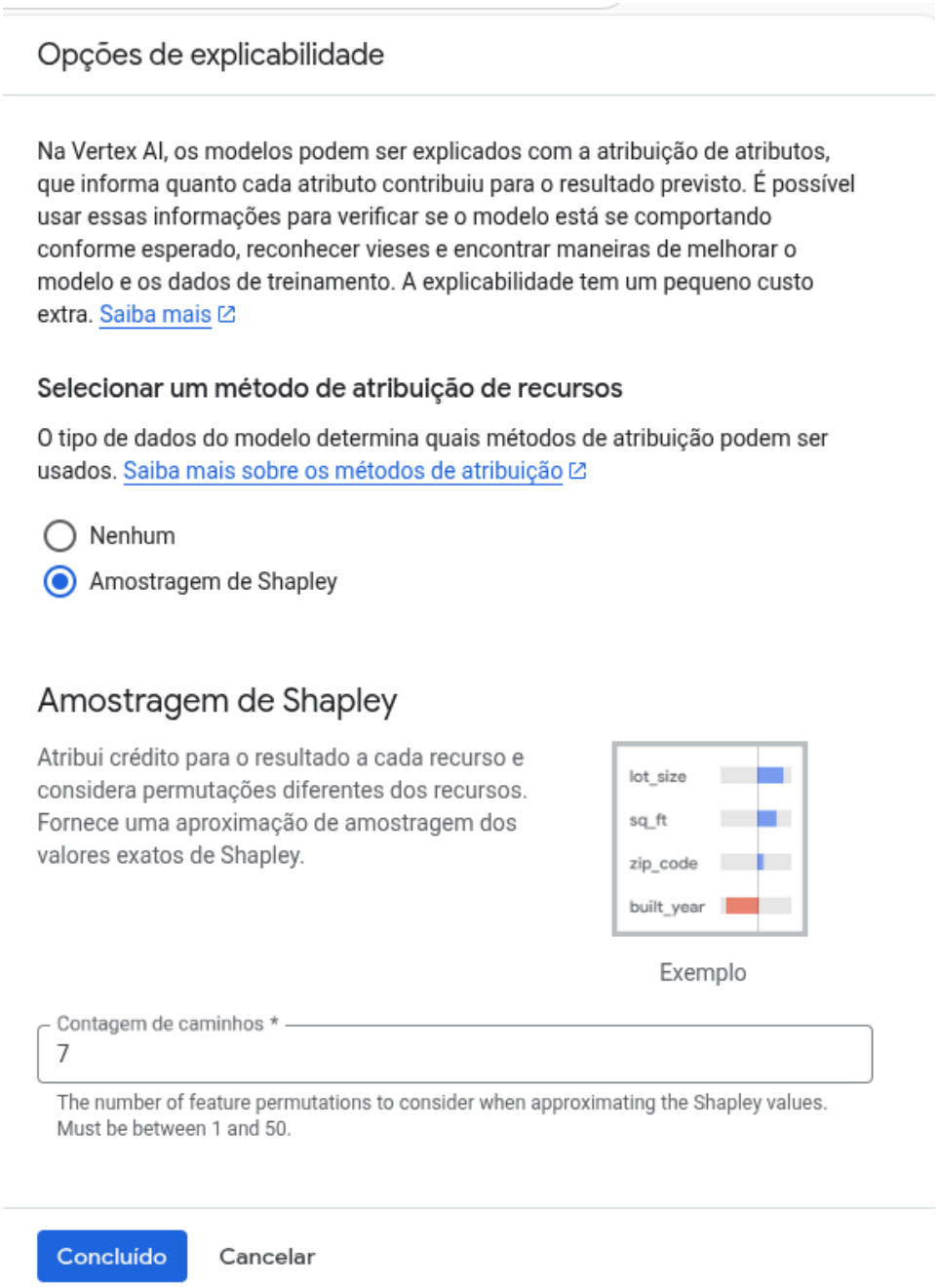
1. **Região do Vertex:** `us-central1`.
2. **Bucket de staging** regional criado (ex.: `residencia-staging-uc1-<sufixo>`).
3. Dataset tabular criado a partir do **CSV no GCS** (`train_labeled.csv`).
4. Job **AutoMLTabularTrainingJob** executado com `optimization_objective = maximize-au-roc`, *split* 80/10/10, `budget_milli_node_hours ≈ 3000`.
5. Modelo treinado disponível em **Vertex** → **Models** (métricas em **Evaluation**). *(Deploy online deixado para etapa posterior para evitar custo contínuo.)*

Observação de regiões: BigQuery e dados em **US**; Vertex em **us-central1** — por isso a fonte do Vertex foi **GCS (CSV)**, evitando incompatibilidades regionais do BigQuery com Vertex.

```
python - <<'PY'
from google.cloud import aiplatform
```

```
PROJECT = "residencia-474423"
LOCATION = "us-central1"  # região do Vertex
STAGING_BUCKET = "gs://${STAGING_BUCKET}"  # preenchido pelo shell
GCS_TRAIN = "gs://residencia-bucket/bmp/train_labeled.csv"  # seu CSV em US
TARGET_COL = "label"

print("Inicializando Vertex...")
aiplatform.init(project=PROJECT, location=LOCATION, staging_bucket=STAGING_BUCKET)

print("Criando Dataset tabular a partir do GCS...")
dataset = aiplatform.TabularDataset.create(
    display_name="bmp_train_labeled",
    gcs_source=[GCS_TRAIN],
)
print("Dataset:", dataset.resource_name)

print("Configurando job AutoML Tabular...")
job = aiplatform.AutoMLTabularTrainingJob(
    display_name="bmp_automl_auc",
    optimization_prediction_type="classification",
    optimization_objective="maximize-au-roc",
)

print("Iniciando treinamento (isso executa em background no Vertex)...")
model = job.run(
    dataset=dataset,
    target_column=TARGET_COL,
    training_fraction_split=0.80,
    validation_fraction_split=0.10,
    test_fraction_split=0.10,
    budget_milli_node_hours=3000,   # aqui vai o dinheirinho
    disable_early_stopping=False,
    sync=True,  # aguarda terminar nesta sessão; troque para False se preferir assíncrono
)

print("Modelo treinado:", model.resource_name)
print("Avaliações disponíveis no console do Vertex (Models > Evaluation).")

# Deploy imediato -- comente se não quiser criar endpoint agora
print("Fazendo deploy em endpoint para predição online...")
endpoint = model.deploy(machine_type="n1-standard-2")
print("Endpoint:", endpoint.resource_name)

PY
```

Durante o deployment foi possível ter contato com ferramentas de explicabilidade.



5) Implantação do modelo

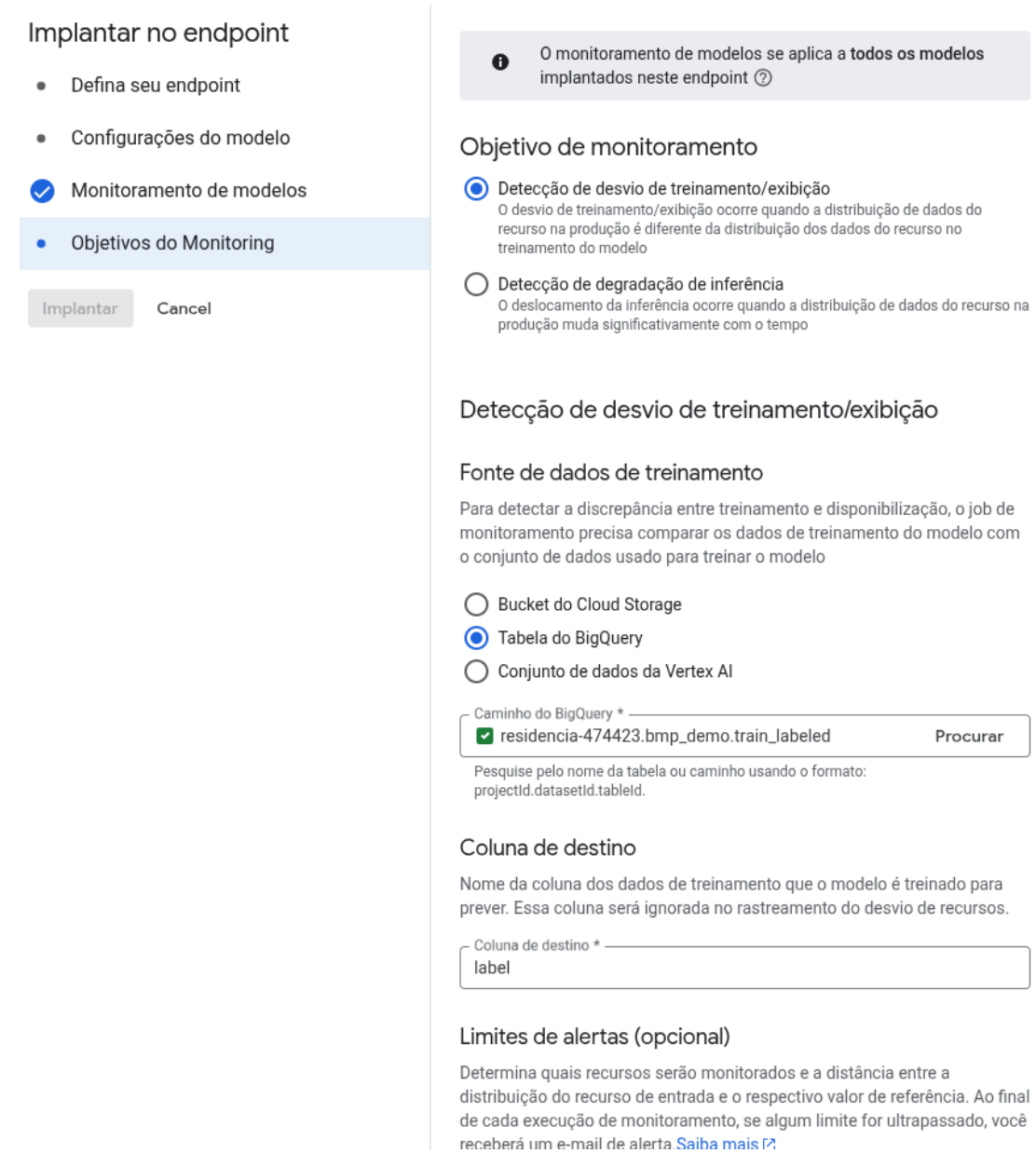
Endpoint: criado no **Vertex AI** em `us-central1` (sem tráfego). Deploy do melhor modelo de AutoML Tabular com `min_replica_count=0` e `max_replica_count=1` (controle de custo). **Tráfego & custo**: mantido **0%** até definir **threshold operacional**; uso apenas para **smoke tests**.

Esquema de requisição: enviar **todas as features** usadas no treino (exceto `label`). **Tipos**: campos **não-delta** como **string** (ex.: `"bun": "16"`), campos **delta** como **float** (ex.: `"chloride_delta_prior": 8.1`). Incluir identificadores (ex.: `patient_id`) como **string**.

```python
# deploy_endpoint.py
from google.cloud import aiplatform

PROJECT  = "residencia-123"
LOCATION = "us-central1"         # Vertex = região
MODEL_DISPLAY_NAME = "bmp_automl_auc"
ENDPOINT_NAME = "bmp-endpoint"          # nome do endpoint

def main():
    aiplatform.init(project=PROJECT, location=LOCATION)

    # pega o modelo mais recente com esse display_name
    models = aiplatform.Model.list(
        filter=f'display_name="{MODEL_DISPLAY_NAME}"',
        order_by="create_time desc",
    )
    if not models:
        raise SystemExit(f"Modelo '{MODEL_DISPLAY_NAME}' não encontrado.")
    model = models[0]
    print("Modelo encontrado:", model.resource_name)

    # cria endpoint se não existir
    endpoints = aiplatform.Endpoint.list(
        filter=f'display_name="{ENDPOINT_NAME}"',
        order_by="create_time desc",
    )
    endpoint = endpoints[0] if endpoints else aiplatform.Endpoint.create(display_name=ENDPOINT_NAME)
    print("Endpoint:", endpoint.resource_name)

    # deploy (com autoscaling)
    deployed = model.deploy(
        endpoint=endpoint,
        machine_type="n1-standard-2",
        min_replica_count=1,
        max_replica_count=3,
        traffic_percentage=100,
        sync=True,
    )
    print("Deploy concluído.")
    print("Endpoint ID:", endpoint.name)

if __name__ == "__main__":
    main()
```

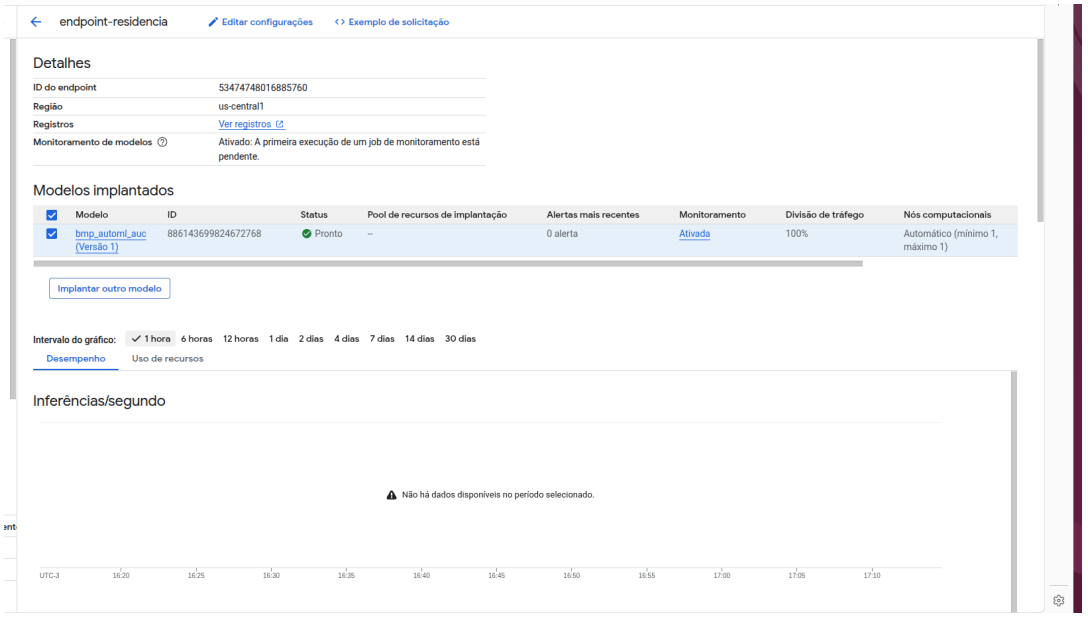
```
# ajuste o ENDPOINT_ID que aparece na tela
export PROJECT_ID="residencia-123"
export LOCATION="us-central1"
```

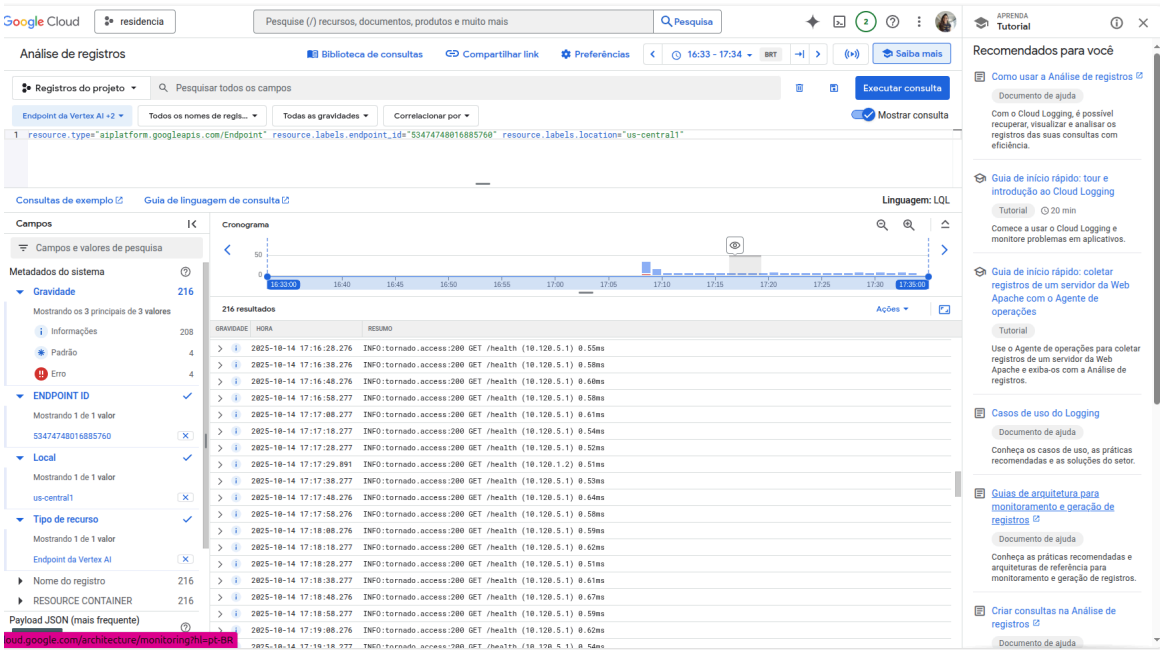
```
export ENDPOINT_ID="141241241"   # <-- troque pelo seu
export THRESHOLD="0.35"                  # ajuste depois

python - <<'PY'
import os, csv, json
from google.cloud import aiplatform

PROJECT  = os.environ["PROJECT_ID"]
LOCATION = os.environ["LOCATION"]
ENDPOINT = os.environ["ENDPOINT_ID"]
THRESH   = float(os.environ.get("THRESHOLD","0.5"))

# baixa 1 amostra do CSV de validação
import subprocess, pandas as pd
subprocess.run(["bash","-lc","gsutil cp gs://residencia-bucket/bmp/validation_labeled.csv
/tmp/validation.csv"], check=True)
df = pd.read_csv("/tmp/validation.csv").head(1)
if "label" in df.columns: df = df.drop(columns=["label"])
# substitui NaN por None (JSON)
inst = df.iloc[0].where(pd.notna(df.iloc[0]), None).to_dict()

aiplatform.init(project=PROJECT, location=LOCATION)
endpoint = aiplatform.Endpoint(ENDPOINT)
resp = endpoint.predict(instances=[inst])

pred = resp.predictions[0]
scores = pred.get("scores") or pred.get("probabilities") or []
p_pos = float(scores[1]) if len(scores)>1 else float(scores[0])
print(json.dumps({"p_pos": round(p_pos,6), "label_hat": int(p_pos>=THRESH)}, indent=2))
PY
```


Implantar no endpoint
Defina seu endpoint
Configurações do modelo
Monitoramento de modelos
Objetivos do Monitoring
Implantar
Cancel
O monitoramento de modelos se aplica a todos os modelos implantados neste endpoint
Objetivo de monitoramento
Detecção de desvio de treinamento/exibição
O desvio de treinamento/exibição ocorre quando a distribuição de dados do recurso na produção é diferente da distribuição dos dados do recurso no treinamento do modelo
Detecção de degradação de inferência
O deslocamento da inferência ocorre quando a distribuição de dados do recurso na produção muda significativamente com o tempo
Detecção de desvio de treinamento/exibição
Fonte de dados de treinamento
Para detectar a discrepância entre treinamento e disponibilização, o job de monitoramento precisa comparar os dados de treinamento do modelo com o conjunto de dados usado para treinar o modelo
Bucket do Cloud Storage
Tabela do BigQuery
Conjunto de dados da Vertex AI
Caminho do BigQuery *
residencia-474423.bmp_demo.train_labeled
Procurar
Pesquise pelo nome da tabela ou caminho usando o formato: projectId.datasetId.tableId.
Coluna de destino
Nome da coluna dos dados de treinamento que o modelo é treinado para prever. Essa coluna será ignorada no rastreamento do desvio de recursos.
Coluna de destino *
label
Limites de alertas (opcional)
Determina quais recursos serão monitorados e a distância entre a distribuição do recurso de entrada e o respectivo valor de referência. Ao final de cada execução de monitoramento, se algum limite for ultrapassado, você
Saiba mais

6) Lições resumidas

- **Regiões importam**: alinhar *storage* / BQ / Vertex com ponte via GCS.
- **Checklist salva tempo**: sempre checar `location`, nomes de bucket/dataset e permissões antes de depurar.
- **Primeiro batch, depois online**: fixar corte e métricas antes de pagar por endpoint.





APÊNDICE 8

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 23 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana 8, dei continuidade aos estudos em **MLOps**, com ênfase em **estratégias de Deploy Release e Monitoramento**.

Na Semana passada, iniciei a implantação do trabalho ***Validating, Implementing, and Monitoring Machine Learning Solutions in the Clinical Laboratory Safely and Effectively***. Porém, treinar ~3 node hours no AutoML + outras operações custou **US$ 50**. Para não depender de novo treinamento a cada ajuste, decidi **servir o modelo original em R** diretamente no Google Cloud, já que o meu foco **não** é treinar modelos.

Contexto importante: modelo offline e transposição de domínio

O modelo original é **offline**: foi treinado e validado **fora do GCP**, no **R/tidymodels**, e o uso era **manual** (rodava scripts localmente). Meu trabalho agora é uma **transposição de domínio**: estou **levando o que foi feito à mão** para um **ambiente gerenciado no GCP**, mantendo o mesmo comportamento do modelo e adicionando:

- **observabilidade** (métricas, logs, alertas),
- **governança** (versionamento e possibilidade de rollback).

O que eu fiz

- Peguei o modelo do projeto **BMP** (R/tidymodels em `.rds`) e criei uma **API** com **plumber** (recebe JSON, devolve predição).
- **Empacotei em container** e publiquei no **Cloud Run**.
- Registrei como **Model/Endpoint** no **Vertex AI** (endereço oficial para predições).
- **Testei predição via REST** no Vertex.

Por que isso ?

- **Fidelidade científica:** uso **o mesmo artefato** validado (.rds).

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próximos passos (Semana 9)

Continuar a implementação do trabalho, com os componentes faltantes:

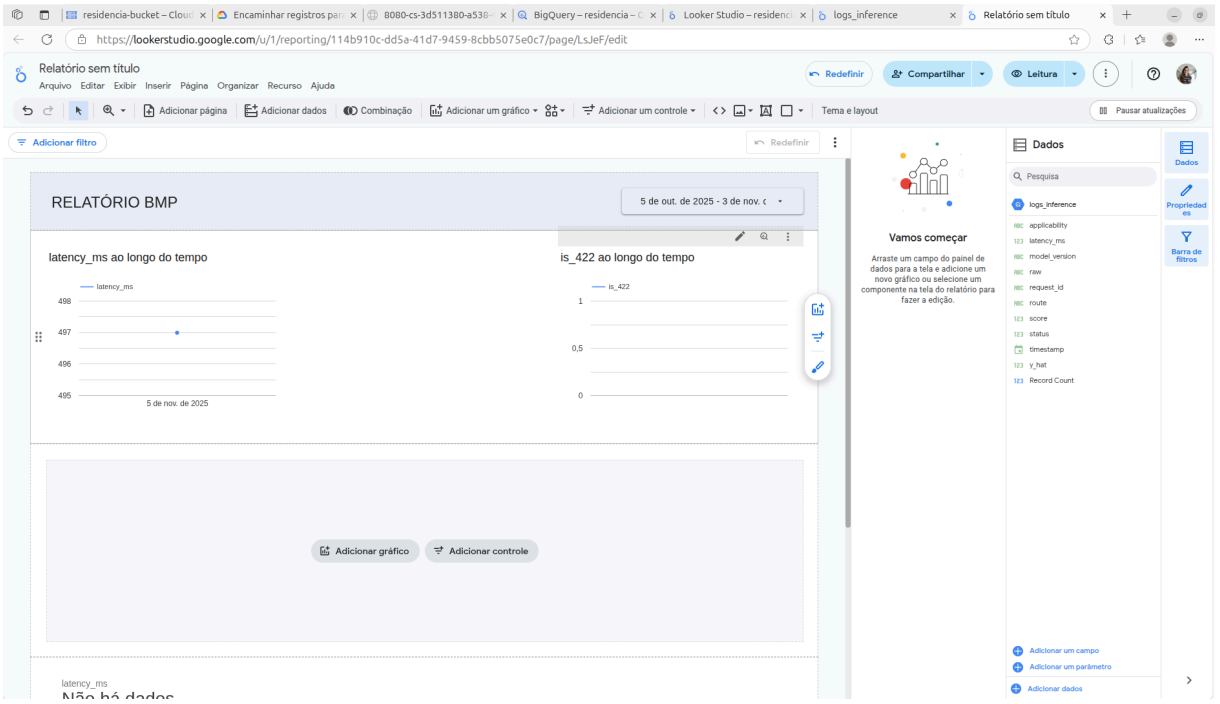
- Explicar modelos e predições (XAI);
- Justiça algorítmica (Fairness);
- Aplicabilidade do modelo (quando NÃO prever).

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Primeiros passos Looker Studio



APÊNDICE 9

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 5 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana 9, dei continuidade aos estudos em **MLOps**, com ênfase em **estratégias de Deploy Release e Monitoramento**.

Na Semana passada, migrei o **modelo validado em R (.rds)** para a **GCP** sem retreinar: empacotei como **serviço HTTP** (plumber), coloquei em **container**, publiquei no **Cloud Run** e registrei no **Vertex AI** para versionar e orquestrar releases. Assim, mantenho o **mesmo comportamento** do modelo validado e ganho governança (canário/rollback) e **observabilidade** (logs e métricas centralizados).

Explicando os pilares

Applicability (segurança de uso)

- Eu comparo os novos exemplos com a "nuvem" de treino (mesmo recipe).
- Se a **distância** (ex.: **Mahalanobis**) passa do **cutoff**, marco como **OOD** e **abstenho (422)**, garantindo que o modelo só responde onde foi validado.

Explainability (XAI)

- **Local** (caso a caso): mostro quanto cada variável empurrou a decisão para cima ou para baixo naquela previsão (pense como pontos que somam/subtraem na nota final).

- **Global** (visão geral): mostro quais variáveis mais pesam, e como variar cada uma costuma mudar a previsão (ex.: "aumentar a glicose geralmente aumenta/diminui o risco e em quais faixas isso acontece").

Fairness (equidade)

- Cruzo **logs de previsões** (no BigQuery) com a **resposta correta** e calculo métricas por **subgrupos** (ex.: faixa etária, sexo).

- As métricas:

- **Sensibilidade:** entre os positivos reais, **quantos** o modelo acerta.
- **Especificidade:** entre os negativos reais, **quantos** o modelo acerta.
- **Precisão nos positivos (PPV):** quando o modelo diz "positivo", **com que frequência** acerta.
- **Precisão nos negativos (NPV):** análogo para "negativo".
- **Falso positivo (FPR):** alarmes falsos.
- **Flag rate:** quanto o modelo **marca como positivo** (quão "agressivo" está).

- O que monitoro como **sinais de injustiça (gaps)**:
  - **Paridade demográfica:** taxas de "positivo" muito diferentes entre grupos?
  - **Igualdade de oportunidades:** acertos/erros (sensibilidade e falsos positivos) parecidos entre grupos?
  - **Paridade preditiva:** quando o modelo diz "positivo", ele acerta **igual** para todos os grupos?

- Se um gap **passa do limiar** combinado, **disparo alerta** para revisão.

O que foi feito

- **Applicability (abstain seguro) dentro do `app.R`**
  - Implementei a checagem de **aplicabilidade** no próprio serviço.
  - Método: distância **Mahalanobis**, calculada após o **mesmo pré-processo do recipe** do `workflow`.
  - Política: se fora da região de aplicabilidade → **HTTP 422** (abstain) com `{"error":"out_of_distribution"}`.
  - **Benefício**: evita outputs confiantes fora do domínio (segurança clínica).

- **Logs estruturados → Cloud Logging → BigQuery**
  - Passei a emitir **JSON estruturado** no stdout (campos: `route`, `status`, `latency_ms`, `score`, `applicability`, `y_hat`, `model_version`, `request_id`, `ts`).
  - Criei **Log Sink**(regra de exportação de logs do Cloud Logging) filtrando o serviço `bmp-plumber`, e escrevendo no dataset `bmp(Big Query)`.
- **XAI**
  - **Local (on-demand):** rota `POST /explain` preparada para **SHAP local** a partir do mesmo `workflow`.
  - **Global (batch):** criei um **Cloud Run Job** que lê `model_workflow.rds` + `train_labeled.csv` no **GCS**, gera:
    - **SHAP global** (beeswarm + top-features),
    - **PDP(Partial Dependence Plot)/ALE(Accumulated Local Effects)**
    - Publica artefatos (PNGs/JSON) em `gs://bmp-xai-artifacts/...`.

- Comecei a elaboração de **Dashboards no Looker Studio** (imagens do GCS + tabelas do BQ).

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próximos passos (Semana 10)

Finalizar a implementação do trabalho:

- Agendar o job de explicabilidade global (Cloud Run Jobs + Scheduler).
- Criar métricas/alertas:
  - Latência p95 do serviço,
  - Taxa de 422 (fora de domínio),
  - Disponibilidade.
- Fairness completo: agendar job semanal que cruza com a resposta correta oficial (quando disponível) e grava métricas por subgrupo em uma tabela dedicada (`metrics_fairness`).
- Dashboard final no Looker Studio unindo: latência/erros, aplicabilidade, importâncias globais e fairness.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

APÊNDICE 10

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 5 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Maria Eduarda Silva Borba

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na Semana passada, apresentei os três pilares que nortearam o fechamento do projeto: **Applicability**, **Explainability** e **Fairness** — dimensões fundamentais para garantir a confiabilidade e a transparência de um modelo em produção. O objetivo era construir um pipeline de deploy que não apenas funcionasse de forma automatizada, mas que também oferecesse interpretações, limites de aplicabilidade e métricas de equidade.

Nesta Semana, finalizei esse desenvolvimento. Concluí o **deploy do modelo preditivo** utilizando o serviço **Cloud Run**, empacotando-o em um **container Docker** — um ambiente padronizado que garante que o modelo funcione da mesma forma em qualquer máquina.
Esse contêiner foi publicado no **Artifact Registry** e integrado a uma estrutura de **observabilidade**, com **BigQuery**, **Cloud Logging** e **Cloud Monitoring**, que permitem acompanhar o comportamento do sistema em tempo real.

Implementei políticas automáticas de monitoramento para **latência**, **erros 422**, que indicam quando o modelo recebe dados muito diferentes do que foi treinado, e **disponibilidade do serviço**, para garantir que ele permaneça ativo e estável.

Além disso, configurei a geração automática de **métricas de explicabilidade global** (XAI Global). Esse módulo mostra, de forma agregada, **quais variáveis mais influenciam as previsões do modelo**, facilitando a interpretação e aumentando a transparência.

Com essa infraestrutura, agora é possível acompanhar **em tempo real** o desempenho e o comportamento do modelo em produção, estabelecendo uma base sólida para criar **painéis visuais e relatórios interativos** no Looker Studio. Os resultados podem ser encontrados no documento.

Desde o início, defini que meu foco seria compreender e aplicar, na prática, os princípios de **MLOps**, o conjunto de práticas que une **ciência de dados, engenharia de software e operações em nuvem** para garantir que modelos de IA possam ser implantados, atualizados e monitorados de forma contínua e confiável.
Ao longo das semanas, aprofundei meus estudos no **ecossistema do Google Cloud Platform (GCP)**, consolidando uma visão prática sobre **pipelines de machine learning** escaláveis, interpretáveis e

monitoráveis.

Essa experiência me permitiu compreender como os pilares de **reprodutibilidade**, **automação** e **observabilidade** se articulam em um ambiente de MLOps moderno.
Também reforçou a importância de garantir **explicabilidade e responsabilidade** no ciclo de vida dos modelos, abrindo caminho para a integração futura de **métricas de equidade (fairness)** e **dashboards analíticos** no Looker Studio.

Em resumo, este projeto me proporcionou uma vivência completa — do contêiner ao monitoramento, passando pela instrumentação de métricas e explicabilidade — consolidando uma base sólida para atuar na interseção entre **engenharia de machine learning** e **operações em nuvem**.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

O ambiente do GCP oferece componentes gerenciados e integrações "prontas para produção" que aceleram o desenvolvimento, mas essa conveniência tem um preço "altíssimo"!

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Deploy, Observabilidade e XAI no GCP (Cloud Run)

Serviço: **bmp-r** · Linguagem: **R / plumber** · Foco: **Deploy + Logs (BQ) + Monitoring + XAI batch + Fairness**

0) Pré-requisitos

- Projeto GCP ativo e faturamento habilitado.
- APIs habilitadas:
  `run.googleapis.com`, `artifactregistry.googleapis.com`, `cloudbuild.googleapis.com`, `logging.googleapis.com`, `bigquery.googleapis.com`, `monitoring.googleapis.com`, `cloudscheduler.googleapis.com`, `iamcredentials.googleapis.com`.
- `gcloud` + `bq` + `gsutil` configurados localmente.
- Artefatos no diretório:
  `Dockerfile`, `app.R`, `model_workflow.rds`, `applicability_refs.rds`, `xai_global.R`, `background_xai.csv`, `bmp_train_labeled.csv`.

Obs: Todos os artefatos necessários podem ser encontrados no repositório.

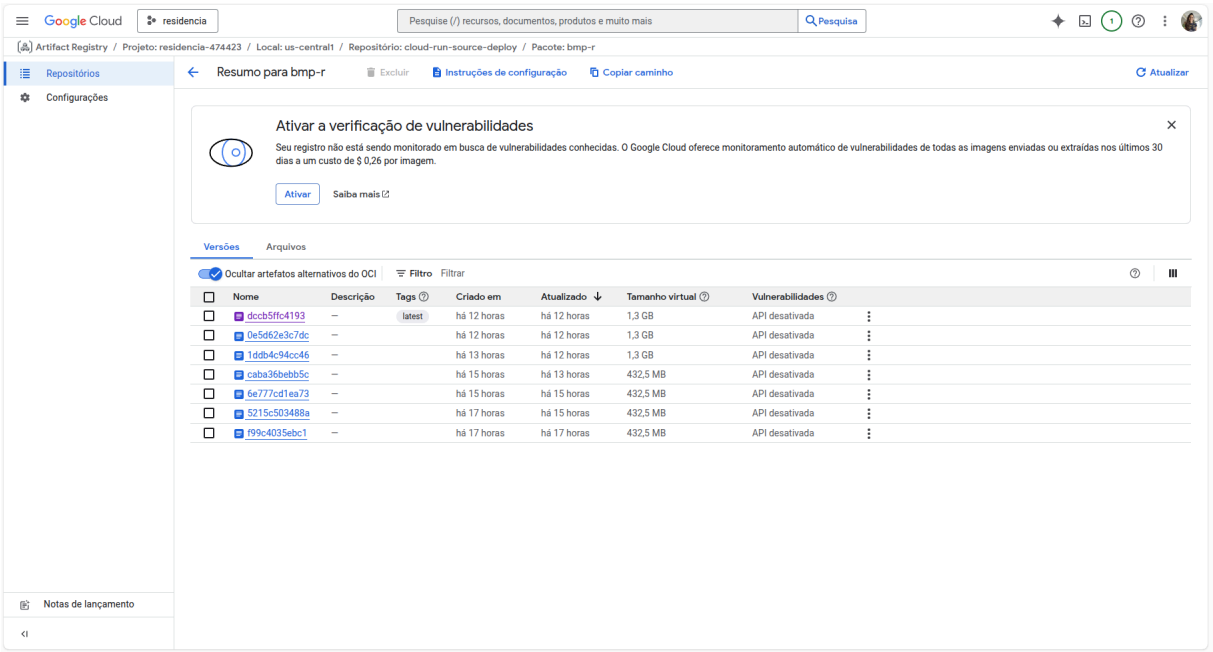
Variáveis padrão

```
export PROJECT_ID="residencia-474423" # mude conforme o seu projeto

export REGION="us-central1"

export REPO="ml-apps"       # nome do repo no Artifact Registry

export IMAGE="bmp-plumber-xai"

export SERVICE="bmp-r"

export DATASET="bmp"

export SINK_NAME="bmp-r-bq-sink"

export BQ_TABLE_LOGS="logs_inference"
```

```
export GCS_BUCKET="gs://residencia-bucket"
```

1) Artifact Registry (opcional, se quiser buildar local e push)

```
gcloud config set project $PROJECT_ID

gcloud artifacts repositories create $REPO \
  --repository-format=docker --location=$REGION || true


# build local

docker build -t $REGION-docker.pkg.dev/$PROJECT_ID/$REPO/$IMAGE:latest .


# push

gcloud auth configure-docker $REGION-docker.pkg.dev

docker push $REGION-docker.pkg.dev/$PROJECT_ID/$REPO/$IMAGE:latest
```



Versionamento de artefatos.

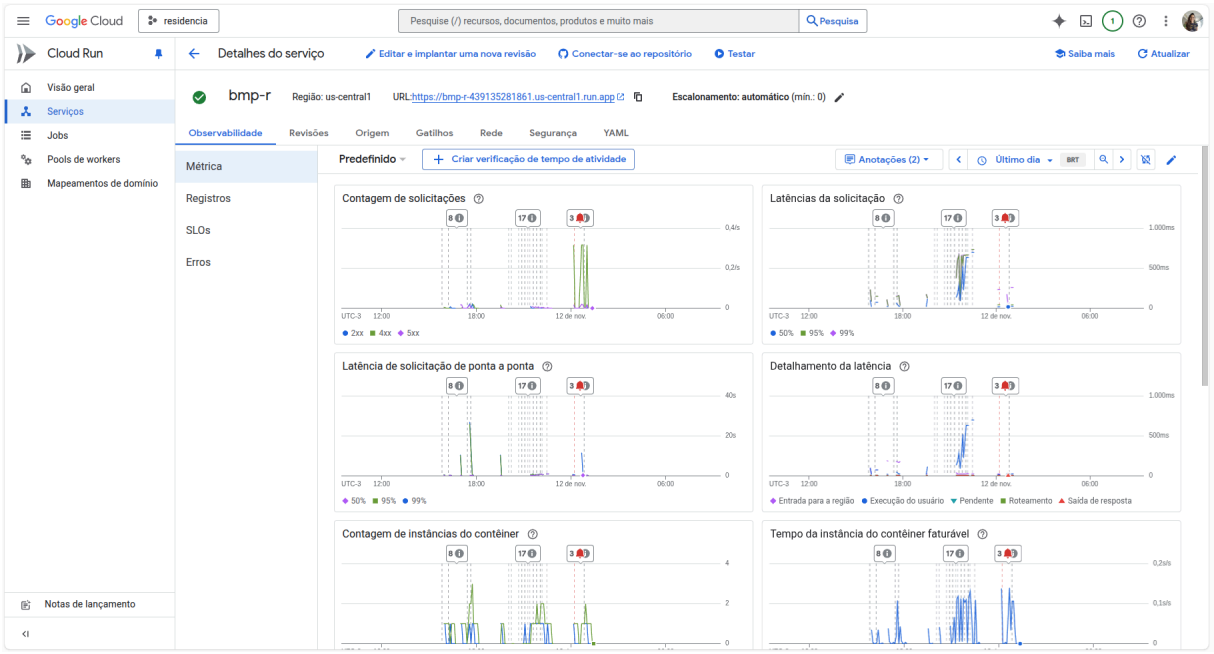
2) Deploy no Cloud Run

> **Importante:** usamos **4 GiB** de memória (tidymodels + .rds) e expomos **PORT 8080**. Com menos GiB faltou memória.

```
gcloud run deploy $SERVICE \

  --image $REGION-docker.pkg.dev/$PROJECT_ID/$REPO/$IMAGE:latest \

  --region $REGION \

  --allow-unauthenticated \

  --memory=4Gi \

  --set-env-vars
MODEL_VERSION=bmp_v1,XAI_BACKGROUND_PATH=/app/background_xai.csv
```

Checagens rápidas:

```
SERVICE_URL=$(gcloud run services describe $SERVICE --region $REGION
--format='value(status.url)')

curl "$SERVICE_URL/healthz/"    # esperado: {"status":"ok", ...}

curl "$SERVICE_URL/schema"      # lista de preditores
```



Observabilidade encontrada no deploy do Cloud Run.

3) Logging → BigQuery (sink)

3.1 Criar dataset e sink (ou recriar)

```
bq --location=US mk -d --description "BMP logs e métricas"
$PROJECT_ID:$DATASET || true


gcloud logging sinks create $SINK_NAME \
  "bigquery.googleapis.com/projects/$PROJECT_ID/datasets/$DATASET" \
  --log-filter='resource.type="cloud_run_revision" AND
resource.labels.service_name="'$SERVICE'"' \
  || echo "Sink já existe"
```

Se já existia e você apagou tabelas manualmente, o sink recria as tabelas diárias (`run_googleapis_com_stdout_YYYYMMDD`, etc.) conforme novos logs chegarem.

3.2 Tabela “afinada” de inferências (campos amigáveis)

A tabela abaixo é um **resumo** feito a partir do `stdout` com JSON estruturado que o `app.R` escreve (ts, route, status, latency_ms, applicability, model_version, request_id, **pred_class**, **score** se disponível).

```
bq query --use_legacy_sql=false "

CREATE OR REPLACE TABLE \`$PROJECT_ID.$DATASET.$BQ_TABLE_LOGS\` AS

SELECT

  TIMESTAMP(JSON_VALUE(textPayload, '$.ts'))              AS ts,

  JSON_VALUE(textPayload, '$.route')                      AS route,

  CAST(JSON_VALUE(textPayload, '$.status') AS INT64)      AS status,

  CAST(JSON_VALUE(textPayload, '$.latency_ms') AS FLOAT64) AS latency_ms,

  JSON_VALUE(textPayload, '$.applicability')              AS
applicability,

  JSON_VALUE(textPayload, '$.model_version')              AS
model_version,

  JSON_VALUE(textPayload, '$.request_id')                 AS request_id,

  JSON_VALUE(textPayload, '$.pred_class')                 AS pred_class,

  CAST(JSON_VALUE(textPayload, '$.score') AS FLOAT64)     AS score

FROM \`$PROJECT_ID.$DATASET.run_googleapis_com_stdout_*\`

WHERE JSON_VALUE(textPayload, '$.route') IS NOT NULL;

"
```

Depois, para manter sempre atualizada, salve a consulta acima como **view materializada** ou rode via **Scheduler + bq query** diariamente.

4) Monitoring — Políticas de alerta

Crie uma pasta `monitoring/` com os JSONs abaixo.

4.1 Latência (usando métrica gerenciada do Cloud Run)

Para latência, recomendamos **usar a métrica gerenciada** do Cloud Run (Request Latency). Exemplo com **média** de 5m > 1200 ms:

`monitoring/policy_latency.json`

```
{
  "displayName": "bmp-r - Latência média > 1200ms (5m)",
  "combiner": "OR",
  "enabled": true,
  "conditions": [{
    "displayName": "REQ LATENCY mean > 1200ms",
    "conditionMonitoringQueryLanguage": {
      "query": "fetch cloud_run_revision\n| metric
'run.googleapis.com/request_latencies'\n| filter (resource.service_name ==
'bmp-r' && resource.location == 'us-central1')\n| align mean_aligner(5m)\n|
group_by [],\n    [val: mean(value.request_latencies)]\n| condition val >
1.2"
    },
    "duration": "0s",
    "trigger": { "count": 1 }
  }],
  "notificationChannels": []
}
```

Criar:

```
gcloud alpha monitoring policies create
--policy-from-file=monitoring/policy_latency.json
```

4.2 Taxa de 422 (Out-of-Distribution)

Usamos **MQL** lendo o **código de status** da métrica gerenciada `run.googleapis.com/request_count`. Ajuste o limiar.

`monitoring/policy_422.json`

```
{

  "displayName": "bmp-r - Taxa 422 > 20% (5m)",

  "combiner": "OR",

  "enabled": true,

  "conditions": [{

    "displayName": "422 rate > 0.2",

    "conditionMonitoringQueryLanguage": {

      "query": "fetch cloud_run_revision\n| metric
'run.googleapis.com/request_count'\n| filter (resource.service_name ==
'bmp-r' && resource.location == 'us-central1')\n| align rate(5m)\n|
group_by [], [tot: sum(value.request_count)]\n| { fetch
cloud_run_revision\n    | metric 'run.googleapis.com/request_count'\n    |
filter (resource.service_name == 'bmp-r' && resource.location ==
'us-central1' && metric.response_code_class == '4xx' &&
metric.response_code == 422)\n    | align rate(5m)\n    | group_by [],
[err: sum(value.request_count)] }\n| ratio_err: err / tot\n| condition
ratio_err > 0.2"

    },

    "duration": "0s",

    "trigger": { "count": 1 }

  }],
```

```
  "notificationChannels": []

}
```

Criar:

```
gcloud alpha monitoring policies create
--policy-from-file=monitoring/policy_422.json
```

4.3 Uptime do `/healthz`

`monitoring/policy_uptime.json`

```
{

  "displayName": "bmp-r - Uptime /healthz falhou",

  "combiner": "OR",

  "enabled": true,

  "conditions": [{

    "displayName": "Health check failed",

    "conditionMonitoringQueryLanguage": {

      "query": "fetch uptime_url\n| metric
'monitoring.googleapis.com/uptime_check/check_passed'\n| filter
(resource.project_id == '"PROJECT_ID"')\n| align next_older(1m)\n| group_by
[], [p: max(value.check_passed)]\n| condition p < 1"

    },

    "duration": "0s",

    "trigger": { "count": 1 }

  }],

  "notificationChannels": []

}
```

Crie um **Uptime Check** apontando para `$SERVICE_URL/healthz/` antes.
Substitua `"PROJECT_ID"` no JSON (ou crie via Console e reaproveite).

Criar:

```
gcloud alpha monitoring policies create
--policy-from-file=monitoring/policy_uptime.json
```

5) XAI Global — Cloud Run Job + Scheduler

5.1 Job

```
gcloud run jobs create xai-global-job \

  --image $REGION-docker.pkg.dev/$PROJECT_ID/$REPO/$IMAGE:latest \

  --region $REGION \

  --task-timeout=3600 \

  --set-env-vars MODEL_VERSION=bmp_v1 \

  --command "Rscript" --args "/app/xai_global.R" \

  --memory=4Gi
```

5.2 Scheduler diário (03:00 BRT)

Precisa de uma **service account** com permissão de invocar jobs de Cloud Run.

```
export SA_EMAIL="scheduler-invoker@$PROJECT_ID.iam.gserviceaccount.com"


gcloud iam service-accounts create scheduler-invoker --display-name
"Scheduler Invoker"

gcloud projects add-iam-policy-binding $PROJECT_ID \

  --member="serviceAccount:$SA_EMAIL" \

  --role="roles/run.invoker"


gcloud scheduler jobs create http xai-global-daily \
```

```
  --schedule="0 3 * * *" \
  --time-zone="America/Sao_Paulo" \

--uri="https://$REGION-run.googleapis.com/apis/run.googleapis.com/v1/namespaces/$PROJECT_ID/jobs/xai-global-job:run" \
  --http-method=POST \
  --oauth-service-account-email="$SA_EMAIL" \
  --oauth-token-scope='https://www.googleapis.com/auth/cloud-platform'
```

6) Fairness — Tabela e rotina diária

6.1 Tabelas auxiliares (exemplo)

- **Ground truth**: `bmp.ground_truth` (colunas mínimas: `request_id`, `target` (0/1), `sex`, `age`).
- **Logs de inferência**: `bmp.logs_inference` (já criado).

6.2 Consulta diária (cria/atualiza métricas por subgrupo)

```
bq query --use_legacy_sql=false "
CREATE OR REPLACE TABLE \`$PROJECT_ID.$DATASET.metrics_fairness\` AS
WITH base AS (
  SELECT
    li.ts,
    DATE(li.ts) AS dt,
    gt.sex,
    gt.age,
    CASE
      WHEN gt.age < 30 THEN '<30'
```

```
      WHEN gt.age BETWEEN 30 AND 49 THEN '30-49'
      WHEN gt.age BETWEEN 50 AND 69 THEN '50-69'
      ELSE '70+'
    END AS age_bucket,
    SAFE_CAST(gt.target AS INT64) AS y_true,
    CASE WHEN li.pred_class = '.pred_1' THEN 1 ELSE 0 END AS y_pred
  FROM \`$PROJECT_ID.$DATASET.$BQ_TABLE_LOGS\` li
  JOIN \`$PROJECT_ID.$DATASET.ground_truth\` gt
    USING (request_id)
  WHERE li.route = '/predict'
),
agg AS (
  SELECT
    dt, sex, age_bucket,
    COUNT(*) AS n,
    SUM(CASE WHEN y_true=1 AND y_pred=1 THEN 1 ELSE 0 END) AS tp,
    SUM(CASE WHEN y_true=0 AND y_pred=1 THEN 1 ELSE 0 END) AS fp,
    SUM(CASE WHEN y_true=1 AND y_pred=0 THEN 1 ELSE 0 END) AS fn,
    SUM(CASE WHEN y_true=0 AND y_pred=0 THEN 1 ELSE 0 END) AS tn,
    AVG(y_pred) AS flag_rate
  FROM base
  GROUP BY 1,2,3
)
SELECT
```

```
  dt, sex, age_bucket, n, flag_rate,
  SAFE_DIVIDE(tp, tp+fn) AS tpr,
  SAFE_DIVIDE(tn, tn+fp) AS tnr,
  SAFE_DIVIDE(fp, fp+tn) AS fpr,
  SAFE_DIVIDE(tp, tp+fp) AS ppv,
  SAFE_DIVIDE(tn, tn+fn) AS npv
FROM agg;
"
```

Para **agendamento**, use um **Cloud Scheduler** que chama um **Cloud Run Job** executando `bq query` (ou um **Workflow** se preferir).

7) Testes rápidos

7.1 Health & Schema

```
curl "$SERVICE_URL/healthz/"
curl "$SERVICE_URL/schema"
```

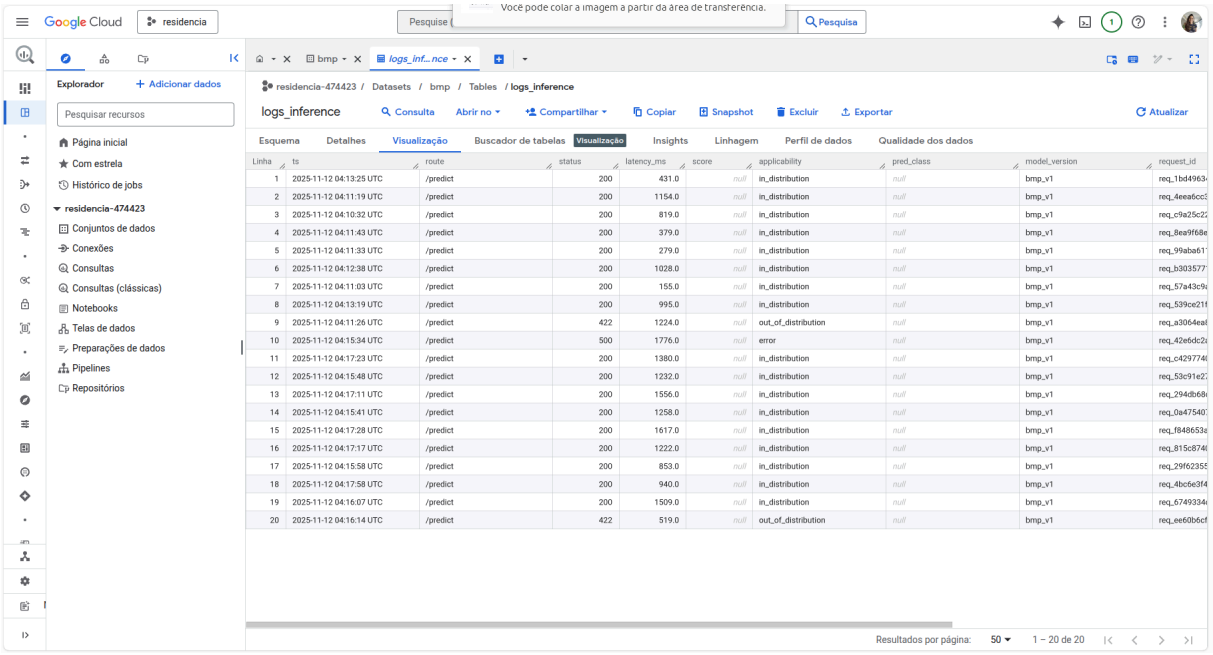
7.2 Predict (exemplo)

```
curl -s -X POST "$SERVICE_URL/predict/" \
  -H "Content-Type: application/json" \
  -d '{
    "sodium": 139, "chloride": 107, "potassium_plas": 4.3, "co2_totl": 25,
    "bun": 15, "creatinine": 0.8, "calcium": 9.5, "glucose": 90
  }' | jq .
```

7.3 BigQuery — últimas linhas

```
bq query --use_legacy_sql=false "
```

```
SELECT ts, route, status, latency_ms, applicability, pred_class, score,
model_version

FROM \`$PROJECT_ID.$DATASET.$BQ_TABLE_LOGS\`

ORDER BY ts DESC

LIMIT 20;"
```



Exemplo visualização Big Query.

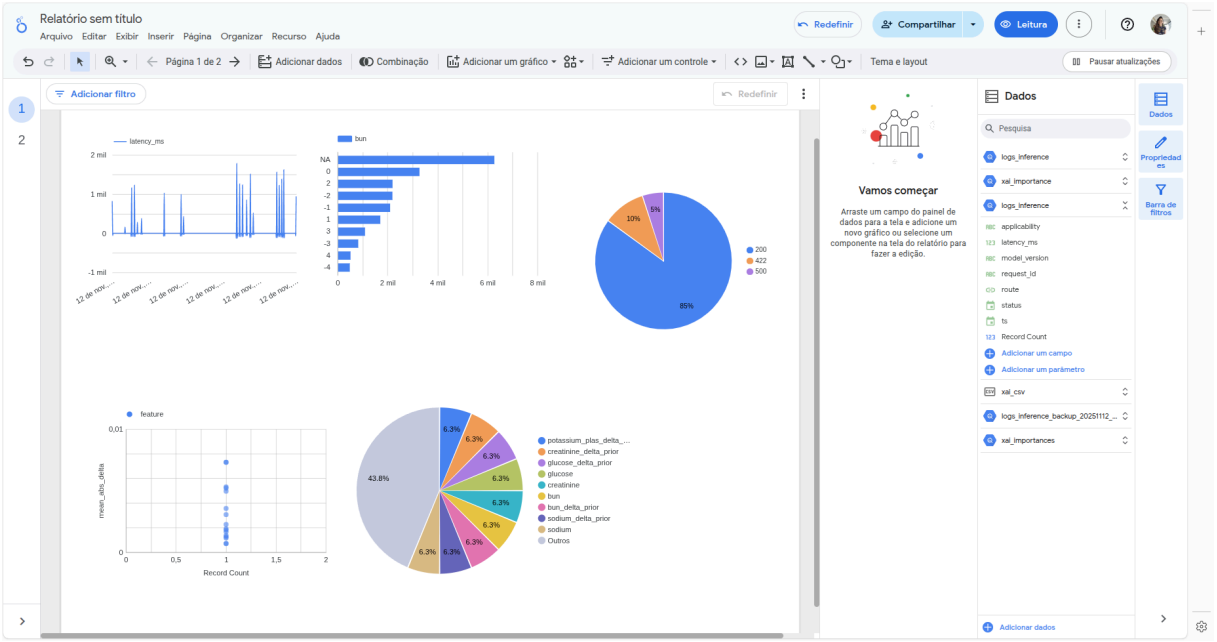
8) Atualização, rollback e troubleshooting

- **Atualizar imagem**: re-build → push → `gcloud run deploy ...`.
- **Rollback**:
  ```
  gcloud run services list-revisions --service $SERVICE --region
  $REGION
  gcloud run services update-traffic $SERVICE --to-revisions REV=100
  --region $REGION
  ```
- **Logs do serviço**:
  ```
  gcloud logging read 'resource.type="cloud_run_revision" AND
  resource.labels.service_name="'$SERVICE'"' --limit=50
  --format='value(textPayload)'
  ```

- **Erros comuns**
  - **MemoryExceeded**: aumente `--memory` (2–4 GiB).
  - **PORT**: API deve **ouvir em 8080** (`pr$run(...,port=as.integer(Sys.getenv('PORT','8080')))`).
  - **BQ vazio**: aguarde a criação das tabelas diárias `run_googleapis_com_*` e **rode a query de consolidação** p/ `logs_inference`.

9) Looker Studio

- **Fontes**:
  - BigQuery: `bmp.logs_inference`, `bmp.metrics_fairness`, (artefatos XAI do GCS se quiser via CSV).
  - Monitoring: latência / uptime.
- **Gráficos**:
  - Série de **latência** (média ou p95 gerenciado).
  - **% 422** por dia (barra/linha).
  - **Status code** (2xx/4xx/5xx/422) empilhado.
  - **Aplicabilidade** (in vs out) + **versão do modelo**.
  - **Importância Global** (Top-N features).
  - **Fairness**: TPR/TNR/FPR/PPV/NPV/flag_rate por **sexo × faixa etária**.



Exemplo do serviço Looker Studio

10) Observações

- **XAI offline**: custo previsível, não impacta latência do `/predict`.
- **Campos nos logs**: mantenha `pred_class` e `score` no JSON do `stdout` (o SQL já extrai).
- **Reprodutibilidade**: pin de versão (`MODEL_VERSION`) e `request_id` em cada inferência.
- **Ciclo de dados**: fairness requer **ground truth** unido por `request_id` (ou outra chave estável).