



JORDAN ROBSON RIBEIRO BASTOS

**BUSINESS INTELLIGENCE E MACHINE LEARNING NA MELHORIA
DA PERFORMANCE OPERACIONAL: UM ESTUDO PRÁTICO EM
EQUIPE CORPORATIVA.**

Balneário Camboriú

2025

JORDAN ROBSON RIBEIRO BASTOS

**BUSINESS INTELLIGENCE E MACHINE LEARNING NA MELHORIA
DA PERFORMANCE OPERACIONAL: UM ESTUDO PRÁTICO EM
EQUIPE CORPORATIVA.**

Trabalho de conclusão de curso apresentado ao curso de Sistemas de Informação do Centro Universitário Avantis de Balneário Camboriú para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Evandro Luís Viapiana.

Balneário Camboriú

2025

JORDAN ROBSON RIBEIRO BASTOS

**BUSINESS INTELLIGENCE E MACHINE LEARNING NA MELHORIA
DA PERFORMANCE OPERACIONAL: UM ESTUDO PRÁTICO EM
EQUIPE CORPORATIVA.**

Trabalho de conclusão de curso apresentado ao curso de Sistemas de Informação do Centro Universitário Avantis de Balneário Camboriú para a obtenção do título de Bacharel em Sistemas de Informação.

Balneário Camboriú (SC), __ de Dezembro de 2025.

Prof. Evandro Luís Viapiana. – Orientador.

BANCA EXAMINADORA

Prof. Nome do Professor – Avaliador.

Prof. Nome do Professor – Avaliador.

Prof. Nome do Professor – Avaliador.

AGRADECIMENTOS

A realização deste Trabalho de Conclusão de Curso foi uma jornada de grande aprendizado e dedicação, que contou com o apoio fundamental de diversas pessoas. A elas, registro meus sinceros agradecimentos.

Agradeço à minha família, pelo apoio constante e pela compreensão durante todo o período da graduação e, em especial, na fase de desenvolvimento deste projeto.

À minha noiva, Dienifer, que foi meu alicerce durante esta caminhada, cuja paciência e incentivo constante foram fundamentais para que eu mantivesse o foco e concluisse este TCC com determinação.

Ao meu orientador, Professor Evandro Luís Viapiana, registro minha profunda gratidão pelas orientações precisas, pela disponibilidade em todos os momentos e pelas contribuições essenciais que elevaram o nível deste trabalho.

Estendo meus agradecimentos ao corpo docente de Sistemas de Informação, pelos conhecimentos compartilhados ao longo da graduação.

Aos meus colegas de classe e de trabalho, agradeço pelo companheirismo, pelas trocas de ideias e pelo apoio mútuo, que tornaram a caminhada mais colaborativa.

"Em Deus nós confiamos. Todos
os outros devem trazer dados."

W. Edwards Deming

RESUMO

Este trabalho apresenta o desenvolvimento e a avaliação de uma solução integrada de *Business Intelligence* (BI) e *Machine Learning* (ML) para a melhoria da performance operacional de uma equipe de certificação digital no setor de crédito imobiliário. A metodologia partiu da identificação de gargalos operacionais (como retrabalho e falta de visibilidade) e abrangeu um processo de engenharia de dados (ETL) para extrair e transformar dados de plataformas como Pipefy e Snowflake. Esses dados foram modelados no Power BI (em Floco de Neve), onde foi desenvolvida uma solução de BI com seis dashboards para análise diagnóstica. Paralelamente, foi treinado um modelo de Machine Learning (*RandomForestClassifier*) em Python, que atingiu 93% de AUC e 85% de acurácia na previsão de atrasos de SLA. Este modelo foi integrado nativamente ao Power BI via *script Python*, culminando em um sétimo dashboard focado em gestão proativa de risco. A solução final, já em uso na empresa, transforma a gestão da equipe de reativa para preditiva, permitindo a priorização de solicitações com alto risco de atraso.

Palavras-Chaves: Business Intelligence. Machine Learning. Lead Time. Otimização de Processos.

ABSTRACT

This paper presents the development and evaluation of an integrated Business Intelligence (BI) and Machine Learning (ML) solution for improving the operational performance of a digital certification team in the real estate credit sector. The methodology began with the identification of operational bottlenecks (such as rework and lack of visibility) and encompassed a data engineering (ETL) process to extract and transform data from platforms like Pipefy and Snowflake. This data was modeled in Power BI (using a Snowflake Schema), where a BI solution with six dashboards was developed for diagnostic analysis. Concurrently, a Machine Learning model (RandomForestClassifier) was trained in Python, which achieved 93% AUC and 85% accuracy in predicting Service Level Agreement (SLA) delays. This model was natively integrated into Power BI via a Python script, culminating in a seventh dashboard focused on proactive risk management. The final solution, already in use at the company, transforms the team's management from reactive to predictive, enabling the prioritization of requests with a high risk of delay.

Keywords: *Business Intelligence. Machine Learning. Lead Time. Process Optimization*

LISTA DE ILUSTRAÇÕES

Figura 1 - Elementos Intervenientes No Processo De Tomada De Decisão.....	21
Figura 2 - Níveis Da Tomada De Decisão Em Uma Organização.....	23
Figura 3 - Níveis de Maturidade no Uso da Informação nas Organizações.....	25
Figura 4 - Componentes Do Power Bi.....	29
Figura 5 - Resumo Funcionamento Snowflake.....	33
Figura 6 - Fluxo De Fases Dentro Do Sistema Pipefy.....	34
Figura 7- Visão Geral De Como Funciona O Processo De Etl.....	35
Figura 8- Visão Geral do funcionamento da Azure Machine Learning.....	36
Figura 9 - Tela inicial do Azure Machine Learning Studio.....	38
Figura 10 - Ilustração do Crescimento Leaf-wise do LightGBM.....	41
Figura 11 - Fluxograma "As-Is" do Processo de Certificação Digital.....	45
Figura 12 - Janela de Conectores de Dados do Power BI Desktop.....	52
Figura 13 - Interface do Editor do Power Query com as Consultas Carregadas.....	53
Figura 14 - Consulta Power Query para Criação da tabela DATA_ATUALIZAÇÃO.....	55
Figura 15 - Código M para Coluna Personalizada Idade Cliente.....	56
Figura 16 - Código M para Coluna Personalizada Período do Mês.....	56
Figura 17 - Diagrama do Modelo Relacional Floco de Neve (Snowflake Schema).....	58
Figura 18 - Código DAX da Coluna Calculada SLA_Dias_Uteis.....	60
Figura 19 - Código DAX da Medida Total Solicitações (Criadas).....	60
Figura 20 - Código DAX da Medida Total Solicitações (Concluídas).....	61
Figura 21 - Código DAX da Medida Média Lead Time Total (Emitidos).....	61
Figura 22 - Código DAX da Medida Concluídos < 3 dias.....	62
Figura 23 - Código DAX da Medida % Concluídos < 3 dias.....	62
Figura 24 - Tela "Análise Solicitações" Análise de Solicitações Criadas e Arquivamento.....	63
Figura 25 - Tela "Análise Concluídas" Performance de Saída e Certificadoras.....	64
Figura 26 - Tela "Indicadores" Foco em Performance e SLA.....	65
Figura 27 - Tela "Tempo Médio Por Etapa" Análise de Gargalos.....	66
Figura 28 - Tela "Tempo Médio por Regional" Análise de Desempenho por Equipe Comercial.....	67
Figura 29 - Tela "Certificado Digital" - Visão Operacional Detalhada Fila de Trabalho.....	68
Figura 30 - Código Python: Instalação e Importação das Bibliotecas.....	70
Figura 31 - Código Python: Para Carregar o Arquivo .xlsx no Ambiente.....	71
Figura 32 - Código Python: Definição da Variável-Alvo e Listas de Features.....	72
Figura 33 - Código Python: Filtragem, Separação (X/y) e Imputação de Nulos.....	74
Figura 34 - Código Python: Construção do Pipeline de Pré-processamento e Modelagem.....	76
Figura 35 - Representação Textual da Arquitetura do Pipeline.....	76
Figura 36 - Código Python: Divisão e Treinamento do Modelo.....	77
Figura 37 - Código Python: Avaliação do Modelo e Saída do Código.....	78
Figura 38 - Matriz de Confusão dos Resultados de Teste.....	79
Figura 39 - Curva ROC e Métrica AUC.....	80
Figura 40 - Curva Precision-Recall e Métrica AP.....	81
Figura 41 - Análise de Métricas vs. Ponto de Corte.....	82

Figura 42 - Código Python: Salvando o Modelo Treinado para Implantação.....	83
Figura 43 - 1º Parte do Script Python executado no Power Query para ML.....	84
Figura 44 - 2º Parte do Script Python executado no Power Query para ML.....	85
Figura 45 - Dashboard "ML - Gestão Proativa de Risco SLA"	86
Figura 46 - Código DAX: Coluna de Correção da Probabilidade.....	88
Figura 47 - Código DAX: Coluna de Status de Risco (Lógica Principal).....	88
Figura 48 - Código DAX: Medidas de Agregação do Risco.....	89

LISTA DE TABELAS

Tabela 1 - Etapas de um processo decisório.....	21
Tabela 2 - Categorias e Tipos de Fontes de Dados Suportadas pelo Power BI.....	28
Tabela 3 - Tipos de Tarefas de ML na Scikit-Learn.....	40
Tabela 4 - Mapeamento de Gargalos do Processo de Certificação.....	47
Tabela 5 - Dicionário de Dados dos Atributos Relevantes.....	51
Tabela 6 - Justificativa da Escolha das Features do Modelo de ML.....	74

LISTA DE ABREVIATURAS

ABC:	(Linguagem de programação)
API:	Application Programming Interface
AP:	Average Precision
AUC:	Area Under the Curve
AutoML:	Automated Machine Learning
BD:	Banco de Dados
BI:	Business Intelligence
CNH:	Carteira Nacional de Habilitação
CPU:	Central Processing Unit
CRM:	Customer Relationship Management
CSV:	Comma Separated Values
DAX:	Data Analysis Expressions
ERPs:	Enterprise Resource Planning
ETL:	Extract Transform Load
GHz:	GigaHertz
JSON:	JavaScript Object Notation
KPI:	Key Performance Indicator
LGPD:	Lei Geral de Proteção de Dados
LightGBM:	Light Gradient Boosting Machine
Matplotlib:	(Biblioteca de visualização de dados em Python)
ML:	Machine Learning
MLOps:	Machine Learning Operations
NumPy:	(Biblioteca numérica para Python)
ODBC:	Open Database Connectivity
OData:	Open Data Protocol
Pandas:	(Biblioteca de análise de dados para Python)
PDF:	Portable Document Format
PK:	Chave Primária
R:	(Linguagem de programação estatística)
RAM:	Random Access Memory
REST:	Representational State Transfer
ROC:	Receiver Operating Characteristic

SciPy:	(Biblioteca científica para Python)
SLA:	Acordo de Nível de Serviço
SQL:	Structured Query Language
vs.:	Versus
VS	Code: Visual Studio Code
x64:	(Arquitetura de processador)
XML:	Extensible Markup Language
.pbix:	(Extensão de arquivo do Power BI Desktop)
.pkl:	Pickle (extensão para serialização de objetos Python)
.xlsx:	(Extensão de arquivo do Microsoft Excel)

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	13
1.1.2 OBJETIVOS ESPECÍFICOS.....	14
1.2 JUSTIFICATIVA.....	15
2 FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 TOMADA DE DECISÃO.....	18
2.2 BUSINESS INTELLIGENCE.....	21
2.3 POWER BI.....	23
2.4 QUALIDADE DOS DADOS.....	28
2.5 FONTE DE DADOS.....	29
2.5.1 SNOWFLAKE.....	30
2.5.2 PIPEFY.....	30
2.6 ETL (EXTRACT TRANSFORM LOAD).....	31
2.7 AZURE MACHINE LEARNING.....	32
2.7.1 AZURE MACHINE LEARNING STUDIO.....	34
2.8 PYTHON.....	35
2.8.1 SCIKIT-LEARN.....	36
2.8.2 LIGHTGBM.....	37
3 DESENVOLVIMENTO.....	39
3.1 AMBIENTE DE DESENVOLVIMENTO E FERRAMENTAS UTILIZADAS.....	39
3.1.1 HARDWARE.....	39
3.1.2 SOFTWARE.....	39
3.1.3 BIBLIOTECAS E TECNOLOGIAS DE APOIO.....	40
3.1.4 INTEGRAÇÃO ENTRE FERRAMENTAS.....	41
3.1.5 MÉTODO DE VERSIONAMENTO E ORGANIZAÇÃO DE ARQUIVOS.....	42
3.2 ANÁLISE E MAPEAMENTO DO PROCESSO.....	42
3.2.1 DESCRIÇÃO DO PROCESSO.....	42
3.2.2 IDENTIFICAÇÃO DE GARGALOS E DEFINIÇÃO DA MÉTRICA PRINCIPAL.....	44
3.2.3 LEVANTAMENTO E ESTRUTURA DAS FONTES DE DADOS.....	46
3.3 CONSTRUÇÃO DA SOLUÇÃO DE BUSINESS INTELLIGENCE (POWER BI).....	49
3.3.1 PROCESSO DE ETL (EXTRAÇÃO, TRANSFORMAÇÃO E CARGA).....	49
3.3.2 MODELAGEM RELACIONAL DE DADOS.....	55
3.3.3 CRIAÇÃO DE COLUNAS CALCULADAS E MEDIDAS (DAX).....	57
3.3.4 DESENVOLVIMENTO DOS DASHBOARDS E VISUALIZAÇÕES.....	61
3.4 IMPLEMENTAÇÃO DO MODELO PREDITIVO DE MACHINE LEARNING.....	67
3.4.1 AMBIENTE, METODOLOGIA E PREPARAÇÃO DOS DADOS.....	67
3.4.2 DEFINIÇÃO DO PIPELINE DE MODELAGEM.....	73
3.4.3 TREINAMENTO E AVALIAÇÃO DOS RESULTADOS.....	75
3.4.4 ANÁLISE AVANÇADA DE PERFORMANCE DO MODELO.....	78
3.4.5 IMPLANTAÇÃO E RETROALIMENTAÇÃO.....	81
3.4.6 DASHBOARD DE GESTÃO PROATIVA DE RISCO (ML).....	84
3.4.7 DETALHAMENTO DAS MEDIDAS E DAS COLUNAS CALCULADAS DA ML.....	85
4 CONSIDERAÇÕES FINAIS.....	88

REFERÊNCIAS.....91

1 INTRODUÇÃO

A transformação digital tem impulsionado mudanças significativas na forma em que as empresas trabalham, essencialmente em setores que dependem fortemente da análise de dados e da tomada de decisões rápidas e assertivas, em um cenário altamente competitivo e em constante evolução. Para tomar decisões informadas e eficazes, os colaboradores precisam de acesso a informações relevantes e atualizadas sobre o desempenho organizacional. É aqui que entram as ferramentas de análises de dados e visualizações (*Business Intelligence*). Essas ferramentas são utilizadas para análises de dados de variadas fontes para auxiliar na tomada de decisões mais assertivas.

Nesse cenário, a engenharia dos dados emerge como um pilar para capacitar colaboradores e empresas para enfrentarem esses desafios de maneira eficaz. A análise para a construção dos fluxos resultará na entrega de atividades dentro do prazo definido. A métrica de *lead time* sendo o tempo total necessário para completar um processo, desde o início até a sua conclusão, onde no contexto operacional representa o intervalo entre o momento em que uma tarefa é iniciada até ela ser finalizada, prova-se eficaz. A vantagem do *Lead Time* é prover informações essenciais para o gerenciamento de tarefas. Desta forma é possível identificar gargalos e possíveis erros no fluxo de trabalho, entender as dificuldades e a capacidade da equipe e melhorar as expectativas das entregas gerando prazos mais concisos. (Amorim, 2023)

Quando o tema é sobre informações baseadas em dados, algumas adversidades aparecem no caminho. Independentemente de onde os dados são extraídos, as preocupações são quais tecnologias serão utilizadas para armazenar e visualizar os resultados das análises dos dados. Outrossim, cada equipe ou cada organização contém um tipo específico de problema. A métrica de *lead time* possibilita a gestão das entregas empresariais. Portanto, cada empresa que coleta e trata seus próprios dados, irá deter o poder de afeiçoá-los da maneira que achar melhor. (Amorim, 2023)

Portanto, cada ação tomada deve ser embasada nos indicadores implementados na empresa. Isso aumenta a probabilidade de tomar decisões acertadas e permite que a empresa avalie seu desempenho no mercado, bem como determinar se suas atividades estão gerando retornos positivos. (Da Silva Leão, 2023)

No setor de crédito imobiliário, onde operações como o certificado digital de clientes e a prevenção de fraudes exigem precisão, agilidade e segurança no processo, o uso e a manipulação inteligente dos dados pode representar um diferencial individualizador. Equipes

que atuam nessas áreas lidam com uma alta demanda diária, processos sensíveis e prazos rigorosos, para a maior satisfação dos clientes, o que torna a gestão do tempo um desafio constante.

É nesse contexto que tecnologias como *Power BI* e *Machine Learning* podem ser integradas para fornecer uma solução completa de apoio à gestão de tempo e desempenho da performance operacional da equipe. Cruzamento entre os dados operacionais e algoritmos de aprendizado de máquina tem o potencial de oferecer *insights* valiosos, como a previsão de atrasos, identificação de padrões de ineficiência e geração de alertas automáticos para tomada de decisão ágil e fundamentada, visando a redução do tempo de atuação da equipe em cada processo.

Dentro deste contexto o presente trabalho propõe fazer uma contribuição na área de gestão operacional e análise de dados, por meio do desenvolvimento e avaliação de uma solução baseada em sistemas de *Business Intelligence* e *Machine Learning* para a melhoria da performance operacional de uma equipe responsável pelo processo de certificação digital dos clientes em uma empresa do setor de crédito imobiliário.

1.1 OBJETIVOS

Desenvolver e avaliar uma solução utilizando *Business Intelligence* e técnicas de *Machine Learning* com o objetivo de otimizar o tempo de atuação das equipes responsáveis pelos processos de certificação digital dos clientes, este trabalho visa verificar, através da análise de dados, quais indicadores podem ser utilizados para identificar as equipes comerciais que necessitam de mais atenção e treinamento. Assim, busca-se promover uma gestão mais eficiente e rápida, totalmente orientada por informações geradas a partir dos dados, dentro de uma empresa do setor de crédito imobiliário.

1.1.2 OBJETIVOS ESPECÍFICOS

1. Analisar os processos operacionais da equipe mapeando fluxos de trabalho relacionados à certificação digital e antifraude, identificando os gargalos e etapas que impactam o tempo de atuação da equipe e levantamento de dados relevantes para análise.
2. Aplicar a métrica de *Lead Time* calculando o tempo médio de execução das atividades e acompanhamento da evolução da performance ao longo do tempo.

3. Desenvolver *dashboards* interativos no *Power BI* integrando diferentes fontes de dados operacionais visualizando em tempo real do desempenho da equipe.
4. Implementar modelos de *Machine Learning* através do treinamento de modelos preditivos com dados históricos para retornar previsão de atrasos nos processos e possível identificação de padrões de ineficiência.
5. Criar uma base de dados unificada e confiável dentro do BI realizando a estruturação de dados brutos em formato utilizável para análise, respeitando e analisando possíveis relação entre as bases de dados e, com isso, garantir a qualidade e consistência das informações.
6. Estabelecer indicadores operacionais e estratégicos de eficiência da equipe e indicadores para acompanhamento gerencial.
7. Avaliar a eficácia da solução implantada comparando métricas antes e depois da aplicação da ferramenta e coletando *feedbacks* da equipe usuária, validando os resultados preditivos com a realidade operacional.
8. Elaborar recomendações para melhoria contínua, com base nos dados obtidos e sugestões de ajustes nos processos e práticas de trabalho.
9. Explorar a escalabilidade da solução e as possibilidades de aplicação da ferramenta em outros setores da empresa.

1.2 JUSTIFICATIVA

Atualmente, no mundo dos negócios, onde a concorrência é acirrada e os dados são cruciais, a eficiência das operações deixou de ser um mero bônus para se tornar essencial para a sobrevivência e o desenvolvimento das empresas. No ramo de crédito imobiliário, onde a validação digital e a proteção contra fraudes demandam rapidez, exatidão e segurança, o tempo que as equipes levam para responder pode afetar significativamente a satisfação do cliente e os resultados da companhia.

A revolução digital tem transformado profundamente a forma como as empresas decidem, tornando indispensável o uso de ferramentas tecnológicas para aproveitar o valor estratégico dos dados. Nesse cenário, a combinação de *Business Intelligence* e *Machine Learning* se apresenta como uma opção eficiente para otimizar o gerenciamento do tempo e a melhoria da performance das equipes. O uso de indicadores como o *Lead Time* facilita a identificação de gargalos nos processos e o aprimoramento do planejamento de tarefas (Amorim, 2023, apud Matos, 2022).

A motivação deste estudo reside na necessidade concreta de aprimorar a gestão operacional de equipes que lidam com alta demanda de trabalho e prazos restritos, como é o caso da área de riscos e a área comercial da empresa. Através da coleta, análise e visualização de dados, busca-se oferecer uma solução prática que possibilite identificar padrões de ineficiência e prever atrasos, visando uma tomada de decisão mais ágil e embasada com o intuito de gerenciar o tempo de forma mais eficiente e elevar a performance da equipe.

Adicionalmente, esta pesquisa enriquece ao demonstrar, de forma aplicada, como a integração entre engenharia de dados, métricas de desempenho e inteligência artificial pode impulsionar o aumento da produtividade e a transparência nos processos. A facilitação do acesso à informação por meio de painéis interativos aprimora a comunicação entre equipe e gestor, promovendo decisões baseadas em informações geradas por dados.

Com isso, espera-se não só diminuir o tempo de atuação da equipe e impulsionar o desempenho dela nos processos analisados, mas também proporcionar treinamentos para equipes da área comercial que têm mais dificuldades nestes processos e também ter um modelo adaptável que possa ser utilizado em outros setores com desafios similares. A solução proposta será desenvolvida com base em dados reais, respeitando a lei geral de proteção de dados (LGPD) a estrutura operacional da empresa e seus protocolos internos, o que aumenta a relevância prática do estudo e sua aplicabilidade.

Finalmente, este trabalho visa unir teoria e prática, contribuindo tanto para a área acadêmica quanto para a resolução de um problema concreto enfrentado no dia a dia da organização. A escolha do tema se justifica, portanto, por sua relevância, atualidade e potencial de impacto positivo na performance da equipe e, evidentemente, nos processos internos da empresa. A crescente quantidade de dados disponíveis atualmente exige ferramentas capazes de processá-los de forma eficiente e significativa. Sistemas inteligentes informatizados, têm se destacado nesse cenário por sua habilidade em analisar grandes volumes de dados e sintetizá-las de maneira clara e acessível e o transformando em informação. Essa capacidade não apenas facilita a compreensão de conteúdos complexos, mas também apoia e otimiza o trabalho humano, permitindo uma tomada de decisão mais embasada e ágil.

2 FUNDAMENTAÇÃO TEÓRICA

A crescente digitalização das empresas corporativas e a grande ampliação dos volumes de dados disponíveis deram origem à sociedade da informação. Neste sentido, verifica-se que os dados passaram a exercer papel central nos processos de gestão, sendo considerados um ativo estratégico para a competitividade no mercado e a inovação organizacional.

Segundo (Castells; Cardoso, 2005, P.423), “Sem inovação organizacional, a inovação tecnológica não chegará a constituir-se como fator de desenvolvimento e fonte efetiva de competitividade” e com essa inovação organizacional que é a transformação nos métodos de negócio da empresa por meio de ideias que podem ser retiradas de informação geradas por dados vem cada vez mais aprimorando a inovação tecnológica.

Com o crescimento do volume de dados gerados na empresa, tornou-se essencial o uso de métodos que permitam transformar esses dados em informações relevantes e, posteriormente, em conhecimento útil para a tomada de decisão e elaboração de ideias. Essa transformação é um processo estruturado de extração, transformação, análise e visualização dos dados, podendo ser potencializada pelo uso de técnicas de *Machine Learning*, que permitem identificar padrões e gerar previsões automatizadas com base em dados históricos.

A tomada de decisão baseada em dados permite maior agilidade, precisão e embasamento nas ações empresariais, favorecendo a adaptação a contextos dinâmicos e competitivos. (Faria, 2022) Ferramentas de *Business Intelligence* por exemplo, possibilitam essa análise integrada transformando dados operacionais em informações estratégicas.



Figura 1 - Elementos Intervenientes No Processo De Tomada De Decisão

Fonte: (Angeloni, 2003)

A figura 1 apresenta os elementos que podem intervir no processo de análise e de tomada de decisão, onde ilustra que o dado gerando a informação junto com o conhecimento, decisão e comunicação devem ser considerados pontos interligados que agregam valor, sendo requisitos chaves da tomada de decisão. (Faria, 2022, apud Angeloni, 2003)

2.1 TOMADA DE DECISÃO

A preocupação com o futuro é uma constante na vida humana. Desde cedo, as pessoas se veem diante de escolhas importantes: qual carreira seguir, o que estudar, se devem casar ou não, entre tantas outras decisões que moldam seu futuro. Ao longo da história, diferentes sociedades buscaram formas de orientar essas decisões, muitas vezes recorrendo à observação cuidadosa do mundo ao seu redor.

Segundo (Freitas, 2021, apud Osinski et al., 2016) a definição de um processo decisório ocorre em 6 etapas:

Etapa	Definição
1	Análise e identificação da situação: identificar claramente a situação que o ambiente do problema se encontra, por meio do levantamento de informações, com o objetivo de chegar a uma decisão sólida;
2	Desenvolver alternativas: com base nos levantamentos realizados na coleta de dados, pode-se elencar alternativas para a resolução dos problemas;
3	Comparação entre alternativas: identificar vantagens e desvantagens de cada opção;
4	Classificação de riscos: avaliar os riscos envolvidos em cada uma das alternativas, de modo a antecipar as principais consequências de cada decisão, buscando apontar a que representa menor risco;
5	Escolha da alternativa: selecionar aquela que resolve o problema, baseado nas vantagens, desvantagens e riscos analisados;
6	Execução e avaliação: comparação dos resultados obtidos com a alternativa escolhida frente às previsões anteriores.

Tabela 1 - Etapas de um processo decisório.

Fonte: (Freitas, 2021, apud Osinski et al., 2016)

Povos indígenas, por exemplo, ao buscar e orientar suas decisões interpretam sinais da natureza como o sol, a lua e os ventos, tendo eles como guias para suas escolhas. Outras civilizações recorrem ao mar, às mudanças climáticas ou aos conselhos dos mais velhos. Em comum, todas buscam referências externas que sirvam como parâmetros confiáveis para agir com mais segurança. Esses sinais e eventos são, na essência, fontes de informação. E é justamente essa busca por informação que permanece como base fundamental do processo decisório até os dias de hoje.

Segundo Rayanne (Araujo, 2020, apud Castro, 1995) a palavra decisão é definida como “Uma escolha entre soluções possíveis, que se assenta num processo subjacente, elaborado a partir do conhecimento e apreensão da informação necessária, da aplicação de regras de decisão, da reflexão e confirmação da escolha visada”, onde o ato de tomar decisões é crucial para as organizações. Esta atividade acontece a todo tempo, em todos os níveis, e influencia diretamente o desempenho da organização (Araujo, 2020, apud Kladis e De Freitas, 1995).

A tomada de decisão é sempre necessária, quando se está diante de um problema que possui mais de uma alternativa que pode ser eleita a melhor, ainda que este problema possua um ato de solução, têm-se as alternativas de se tomá-lo ou não. A decisão faz parte da gestão e muitas vezes seu estudo é negligenciado, inclusive por aqueles responsáveis por tomar decisões importantes nas organizações como os executivos e os negociadores. (Pereira, 2019)

Para uma organização em busca de entrega de resultados, existem três níveis diferentes de tomada de decisão que vão impulsionar todas as equipes da organização para a concretização dos seus objetivos e de suas metas.



Figura 2 - Níveis Da Tomada De Decisão Em Uma Organização.

Fonte: (Préve, 2010)

A figura 2 representa os três níveis de tomada de decisões dentro de uma organização onde Altamiro (Préve, 2010) define elas em:

- Decisões estratégicas são aquelas que determinam os objetivos da organização, seus propósitos e sua direção, sendo uma função exclusiva da alta administração.
- Decisões táticas (ou administrativas) são as tomadas em um nível abaixo das decisões estratégicas. Normalmente são tomadas pela gerência intermediária, como gerentes de divisão ou de departamentos.
- Decisões operacionais são as tomadas no nível mais baixo da estrutura organizacional, no campo da supervisão ou operacional de uma organização, e se referem ao curso de operações diárias.

Portanto, na tomada de decisão, a qualidade da decisão é fator fundamental, porque afeta o futuro da organização. A racionalidade do gestor é necessária nas tomadas de decisão, mas deverá estar em consonância com as informações advindas do diagnóstico do problema,

bem como na análise das informações geradas por dados e avaliação das alternativas, no sentido de evitar problemas maiores e não ter a melhora na entrega de resultados.

2.2 BUSINESS INTELLIGENCE

Em um cenário corporativo onde cada vez mais está se tornando orientado por dados o *Business Intelligence* (BI), que em português trás o significado “Inteligência de Negócios”, surgiu na década de 1980 e foi registrado pelo Gartner Group, como uma metodologia estratégica para agilizar e dar segurança ao processo de tomada de decisão, onde passou ao longo dos anos por evolução e melhorias, sendo seu principal objetivo ser um método que coleta, organiza, analisa e monitora dados. (Dos Santos, 2022)

De acordo com Ronaldo (Braghittoni, 2017),

Business Intelligence (ou BI) é um termo cunhado por Howard Dresner do Gartner Group, em 1989, para descrever um conjunto de conceitos e métodos para melhorar o processo de tomada de decisão das empresas, utilizando-se de sistemas fundamentados em fatos e dimensões.

O *Business Intelligence* é uma metodologia e não uma ferramenta. Isso implica que sua implementação pode ser realizada por meio de diversas soluções tecnológicas disponíveis para controle e análise de dados. Assim, desde que se domine a metodologia e os princípios do BI, é possível aplicar seus conceitos utilizando diferentes ferramentas de controle de dados.

Com isso, resumidamente determina-se que o termo *Business Intelligence*, é um conjunto de conceitos e metodologias que, fazendo uso de dados extraídos de uma organização, apoiam a tomada de decisões. Reforça-se que o objetivo do *Business Intelligence* é transformar dados em conhecimento, que suporta o processo decisório com o objetivo de gerar vantagens competitivas. (Cruz, 2014)

Desavisado	Tático	Focado	Estratégico	Universal
Anarquia de informação: Dados espalhados pela organização em diversos sistemas e em diversos arquivos (excel, word, txt etc.)	Distribuição de informação: Sistemas espalhados sem centralização das regras e das informações. Diversas formas de entender uma mesma informação.	Foco em algumas informações: Informações centralizadas para determinados focos. Áreas com bastante maturidade de consumo de informações e outras nem tanto.	Demandas estratégicas de informação: A obtenção e análise de informações é direcionada por estratégias da empresa, com Governança, "frameworks", e padrões.	Informação para todos: Toda a informação gerada na empresa é ubíqua e permeia todos os níveis, inclusive parceiros, clientes e fornecedores.

Figura 3 - Níveis de Maturidade no Uso da Informação nas Organizações

Fonte: (Braghittoni, 2017)

A Figura 3 mostra como as empresas podem evoluir no uso da informação, desde um cenário desorganizado até um estágio em que os dados realmente fazem parte da rotina e das decisões de todos. No começo, as informações estão espalhadas por vários arquivos e sistemas, sem muita integração ou controle. Aos poucos, com mais organização e foco, a empresa começa a centralizar os dados e usá-los de forma mais inteligente.

Se uma empresa não possui uma solução onde usa a metodologia de *Business Intelligence*, é bem provável que ela se encontre entre os níveis “Desavisado” ou “Tático”, ou seja os dados estão espalhados em sistemas e planilhas descentralizadas. Com isso torna-se comum dois departamentos chegarem a números diferentes para o que deveria ser a mesma informação. Quando é implementado de alguma forma o método de *Business Intelligence* por uma equipe em uma organização, passa-se para o nível “Focado”, ou seja a informação e o conhecimento passam a ter a mesma relevância no dia a dia. Segundo (Braghittoni, 2017), no entanto somente a equipe que a implementou terá as informações centralizadas para determinados focos, sendo que as outras áreas seguem sem a maturidade de consumo de informações.

Com o tempo, essa maturidade cresce e a informação passa a ser tratada como algo estratégico, com regras claras, boas práticas e apoio das lideranças. No estágio mais avançado, chamado de “Universal”, a informação é acessível, confiável e compartilhada com todos que precisam dela, inclusive clientes, fornecedores e parceiros, e em todas as áreas da empresa. Esse caminho ajuda a transformar dados soltos em informações mais certeiras e alinhadas com os objetivos do negócio, apoiando na melhoria da tomada de decisão dentro de uma organização.

2.3 POWER BI

O Power BI é uma plataforma unificada e escalonável de *Business Intelligence* desenvolvida pela Microsoft e teve seu lançamento no ano de 2015, caracterizada pela combinação da facilidade de uso com recursos avançados de análise de dados. Voltado tanto para o autoatendimento quanto para o uso empresarial, o Power BI permite conectar-se a uma ampla variedade de fontes de dados, estruturadas ou não, facilitando a criação de relatórios dinâmicos e *dashboards* interativos. (Microsoft:..., 2025)

O Power BI é uma das plataformas mais completas e acessíveis quando o assunto é análise e visualização de dados no ambiente corporativo. Seu principal objetivo é ajudar pessoas e empresas a transformarem dados brutos em informações claras, úteis e fáceis de entender. Com uma interface intuitiva e funcionalidades poderosas, ele permite a criação de relatórios e *dashboards* interativos que facilitam a leitura dos dados e a tomada de decisões. O Power BI contribui para criar uma cultura mais analítica nas organizações, onde as decisões são guiadas por dados e não pelo que vemos nos dias atuais em algumas organizações onde as decisões são guiadas apenas pela intuição.

Para entender todo o potencial dessa ferramenta, é importante conhecer suas principais funcionalidades que, de acordo com Manuelle, (Ribeiro, 2024) se apresentam como:

- Visualização de Dados: Permite criar gráficos interativos e *dashboards* personalizados que facilitam a análise detalhada das informações. Esses painéis podem ser facilmente compartilhados com outras pessoas e adaptados às necessidades de cada usuário. Além disso, é possível utilizar visualizações personalizadas disponíveis no Microsoft AppSource ou até mesmo criar as suas próprias, oferecendo ainda mais flexibilidade na forma de apresentar os dados.
- Conexão com diversas fontes de dados: O Power BI se conecta a uma grande variedade de fontes, como bancos de dados SQL, Azure, serviços *online* (como Google Analytics e Salesforce), planilhas do Excel, entre muitos outros. Também permite configurar atualizações automáticas e programadas dos dados, garantindo que os relatórios e *dashboards* estejam sempre com as informações mais recentes.
- Modelagem de Dados: Com o uso do Power Query, o Power BI oferece ferramentas intuitivas para preparar e transformar os dados de forma prática. É possível criar modelos complexos com relacionamentos entre tabelas, além de

usar colunas calculadas e medidas personalizadas, facilitando análises mais profundas e detalhadas mesmo com grandes volumes de informação.

- Análises de Dados: O Power BI utiliza a linguagem *DAX (Data Analysis Expressions)*, que permite criar medidas e colunas calculadas para análises mais precisas. Além disso, é possível integrar *scripts* em R e Python, o que amplia as possibilidades de realizar análises estatísticas e aplicar técnicas de *machine learning* diretamente nos relatórios.
- Compartilhamento e Colaboração: A plataforma permite publicar relatórios *online* por meio do serviço Power BI, facilitando o acesso e o compartilhamento das informações com outros usuários. Também é possível criar aplicativos personalizados que funcionam em dispositivos móveis, além de definir permissões específicas para controlar quem pode visualizar ou interagir com os relatórios e os dados.
- Integração com outras ferramentas Microsoft: O Power BI se conecta facilmente com outras soluções da Microsoft, como o Excel permitindo importar e exportar dados com agilidade, o Teams e o SharePoint, facilitando a colaboração entre equipes e o compartilhamento de *insights* dentro do ambiente corporativo.
- Segurança e Governança: Oferece recursos avançados de segurança para proteger dados sensíveis, permitindo definir quem pode acessar cada informação. Além disso, conta com funcionalidades de auditoria, monitoramento e conformidade com normas e regulamentações, garantindo maior controle e confiança no uso da ferramenta.

É justamente por reunir essas funcionalidades que fazem dela uma grande aliada na rotina de quem precisa tomar decisões com mais agilidade e precisão. E naturalmente ser uma das ferramentas de *Business Intelligence* mais utilizadas no mundo.

Segundo a Microsoft (Microsoft:..., 2025) o Power BI é capaz de extrair e conectar-se a diversas categorias e fontes de dados, como mostra a tabela 2.

Categoria	Descrição Geral da Categoria
Arquivo	Conexão com arquivos como Excel, CSV, XML, JSON, PDF, entre outros.
Banco de Dados	Suporte a diversos bancos como SQL Server, Oracle, MySQL, PostgreSQL, SAP HANA, Amazon Redshift, entre outros.
Microsoft Fabric	Acesso a modelos semânticos, fluxos de dados, <i>datamarts</i> , <i>lakehouses</i> e métricas.
Power Platform	Integração com Dataverse, fluxos de dados e serviços herdados do Power BI.
Azure	Conexão com serviços da nuvem Azure, como SQL, Cosmos DB, Synapse, Data Lake, entre outros.
Serviços Online	Integração com plataformas como SharePoint Online, Dynamics 365, Google Analytics, Salesforce, GitHub, entre outros.
Outros	Suporte a fontes como Web, OData, Active Directory, scripts R/Python, ODBC, Google Sheets, APIs diversas e muito mais.

Tabela 2 - Categorias e Tipos de Fontes de Dados Suportadas pelo Power BI

Fonte: (Microsoft:..., 2025)

A grande variedade de fontes de dados que o Power BI torna a ferramenta muito versátil, permitindo integrar informações de diferentes sistemas e serviços em um só lugar, interligando dados de diferentes fontes, como se fossem tabelas que compartilham um mesmo campo em comum. Isso facilita combinar informações que estão espalhadas em sistemas e fontes de dados distintas, criando *dashboards* e relatórios integrados e completos sempre atualizados, ajudando equipes a tomar decisões rápidas e baseadas em dados reais.

O Power BI de acordo com Walter (Lucena, 2022) é composto por sete componentes principais:

- Power BI Desktop: Trata-se do aplicativo para computadores com sistema operacional *Windows*, considerado a base da plataforma. Nele são desenvolvidos inicialmente os painéis (*dashboards*), que posteriormente podem ser publicados no ambiente online do Power BI.
- Power BI Service: Representa a versão baseada em nuvem, onde os painéis publicados podem ser acessados, gerenciados e compartilhados por meio da internet.

- Aplicativos móveis do Power BI: São as versões do Power BI destinadas a dispositivos móveis, com aplicativos disponíveis para sistemas Android, iOS e também para tablets e smartphones com *Windows*.
- Power BI Gateway: Ferramenta responsável por estabelecer a conexão entre dados armazenados localmente em servidores internos e o serviço em nuvem do Power BI, garantindo a atualização automática das informações.
- Power BI Embedded: Utiliza a *API REST* para permitir que relatórios e dashboards criados no Power BI sejam incorporados em aplicações personalizadas, independentemente de o usuário final possuir ou não uma conta na plataforma.
- Power BI Report Server: É uma alternativa ao serviço em nuvem, permitindo que empresas publiquem e visualizem relatórios em servidores locais, ideal para organizações com restrições ao uso de recursos em nuvem.
- Power BI Visuals Marketplace: Um repositório de elementos visuais personalizados, incluindo visualizações baseadas em linguagem R, que podem ser integradas ao *Power BI Desktop* para ampliar as possibilidades de análise.

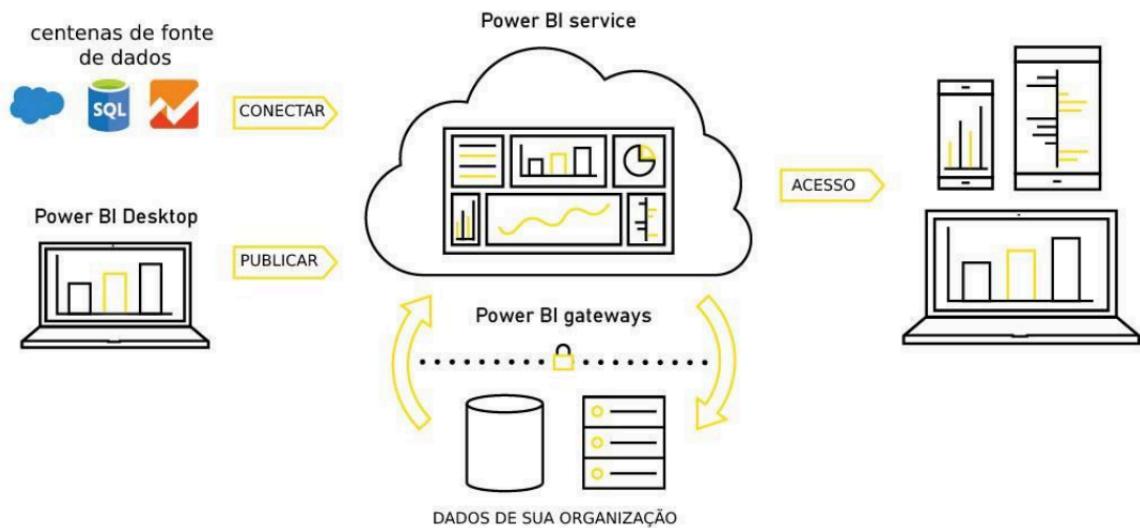


Figura 4 - Componentes Do Power Bi

Fonte: (Lucena, 2022)

Cada componente do Power BI exerce um papel essencial no processo de criação e publicação de dashboards. O fluxo tem início na conexão com as fontes de dados, que são tratadas e transformadas no Power BI Desktop. A partir daí, o Power BI Service assegura que os indicadores permaneçam atualizados, realizando sincronizações periódicas com os

servidores de origem por meio da configuração dos gateways. Uma vez publicado, o painel pode ser acessado por usuários autenticados ou por meio de links gerados com o uso do Power BI Embedded, recurso que facilita o compartilhamento online dos relatórios.

Então basicamente o Power BI é uma plataforma que transforma dados em informações visuais e interativas por meio de relatórios e painéis (*dashboards*), que permite conectar-se com várias fontes de dados, fazer o tratamento dos dados obtidos e por fim criar as visualizações das informações geradas pelos dados analisados, que ajudam na tomada de decisão. É muito utilizado nas empresas para monitorar indicadores, descobrir padrões, gerar insights estratégicos e transformá-los em forma visual.

Entendendo o Funcionamento

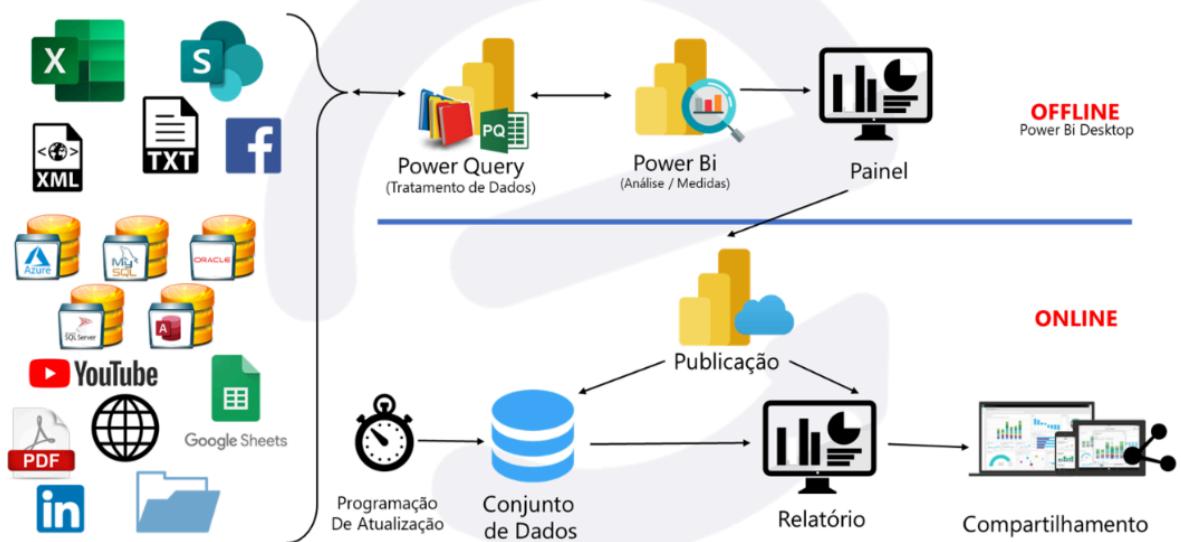


Figura 5 - Visão Geral do Funcionamento do Power BI

Fonte: (Da Silva Leão, 2023)

2.4 QUALIDADE DOS DADOS

A eficácia de qualquer projeto de *Business Intelligence* dentro de uma organização depende diretamente da qualidade dos dados utilizados. Tabelas incompletas, dados desatualizados ou inconsistentes podem comprometer a análise das informações e levar a tomadas de decisões equivocadas. Desta forma, garantir a integridade, a confiabilidade e a precisão dos dados é um passo essencial antes de qualquer interpretação. Podemos fazer tudo isso dentro do *Power BI* pois a ferramenta auxilia ativamente na limpeza, na transformação e

no tratamento dos dados, contribuindo de forma clara e objetiva ao gerar relatórios e informações.

Segundo pesquisa realizada por Julia (Rittmann, 2023), em qualidade dos dados existem seis dimensões amplamente reconhecidas no mercado:

- Completude: refere-se à necessidade de que os dados estejam totalmente preenchidos, sem lacunas, garantindo que os relatórios e análises reflitam a realidade com exatidão.
- Conformidade: diz respeito à obediência a formatos preestabelecidos, assegurando que os dados sejam compreensíveis e sigam padrões definidos.
- Consistência: implica na uniformidade das informações, evitando conflitos ou contradições entre registros que deveriam apresentar os mesmos valores.
- Precisão: indica o grau de exatidão dos dados, assegurando que os valores registrados sejam confiáveis e representem corretamente os fatos.
- Duplicação: diz respeito à identificação e eliminação de registros repetidos, que podem comprometer a análise ao inflar ou distorcer os resultados.
- Integridade: garante que todas as partes essenciais da informação estejam presentes e funcionais dentro de um registro, permitindo seu uso completo e correto.

Em resumo, a qualidade dos dados não é somente um requisito técnico, mas sim um fator determinante para o sucesso de qualquer análise ou projeto. Quando os dados têm qualidade e se enquadram nas seis dimensões acima, o processo de geração de *insights* se torna mais eficaz, permitindo tomadas de decisões mais acertadas. Por outro lado, dados com falhas e sem integridade podem gerar *insights* e tomadas de decisões equivocadas e impactar negativamente os resultados.

Por fim, antes de qualquer visualização ou criação de relatórios, é fundamental reservar um tempo e atenção para a preparação e validação dos dados.

2.5 FONTE DE DADOS

A base de qualquer projeto de análise de dados é definida pelas fontes de onde os dados são retirados. Antes mesmo de pensar na visualização dos indicadores ou na criação dos *dashboards*, é necessário identificar onde estão armazenados os dados que alimentam os sistemas.

As fontes podem variar bastante conforme o sistema utilizado, o contexto, a capacidade de pagamento e a necessidade de cada projeto. Entre as mais comuns estão arquivos simples como planilhas eletrônicas (Excel, CSV), bancos de dados relacionais (MySQL, SQL Server e Oracle), sistemas empresariais (ERPs, CRMs) e arquivos estruturados (JSON, XML). Além disso, plataformas de dados baseadas em nuvem (*Snowflake*) estão se tornando cada vez mais presentes no mercado de soluções modernas de análise.

CRM (*Customer Relationship Management*), ou Gestão de Relacionamento com o Cliente, é uma estratégia de negócios e um sistema de software focado na gestão de todas as interações da empresa com seus clientes atuais e potenciais. Ele armazena dados cruciais sobre clientes, histórico de interações, status de vendas e propostas. No contexto deste trabalho, sistemas de CRM são fontes de dados vitais para o *Business Intelligence*, pois fornecem a matéria-prima para a análise de performance comercial e de atendimento, exatamente como os dados da "Proposta CRM" e "Registro Contrato" que foram extraídos do Snowflake.

ERP (*Enterprise Resource Planning*), ou Planejamento dos Recursos da Empresa, refere-se a sistemas de *software* que integram e gerenciam os processos de negócios centrais de uma organização, como finanças, contabilidade, recursos humanos, estoque e operações. Para o *Business Intelligence*, o ERP é frequentemente a "fonte única da verdade" para dados financeiros e operacionais. Ele permite que as análises de BI cruzem informações de diferentes departamentos, garantindo uma visão holística da performance empresarial que vai além de um único processo.

Além da identificação da fonte, é fundamental garantir que a coleta de dados ocorra de forma segura e eficiente, respeitando padrões de qualidade e integridade. Só assim será possível obter resultados confiáveis nas etapas seguintes da análise.

2.5.1 SNOWFLAKE

O SnowFlake é uma plataforma de banco de dados compartilhado, que permite o processamento, armazenamento e a análise de grande volumes de dados de forma escalável e eficiente, sendo um exemplo seguro de compartilhamento de dados (Snowflake:..., 2025). Diferencia-se das arquiteturas de *Data Warehouse* tradicionais por ter uma abordagem moderna baseada em uma arquitetura *Multi-cluster*, que separa logicamente o armazenamento de dados do processamento operacional, permitindo que diferentes cargas de trabalho sejam

executadas de forma independente, otimizando a utilização de recursos e reduzindo custos.

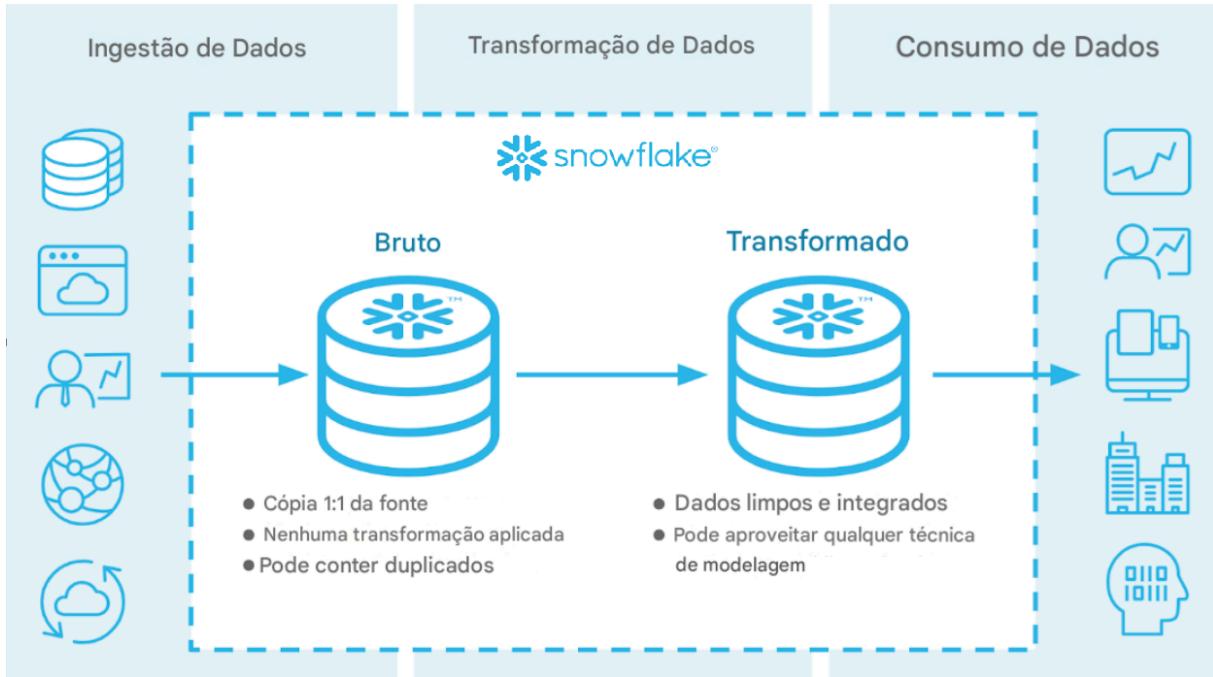


Figura 5 - Resumo Funcionamento Snowflake

Fonte: Adaptado de (SNOWFLAKE:..., 2025)

A Figura 5 ilustra o fluxo de dados na plataforma Snowflake, que ocorre em três etapas: ingestão, transformação e consumo. Inicialmente, os dados são carregados de plataformas operacionais, após passam por um processo de transformação, no qual são limpos e integrados para suportar diferentes modelos analíticos e, por fim, esses dados preparados são consumidos por ferramentas de BI, relatórios e aplicações de *Machine Learning*.

2.5.2 PIPEFY

O Pipefy é uma plataforma de gestão e automação de processos que permite às organizações modelar e controlar seus fluxos de trabalho de forma visual e personalizável. Classificada como uma ferramenta *low-code* ou *no-code*, ela possibilita que usuários sem conhecimento técnico aprofundado criem *pipelines*, definam regras de automação e integrem diferentes sistemas corporativos.(Pipefy:..., 2025).

A proposta central do Pipefy é transformar processos manuais e desestruturados em fluxos digitais padronizados, aumentando a eficiência operacional e reduzindo erros. A plataforma é especialmente útil em áreas como gestão de solicitações, controle de *compliance*, atendimento ao cliente e em operações de *back-office*.

A figura 6 demonstra o fluxo que já está sendo utilizado pela equipe de certificado digital, permitindo uma visualização clara e detalhada de uma parte do processo operacional que é conduzido no dia a dia. A partir desta imagem é possível verificar visualmente como o Pipefy pode ser utilizado na resolução de solicitações.

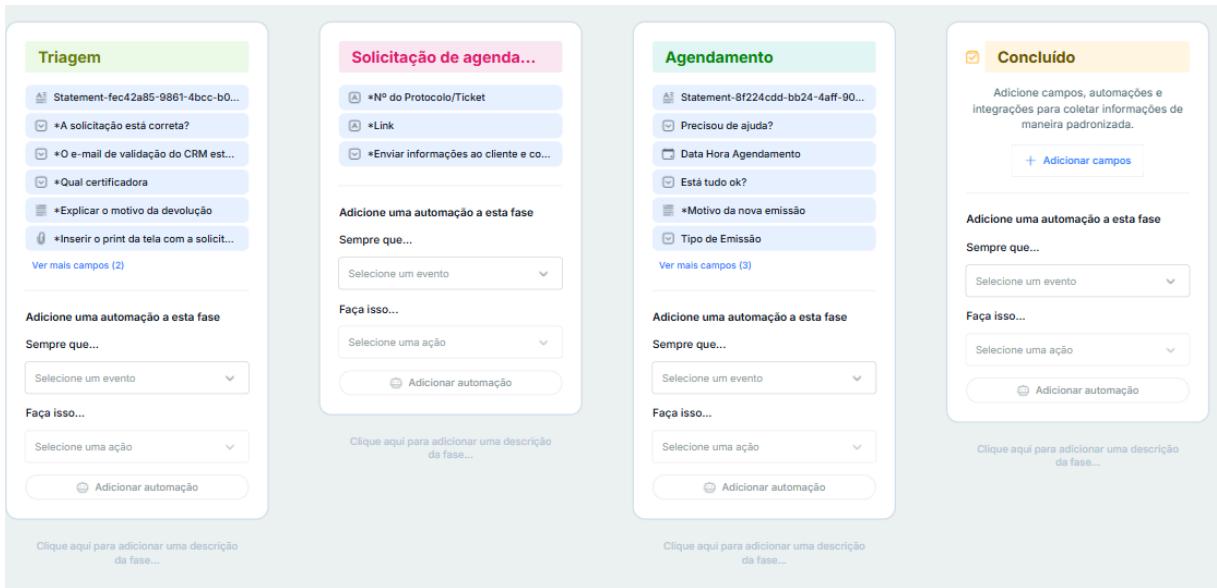


Figura 6 - Fluxo De Fases Dentro Do Sistema Pipefy

Fonte: Elaborado pelo autor deste trabalho.

2.6 ETL (EXTRACT TRANSFORM LOAD)

O ETL (*Extract Transform Load*) é um processo que consiste em extrair dados de um sistema de banco de dados (BD), realizando o seu processamento e modificação para que possam ser posteriormente inseridos em outro banco de dados, geralmente *Data Warehouse*, como mostrado na figura 7. Isso garante que os dados estejam limpos, consistentes e prontos

para gerar *insights* assertivos.

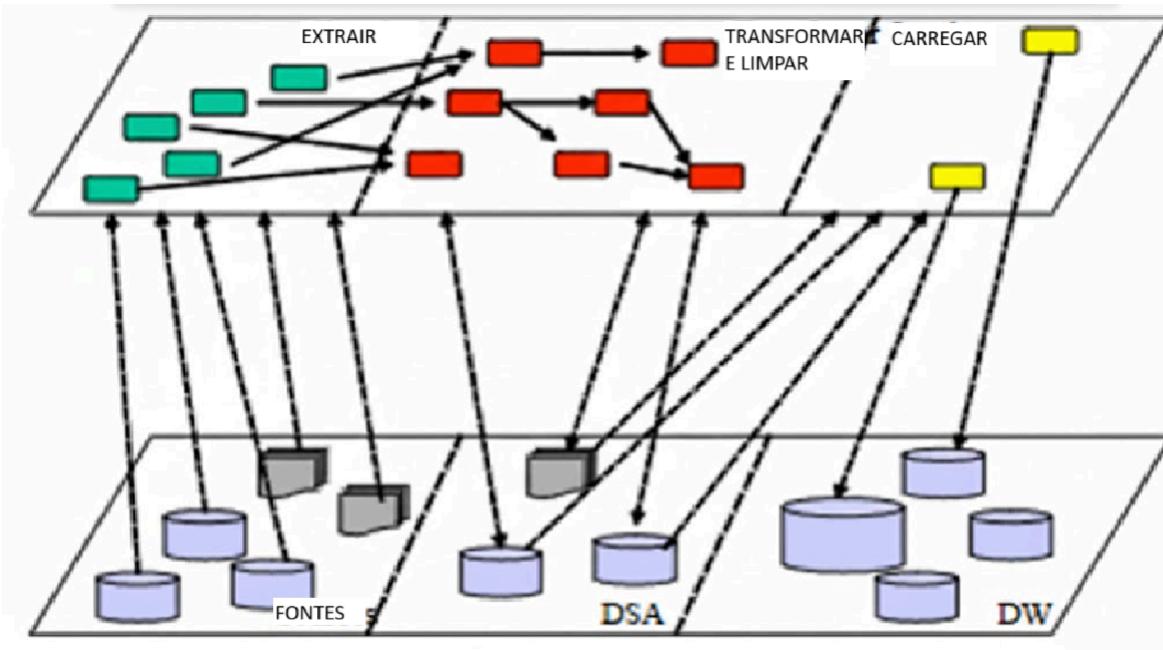


Figura 7- Visão Geral De Como Funciona O Processo De Etl

Fonte: Adaptado de (Ferreira, 2010)

O processo de ETL pode ser realizado diretamente no Power BI, utilizando a ferramenta integrada *Power Query*. Com ela é possível executar as etapas de extração, transformação e carga de dados de maneira prática e acessível, sem a necessidade de ferramentas externas. (Neto, 2021)

A aplicação de um processo de ETL eficiente é fundamental. Ele não só permite a centralização e o enriquecimento dos dados, mas também garante a qualidade, a governança e a rastreabilidade das informações utilizadas em todas as análises. Com isso, as decisões tomadas a partir desses dados se tornam mais confiáveis e estratégicas para a organização.

2.7 AZURE MACHINE LEARNING

Azure é a plataforma de serviços de computação em nuvem da Microsoft. A Azure Machine Learning é um serviço em nuvem para acelerar e gerenciar o fluxo de projetos de *Machine Learning*. Pode ser usada em trabalho cotidiano para o treinamento, implantação de modelos e o gerenciamento de MLOps (operações de aprendizado de máquina). (Microsoft:..., 2025)

Uma das vantagens primordiais da Azure Machine Learning reside na facilidade com que o desenvolvimento de modelos se encaixa em um padrão de trabalho cíclico e bem definido para a construção de soluções de análise preditiva. Essa característica promove a acessibilidade, tanto para aqueles que estão começando quanto para profissionais experientes na área de ciência de dados. (Carvalho, 2018)

Conforme explicitado pela Microsoft (Microsoft:..., 2025) em sua página oficial, a plataforma possui como vantagens principais:

- Colaboração facilitada: Ambientes compartilhados para *notebooks*, computação, dados e desenvolvimento unificado.
- Desenvolvimento responsável: Modelos para equidade e explicabilidade, acompanhados de rastreamento e auditabilidade para garantir a conformidade regulatória.
- Implantação e gestão eficaz: Implementação rápida e fácil de modelos de ML em escala com MLOps para gerenciamento e governança eficientes.
- Execução flexível e segura: Cargas de trabalho de *Machine Learning* em qualquer lugar com governança, segurança e conformidade internas.

Com essas vantagens, a plataforma elimina a necessidade de instalação e execução de aplicações locais nos dispositivos dos usuários. Além disso, libera os usuários das responsabilidades de manutenção e suporte do ambiente computacional subjacente. O acesso à plataforma e a sua utilização ocorrem por meio de um navegador web, que estabelece conexão com o ambiente de desenvolvimento. Dentro deste ambiente, torna-se viável a criação de modelos de análise e predição, customizados para atender às demandas específicas de cada organização. (Carvalho, 2018)

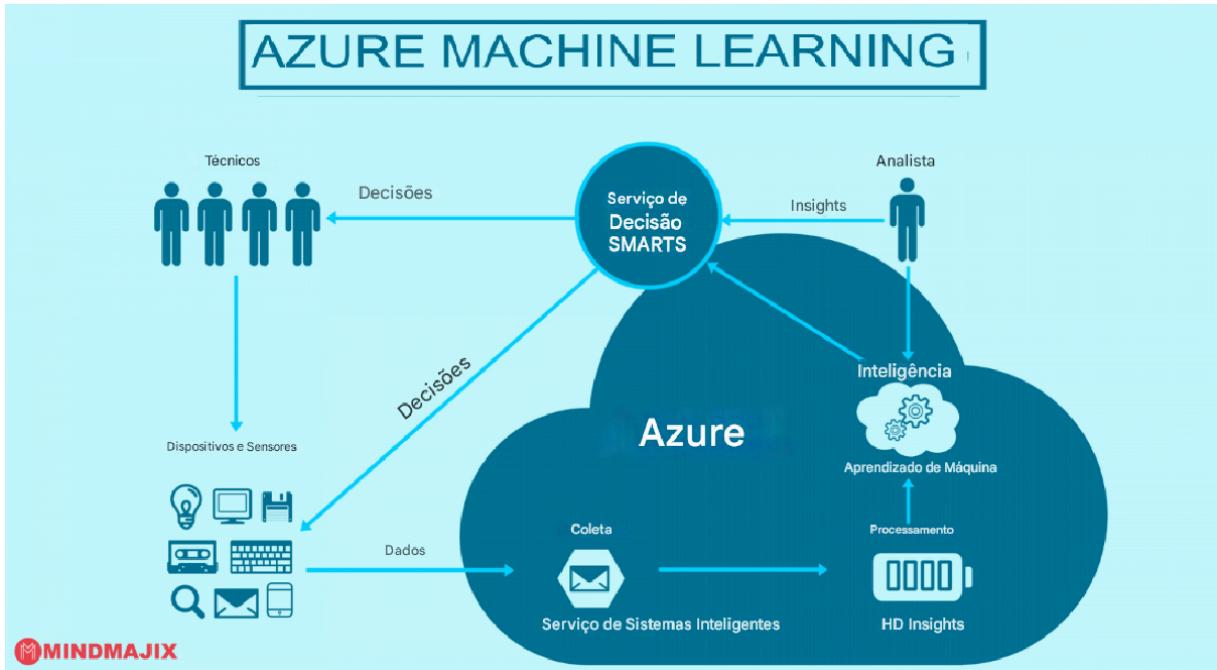


Figura 8- Visão Geral do funcionamento da Azure Machine Learning

Fonte: Adaptado de (Velagana, 2023)

A figura 8 exemplifica o fluxo de trabalho da Azure Machine Learning, mostrando como dados de dispositivos e sensores são coletados e processados na nuvem. A partir desses dados, a inteligência é gerada através do aprendizado de máquina, permitindo que análises resultem em *insights* valiosos. Essas informações, juntamente com decisões automatizadas, são aplicadas de volta aos técnicos e dispositivos, formando um ciclo contínuo de aprimoramento e otimização impulsionado pela nuvem.

2.7.1 AZURE MACHINE LEARNING STUDIO

O Azure Machine Learning Studio se posiciona e se destaca como um ambiente integrado que abrange todas as fases do ciclo de vida de projetos de Machine Learning, desde o desenvolvimento e treinamento até a implantação e gerenciamento. Possui funcionalidades para uso com pouco código ou sem código fonte para treinamento. (Figueiredo, 2020)

Segundo o que é apresentado no website da Microsoft (Microsoft:..., 2025) a plataforma se destaca por oferecer os seguintes benefícios primordiais:

- Ambiente Flexível para Desenvolvimento de Código: A plataforma oferece a conveniência de *notebooks* interativos baseados em Jupyter, executados em servidores gerenciados, que se integram perfeitamente ao estúdio. Para maior flexibilidade, o desenvolvimento de código pode ser estendido a ambientes

como o VS Code, seja na versão web ou *desktop*, permitindo que cientistas e engenheiros de dados trabalhem no ambiente de sua preferência.

- Visualização e Otimização de Experimentos: Para garantir a eficácia e o desempenho dos modelos, o Studio proporciona ferramentas robustas de visualização de métricas de execução. Isso permite uma análise aprofundada dos experimentos, facilitando a otimização contínua dos algoritmos e do processo de treinamento.
- Desenvolvimento de Modelos sem Código (Designer): O Designer do Azure Machine Learning é uma ferramenta intuitiva que democratiza o desenvolvimento de modelos. Por meio de uma interface visual de arrastar e soltar, usuários podem construir e implantar *pipelines* complexos de ML sem a necessidade de escrever código, ideal para engenheiros que buscam agilidade e para aqueles com menos experiência em programação.
- Automatização de Machine Learning (AutoML): A *interface* de usuário dedicada ao Machine Learning Automatizado (AutoML) simplifica drasticamente o processo de experimentação, permitindo a criação de modelos de alta performance com mínima intervenção humana. Essa funcionalidade acelera a descoberta de melhores algoritmos e hiperparâmetros.
- Gestão Coordenada de Rotulagem de Dados: Reconhecendo a importância da qualidade dos dados, o Studio oferece recursos para coordenar eficientemente projetos de rotulagem. Isso é particularmente útil para a preparação de grandes volumes de dados de imagens e textos, assegurando que os conjuntos de treinamento sejam precisos e consistentes.

Dessa forma, a sinergia dessas funcionalidades estabelece o Azure Machine Learning Studio como um ambiente completo e acessível para o desenvolvimento e gerenciamento de inteligência artificial, atendendo a um amplo espectro de usuários, desde codificadores experientes a adeptos de abordagens visuais e automatizadas.

Na figura 9 é possível ver-se a tela inicial do Azure Machine Learning Studio.

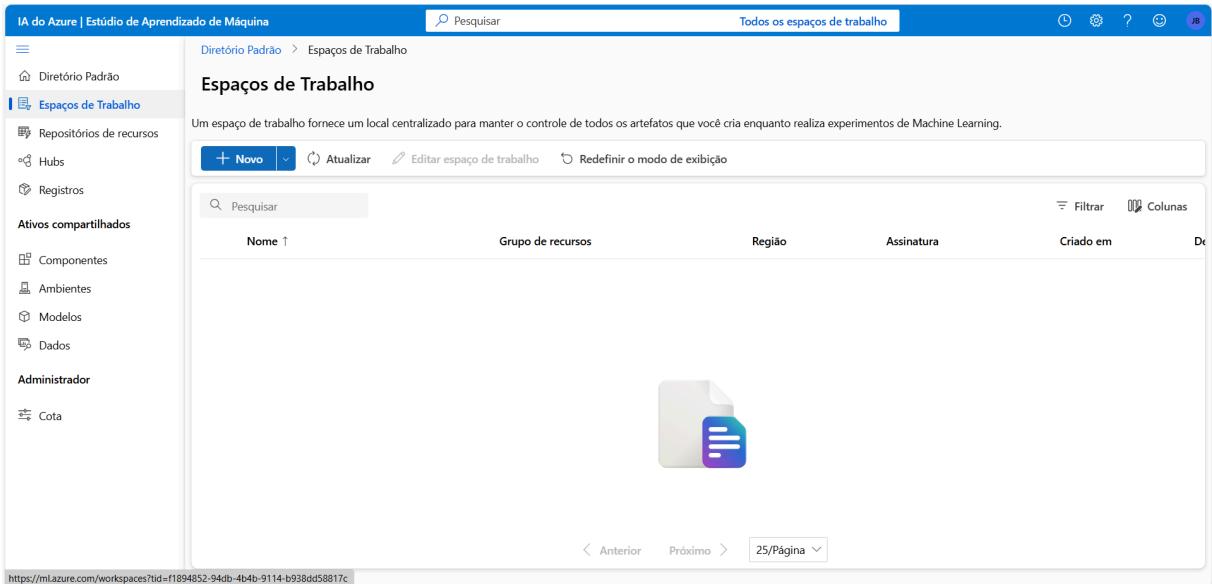


Figura 9 - Tela inicial do Azure Machine Learning Studio

Fonte: Elaborado pelo autor deste trabalho.

2.8 PYTHON

Python é uma linguagem de programação de alto nível, desenvolvida no início dos anos 1990 por Guido Van Rossum, na Holanda, para se tornar sucessor de uma linguagem chamada ABC. (Phyton:..., 2025) Ela foi projetada com um filosofia que enfatiza a legibilidade em seu código, utilizando uma sintaxe clara e concisa, tornando uma linguagem mais fácil de aprender e de ser utilizada.

Além disso, Python é uma linguagem interpretada de propósito geral e de multiplataforma, significando que seu código pode ser utilizado e executado em diferentes sistemas operacionais como Linux, Windows e MacOS sem a necessidade de grandes modificações.

A capacidade do Python de se integrar com outras linguagens de programação e seu vasto e consolidado ecossistema de bibliotecas figuram entre suas características mais notáveis. (Lopes. 2019) As bibliotecas Scikit-learn e LightGBM, que serão discutidas na sequência, representam uma completa e poderosa caixa de ferramentas para ciência de dados. Elas são projetadas com foco em análise, oferecendo funções altamente otimizadas para processamento de dados e configuração eficiente de memória, garantindo que suas operações de *script* rodem com o melhor desempenho possível, especialmente em tarefas de *Machine Learning*.

2.8.1 SCIKIT-LEARN

É uma das bibliotecas mais fundamentais para um clássico projeto de *Machine Learning* em Python. Teve sua trajetória iniciada no ano de 2007 como um projeto do Google Summer of Code por David Cournapeau. Logo se tornou um projeto comunitário, desenvolvido por um grande grupo de pessoas e pesquisadores em todo o mundo. (Scikit-Learn:..., 2025)

A biblioteca se diferencia por oferecer uma API clara e de fácil utilização, facilitando a implementação de algoritmos avançados de aprendizado de máquina. Construída com base em bibliotecas consolidadas como NumPy, SciPy e Matplotlib, ela disponibiliza uma ampla variedade de algoritmos voltados para as principais tarefas de *Machine Learning*, como as mostradas na tabela 3:

Tipo de Tarefas	Descrição	Exemplo Prático
Classificação	Usada para categorizar dados em classes distintas.	Identificar se um e-mail é "spam" ou "não spam".
Regressão	Serve para prever valores contínuos.	Estimar o preço de um imóvel ou automóvel com base em suas características.
Clustering (Agrupamento)	Foca em descobrir padrões e agrupar dados semelhantes que não possuem rótulos predefinidos.	Segmentar clientes com comportamentos parecidos.
Redução de Dimensionalidade	Ajuda a simplificar conjuntos de dados complexos ao reduzir o número de variáveis, mantendo informações importantes.	Análise de Componentes Principais.
Pré-processamento de Dados	Oferece funções essenciais para limpar e preparar os dados antes da modelagem.	Normalização, padronização, tratamento de valores ausentes.
Seleção de Modelos	Fornece ferramentas para avaliar e escolher o melhor modelo para sua tarefa.	Validação cruzada, ajuste de hiperparâmetros.

Tabela 3 - Tipos de Tarefas de ML na Scikit-Learn.

Fonte: Elaborado pelo autor deste trabalho.

Sua popularidade não se refere apenas à sua abrangência, mas também à sua excelente documentação e à sua facilidade de integração com outras bibliotecas do

ecossistema Python de ciência de dados, tornando uma das escolhas ideais tanto para iniciantes quanto para profissionais experientes que buscam uma solução robusta e eficiente para a maioria dos problemas de *Machine Learning*. (Scikit-Learn:..., 2025)

2.8.2 LIGHTGBM

O LightGBM é uma biblioteca de aprendizado de máquina notável por sua base em árvores de decisão e pela utilização da avançada técnica de *gradient boosting*. Foi projetado para processar grandes volumes de dados de forma rápida e eficiente, tendo ganhado vasta aceitação e é amplamente empregada em diversas áreas da ciência de dados e do *Machine Learning*. (Cruz, 2025) Tem como seus principais uso em aplicações como:

- Previsão de Séries Temporais: É uma ferramenta amplamente empregada para prever valores futuros em dados sequenciais, como a estimativa de demandas de mercado ou a projeção de indicadores econômicos.
- Classificação e Regressão: Destaca-se como uma escolha popular em diversas tarefas, incluindo a classificação de clientes, a previsão de *churn* (rotatividade de clientes) e a estimativa de preços.
- Detecção de Fraudes: Devido à sua notável eficiência, é frequentemente utilizado para identificar transações suspeitas em tempo real.
- Recomendação de Produtos: Contribui para a personalização das experiências dos usuários em plataformas, baseando-se em dados históricos de comportamento.

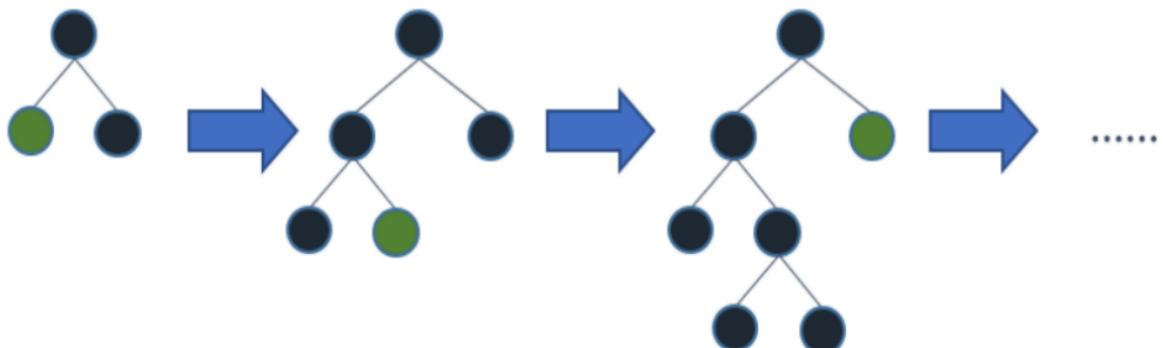


Figura 10 - Ilustração do Crescimento Leaf-wise do LightGBM.

Fonte: Adaptado de (LightGBM:..., 2025)

A Figura 10 ilustra a estratégia de crescimento *leaf-wise* (por folha) empregada pelo algoritmo LightGBM. Ao contrário de outros algoritmos baseados em árvores que tipicamente expandem a árvore *level-wise (por nível)*, adicionando nós em todas as ramificações de uma mesma profundidade, o LightGBM foca na expansão do nó que promete a maior redução de perda. Essa abordagem é visualmente demonstrada na sequência de árvores, onde cada nova expansão representada pelos nós verdes, ocorre na folha que oferece o maior ganho de informação ou a maior diminuição do erro. O resultado são árvores mais profundas e assimétricas, o que contribui para uma otimização mais eficaz do desempenho do modelo.

3 DESENVOLVIMENTO

O capítulo anterior (Fundamentação Teórica) cobriu o "o quê" e o "porquê". Este capítulo de Desenvolvimento foca no "como".

Aqui, vou detalhar o passo a passo de como a solução foi construída, desde a primeira ideia no planejamento até a ferramenta pronta, em aplicação prática. Meu objetivo é mostrar de forma clara e estruturada como cada fase foi conduzida. A ideia é que, ao final, a metodologia que eu usei esteja clara o suficiente para que outras pessoas possam usá-la como referência ou adaptá-la para suas próprias necessidades, seja no trabalho ou em outros projetos.

3.1 AMBIENTE DE DESENVOLVIMENTO E FERRAMENTAS UTILIZADAS

Neste capítulo serão apresentados os detalhes técnicos referentes ao *hardware*, *softwares* e plataformas empregados no desenvolvimento do projeto, destacando a infraestrutura e os recursos utilizados para sua implementação.

3.1.1 HARDWARE

O ambiente de desenvolvimento foi configurado em um *notebook* Dell Vostro com as seguintes especificações:

- Processador: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz (1.19 GHz);
- 16 GB de memória RAM;
- Arquitetura 64 bits, processador baseado em x64;
- Sistema operacional Windows 11 Enterprise versão 25H2

O desempenho do equipamento foi adequado para execução das ferramentas utilizadas no projeto.

3.1.2 SOFTWARE

O desenvolvimento do projeto utilizou um conjunto de ferramentas integradas que possibilitaram a construção, análise e visualização dos dados de forma eficiente. São eles:

- Power BI Desktop: empregado para o desenvolvimento local dos *dashboards*, processo de extração, transformação, carregamento e modelagem dos dados, junto com a criação de medidas e consultas.

- Power BI *Workspace*: utilizado para a publicação e compartilhamento dos *dashboards*, além de permitir o agendamento automático das atualizações.
- Pipefy: utilizado como principal fonte de dados operacionais da equipe de certificados em si, fornecendo planilhas extraídas dos fluxos de trabalho relacionadas a emissão de certificado digital dos clientes.
- Google Colab: em conjunto com a linguagem Python, foi utilizado como ambiente para desenvolvimento e treinamento do modelo da *Machine Learning* possibilitando a manipulação e análise avançada dos dados.
- Snowflake: serviu como fonte de dados secundária para puxar os dados do sistema que roda o Customer Relationship Management (CRM) da empresa.

3.1.3 BIBLIOTECAS E TECNOLOGIAS DE APOIO

No desenvolvimento da *Machine Learning*, foram utilizadas bibliotecas Python para manipulação, pré-processamento, modelagem e visualização dos dados. As principais bibliotecas empregadas incluem:

- Scikit-learn: Empregada na criação, avaliação e validação dos modelos de aprendizado de máquina, além de fornecer ferramentas para pré-processamento, como codificação de variáveis categóricas (OneHot Encoder) e divisão de conjuntos de treino e teste (train_test_split).
- Pandas: utilizada para manipulação, limpeza e organização dos dados, permitindo a construção de *dataframes* e operações sobre tabelas de forma eficiente.
- RandomForestClassifier: algoritmo de *Machine Learning* utilizado para classificação, integrado aos *pipelines* do Scikit-learn para otimização do fluxo de processamento.
- Joblib: utilizada para salvar e carregar *pipelines* de processamento e modelos treinados, garantindo reproduzibilidade e facilidade de reutilização.
- Matplotlib e Seaborn: bibliotecas de visualização empregadas para gráficos exploratórios e apresentação da matriz de confusão, permitindo análise clara do desempenho dos modelos.

3.1.4 INTEGRAÇÃO ENTRE FERRAMENTAS

O fluxo de dados do projeto foi desenhado para integrar as diferentes ferramentas de forma coesa. A base do processo é a extração de dados realizada pelo Power BI, que se conecta tanto ao *Data Warehouse* Snowflake (para obter informações de CRM e corporativas) quanto a planilha exportada do Pipefy, que contém o histórico do processo de certificação dos clientes.

No ambiente do Power BI Desktop, os dados brutos são transformados e modelados, passando por etapas de limpeza e pela criação de medidas essenciais para a análise. A vertente de *Machine Learning* executada no Google Colab, utilizando os dados de certificados da planilha do Power BI onde consta os dados de certificados já com as colunas e medidas calculadas para treinar e validar os modelos preditivos.

A *Machine Learning*, então, é importada para o Power BI, agregando uma camada de inteligência aos *dashboards*. Finalmente, os painéis são publicados no Power BI Service, onde as atualizações automáticas são agendadas para manter a consistência e a atualidade das informações acessíveis à equipe.

3.1.5 MÉTODO DE VERSIONAMENTO E ORGANIZAÇÃO DE ARQUIVOS

Para a gestão dos arquivos do projeto, foi implementado um método de organização no Microsoft OneDrive. Esta plataforma serviu como um repositório único para todos os documentos como planilhas de dados e o modelo de *Machine Learning*, o que permitiu o *backup* contínuo e o acesso controlado pela equipe. Foi estabelecido um padrão para os arquivos, incluindo nomenclatura padronizada e separação por categoria de informação para facilitar a identificação e o manuseio.

O próprio arquivo de trabalho do Power BI foi mantido neste ambiente, em uma pasta específica, para prevenir perdas e controlar as versões do relatório. O acesso ao repositório foi gerenciado por meio de permissões, onde a capacidade de edição foi concedida apenas ao autor e sua coordenadora e o acesso para visualização foi estendido aos demais envolvidos.

A organização geral do projeto seguiu uma estrutura de pastas lógica, relatórios, arquivos de trabalho do Power BI e modelos de *Machine Learning*, o que simplifica a manutenção futura, a atualização contínua e a reprodução do trabalho.

3.2 ANÁLISE E MAPEAMENTO DO PROCESSO

Antes de qualquer implementação tecnológica, realizei uma etapa de análise e mapeamento do processo de certificação digital dos clientes, a fim de compreender a fundo a realidade operacional vigente. A análise foi conduzida por meio de conversas com os gestores e colaboradores da equipe, o que permitiu realizar um mapeamento do fluxo da solicitação.

3.2.1 DESCRIÇÃO DO PROCESSO

O processo operacional da equipe de certificado digital é gerenciado primariamente através da ferramenta Pipefy e segue um fluxo de etapas bem definidas. A Figura 11 ilustra visualmente este fluxo, desde a criação da solicitação até a sua conclusão ou arquivamento.

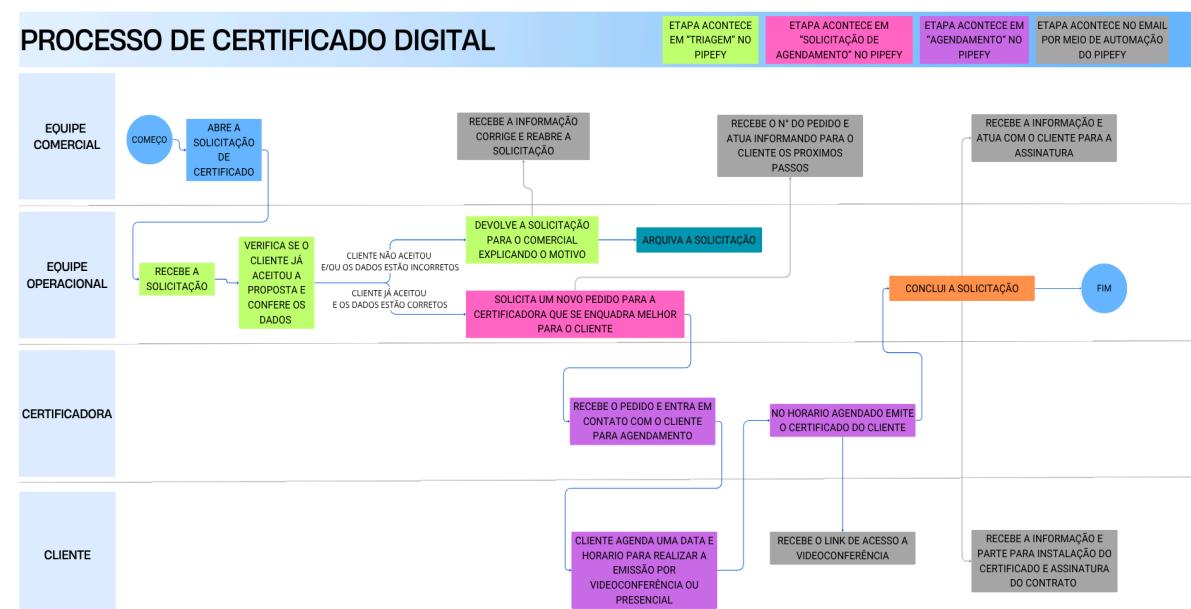


Figura 11 - Fluxograma "As-Is" do Processo de Certificação Digital.

Fonte: Elaborado pelo autor deste trabalho.

Conforme ilustrado na Figura 11, o processo é composto pelas seguintes macro etapas:

- **Abertura da Solicitação:** a solicitação é criada no Pipefy pela equipe comercial. Neste ponto, o *card* contém as informações cadastrais iniciais do cliente, como nome, data de nascimento, telefone, e-mail etc.

- Triagem e Análise Inicial: a equipe de certificação digital recebe a solicitação. A primeira ação é verificar duas condições: se o cliente já aceitou a proposta no sistema corporativo e se os dados cadastrais estão corretos.
- Arquivamento (Pendência Comercial): caso o cliente ainda não tenha aceitado a proposta ou se forem identificadas inconsistências nos dados, a solicitação é devolvida ao comercial com a devida justificativa e, em seguida, arquivada no Pipefy. Este arquivamento notifica a equipe comercial, que, após corrigir a pendência, deve iniciar uma nova solicitação.
- Abertura do Pedido na Certificadora: com os dados validados e o aceite da proposta confirmado, a equipe interna inicia a abertura do pedido do certificado digital junto à certificadora parceira. A escolha da certificadora pode considerar, por exemplo, o posto de atendimento mais próximo do endereço do cliente, caso a emissão presencial seja mandatória.
- Agendamento: após a abertura do pedido, o processo segue para a fase de agendamento. Nesta etapa, a certificadora contata o cliente para verificar a disponibilidade de emissão imediata ou para definir a melhor data e horário para o atendimento (realizado por videoconferência).
- Emissão do Certificado: na data agendada, a certificadora conduz a videoconferência com o cliente. Ao finalizar o processo de validação, o certificado é emitido e o *status* é finalizado no sistema da própria certificadora.
- Conclusão (Sistema Pipefy): após a confirmação de que o certificado foi emitido pela certificadora, a solicitação no Pipefy é movida manualmente pela equipe operacional para a coluna "Concluído", finalizando formalmente o processo no sistema de gestão de tarefas.

3.2.2 IDENTIFICAÇÃO DE GARGALOS E DEFINIÇÃO DA MÉTRICA PRINCIPAL.

A partir do mapeamento do processo e das conversas com a equipe, foi possível identificar os principais desafios e gargalos que impactam diretamente o tempo de atuação da equipe e a eficiência do processo. Estes desafios estão resumidos na tabela 4, mostrada abaixo:

GARGALO IDENTIFICADO	DESCRIÇÃO DO PROBLEMA	IMPACTO NO PROCESSO
Retrabalho (Pendências Comerciais)	O processo é arquivado se a equipe comercial envia dados inconsistentes ou se o cliente não deu o aceite. Isso força a criação de uma nova solicitação.	Aumenta drasticamente o tempo percebido pelo cliente. Gera trabalho duplicado para a equipe comercial e para a equipe de certificação (nova triagem). Dificulta a medição do desempenho, pois a solicitação "original" é perdida
Dependência Externa (Certificadora)	A equipe de certificação depende da disponibilidade da certificadora para as fases de "Agendamento" e "Emissão". Não há controle interno sobre esse tempo.	O processo pode ficar parado por dias aguardando a certificadora, sem visibilidade para a gestão interna. Aumenta o <i>Lead Time Total</i> , mesmo que a equipe interna tenha sido ágil.
Etapa Manual de Conclusão	A equipe operacional precisa mover manualmente a solicitação para "Concluído" no Pipefy após a certificadora finalizar o processo em outro sistema.	Risco de esquecimento ou atraso na baixa do <i>card</i> , gerando dados de <i>Lead Time</i> incorretos (parecendo que demorou mais do que realmente demorou). Consome tempo da equipe em uma tarefa de baixo valor agregado.
Falta de Visibilidade Gerencial	Gestores não conseguem responder rapidamente: "Qual equipe comercial gera mais retrabalho?" ou "Qual o tempo médio de atuação real da nossa equipe, excluindo o tempo da certificadora?".	Dificuldade em tomar decisões baseadas em dados. Impossibilidade de focar treinamentos nas equipes comerciais que mais precisam.

Tabela 4 - Mapeamento de Gargalos do Processo de Certificação.

Fonte: Elaborado pelo autor deste trabalho.

Para endereçar esses desafios e fornecer uma base quantitativa para a análise, este trabalho formaliza a métrica de *Lead Time* como o principal Indicador-Chave de Desempenho (KPI).

Este trabalho foca em otimizar o *Lead Time* Operacional, pois ele mede diretamente a eficiência da equipe interna, e em analisar as causas do retrabalho, que impactam negativamente o *Lead Time* Total.

3.2.3 LEVANTAMENTO E ESTRUTURA DAS FONTES DE DADOS

Para viabilizar o cálculo das métricas definidas e a análise dos gargalos, realizei um levantamento das fontes de dados disponíveis. Conforme mencionado na seção 3.1.4, os dados são oriundos de planilhas de exportação do Pipefy (processo) e do Snowflake (dados corporativos e de CRM).

A Tabela 5 detalha os principais campos (atributos) que formam a base de dados deste projeto e que foram utilizados nas etapas de ETL e modelagem.

Nome do Campo	Fonte de Dados	Tipo de Dado	Descrição e Propósito no Projeto
Código	Pipefy (planilha)	Numérico	Identificador único de cada solicitação. Chave primária.
Fase atual	Pipefy (planilha)	Texto	Fase atual da solicitação (“Triagem” “concluído” “arquivado”)
Criado em	Pipefy (planilha)	Data/Hora	Data e hora em que a solicitação foi criada.. Início do <i>Lead Time</i> .
Nº da Proposta/Contrato	Pipefy (planilha)	Texto	Identificador da proposta. É a chave de relação (chave estrangeira) usada para conectar os dados da solicitação (Pipefy) à tabela de proposta do CRM (Snowflake).
Área solicitante	Pipefy (planilha)	Texto	Nome da equipe comercial que originou a solicitação. Campo-chave para o objetivo de treinamento.

Nome do solicitante	Pipefy (planilha)	Texto	Nome do consultor ou responsável que abriu a solicitação. Campo chave para uma análise individual.
Nome do cliente	Pipefy (planilha)	Texto	Nome do cliente para identificação e acompanhamento do comercial com a solicitação.
Cliente possui CNH	Pipefy (planilha)	“Sim” ou “Não”	Campo para análise de possível cliente presencial ou por videoconferência
Emissão do certificado	Pipefy (planilha)	Texto	Campo para identificar 1ºemissão ou 2ºemissão.
A solicitação está correta?	Pipefy (planilha)	Texto	Campo utilizado para verificar se a solicitação foi aberta corretamente pelo comercial.
Qual Certificadora?	Pipefy (planilha)	Texto	Campo para identificar qual foi a certificadora que foi aberto o pedido do cliente.
Nº do protocolo/ticket	Pipefy (planilha)	Numérico	Campo de identificação do número do pedido junto com a certificadora. (Único para cada cliente)
Tipo de emissão	Pipefy (planilha)	“Presencial” ou “videoconferência”	Campo onde realmente consta se o cliente emitiu presencial ou por videoconferência. (preenchido somente ao concluir a solicitação)
Última vez que entrou na fase agendamento	Pipefy (planilha)	Data/Hora	Data e hora em que o pedido foi enviado à certificadora. Fim da análise do <i>Lead Time</i> de criado até abertura do pedido.

Última vez que entrou na fase Concluído	Pipefy (planilha)	Data/Hora	Data e hora em que o card entrou pela última vez na fase "Concluído" no Pipefy. Fim do <i>Lead Time</i> Total.
Tempo total na fase agendamento	Pipefy (planilha)	Numérico	Tempo total que o cliente ficou entre a abertura do pedido e a conclusão ou arquivamento do mesmo.
Motivo do arquivamento	Pipefy (planilha)	Texto	Motivo do arquivamento da solicitação Crucial para análise de gargalos.
Precisou de ajuda?	Pipefy (planilha)	Texto	Campo onde é preenchido se o comercial ou o cliente precisou de alguma ajuda da equipe operacional.
Número do contrato	Snowflake (Registro Contrato)	Numérico	Identificador único do contrato. Chave Primária (PK) da tabela de Contratos
Data de assinatura	Snowflake (Registro Contrato)	Data	Campo onde consta a data de assinatura do cliente no contrato.
Proposta	Snowflake (Proposta CRM)	Texto	Identificador único da proposta. É a chave de relação (chave Primária) usada para conectar os dados da proposta do CRM (Snowflake) à solicitação (Pipefy).
Núm. Contrato	Snowflake (Proposta CRM)	Numérico	Atributo de contrato de proposta CRM (Snowflake). É utilizado para enriquecer a tabela de 'Solicitações' (Pipefy) por meio de mesclagem (proposta) e, subsequentemente, serve como Chave Estrangeira

			para o relacionamento com a tabela de Registro Contrato (Snowflake).
Estado da Proposta	Snowflake (Proposta CRM)	Texto	Campo para verificar o status da proposta. “ativa” “cancelada” “expirada”.
Data Feriado	Feriado (planilha)	data	Campo usado para retirar feriados do <i>Lead Time</i> em dias úteis.

Tabela 5 - Dicionário de Dados dos Atributos Relevantes.

Fonte: Elaborado pelo autor deste trabalho.

Com o processo mapeado, os gargalos priorizados e as fontes de dados estruturadas, a próxima etapa consistiu na construção efetiva da solução de *Business Intelligence* para tratar, modelar e visualizar estas informações.

3.3 CONSTRUÇÃO DA SOLUÇÃO DE BUSINESS INTELLIGENCE (POWER BI)

Após o mapeamento do processo e a definição das fontes de dados, iniciei a construção da solução no Power BI. Este processo foi dividido em três etapas principais: tratamento e carregamento dos dados (ETL), modelagem e criação de medidas (DAX) e de colunas calculadas e o desenvolvimento dos painéis visuais (*Dashboards*).

3.3.1 PROCESSO DE ETL (EXTRAÇÃO, TRANSFORMAÇÃO E CARGA)

O primeiro passo para a construção da solução consistiu na implementação do processo de Extração, Transformação e Carregamento (ETL). Esta fase foi inteiramente desenvolvida no Power Query, o editor integrado ao Power BI Desktop, e teve como objetivo preparar os dados brutos de diferentes fontes para que se tornassem um conjunto de dados coeso, limpo e pronto para a modelagem.

O processo foi estruturado da seguinte forma:

Extração (Conexão com as Fontes): o processo de ingestão de dados iniciou-se com a seleção dos conectores adequados na interface "Nova Fonte" do Power BI Desktop. A ferramenta disponibiliza uma vasta gama de conectores nativos, permitindo a integração de múltiplas fontes, conforme ilustra a Figura 12.

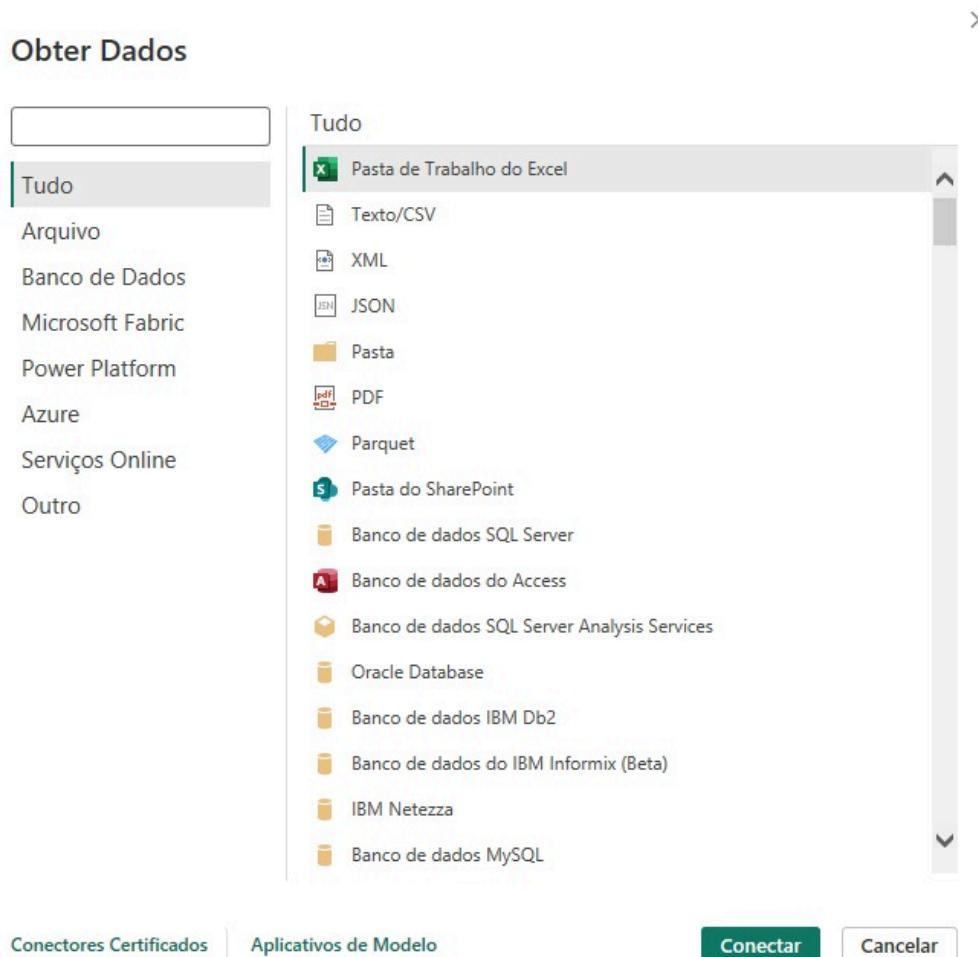


Figura 12 - Janela de Conectores de Dados do Power BI Desktop.

Fonte: Captura de tela do Power BI Desktop (2025).

Para os dados operacionais do Pipefy, provenientes de uma planilha, foi selecionado o conector Pasta de trabalho do Excel, configurado para acessar o arquivo de exportação armazenado no repositório do OneDrive. Para os dados corporativos, foi estabelecida uma conexão direta com o *Data Warehouse* da empresa utilizando o conector Snowflake, também nativo no Power BI.

Após a seleção, os dados foram carregados no Editor do Power Query para a etapa de Transformação. A Figura 13 ilustra a interface do editor com o painel "Consultas" à esquerda, listando todas as tabelas carregadas no projeto.

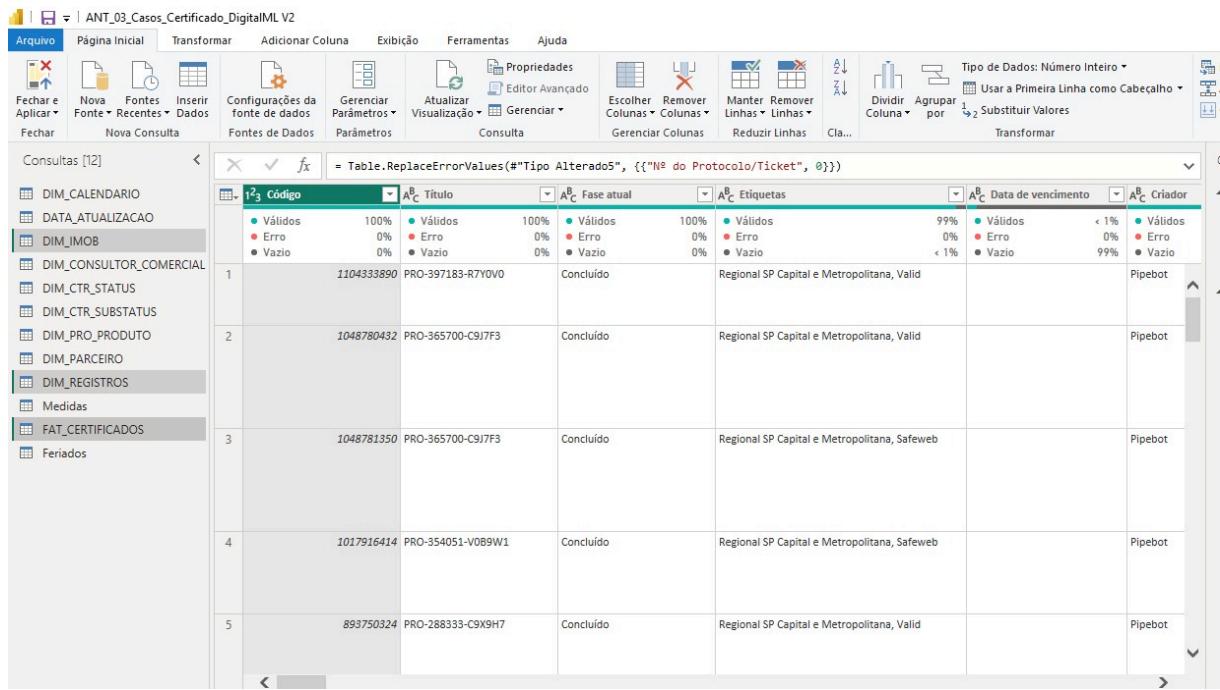


Figura 13 - Interface do Editor do Power Query com as Consultas Carregadas.

Fonte: Captura de tela do Power BI Desktop (2025).

Conforme a Figura 13, as tabelas selecionadas em destaque são as tabelas principais. As demais são tabelas de apoio do Snowflake; como é um *Data Warehouse* robusto, ele se desmembra em múltiplas tabelas (normalização) para otimizar o armazenamento, utilizando chaves estrangeiras (IDs) para campos que se repetem. As Tabelas relacionadas como “Feriado” “Medidas” “DATA_ATUALIZAÇÃO” “DIMCALENDARIO” são tabelas criadas para apoiar a modelagem e serão detalhadas nas seções seguintes.

Após a ingestão dos dados no Power BI, iniciou-se a etapa de transformação. Esta fase é a mais crítica do processo de ETL, pois é onde os dados brutos são higienizados, padronizados e enriquecidos. É um processo cauteloso e muito bem construído que demanda tempo e atenção para garantir a qualidade e a integridade das informações que alimentarão o modelo. As principais etapas de tratamento foram divididas da seguinte forma:

Higienização e Padronização de Dados: As primeiras ações de tratamento se concentraram na limpeza dos dados. Isso incluiu o preenchimento de valores nulos (quando aplicável) e a padronização de colunas de texto, como por exemplo, o campo [Motivo do arquivamento], para garantir a consistência das categorias de erro. Adicionalmente, foi feita a correção dos tipos de dados para otimizar o modelo, assegurando que datas fossem reconhecidas como

Data/Hora exemplo das colunas: Criado em, Última vez que entrou na fase Concluído; e algumas colunas como números inteiros tendo como exemplo os campos de: Código, N° do protocolo/ticket.

Criação de Tabelas de Apoio: Paralelamente ao tratamento dos dados principais, foram geradas tabelas auxiliares diretamente no Power Query para apoiar a modelagem e as análises de tempo:

- DIM_CALENDARIO: Para habilitar as análises de inteligência de tempo, foi implementada esta dimensão de tempo padronizada. A tabela foi gerada no Power Query a partir de uma função que define um intervalo dinâmico de datas e foi expandida para incluir múltiplos atributos de tempo, como Ano, Mês, Semestre, Bimestre e indicadores de fim de semana.
- Tabela de Feriados: Foi importada e tratada uma tabela simples contendo os feriados nacionais e os feriados de folga da equipe operacional. O propósito desta tabela é permitir o cálculo de métricas de desempenho em dias úteis, cruzando-a com a DIM.CALENDARIO para excluir fins de semana e feriados da contagem.
- DATA_ATUALIZAÇÃO: Para esta tabela, foi utilizada uma abordagem robusta para refletir a data da carga de dados no *Data Warehouse*, em vez da hora de atualização do próprio relatório. Foi executada uma consulta SQL nativa diretamente na fonte Snowflake, utilizando a função Value.NativeQuery do Power Query. Esta consulta foi otimizada para buscar o valor máximo de data da coluna de registro de dados na tabela Fato principal do *Data Warehouse*. A consulta em linguagem M, com os nomes de servidor e tabelas substituídos por tabela genérica para fins de replicação, é demonstrada abaixo:

```

1   = Value.NativeQuery(
2 // 1. Conexão com a fonte Snowflake
3 | Snowflake.Databases(
4 | | "seuservidor.real.snowflakecomputing.com", // URL do Servidor
5 | | "SEU_WAREHOUSE" // Warehouse
6 |
7 | | [Name="SEU_BANCO_DE_DADOS"] // Banco de Dados
8 | | }[Data],
9 // 2. Consulta SQL Nativa enviada ao Snowflake
10 | "select#(lf)
11 | | to_varchar(
12 | | | max([sua_coluna_de_data_hora])::datetime,
13 | | | 'DD/MM/YYYY HH24:MI'
14 | | ) as ""Data Atualização dos Dados"""
15 | | #(lf)from [seu_esquema].[sua_tabela_de_fatos],
16 | | null,
17 | | [EnableFolding=true]
18 )

```

Figura 14 - Consulta Power Query para Criação da tabela DATA_ATUALIZAÇÃO.

Fonte: Elaborado pelo autor deste trabalho.

Criação de Colunas Personalizadas: Finalmente, uma etapa crucial da transformação foi a criação de Colunas Personalizadas no Power Query. Esta etapa, também conhecida como Engenharia de Atributos, foi executada com um duplo objetivo:

- Simplificar as análises e a criação de medidas DAX complexas no Power BI.
- Preparar o conjunto de dados para a etapa de *Machine Learning*, cujo objetivo é prever solicitações com alto risco de atraso no SLA.

As colunas personalizadas mais relevantes criadas nesta fase Inclui:

```

1  v = Table.AddColumn(#"Linhas Filtradas", "Idade Cliente", each let
2      DataNascimentoRaw = [Data de Nascimento], // Coluna de origem
3      DataAtual = Date.From(DateTime.LocalNow()),
4
5      // Converte a DataNascimento para tipo Date, lidando com nulos e erros
6      DataNascimento = try Date.From(DataNascimentoRaw) otherwise null,
7
8  v
9  | CalculaIdade =
10 |     if DataNascimento = null or DataNascimento > DataAtual then
11 |         null // Retorna null se DataNascimento for nula/inválida ou futura
12 |     else
13 |         let
14 |             AnoNascimento = Date.Year(DataNascimento),
15 |             MesNascimento = Date.Month(DataNascimento),
16 |             DiaNascimento = Date.Day(DataNascimento),
17 |
18 |             AnoAtual = Date.Year(DataAtual),
19 |
20 |             // Determina a data do aniversário neste ano, ajustando para 29 de fevereiro
21 |             AniversarioEsteAno =
22 |                 if MesNascimento = 2 and DiaNascimento = 29 and Date.DaysInMonth(#date(AnoAtual, 2, 1)) = 28 then
23 |                     // Se for 29/Fev e o AnoAtual NÃO é bissexto, considera 28/Fev para o cálculo
24 |                     #date(AnoAtual, 2, 28)
25 |                 else
26 |                     // Caso contrário, usa a data de nascimento normal neste ano
27 |                     #date(AnoAtual, MesNascimento, DiaNascimento),
28 |
29 |             AnosTotais = AnoAtual - AnoNascimento,
30 |             IdadeFinal = if AniversarioEsteAno > DataAtual then AnosTotais - 1 else AnosTotais
31 |         in
32 |             IdadeFinal
33 |     CalculaIdade)

```

Figura 15 - Código M para Coluna Personalizada Idade Cliente.

Fonte: Elaborado pelo autor deste trabalho.

Esta personalização foi criada para retornar a idade do cliente, como só tínhamos a data de nascimento como coluna na planilha de certificados, esse código na linguagem M direto no *power query*, nos retornou uma nova coluna com a idade de cada cliente, padronizando e facilitando a leitura para a futura *Machine Learning*. Com alguns ajustes para pequenos erros que apareceram, por exemplo os clientes que nasceram dia 29 de fevereiro ele corrige para que não retorne erros.

```

1  = Table.AddColumn(#"Personalização Adicionada1", "Período do Mês", each let
2      DataCriacao = [Criado em], // Referência à sua coluna existente
3      DiaDoMes = if DataCriacao = null then null else Date.Day(DataCriacao),
4
5      Período = if DiaDoMes = null then null else
6          if DiaDoMes >= 1 and DiaDoMes <= 7 then "1º Semana"
7          else if DiaDoMes >= 8 and DiaDoMes <= 14 then "2º Semana"
8          else if DiaDoMes >= 15 and DiaDoMes <= 21 then "3º Semana"
9          else if DiaDoMes >= 22 and DiaDoMes <= 28 then "4º Semana"
10         else if DiaDoMes >= 29 then "5º Semana"
11         else null // Fallback para qualquer caso não coberto
12     in
13     Período)

```

Figura 16 - Código M para Coluna Personalizada Período do Mês.

Fonte: Elaborado pelo autor deste trabalho.

Esta personalização foi criada para segmentar o tempo em semanas. Assim tem-se uma visão de quantidade de solicitações criadas em cada semana do mês, como as solicitações vem da equipe comercial que é responsável por bater metas mensais, onde todo final de mês tende a ser com um número maior de solicitações criadas onerando assim a equipe operacional. Foi escolhida a segmentação por semana por conta de uma campanha do comercial que se baseia em semanas. Da mesma forma, foi feita outra personalização para verificar a quantidade de solicitações concluídas, mas foi segmentada em dezenas do dia 01 a 10, 11 a 20, 21 a 31, para ser mais fácil a visualização e teve como data de análise a coluna “Última vez que entrou na fase Concluído”. O objetivo destas novas colunas é permitir a análise de picos de demanda e fluxo de entrega e também será um bom dado para análise de aprendizado da *Machine Learning*.

Carregamento (Load): Ao final de todas as etapas de transformação, as tabelas tratadas, mescladas e enriquecidas (como FAT_CERTIFICADOS, DIM_CALENDARIO, Feriados, etc.) foram carregadas no modelo de dados do Power BI Desktop. Este processo concluiu a fase de ETL, resultando em um conjunto de dados limpo, otimizado e relacional, pronto para a próxima etapa: a Modelagem Relacional.

3.3.2 MODELAGEM RELACIONAL DE DADOS.

Após o carregamento dos dados tratados pelo Power Query, a próxima etapa foi estabelecer os relacionamentos entre as tabelas no Power BI. Esta fase é crucial para que os filtros e cálculos DAX funcionem corretamente.

Neste projeto, foi adotado um Modelo Floco de Neve (*Snowflake Schema*). Esta estrutura, que reflete a modelagem altamente normalizada do *Data Warehouse (Snowflake)* de origem, caracteriza-se por ter tabelas de dimensão que se desmembraram em outras tabelas de sub-dimensão. Como descrito na seção 3.3.1, em vez de uma única tabela de dimensão, existem tabelas de apoio conectadas a outras dimensões, uma prática que otimiza o armazenamento no *Data Warehouse* de origem.

A Figura 17 ilustra o diagrama do modelo de dados final, evidenciando as múltiplas tabelas de dimensão e suas sub-dimensões conectadas à tabela Fato central (FAT_CERTIFICADOS).

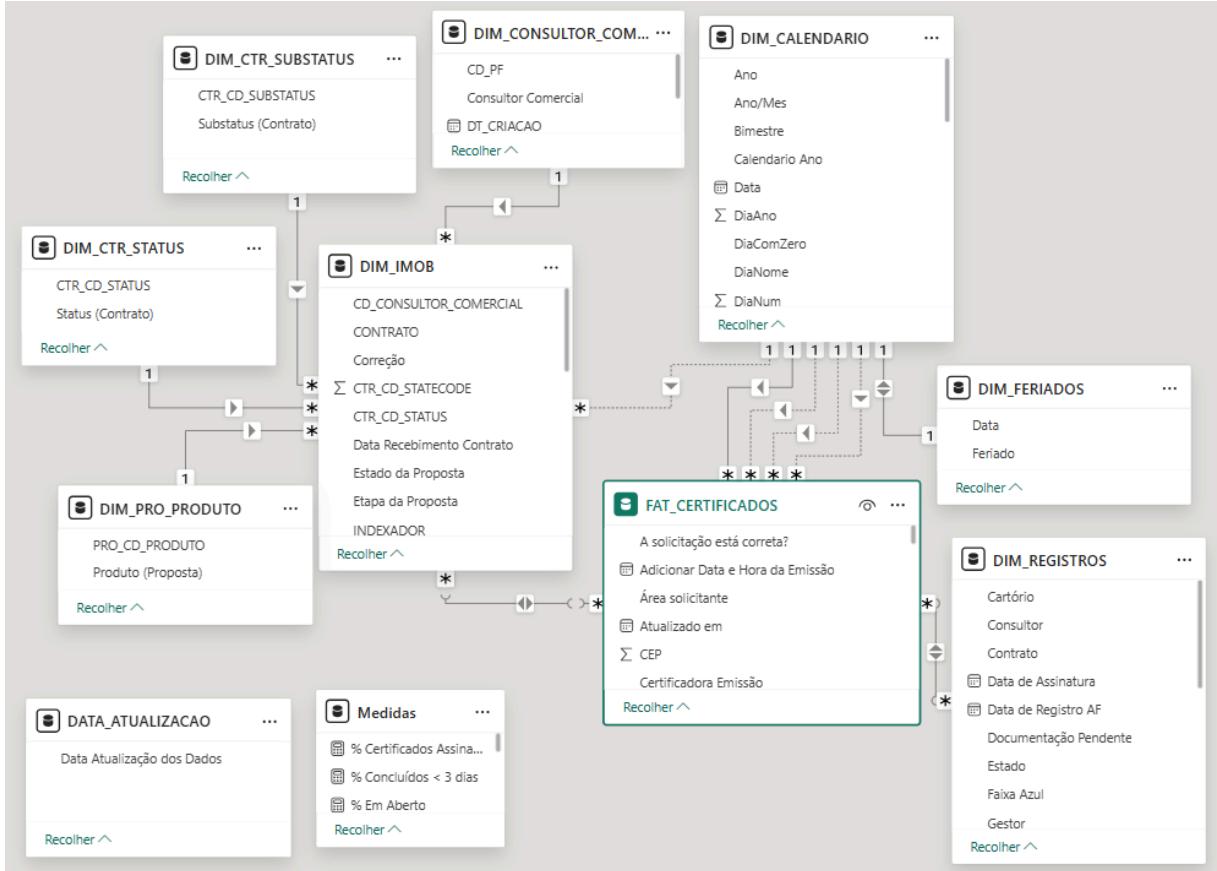


Figura 17 - Diagrama do Modelo Relacional Floco de Neve (Snowflake Schema).

Fonte: Elaborado pelo autor deste trabalho.

O estabelecimento correto desses relacionamentos foi essencial para garantir a integridade das consultas. Devido à complexidade dos dados corporativos e à natureza do modelo Floco de Neve, foram utilizadas diferentes cardinalidades, incluindo relacionamentos um-para-muitos (1:*) e um-para-um (1:1), e, em casos específicos, muitos-para-muitos (:). A configuração precisa dessas cardinalidades e das direções de filtragem permitiu que a tabela fato FAT_CERTIFICADOS fosse corretamente filtrada por atributos de todas as tabelas de dimensão, independentemente do seu nível na hierarquia do modelo.

Um ponto de destaque adicional na modelagem, e que é fundamental para a solução, é o tratamento da DIM_CALENDARIO. Esta tabela atua como uma dimensão de múltiplos papéis, pois ela se conecta à tabela FAT_CERTIFICADOS mais de uma vez, representando eventos distintos no tempo. No diagrama, mostrado na Figura 16, é possível observar uma relação ativa em linha sólida, ligando a DIM_CALENDARIO[Data] à coluna FAT_CERTIFICADOS[Criado Em], e outras relações inativas em linhas pontilhadas, como a ligação com a coluna [Última vez que entrou na fase Concluído]. Esta abordagem de modelagem é um pré-requisito técnico que permite, posteriormente, o uso da função

USERELATIONSHIP nas medidas DAX. Dessa forma, é possível calcular e comparar corretamente o fluxo de solicitações "Criadas" e "Concluídas" dentro do mesmo período (como Mês ou Semana), utilizando a mesma dimensão de calendário.

3.3.3 CRIAÇÃO DE COLUNAS CALCULADAS E MEDIDAS (DAX)

Com o modelo relacional Floco de Neve (*Snowflake Schema*) e as dimensões de múltiplos papéis (DIM_CALENDARIO) devidamente estruturados na seção anterior, a próxima etapa foi a construção da camada semântica e analítica. Esta camada é o "cérebro" da solução, sendo responsável por transformar os dados brutos e já tratados em métricas de negócio e análise de desempenho, que respondem diretamente aos gargalos identificados na seção 3.2.2. A linguagem utilizada para esta finalidade foi a DAX (*Data Analysis Expressions*). O DAX opera sobre o modelo de dados já carregado e permite dois tipos principais de cálculos, ambos explorados neste projeto:

- Colunas Calculadas: São cálculos que operam em contexto de linha (linha a linha) durante a atualização dos dados. Elas armazenam um valor estático em uma nova coluna na tabela, sendo ideais para atributos fixos de uma solicitação, como o SLA_Dias_Uteis que será demonstrado abaixo.
- Medidas: São cálculos dinâmicos que operam em contexto de filtro (calculados em tempo real) com base nas interações do usuário no *dashboard*. Elas não armazenam dados e são usadas para agregar valores, como calcular uma média ou uma taxa.

Nesta seção, serão detalhadas as principais fórmulas DAX (*Data Analysis Expressions*) criadas tanto para medidas tanto para colunas calculadas, com destaque para a ativação das relações inativas (mencionadas na seção 3.3.2) através da função *USERELATIONSHIP*, um passo essencial para analisar o fluxo de criações e conclusões no mesmo período.

Colunas Calculadas (DAX): A principal coluna calculada foi incluída na tabela FAT_CERTIFICADOS com o objetivo de determinar o SLA (Acordo de Nível de Serviço) em dias úteis para cada solicitação. Esse cálculo foi implementado como coluna, pois sua execução em tempo real por meio de uma medida seria menos eficiente.

```

1 SLA_Dias_Uteis =
2 VAR DataInicial = FAT_CERTIFICADOS[Criado em]
3 VAR DataFinal = FAT_CERTIFICADOS[Última vez que entrou na fase Concluído]
4 RETURN
5 COUNTROWS(
6   FILTER(
7     'DIM_CAENDARIO',
8     'DIM_CAENDARIO'[Data] >= DataInicial &&
9     'DIM_CAENDARIO'[Data] <= DataFinal &&
10    'DIM_CAENDARIO'[FinalSemana] = "Não" &&
11    'DIM_CAENDARIO'[FERIADO] = "Não"
12  )
13 )-1
14

```

Figura 18 - Código DAX da Coluna Calculada SLA_Dias_Uteis

Fonte: Elaborado pelo autor deste trabalho.

Esta coluna calculada foi criada para demonstrar o SLA em dias úteis, filtrada nela os finais de semana que vem da base da planilha DIM_CAENDARIO e os feriados da base da planilha FERIADOS para contagem de *Lead Time* total entre as datas das colunas “criado em” e “Última vez que entrou na fase Concluído” da planilha principal FAT_CERTIFICADOS.

Medidas (DAX): Por fim, com o propósito de atender às demandas analíticas e sustentar os elementos visuais do *dashboard*, foram elaboradas medidas DAX. Para garantir melhor organização e manutenção do modelo, essas medidas foram agrupadas em uma tabela específica denominada “Medidas”. As principais medidas criadas no contexto deste projeto estão apresentadas abaixo:

```

1 Total_Solicitações = CALCULATE(
2   COUNT('FAT_CERTIFICADOS'[Código]),
3   USERELATIONSHIP(DIM_CAENDARIO[Data], 'FAT_CERTIFICADOS'[Criado em]))

```

Figura 19 - Código DAX da Medida Total_Solicitações (Criadas).

Fonte: Elaborado pelo autor deste trabalho.

Esta medida é um exemplo da técnica de *role-playing dimension* discutida na seção de modelagem (3.3.2). Ela usa a função *USERELATIONSHIP* para ativar a relação inativa entre a DIM_CAENDARIO e a data de Criação da planilha FAT_CERTIFICADO. Isso permite que o usuário filtre um mês (ex: “Outubro”) e, usando essa medida nos gráficos do *dashboard*, só verá a quantidade de solicitações criadas no determinado mês escolhido, a mesma medida foi replicada para contabilizar as solicitações concluídas no mês, só determinando que a fase

atual da solicitação teria que ser a “concluído” e trocando a relação de data também, como mostra a figura 20.

```

1 Total·Solicitações Concluídas = CALCULATE(
2     COUNT('FAT_CERTIFICADOS'[Código]),
3     'FAT_CERTIFICADOS'[Fase atual] = "Concluído",
4     USERELATIONSHIP(DIMCALENDARIO[Data], 'FAT_CERTIFICADOS'[Primeira vez que entrou na fase Concluído]))
```

Figura 20 - Código DAX da Medida Total Solicitações (Concluidas).

Fonte: Elaborado pelo autor deste trabalho.

Diferentemente da coluna SLA_Dias_Uteis, que considera apenas os dias úteis, a medida na figura 21 foi desenvolvida para calcular a média do *Lead Time* Total em dias corridos. Para isso, é empregada a função AVERAGEX, que percorre cada linha da tabela FAT_CERTIFICADOS, aplicando a função DATEDIFF para obter a quantidade de dias, entre a data de criação e a data de conclusão de cada registro. Ao final, a AVERAGEX retorna a média desses intervalos.

```

1 Média Lead Time Total (Emitidos) =
2 AVERAGEX(
3     FAT_CERTIFICADOS,
4     DATEDIFF(FAT_CERTIFICADOS[Criado em], FAT_CERTIFICADOS[Primeira vez que entrou na fase Concluído], DAY)
5 )
```

Figura 21 - Código DAX da Medida Média Lead Time Total (Emitidos).

Fonte: Elaborado pelo autor deste trabalho.

Essa métrica é essencial para avaliar o tempo total percebido pelo cliente, abrangendo também os finais de semana e feriados.

```

1 Concluídos < 3 dias =
2 CALCULATE(
3     COUNTROWS('FAT_CERTIFICADOS'),
4     FILTER(
5         'FAT_CERTIFICADOS',
6         'FAT_CERTIFICADOS'[Fase atual] = "Concluído"
7         && 'Medidas'[Média Lead Time Total (Emitidos)] <= 3
8     )
9 )
```

Figura 22 - Código DAX da Medida Concluídos < 3 dias.

Fonte: Elaborado pelo autor deste trabalho.

Esta medida foi criada para quantificar o volume de entregas rápidas, definindo um SLA de 3 dias corridos. A função CALCULATE modifica o contexto de filtro, e a função FILTER é usada para iterar sobre a tabela FAT_CERTIFICADOS e aplicar duas condições:

- A fase atual da solicitação deve ser "Concluído".
- A medida Média *Lead Time* Total (Emitidos) calculada no contexto de filtro atual deve ser menor ou igual a 3 dias.

```

1 % Concluídos < 3 dias =
2 DIVIDE(
3     [Concluídos < 3 dias],
4     [Total Solicitações Concluídas],
5     0
6 )

```

Figura 23 - Código DAX da Medida % Concluídos < 3 dias.

Fonte: Elaborado pelo autor deste trabalho.

Por fim, esta medida de KPI transforma o volume absoluto da medida anterior em um percentual de performance. Ela divide o total de concluídos < 3 dias, visto na Figura 22, pela Total Solicitações Concluídas (Figura 20). O uso da função DIVIDE é uma boa prática que garante que o cálculo retorne 0 em vez de um erro caso o denominador seja zero, o que é crucial para a exibição nos visuais.

Com a implementação destas principais colunas calculadas e medidas complexas, a camada analítica e semântica do projeto está concluída. O modelo de dados agora está equipado com todos os KPIs necessários para quantificar o desempenho operacional, medir o SLA em diferentes granularidades (dias úteis e corridos) e, crucialmente, identificar a performance de entrega (% Concluídos < 3 dias). A próxima e última etapa do desenvolvimento é a construção da camada de visualização, onde essas métricas serão traduzidas em *insights* acionáveis e apresentadas de forma interativa nos *dashboards* para os usuários finais.

3.3.4 DESENVOLVIMENTO DOS DASHBOARDS E VISUALIZAÇÕES

A etapa final do desenvolvimento da solução de *Business Intelligence* consistiu na criação da interface visual. Esta é a camada de apresentação, responsável por traduzir o complexo modelo de dados Floco de Neve e as medidas DAX em *insights* açãoáveis para os usuários finais. A solução não é composta por um único *dashboard*, mas sim por um conjunto de relatórios interativos com múltiplas páginas, cada qual com um objetivo analítico específico. A seguir, são detalhadas as principais telas que compõem a solução:

A Primeira tela, “Análise Solicitações”, conforme a figura 24, é dedicada à análise de causa-raiz das entradas do processo e do retrabalho primeiro Gargalo mencionado na seção 3.2.2.



Figura 24 - Tela "Análise Solicitações" Análise de Solicitações Criadas e Arquivamento.

Fonte: Elaborado pelo autor deste trabalho.

Esta tela é fundamental para os objetivos do projeto, pois foca nos problemas de origem:

- **Solicitações Criadas de Certificado por Mês:** Este gráfico de colunas agrupadas e linhas, mostra a quantidade de solicitações que foram criadas por mês usando a medida da figura 19 no valor da coluna e o tempo médio das arquivadas na linha em azul escuro.

- % de Solicitações por Semana: Este gráfico de pizza (% de solicitações por semana), que utiliza a coluna personalizada Período do Mês (Criação), comprova a hipótese de picos de demanda, mostrando a concentração de entradas nas últimas semanas do mês.
- Fase Atual Solicitações: Este gráfico de colunas (abaixo) mostra o volume de solicitações por status final, evidenciando o alto volume de "Concluído" versus "Arquivado". E também pode ser utilizado para visualizar a quantidade de solicitações que ainda não foram concluídas ou arquivadas de determinado mês ao filtrar somente um único mês.
- Motivo Solicitação Arquivada: Este gráfico de colunas responde diretamente ao Gargalo 1, mostrando por que as solicitações são arquivadas (ex: "Cliente desistiu", "Excedeu o prazo").

A segunda tela, "Análise Concluídas", vista na figura 25, foca no resultado final do processo: as solicitações concluídas e a eficiência dos parceiros (Gargalo 2).



Figura 25 - Tela "Análise Concluídas" Performance de Saída e Certificadoras.

Fonte: Elaborado pelo autor deste trabalho.

Esta tela avalia a eficiência da saída da operação:

- Solicitações Concluídas de Certificado por Mês: O gráfico de combinação colunas agrupadas e linhas, mostra o volume de entregas com a quantidade de

solicitações concluídas, e como o *Lead Time Total* (em dias corridos) da medida da figura 21 se comportou mês a mês.

- Tipo de Emissão: O gráfico de pizza detalha a porcentagem de solicitações que foram emitidas por videoconferência ou presencial ao longo do período selecionado na segmentação.
- Quantidade e Tempo Médio por Certificadora: Estes dois gráficos (colunas 100 % empilhadas e colunas clusterizado) são as principais ferramentas de gestão do Gargalo 2 (Dependência Externa). Eles permitem à gestão comparar o volume de entrega e, mais importante, o tempo médio de cada certificadora parceira (ex: Safeweb, DigitalSign), mostrando qual se demonstrou melhor em cada mês para os meses subsequentes manter maior quantidade de solicitações na certificadora que foi mais eficaz.

A terceira tela, "Indicadores", vista na figura 26, serve como um sumário executivo focado nos KPIs de performance e SLA (Acordo de Nível de Serviço).



Figura 26 - Tela "Indicadores" Foco em Performance e SLA.

Fonte: Elaborado pelo autor deste trabalho.

O objetivo desta tela é responder "Como estamos performando?". O KPI principal, "% Concluídos < 3 dias", é exibido com destaque (47,1%), utilizando as medidas DAX detalhadas na seção 3.3.3. Os gráficos de barras inferiores são cruciais, pois comparam o

"Tempo Médio (Dias Corridos)" com o "Tempo Médio (Dias Úteis)", este último calculado a partir da coluna SLA_Dias_Uteis, demonstrando o impacto real de fins de semana e feriados na operação.

A quarta tela, "Tempo Médio Por Etapa", vista na figura 27, foca em decompor o *Lead Time* para identificar onde estão os principais gargalos de tempo no fluxo do processo.



Figura 27 - Tela "Tempo Médio Por Etapa" Análise de Gargalos.

Fonte: Elaborado pelo autor deste trabalho.

Esta tela utiliza os cartões à esquerda para detalhar o tempo gasto em cada macro-etapa. Fica evidente que o maior gargalo operacional das solicitações concluídas está na etapa de agendamento demonstrado no "Tempo Médio - Agendamento (Emitidos)" 4,70 dias, que representa o tempo de espera da certificadora (Gargalo 2). Também é possível quantificar o custo do retrabalho: um processo Arquivado leva, em média, 10,60 dias, um impacto significativamente maior.

A quinta tela, "Tempo Médio Por Regional", vista na figura 28, foca na análise de desempenho segmentada por equipe operacional e consultor individual.

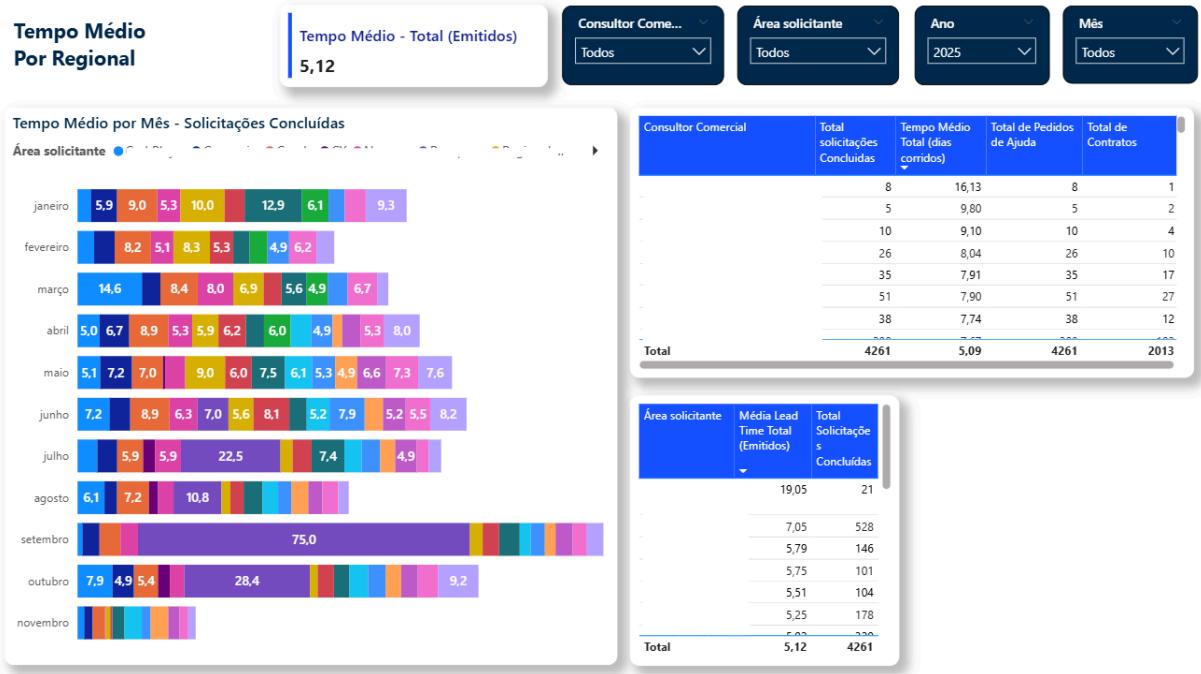


Figura 28 - Tela "Tempo Médio por Regional" Análise de Desempenho por Equipe Comercial.

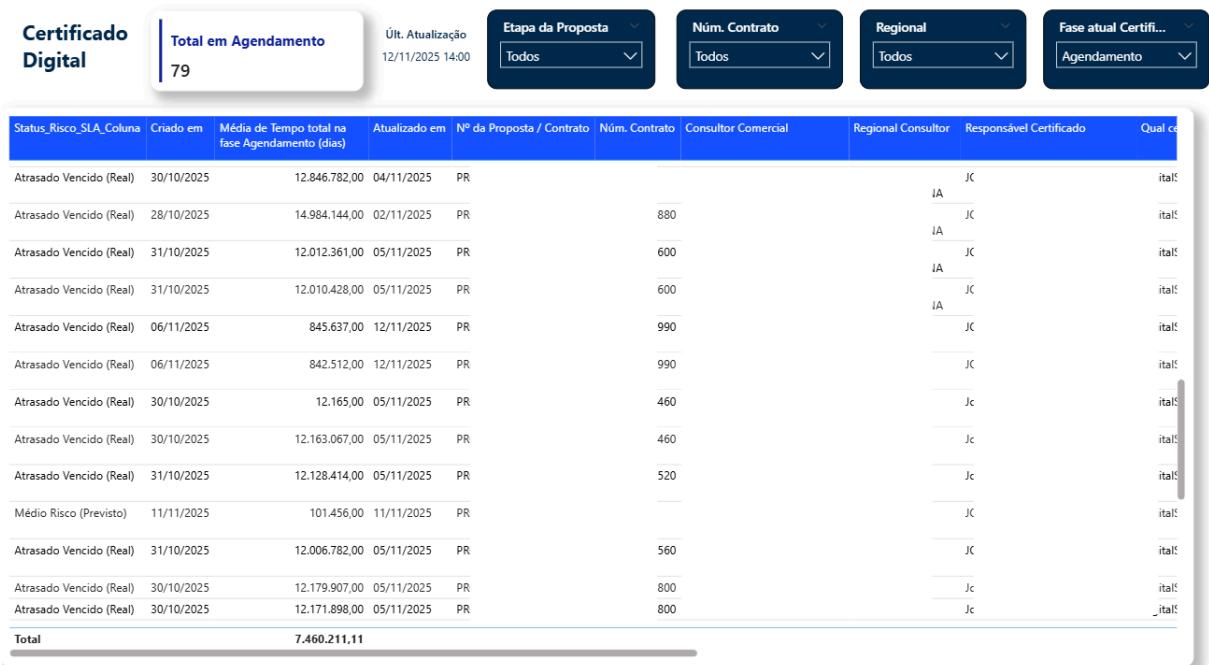
Fonte: Elaborado pelo autor deste trabalho.

Esta tela é essencial para a gestão tática das equipes comerciais, permitindo identificar disparidades de performance e necessidades de treinamento localizadas em equipes ou individual de cada consultor. Os visuais são projetados para detalhar o desempenho:

- **Tempo Médio por Mês - Solicitações Concluídas:** Este gráfico de barras empilhadas é o principal visual da tela. Ele decompõe o tempo médio de conclusão de cada mês pelas equipes comerciais, permitindo ao gestor ver imediatamente quais equipes (ex: "Recuperação de Crédito", "Condo", "Regional RJ") estão impactando a média geral.
- **Tabelas de Detalhamento Consultores e Área solicitante:** As tabelas à direita fornecem os dados brutos de performance, detalhados por Consultor Comercial e resumidos por Área solicitante. Elas permitem que a gestão compare não apenas o *Lead Time* Médio, mas também o volume de solicitações de cada equipe e indivíduo.

Esta visão é fundamental para entender se um tempo médio alto é um problema pontual de um consultor ou um desafio sistêmico de uma regional inteira, direcionando a ação gerencial de forma precisa.

Por fim, a tela "Certificado Digital", vista na figura 28, não é um *dashboard* analítico, mas sim um relatório operacional acionável.



The screenshot shows a table titled 'Certificado Digital' with various filters at the top. The filters include 'Total em Agendamento' (79), 'Últ. Atualização' (12/11/2025 14:00), 'Etapa da Proposta' (Todos), 'Nº da Proposta / Contrato' (Todos), 'Regional' (Todos), and 'Fase atual Certificado' (Agendamento). The table lists 15 rows of data, each containing information about a task: Status_Risco_SLA_Coluna, Criado em, Média de Tempo total na fase Agendamento (dias), Atualizado em, Nº da Proposta / Contrato, Núm. Contrato, Consultor Comercial, Regional Consultor, Responsável Certificado, and Qual certificadora. The last row shows a total of 7.460.211,11.

Status_Risco_SLA_Coluna	Criado em	Média de Tempo total na fase Agendamento (dias)	Atualizado em	Nº da Proposta / Contrato	Núm. Contrato	Consultor Comercial	Regional Consultor	Responsável Certificado	Qual certificadora
Atrasado Vencido (Real)	30/10/2025	12.846.782,00	04/11/2025	PR			JC	italf	
Atrasado Vencido (Real)	28/10/2025	14.984.144,00	02/11/2025	PR	880		IA	italf	
Atrasado Vencido (Real)	31/10/2025	12.012.361,00	05/11/2025	PR	600		IA	italf	
Atrasado Vencido (Real)	31/10/2025	12.010.428,00	05/11/2025	PR	600		IA	italf	
Atrasado Vencido (Real)	06/11/2025	845.637,00	12/11/2025	PR	990		JC	italf	
Atrasado Vencido (Real)	06/11/2025	842.512,00	12/11/2025	PR	990		JC	italf	
Atrasado Vencido (Real)	30/10/2025	12.165,00	05/11/2025	PR	460		Jc	italf	
Atrasado Vencido (Real)	30/10/2025	12.163.067,00	05/11/2025	PR	460		Jc	italf	
Atrasado Vencido (Real)	31/10/2025	12.128.414,00	05/11/2025	PR	520		Jc	italf	
Médio Risco (Previsto)	11/11/2025	101.456,00	11/11/2025	PR			JC	italf	
Atrasado Vencido (Real)	31/10/2025	12.006.782,00	05/11/2025	PR	560		JC	italf	
Atrasado Vencido (Real)	30/10/2025	12.179.907,00	05/11/2025	PR	800		Jc	italf	
Atrasado Vencido (Real)	30/10/2025	12.171.898,00	05/11/2025	PR	800		Jc	italf	
Total		7.460.211,11							

Figura 29 - Tela "Certificado Digital" - Visão Operacional Detalhada Fila de Trabalho.

Fonte: Elaborado pelo autor deste trabalho.

O propósito desta página é permitir a gestão diária da fila de trabalho. Enquanto as outras telas mostram médias e tendência com uma visão tática e estratégica, esta tela mostra o detalhe com uma visão operacional da fila. O gestor pode, por exemplo, usar o filtro "Fase atual Certificado" para selecionar "Agendamento" e ver, em tempo real, as solicitações que estão paradas, com qual responsável, há quantos dias, permitindo uma ação gerencial imediata. Esta página foi feita também pensando na parte da visualização das equipes comerciais, para terem visão de como a solicitação de cada cliente se encontra. Esta tela consta dados do *snowflake* como data de assinatura, etapa do contrato, etapa da proposta e informações da tabela fato de certificados como fase atual do certificado, qual certificadora e o número de protocolo do cliente na certificadora.

Em algumas das telas constam informações de dados que são protegidas pela LGPD, por isso foram censuradas e ocultadas com um quadrado com preenchimento na cor branca.

Com a finalização da camada de visualização, a solução de *Business Intelligence* está completa. O conjunto destas seis telas fornece uma visão 360° da operação desde a causa-raiz (Tela 1), passando pela performance de saída (Tela 2), os KPIs estratégicos (Tela 3), os gargalos de processo (Telas 4 e 5), até a ação operacional diária (Tela 6). Os dashboards

interativos e automatizados oferecem à gestão as ferramentas necessárias para monitorar a performance em tempo real, identificar causas-raiz e tomar decisões baseadas em dados, cumprindo os objetivos propostos para esta parte do projeto. A próxima etapa do trabalho detalha a implementação dos modelos de *Machine Learning*, que utilizam a base de dados tratada e estruturada da tabela fato nesta solução como seu principal insumo.

3.4 IMPLEMENTAÇÃO DO MODELO PREDITIVO DE MACHINE LEARNING.

Com a solução de *Business Intelligence* do capítulo anterior concluída, o projeto entrou em sua próxima fase: a criação de um modelo preditivo. O BI permitiu a análise diagnóstica e a *Machine Learning* tem como objetivo a análise preditiva. O objetivo é usar os dados históricos, já tratados e enriquecidos no Power BI, para prever quais novas solicitações têm alta probabilidade de atrasar o SLA, permitindo uma ação proativa da equipe operacional.

3.4.1 AMBIENTE, METODOLOGIA E PREPARAÇÃO DOS DADOS.

No momento de decidir onde construir o modelo preditivo, levou-se em consideração algumas opções. Embora plataformas de nuvem robustas, como o Azure Machine Learning (discutido na Fundamentação Teórica), fossem uma possibilidade, elas acabaram não sendo viáveis para este projeto por uma questão prática de custos para a empresa.

Por isso, optou-se por usar o Google Colab. Essa decisão foi natural por dois motivos: primeiro, porque já tinha-se familiaridade com a ferramenta por ter-se utilizado-a em uma disciplina da faculdade; segundo, por ser uma plataforma gratuita e que atende perfeitamente às necessidades do projeto. O Colab é um ambiente baseado em *notebooks Jupyter*, ideal para rodar o código Python e as bibliotecas de Ciência de Dados necessárias. Assim, todo o processo de desenvolvimento do modelo foi estruturado dentro dele da seguinte forma:

Importação das Bibliotecas: A figura 30 demonstra a importação das bibliotecas essenciais para o projeto, instalando-as e, em seguida, importando-as para o *script*.

```

▶ # --- 1º passo do TCC Configuração e Importação de Bibliotecas ---

# Instalação de bibliotecas
!pip install pandas scikit-learn joblib lightgbm openpyxl matplotlib seaborn

# Importação das bibliotecas
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, classification_report, roc_auc_score, confusion_matrix
import joblib

# Para visualização da matriz de confusão
import matplotlib.pyplot as plt
import seaborn as sns

# Para upload e leitura de .xlsx no Colab
from google.colab import files
import io
import os

```

Figura 30 - Código Python: Instalação e Importação das Bibliotecas.

Fonte: Elaborado pelo autor deste trabalho.

O código importa as principais ferramentas e bibliotecas de Ciência de Dados:

- Pandas: Para a manipulação e carregamento dos dados (do arquivo .xlsx).
- *Scikit-learn* (sklearn): A principal biblioteca de ML, usada para:
 - train_test_split*: Dividir os dados em treino e teste.
 - RandomForestClassifier*: O algoritmo de modelo escolhido.
 - Pipeline* e *ColumnTransformer*: Para criar o *pipeline* de pré-processamento.
 - OneHotEncoder*: Para tratar as colunas categóricas.
 - metrics*: Para avaliar o modelo (Acurácia, Matriz de Confusão, etc.).
- Matplotlib e Seaborn: Usadas para a visualização dos dados, especificamente a Matriz de Confusão.
- Joblib: Utilizada para salvar (exportar) o modelo treinado em um arquivo .pkl.

A segunda etapa prática do desenvolvimento em Python foi a extração dos dados do ambiente de BI e seu carregamento no Google Colab. Como o Power BI e o Colab não possuem uma conexão nativa direta, foi utilizado um método de exportação manual: o conjunto de dados tratado no Power BI foi exportado para um arquivo Excel (.xlsx) e, em seguida, carregado no Colab como mostra a figura 31.

```

▶ # --- 2º passo TCC Upload do XLSX e Carregamento dos Dados ---

print("Por favor, faça o upload do seu arquivo XLSX (treinoML.xlsx):")
uploaded = files.upload()

XLSX_FILE_NAME = 'treinoML.xlsx' # Ajuste este nome para o seu arquivo, se necessário

# Carregar os dados do arquivo XLSX
print(f"\nCarregando dados de '{XLSX_FILE_NAME}'...")
try:
    df = pd.read_excel(io.BytesIO(uploaded[XLSX_FILE_NAME]), engine='openpyxl')
    print("Dados carregados com sucesso!")
    print(f"Shape dos dados: {df.shape}")
    print("Primeiras 5 linhas:")
    print(df.head())
    print("\nInformações sobre as colunas:")
    df.info()
except KeyError:
    print(f"ERRO: O arquivo '{XLSX_FILE_NAME}' não foi encontrado entre os arquivos enviados.")
    print("Verifique se o nome do arquivo no código (XLSX_FILE_NAME) corresponde ao nome do arquivo que você fez upload.")
    exit()
except Exception as e:
    print(f"ERRO ao carregar o arquivo XLSX: {e}")
    exit(0)

```

Figura 31 - Código Python: Para Carregar o Arquivo .xlsx no Ambiente.

Fonte: Elaborado pelo autor deste trabalho.

Este *script* realiza três tarefas fundamentais:

- Extração (*Upload*): A função `files.upload()` (específica do Google Colab) abre uma interface na saída, para que o arquivo `treinoML.xlsx` seja enviado da máquina local do desenvolvedor para o ambiente de execução do *notebook*. O arquivo é armazenado temporariamente em um dicionário chamado `uploaded`.
- Carregamento (*Loading*): O código utiliza a biblioteca Pandas para ler o arquivo. Acessando o conteúdo binário do arquivo, transforma ele em um arquivo em memória que o Pandas consegue ler, e no final carrega os dados do arquivo em memória para um *dataframe* chamado `df`.
- Verificação (*Data Exploration*): Imediatamente após o carregamento, o script realiza a primeira verificação dos dados (análise exploratória) para garantir que a importação foi bem-sucedida. Confirma-se as dimensões dos dados, conforme a saída do *script*: 9.662 linhas (`df.shape`), e a saída imprimirá as primeiras 5 linhas de cada atributo para uma inspeção visual da estrutura (`df.head`). No final da saída são listadas todas as colunas no arquivo, constando os nomes, contagem de valores que não são nulos e o tipo de dados. Esta saída é a base para a próxima etapa de tratamento e definição das principais colunas que serão usadas para treinamento.

Definição da Variável-Alvo (*Target*): O objetivo do modelo é prever se uma solicitação irá atrasar (ultrapassar 3 dias de SLA). Portanto, a variável-alvo (Y) foi definida como a coluna SLA_Atrasado_3Dias. Esta é uma coluna binária (0 ou 1) que já havia sido calculada na camada de BI, onde '1' significa que o SLA foi violado e '0' que foi cumprido.

```
▶ # --- 3º Passo TCC  Definição de Features --

TARGET_COLUMN = 'SLA_Atrasado_3Dias'

NUMERICAL_FEATURES = [
    'Idade Cliente',
]

CATEGORICAL_FEATURES = [
    'Período do Mês',
    'Nome do Solicitante',
    'Cliente possui CNH?',
    'Área solicitante',
    'Valor do contrato',
    'Etapa da Proposta',
    'Período concluídas',
]

if TARGET_COLUMN not in df.columns:
    raise ValueError(f'A coluna alvo '{TARGET_COLUMN}' não foi encontrada no DataFrame. Verifique o nome da coluna no seu XLSX.')

df[TARGET_COLUMN] = df[TARGET_COLUMN].fillna(-1)
df[TARGET_COLUMN] = df[TARGET_COLUMN].astype(int)
```

Figura 32 - Código Python: Definição da Variável-Alvo e Listas de Features.

Fonte: Elaborado pelo autor deste trabalho.

A figura 32 ilustra o código de configuração inicial das variáveis dentro do ambiente de treinamento da ML.

- Declaração de Variáveis: Primeiramente, as variáveis TARGET_COLUMN (o que o modelo tentará prever), NUMERICAL_FEATURES (colunas numéricas) e CATEGORICAL_FEATURES (colunas de texto) são declaradas. Esta separação é um passo metodológico essencial, pois permite que o *pipeline* de pré-processamento (detalhado adiante) aplique regras de transformação distintas para cada tipo de dados.
- Verificação de Robustez: A instrução (if TARGET_COLUMN not in) atua como uma verificação de segurança, garantindo que a coluna alvo definida realmente exista no *DataFrame* antes de prosseguir.
- Tratamento da Variável-Alvo: O código realiza duas preparações críticas na coluna alvo. Primeiro preenche os valores nulos que representam solicitações ainda em aberto e, portanto, sem SLA definido com o valor -1 como valor temporário. Segundo, ele converte toda a coluna para o tipo inteiro. Esta preparação é fundamental para a etapa de filtragem que virá a seguir.

As *features* (atributos) listadas na figura 32 não foram escolhidas aleatoriamente. Cada uma foi selecionada com base em uma hipótese de negócio sobre sua capacidade de influenciar o risco de atraso no SLA. A tabela 6 detalha esta justificativa.

Feature (Atributo)	Tipo	Justificativa
Idade Do Cliente	Numérica	Hipótese: Clientes com idade mais avançada poderiam ter mais dificuldade no processo de certificação por videoconferência, aumentando o risco de atraso.
Período Do Mês	Categórica	Hipótese: Solicitações criadas nas últimas semanas do mês (devido a metas comerciais) podem ter maior risco de atraso devido ao pico de demanda sobrecarregando a equipe operacional.
Nome do Solicitante	Categórica	Hipótese: A performance individual do consultor (solicitante) é um forte indicador de risco. Consultores que historicamente geram mais retrabalho tendem a gerar solicitações com maior risco de atraso.
Cliente possui CNH?	Categórica	Hipótese: Clientes sem CNH (que exigem validação presencial ou outro método) têm um fluxo de processo diferente e potencialmente mais demorado, impactando o SLA.
Área solicitante	Categórica	Hipótese: Diferentes equipes regionais possuem processos de venda e coleta de documentos distintos, impactando diretamente a qualidade da solicitação e o risco de retrabalho/atrasto.
Valor do contrato	Categórica	Hipótese: O valor ou o tipo de contrato poderia estar associado a um perfil de cliente ou a uma complexidade de processo diferente.
Etapa da Proposta	Categórica	Hipótese: O status da proposta no momento da criação da solicitação de certificado pode indicar uma prioridade ou complexidade diferente do cliente.
Período concluídas	Categórica	Hipótese: Embora usada para análise, esta feature (se usada no treino) poderia indicar se o fluxo de saída (conclusões) de um período impacta o risco dos novos pedidos.

Tabela 6 - Justificativa da Escolha das Features do Modelo de ML

Fonte: Elaborado pelo autor deste trabalho.

```

▶ initial_rows = df.shape[0]
df_train = df[df[TARGET_COLUMN] != -1].copy()
removed_rows = initial_rows - df_train.shape[0]
print(f"\nRemovidas {removed_rows} linhas do treinamento (tickets em aberto ou com dados incompletos para o alvo.)")

if df_train.empty:
    raise ValueError("Não há dados suficientes para treinar o modelo após remover linhas com alvo nulo.")

X = df_train[NUMERICAL_FEATURES + CATEGORICAL_FEATURES].copy()
y = df_train[TARGET_COLUMN].copy()

print("\nVerificando e tratando valores nulos nas FEATURES para o TREINAMENTO:")
for col in NUMERICAL_FEATURES:
    if X[col].isnull().any():
        median_val = X[col].median()
        X[col] = X[col].fillna(median_val)
        print(f" Preenchidos nulos na coluna numérica '{col}' com a mediana {{median_val:.2f}}.")

for col in CATEGORICAL_FEATURES:
    if X[col].isnull().any():
        X[col] = X[col].fillna('Desconhecido')
        print(f" Preenchidos nulos na coluna categórica '{col}' com 'Desconhecido'.")

print("\nStatus de nulos nas FEATURES após tratamento (espera-se 0 nulos):")
print(X.isnull().sum())
if X.isnull().sum().sum() > 0:
    print("AVISO: Ainda existem valores nulos nas features após o tratamento. Verifique os dados.")

```

Figura 33 - Código Python: Filtragem, Separação (X/y) e Imputação de Nulos.

Fonte: Elaborado pelo autor deste trabalho.

A figura 33 demonstra a continuação do pré-processamento, com quatro etapas lógicas:

- Filtragem de Dados para Treinamento: O primeiro passo é criar o DataFrame de treinamento (df_train). O código filtra (df[df[TARGET_COLUMN] != -1]) o DataFrame original, selecionando apenas as linhas onde a coluna alvo não é -1. Isso efetivamente remove todas as solicitações em aberto que tinham NaN e receberam o placeholder -1, garantindo que o modelo aprenda apenas com dados históricos cujo resultado atrasou ou não, já é conhecido. Conforme o output do script, 1.017 linhas foram removidas para este fim, sobrando no restante 8.645 Linhas.
- Separação de Features (X) e Alvo (Y): A partir deste *DataFrame* filtrado, os dados são formalmente separados em duas matrizes: X, que contém apenas as colunas de features os atributos usados para prever ('Idade Cliente', 'Período do Mês', 'Nome do Solicitante', 'Cliente possui CNH?', 'Área solicitante', 'Valor

do contrato', 'Etapa da Proposta', 'Período concluidas'), e Y, que contém apenas a coluna alvo o resultado a ser previsto ('SLA_Atrasado_3Dias').

- Tratamento de Nulos nas Features (Imputação): O código então trata os valores ausentes apenas na matriz X, pois o algoritmo de ML não pode processar valores nulos. A estratégia de imputação foi:
 - Numéricas: Para colunas como Idade Cliente, os valores nulos foram preenchidos com a mediana da coluna. A mediana é usada em vez da média por ser mais robusta a valores extremos.
 - Categóricas: Para colunas como Cliente possui CNH?, os valores nulos foram preenchidos com a string "Desconhecido". Isso cria, na prática, uma nova categoria que o modelo pode aprender a interpretar (ex: "o fato da CNH ser 'Desconhecida' aumenta o risco de atraso?").
- Verificação Final: O comando X.isnull().sum() é uma verificação de sanidade para confirmar que todas as colunas em X agora possuem zero valores nulos, estando prontas para a próxima etapa do pipeline.

3.4.2 DEFINIÇÃO DO PIPELINE DE MODELAGEM

Construção do *Pipeline* e Treinamento do Modelo: Para garantir a reproduzibilidade e a correta aplicação das transformações de dados, foi construído um *Pipeline* do Scikit-learn. Esta é uma prática recomendada que encapsula todo o processo de pré-processamento e modelagem.



```
# --- 4º Passo TCC Criação do Pipeline de Pré-processamento e Modelo ---

preprocessor = ColumnTransformer(
    transformers=[
        ('num', 'passthrough', NUMERICAL_FEATURES),
        ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), CATEGORICAL_FEATURES)
    ],
    remainder='drop'
)

classifier = RandomForestClassifier(n_estimators=100, random_state=42, class_weight='balanced')

model_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', classifier)
])

print("\nPipeline de ML criado com sucesso!")
print(model_pipeline)
```

Figura 34 - Código Python: Construção do Pipeline de Pré-processamento e Modelagem.

Fonte: Elaborado pelo autor deste trabalho.

Este código constrói a arquitetura do modelo de *Machine Learning*.

- Pré-processador (*preprocessor*): Define as regras de transformação dos dados. Ele foi configurado para deixar as colunas numéricas (Idade Cliente) passarem direto (*passthrough*). e converter as colunas de texto (Período do Mês, Área solicitante, etc.) em um formato numérico que o modelo entende, usando a técnica de (*OneHotEncoder*)
- Classificador (*Classifier*): Define o algoritmo que fará as previsões. Foi escolhido o *RandomForestClassifier* (Floresta Aleatória), um modelo robusto, configurado com 100 "árvores" e o parâmetro `class_weight='balanced'` para lidar com possíveis desbalanceamentos entre classes (atrasados vs. não atrasados).
- *Pipeline* do modelo (*model_pipeline*): É o passo final e mais importante. Ele "encapsula" os dois passos anteriores em um único objeto. Isso garante que, ao treinar ou usar o modelo, os dados sempre passem pelo pré-processamento correto antes de chegarem ao classificador, tornando o processo seguro e reproduzível.



```
Pipeline de ML criado com sucesso!
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num', 'passthrough',
                                                   ['Idade Cliente']),
                                                   ('cat', OneHotEncoder(handle_unknown='ignore',
                                                       sparse_output=False),
                                                   ['Período do Mês',
                                                    'Nome do Solicitante',
                                                    'Cliente possui CNH?',
                                                    'Área solicitante',
                                                    'Valor do contrato',
                                                    'Etapa da Proposta',
                                                    'Período concluidas'])])),
              ('classifier',
               RandomForestClassifier(class_weight='balanced',
                                      random_state=42))])
```

Figura 35 - Representação Textual da Arquitetura do Pipeline

Fonte: Elaborado pelo autor deste trabalho.

A saída do console é a confirmação visual de que o que foi solicitado no código da figura 34 foi criado corretamente. Ela funciona como um "recibo" ou um registro textual da arquitetura do *pipeline*, mostrando:

- O passo '*preprocessor*', que contém o *ColumnTransformer* com as *features* exatas que foram usadas (ex: 'Idade Cliente', 'Período do Mês', etc.).
- O passo '*classifier*', que confirma o uso do *RandomForestClassifier* e seus parâmetros definidos (*class_weight='balanced'*, *random_state=42*).

Esta saída é a garantia de que o *pipeline* foi montado conforme planejado, estando pronto para a próxima etapa de divisão e treinamento dos dados.

3.4.3 TREINAMENTO E AVALIAÇÃO DOS RESULTADOS.

Após a definição e construção do pipeline detalhado em 3.4.2, o próximo passo foi a execução do treinamento e a subsequente avaliação de performance do modelo

Divisão e Treinamento dos Dados: Primeiramente, o conjunto de dados (X e Y, com 8.645 amostras) foi dividido em dois subconjuntos, conforme o código na figura 36.

```
▶ # ---5º passo do TCC: Divisão dos Dados e Treinamento do Pipeline ---

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
print(f"\nDados divididos: Treino={X_train.shape}, Teste={X_test.shape}")

print("\nIniciando treinamento do modelo...")
model_pipeline.fit(X_train, y_train)
print("Treinamento concluído!")
```

→ Dados divididos: Treino=(6916, 8), Teste=(1729, 8)
 Iniciando treinamento do modelo...
 Treinamento concluído!

Figura 36 - Código Python: Divisão e Treinamento do Modelo.

Fonte: Elaborado pelo autor deste trabalho.

O código utiliza a função *train_test_split* do Scikit-learn para dividir os dados:

- *test_size = 0.2*: Define que 20% dos dados (1.729 amostras) serão separados para o conjunto de Teste (X_test, y_test), que o modelo não verá durante o treino. Os 80% restantes (6.916 amostras) formam o conjunto de Treinamento (X_train, y_train).
- *stratify=y*: Garante que a proporção de "Atrasados" (1) e "Não Atrasados" (0) seja a mesma tanto no conjunto de treino quanto no de teste.

- `model_pipeline.fit(X_train, y_train)`: Este é o comando que efetivamente executa o treinamento. Ele alimenta o pipeline com os dados de treino, que são pré-processados e usados pelo *RandomForestClassifier* para aprender os padrões.

Avaliação de Performance no Conjunto de Teste: Após o treinamento, o `model_pipeline` treinado foi usado para fazer previsões no conjunto de Teste (os 20% de dados que ele nunca viu). A Figura 37 mostra o código usado para gerar as métricas de performance.

▶ # --- 6º passo TCC: Avaliação do Modelo ---

```
print("\nAvaliação do Modelo no Conjunto de Teste:")
y_pred = model_pipeline.predict(X_test)
y_proba = model_pipeline.predict_proba(X_test)[:, 1]

print(f"Acurácia: {accuracy_score(y_test, y_pred):.4f}")
print(f"AUC (Area Under the Curve): {roc_auc_score(y_test, y_proba):.4f}")
print("\nRelatório de Classificação:\n", classification_report(y_test, y_pred))
```



Avaliação do Modelo no Conjunto de Teste:

Acurácia: 0.8473

AUC (Area Under the Curve): 0.9297

Relatório de Classificação:

	precision	recall	f1-score	support
0	0.85	0.83	0.84	827
1	0.85	0.87	0.86	902
accuracy			0.85	1729
macro avg	0.85	0.85	0.85	1729
weighted avg	0.85	0.85	0.85	1729

Figura 37 - Código Python: Avaliação do Modelo e Saída do Código.

Fonte: Elaborado pelo autor deste trabalho.

Este script funciona como uma avaliação do modelo, comparando as previsões que ele fez com o gabarito real para ver o quanto bem ele se saiu. Primeiro, o script gera a previsão final se o modelo acha que vai atrasar ou não, e também o nível de confiança dessa previsão. Em seguida, ele calcula a acurácia do modelo: o *accuracy_score* é a nota geral a porcentagem total de acertos. Por fim, o *classification_report* gera um boletim detalhado mostrando onde ele é bom e onde ele é ruim e a *confusion_matrix* mostra o placar exato de quantos acertos e erros o modelo teve.

A Figura 38 apresenta a Matriz de Confusão, a ferramenta visual mais importante para avaliar a performance do classificador no conjunto de testes. Ela compara os valores reais (o que de fato aconteceu) com as previsões do modelo (o que o modelo disse que aconteceria).

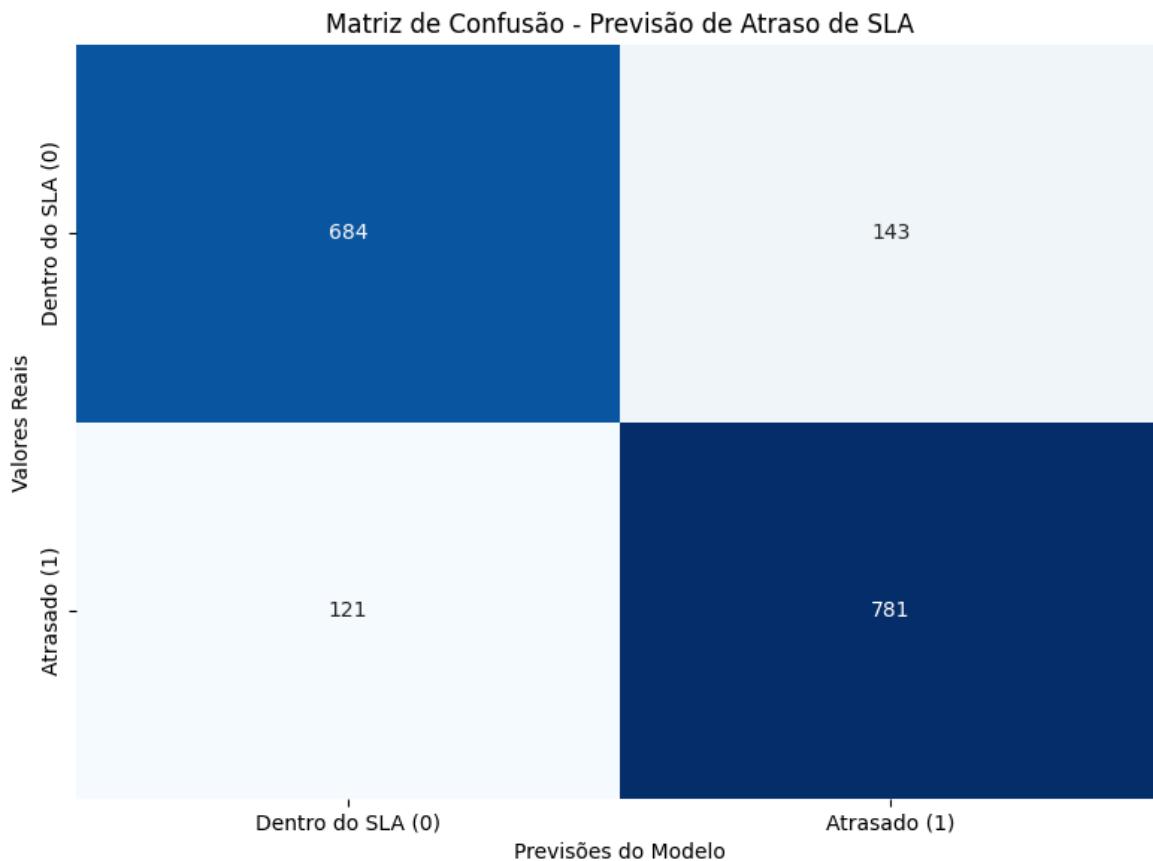


Figura 38 - Matriz de Confusão dos Resultados de Teste.

Fonte: Elaborado pelo autor deste trabalho.

A análise dos quatro quadrantes da matriz revela o seguinte:

- Verdadeiros Negativos - 684 casos: No canto superior esquerdo, temos 684 acertos. O modelo previu "Dentro do SLA" (0) e as solicitações realmente ficaram "Dentro do SLA" (0).
- Verdadeiros Positivos - 781 casos: No canto inferior direito, temos 781 acertos. O modelo previu Atrasado (1) e as solicitações realmente atrasaram (1).
- Falsos Positivos - 143 casos: No canto superior direito, temos 143 alarmes falsos. O modelo previu atrasado (1), mas as solicitações terminaram dentro do SLA (0). Este é um erro de menor impacto, no máximo geraria uma priorização desnecessária da equipe operacional.

- Falsos Negativos - 121 casos: No canto inferior esquerdo, temos os 121 erros mais críticos para o negócio. O modelo previu dentro do SLA (0), mas as solicitações efetivamente atrasaram (1). Estes são os casos que escaparam do radar, que não seriam priorizados e necessitariam de prioridade.

A matriz demonstra a alta eficácia do modelo. Ele acertou em um total de 1.465 casos de um total de 1.729. Mais importante, o número de erros críticos, que são o principal foco do projeto, foi baixo somente 121 casos, comprovando a viabilidade do modelo para priorizar proativamente a fila de trabalho.

3.4.4 ANÁLISE AVANÇADA DE PERFORMANCE DO MODELO

Para uma avaliação mais profunda da capacidade do modelo, para além da Matriz de Confusão, foram geradas as curvas de performance ROC e Precision-Recall.

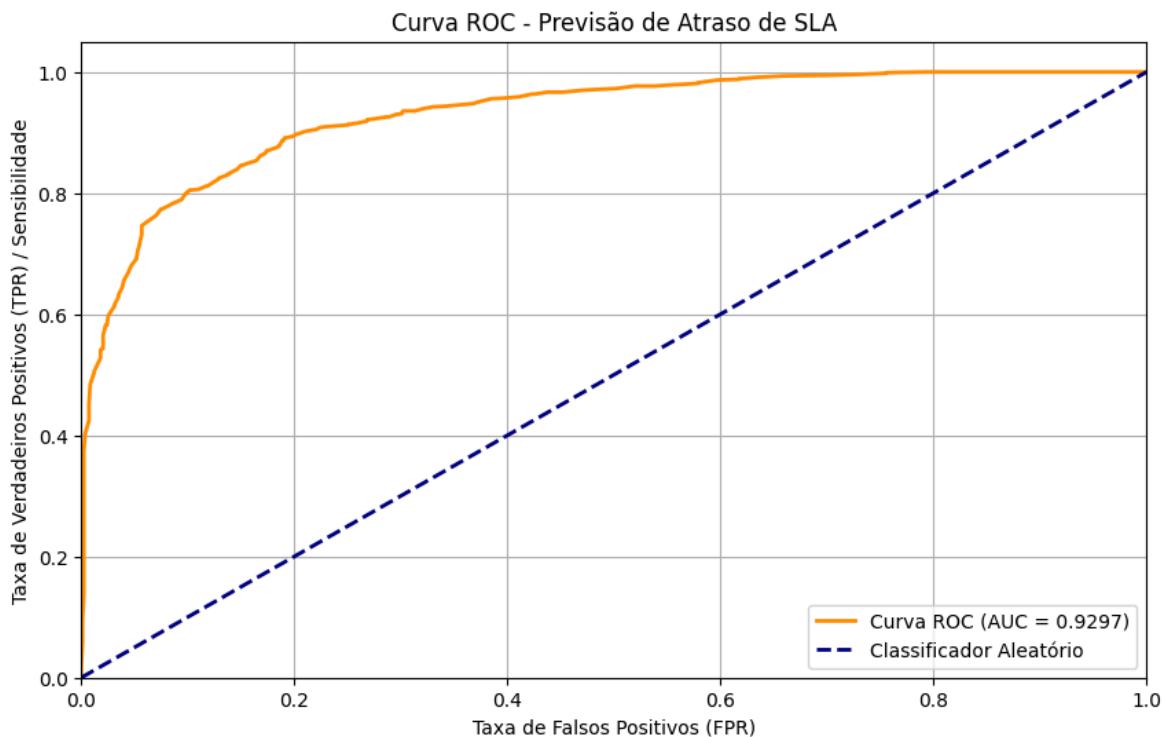


Figura 39 - Curva ROC e Métrica AUC.

Fonte: Elaborado pelo autor deste trabalho.

O resultado mais importante deste gráfico é a AUC (Área sob a Curva), que foi de 0.9297 (ou 93%). Um valor tão próximo de 1.0 indica que o modelo tem um poder discriminatório excelente. Ele é muito superior ao Classificador Aleatório (a linha tracejada),

provando que o modelo aprendeu os padrões de forma eficaz. Quanto mais a curva se afasta da linha tracejada, melhor o modelo.

Enquanto a ROC é uma boa métrica geral, a Curva Precision-Recall, na Figura 40, é especialmente valiosa para este projeto, pois lida bem com dados onde uma classe pode ser mais importante que a outra.

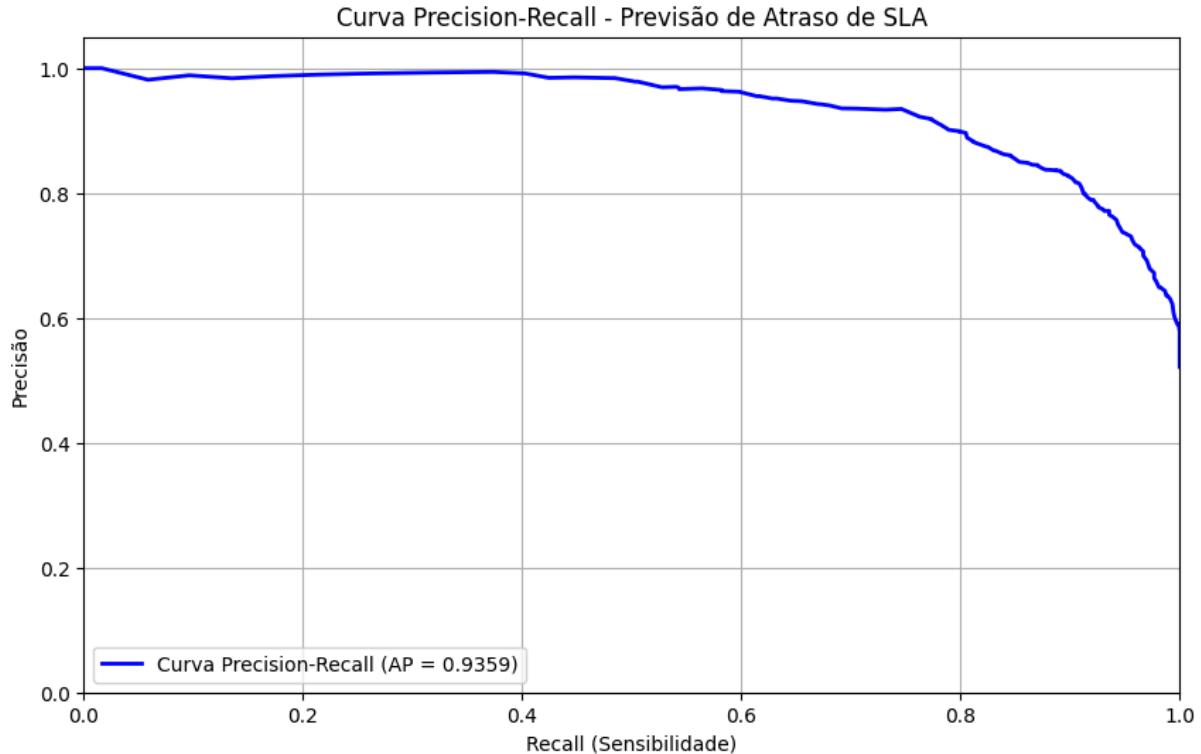


Figura 40 - Curva Precision-Recall e Métrica AP

Fonte: Elaborado pelo autor deste trabalho.

A Média de Precisão, que é a área sob esta curva, foi de 0.9359 (ou 94%). Este valor, também muito próximo de 1.0, demonstra que o modelo consegue manter alta precisão e não gerar muitos alarmes falsos, enquanto também atinge alto Recall encontrando a maioria dos atrasos reais.

Por padrão, um modelo classifica como atrasado (1) se a probabilidade for > 0.5 (50%). O gráfico da Figura 41 investiga se este é o melhor ponto de corte.

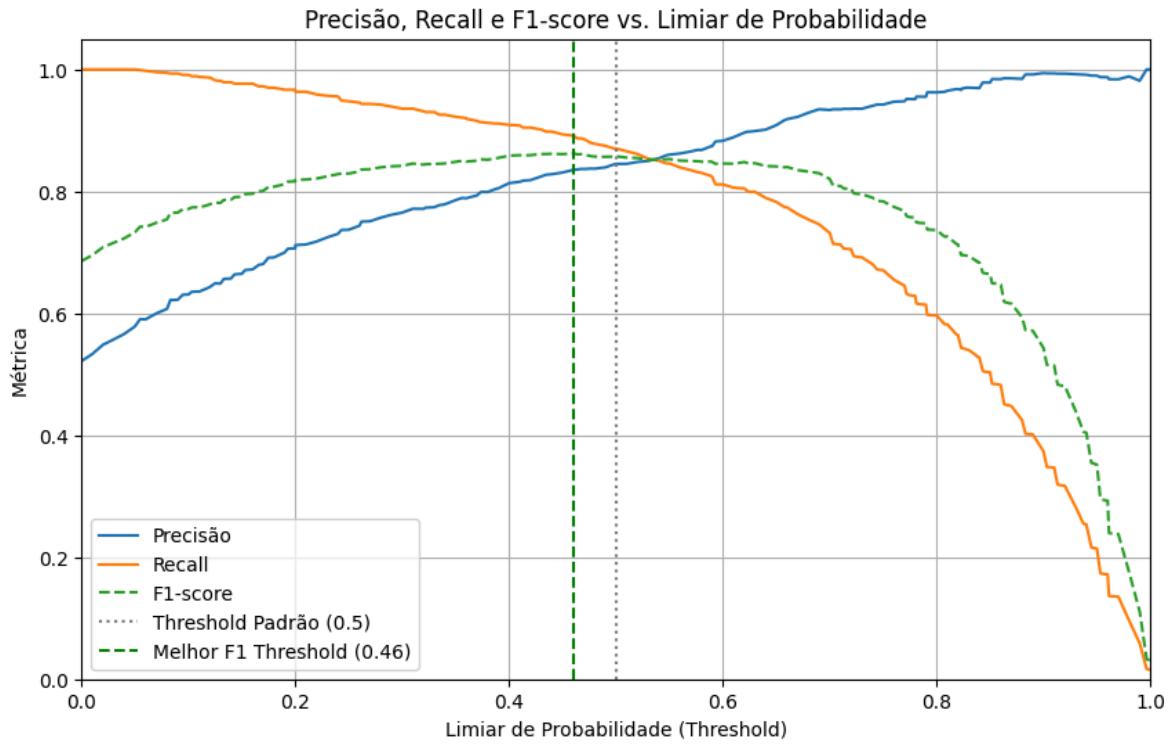


Figura 41 - Análise de Métricas vs. Ponto de Corte.

Fonte: Elaborado pelo autor deste trabalho.

Este gráfico mostra que:

- Aumentar o ponto de corte (ex: 0.8) aumenta a Precisão (linha azul), mas derruba o Recall (linha laranja).
- Diminuir o limiar (ex: 0.2) aumenta o Recall (encontra todos os atrasos), mas derruba a Precisão (gera muitos alarmes falsos).

O F1-Score (linha verde) é a métrica que busca o equilíbrio ideal entre os dois. Conforme indicado, o ponto máximo do F1-Score (0.86) não ocorre no limiar padrão (0.5), mas sim no limiar de 0.46. Isso indica que, para otimizar o equilíbrio do modelo, o ponto de corte ideal seria 46% de probabilidade, e não 50%.

Com a conclusão desta análise estatística, que abrangeu desde a Acurácia e a Matriz de Confusão até as curvas ROC e Precision-Recall, o modelo de *Machine Learning* está estatisticamente validado. Os resultados, especialmente a alta AUC (0.9297) e a AP (0.9359), comprovam que o classificador possui um excelente poder discriminatório e é significativamente eficaz em prever quais solicitações irão atrasar. A análise detalhada dos erros e a identificação do ponto de corte ótimo (0.46) fornecem a confiança necessária em sua

performance. A próxima e última etapa deste capítulo é, portanto, a exportação deste modelo treinado para que ele possa ser operacionalizado e suas previsões retroalimentadas na solução de *Business Intelligence*.

3.4.5 IMPLANTAÇÃO E RETROALIMENTAÇÃO.

Após o treinamento e a avaliação bem-sucedida, a etapa final do processo no Google Colab foi a exportação do pipeline treinado.

O objeto `model_pipeline`, que contém toda a inteligência o pré-processador e o classificador *RandomForest* treinado, foi salvo em um único arquivo binário. A figura 42 demonstra o código utilizado para salvar e baixar este artefato do modelo.

```
▶ # --- 7º Passo para o TCC Salvar o Pipeline Treinado e Baixar para a Máquina Local ---

# Nome do arquivo do modelo a ser salvo no ambiente Colab (e depois baixado)
MODEL_SAVE_NAME = 'ML_certificado_V2.pkl'

# Salvar o pipeline completo
joblib.dump(model_pipeline, MODEL_SAVE_NAME)
print(f"\nPipeline de ML salvo com sucesso no ambiente Colab como: {MODEL_SAVE_NAME}")

# Baixar o arquivo do modelo para sua máquina local
print(f"Fazendo download de '{MODEL_SAVE_NAME}' para sua máquina local...")
files.download(MODEL_SAVE_NAME)
print("Download concluído!")

...
Pipeline de ML salvo com sucesso no ambiente Colab como: ML_certificado_V2.pkl
Fazendo download de 'ML_certificado_V2.pkl' para sua máquina local...
Download concluído!
```

Figura 42 - Código Python: Salvando o Modelo Treinado para Implantação.

Fonte: Elaborado pelo autor deste trabalho.

O código utiliza a função `joblib.dump` para serializar e salvar o `model_pipeline` completo no arquivo `ML_certificado_V2.pkl`. Este arquivo `.pkl` é o produto final do desenvolvimento de *Machine Learning*. Ele pode ser carregado em qualquer outro ambiente Python para fazer previsões em dados novos, sem a necessidade de retreinar o modelo. A função `files.download` do Google Colab foi usada para baixar este artefato.

Ao contrário de um processo externo via banco de dados, a integração do modelo foi feita nativamente dentro do Power Query no Power BI. Foi utilizada a funcionalidade "Executar script do Python" para carregar o modelo `.pkl` e aplicar as previsões durante a própria atualização dos dados.

O processo funciona da seguinte forma:

- O arquivo ML_certificado_V2.pkl foi salvo em um local fixo, acessível pela máquina que realiza a atualização do Power BI (ex: no computador local ou em um servidor de rede).
- Dentro do Power Query, na consulta da FAT_CERTIFICADOS, foi adicionada uma etapa "Executar script do Python".
- Esta etapa executa um script que carrega o modelo .pkl e aplica as previsões nos dados da tabela. A figura 43 e a figura 44 ilustram o conceito do código Python que roda dentro do Power BI.

```

1 #Script Python no Power Query para previsão da ML
2 import pandas as pd
3 import joblib # Para carregar o pipeline salvo
4 import os # Para lidar com caminhos de arquivo
5 # --- 1. Configurações (AJUSTE AQUI!) ---
6 # Caminho COMPLETO para o seu arquivo .pkl salvo.
7 # Use o prefixo 'r' para strings brutas ou barras normais '/' para evitar problemas com caminhos do Windows.
8 MODEL_PATH = r"coloque o caminho do diretório seu modelo aqui" # AJUSTE ESTE CAMINHO DO SEU MODELO .PKL
9 # Estas listas de colunas DEVEM ser EXATAMENTE as mesmas que você usou no script de TREINAMENTO!
10 NUMERICAL_FEATURES = [
11     'Idade Cliente',
12 ]
13 CATEGORICAL_FEATURES = [
14     'Período do Mês',
15     'Nome do Solicitante',
16     'Cliente possui CNH?',
17     'Área solicitante',
18     'Valor do contrato',
19     'Etapa da Proposta',
20     'Período concluídas',
21 ]
22 # --- LISTA DE COLUNAS DE DATA QUE JÁ CHEGAM NO PYTHON EM FORMATO DD/MM/YYYY (Texto) ---
23 DD_MM_YYYY_INPUT_COLUMNS = [
24     'Primeira vez que entrou na fase Concluído',
25 ]
26 # --- LISTA DE OUTRAS COLUNAS DE DATA QUE CHEGAM EM OUTROS FORMATOS (ou como Python datetime) ---
27 # São as colunas de data que o Power Query envia ao Python como tipo 'Date'/'DateTime'
28 OTHER_DATE_COLUMNS_TO_PROCESS = [
29     'Data de Nascimento',
30     'Criado em',
31 ]
32 # Data padrão para preencher NaT no Python antes de formatar para string.
33 # Isso garante que o Power Query não receba um 'None' ou 'NaT' que vira null,
34 DEFAULT_DATE_PLACEHOLDER_PYTHON = pd.Timestamp('1900-01-01')
35 # --- 2. Carregar o Pipeline Completo ---
36 if not os.path.exists(MODEL_PATH):
37     raise FileNotFoundError(f"ERRO: O arquivo do modelo '{MODEL_PATH}' não foi encontrado. Verifique o caminho e se o arquivo foi salvo.")
38 model_pipeline = joblib.load(MODEL_PATH)
39 # --- 3. Preparar Features do Dataset Atual ---
40 all_features_expected = NUMERICAL_FEATURES + CATEGORICAL_FEATURES

```

Figura 43 - 1º Parte do Script Python executado no Power Query para ML.

Fonte: Elaborado pelo autor deste trabalho.

```

41 # Verifica se as features esperadas existem no dataset do Power BI
42 missing_features_in_dataset = [f for f in all_features_expected if f not in dataset.columns]
43 if missing_features_in_dataset:
44     raise ValueError(f"As seguintes features esperadas pelo modelo não foram encontradas nos dados do Power BI: {missing_features_in_dataset}.")
45 df_features = dataset[all_features_expected].copy()
46 # --- 4. Tratar Valores Nulos nas FEATURES (Mesma lógica do treinamento) ---
47 # É crucial que o tratamento de nulos nos dados de previsão seja IGUAL ao do treinamento.
48 for col in NUMERICAL_FEATURES:
49     if df_features[col].isnull().any():
50         median_val = df_features[col].median()
51         df_features[col] = df_features[col].fillna(median_val)
52 for col in CATEGORICAL_FEATURES:
53     if df_features[col].isnull().any():
54         df_features[col] = df_features[col].fillna('Desconhecido')
55 # --- 5. Fazer Previsões Usando o Pipeline ---
56 dataset['SLA_Previsto'] = model_pipeline.predict(df_features)
57 dataset['Probabilidade_Atraso'] = model_pipeline.predict_proba(df_features)[:, 1]
58 # --- 6. TRATAMENTO CRUCIAL: CONVERTER COLUNAS DE DATA PARA TEXTO YYYY-MM-DD ANTES DE RETORNAR ---
59 # 6.1. Processar colunas que chegam em formato DD/MM/YYYY (texto)
60 for col in DD_MM_YYYY_INPUT_COLUMNS:
61     if col in dataset.columns:
62         converted_to_datetime = pd.to_datetime(dataset[col], format='%d/%m/%Y', errors='coerce')
63         filled_datetime = converted_to_datetime.fillna(DEFAULT_DATE_PLACEHOLDER_PYTHON)
64         dataset[col] = filled_datetime.dt.strftime('%Y-%m-%d')
65 # 6.2. Processar outras colunas de data (que chegam como datetime ou outro texto, para YYYY-MM-DD)
66 for col in OTHER_DATE_COLUMNS_TO_PROCESS:
67     if col in dataset.columns:
68         converted_to_datetime = pd.to_datetime(dataset[col], errors='coerce')
69         filled_datetime = converted_to_datetime.fillna(DEFAULT_DATE_PLACEHOLDER_PYTHON)
70         dataset[col] = filled_datetime.dt.strftime('%Y-%m-%d')
71 # --- 7. Retornar o DataFrame atualizado ---
72 Output = dataset

```

Figura 44 - 2º Parte do Script Python executado no Power Query para ML.

Fonte: Elaborado pelo autor deste trabalho.

A figura 43 e a figura 44 detalham o script Python que é executado nativamente dentro do Power Query a cada atualização, para operacionalizar o modelo de *Machine Learning*. O processo é executado em etapas lógicas:

- **Configuração e Carregamento:** Primeiramente, o script define os parâmetros essenciais: o caminho MODEL_PATH para localizar o arquivo .pkl salvo e as listas exatas de colunas numéricas e categóricas que o modelo espera receber. Em seguida, ele carrega o modelo .pkl na memória usando joblib.load.
- **Pré-processamento Idêntico:** O script recebe a tabela de dados (dataset) do Power Query e aplica o mesmo tratamento de valores nulos que foi usado no treinamento preenchendo números com a mediana e textos com "Desconhecido". Esta etapa é vital para garantir a precisão das previsões.
- **Geração das Previsões:** Com os dados preparados, o script utiliza o model_pipeline carregado para gerar duas novas colunas: SLA_Previsto sendo a previsão final de 0 ou 1 e Probabilidade_Atraso a porcentagem de risco.
- **Tratamento de Saída:** Por fim, o script realiza um tratamento de colunas de data, convertendo-as para um formato de texto padrão. Este é um passo técnico necessário para garantir que os dados retornem do Python para o Power Query sem erros.

O Output final deste script é a própria tabela de dados (dataset), agora enriquecida com as previsões de risco, pronta para ser carregada no modelo de dados e utilizada no *dashboard* de Gestão de Risco.

3.4.6 DASHBOARD DE GESTÃO PROATIVA DE RISCO (ML)

A etapa final do Capítulo 3 consistiu em unir as duas soluções (*Business Intelligence* e *Machine Learning*) em uma tela final focada em gestão proativa. Após o script Python (detalhado em 3.4.4) enriquecer a tabela FAT_CERTIFICADOS com as colunas SLA_Previsto e Probabilidade_Atraso.

O objetivo deste painel não é analisar o passado (como as outras seis telas de BI), mas sim fornecer uma visão preditiva do futuro, permitindo que os gestores e a equipe operacional ajam antes que o SLA seja violado.

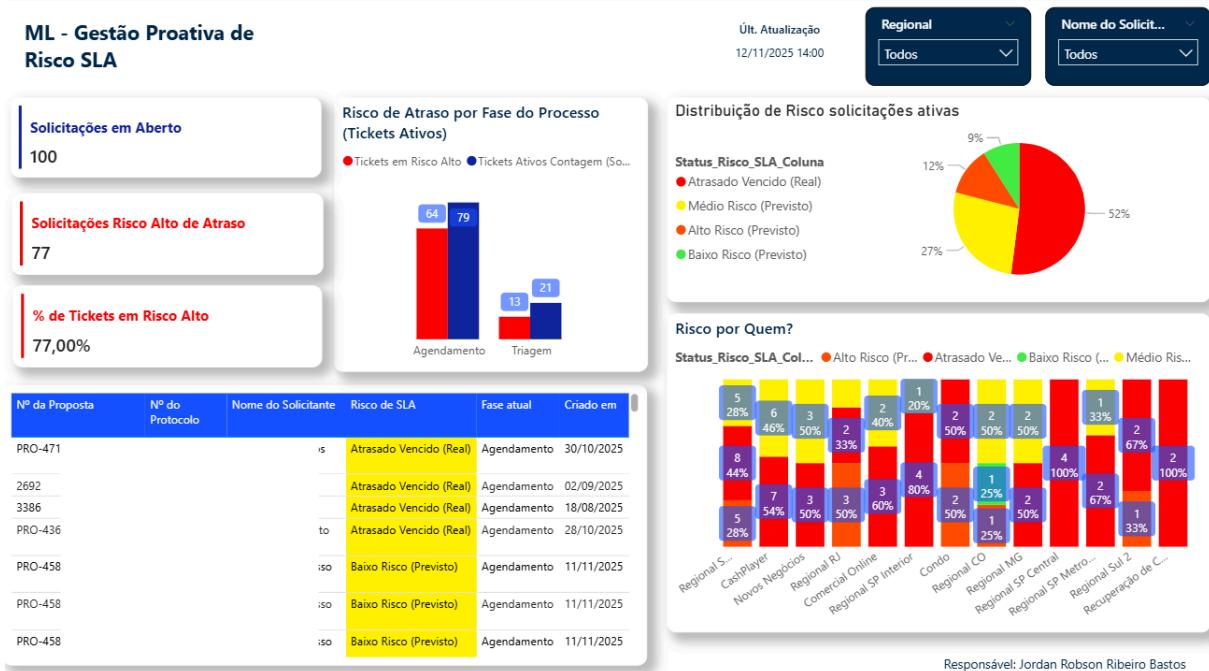


Figura 45 - Dashboard "ML - Gestão Proativa de Risco SLA".

Fonte: Elaborado pelo autor deste trabalho.

A Figura 45 detalha a tela de "Gestão Proativa", que é estruturada para responder a quatro perguntas críticas de gerenciamento de risco:

- Qual é o tamanho do risco?: Os cartões à esquerda fornecem um sumário executivo imediato. De um total de 100 solicitações em aberto, o modelo prevê que 77 (ou 77%) estão em "Risco Alto de Atraso". Isso quantifica o tamanho do problema para o gestor e para a equipe.

- Como o risco está distribuído?: O gráfico Pizza "Distribuição de Risco solicitações ativas" é crucial, pois ele combina dados reais com dados preditivos. Ele mostra que: 12% da fila está Alto Risco (Previsto) 52% atrasado vencido (real). Somados, o gestor sabe que 64% de toda a fila requer intervenção imediata.
- Quem é o responsável pelo risco?: O gráfico "Risco por Quem?" conecta a previsão de ML com a análise de BI. Ele segmenta a distribuição de risco por Regional (Área Solicitante), respondendo à pergunta: "Quais equipes estão com as solicitações de maior risco em sua fila?". Isso permite uma alocação de recursos mais inteligente e uma análise para uma tomada de decisão mais embasada nas informações geradas pelos dados.
- Onde devo agir agora?: A tabela na parte inferior é a ferramenta de ação do dashboard. Ela lista individualmente cada solicitação em aberto, mas está classificada pela nova coluna calculada Risco de SLA (a previsão do modelo) que vou detalhar abaixo. O gestor pode ver exatamente quais propostas estão com "Baixo Risco" e quais já estão "Atrasado Vencido", permitindo uma priorização cirúrgica da fila de trabalho.

3.4.7 DETALHAMENTO DAS MEDIDAS E DAS COLUNAS CALCULADAS PREDITIVAS

Para que o *dashboard* "ML - Gestão Proativa de Riscos SLA" Figura 45 funcionasse, foi necessário criar um conjunto final de colunas e medidas em DAX. O objetivo desses cálculos foi traduzir a saída bruta do Python como Probabilidade_Atraso em lógica de negócio e KPIs acionáveis.

Coluna Condisional de Base: Primeiramente, para focar o *dashboard* apenas na fila de trabalho relevante, criei uma coluna condicional simples no Power Query (ETL) chamada solicitação ativa. Esta coluna atribui o valor 1 para solicitações nas fases "Triagem" ou "Agendamento" e 0 para todas as demais (como "Concluído" ou "Arquivado"). Todas as métricas a seguir utilizam esta coluna como filtro base.

Colunas Calculadas (DAX) para Lógica de Risco: Duas colunas calculadas foram criadas na FAT_CERTIFICADOS para processar a saída do ML:

```

1 Probabilidade_Atraso_Final_Coluna =
2 VAR RawProb = 'FAT_CERTIFICADOS'[Probabilidade_Atraso]
3 VAR MaxValidProbScale = 100
4 VAR DefaultOutlierValue = 0.45
5
6 RETURN
7 IF(
8     RawProb > MaxValidProbScale,
9     DefaultOutlierValue,
10    DIVIDE(RawProb, 100)
11 )

```

Figura 46 - Código DAX: Coluna de Correção da Probabilidade.

Fonte: Elaborado pelo autor deste trabalho.

A figura 46 mostra uma etapa de tratamento de dados. A Probabilidade_Atraso vinda do Python estava em uma escala de 0-100 e continha valores anômalos (outliers) em solicitações antigas que não tinham determinada a base de dados necessária para análise. Este código normaliza a probabilidade para uma escala de 0 a 1 (ex: 95 -> 0.95) e atribui um valor padrão (0.45) para qualquer valor inconsistente, garantindo a integridade do dado.

```

1 Status_Risco_SLA_Coluna =
2 VAR ProbFinal = 'FAT_CERTIFICADOS'[Probabilidade_Atraso_Final_Coluna]
3 VAR ColunaSolicitacaoAtiva = 'FAT_CERTIFICADOS'[solicitacao ativa]
4 VAR DataEntrada = 'FAT_CERTIFICADOS'[Criado em]
5 VAR LimiteSLADias = 3
6 VAR DataLimiteSLA = IF(NOT ISBLANK(DataEntrada), DataEntrada + LimiteSLADias, BLANK())
7 VAR DataAtual = TODAY()
8
9 RETURN
10 IF(
11     ColunaSolicitacaoAtiva = 1 && NOT ISBLANK(DataEntrada) && DataLimiteSLA < DataAtual,
12     "Atrasado Vencido (Real)",
13
14     IF(
15         ProbFinal >= 0.7,
16         "Alto Risco (Previsto)",
17         IF(
18             ProbFinal >= 0.4,
19             "Médio Risco (Previsto)",
20             "Baixo Risco (Previsto)"
21         )
22     )
23 )

```

Figura 47 - Código DAX: Coluna de Status de Risco (Lógica Principal).

Fonte: Elaborado pelo autor deste trabalho.

A figura 47 detalha a lógica de negócios central do dashboard. Esta coluna combina a realidade com a previsão:

- Primeiro a Realidade: Ela primeiro verifica se uma solicitação ativa já está vencida na realidade (se sua data limite de 3 dias já passou). Se sim, ela ignora o ML e rotula como "Atrasado Vencido (Real)".
- Segundo a Previsão: Somente se a solicitação ainda não venceu, a coluna passa a usar a probabilidade do ML limpa para classificar o risco em "Alto" ($\geq 70\%$), "Médio" ($\geq 40\%$) ou "Baixo" (Previsto).

Medidas (DAX) para KPIs do Dashboard: Por fim, as três medidas a seguir agrupadas na figura 48, foram criadas para agregar os dados e alimentar os KPIs cartões e gráficos do *dashboard* de riscos.

```

1  Tickets Ativos Contagem (Somente 1s) =
2  ↘ CALCULATE(
3    | COUNTROWS('FAT_CERTIFICADOS'),
4    | 'FAT_CERTIFICADOS'[solicitacao ativa] = 1
5  )
6
7  Tickets em Risco Alto =
8  ↘ CALCULATE(
9    | COUNTROWS('FAT_CERTIFICADOS'),
10   | 'FAT_CERTIFICADOS'[solicitacao ativa] = 1,
11   | 'FAT_CERTIFICADOS'[SLA_Previsto] = 1
12 )
13
14  % de Tickets em Risco Alto/Atrasados =
15  ↘ DIVIDE(
16    | [Tickets em Risco Alto],
17    | [Tickets Ativos Contagem (Somente 1s)]
18 )

```

Figura 48 - Código DAX: Medidas de Agregação do Risco.

Fonte: Elaborado pelo autor deste trabalho.

A explicação para cada uma delas se dá:

- Tickets Ativos Contagem: Calcula o número total de solicitações na fila, o "100" no dashboard, usando a coluna condicional de solicitação ativa.
- Tickets em Risco Alto: Calcula o numerador do *KPI* o "77". Ele conta as solicitações que estão ativas que o ML previu como "1" (Risco Alto).

- % de Tickets em Risco Alto/Atrasados: Calcula o *KPI* principal o "77%", dividindo o Tickets em Risco Alto pelo total de Tickets Ativos.

Para finalizar o desenvolvimento, a solução completa foi publicada no Power BI Service (Online). Utilizando a função "Publicar" do Power BI Desktop, o arquivo .pbix contendo as sete telas de BI e ML, o modelo de dados e os scripts foi implantado em um Workspace corporativo da área em que atuo. Esta etapa disponibilizou o relatório final de forma segura para os gestores e equipes, que agora podem acessá-lo via navegador ou aplicativo móvel. Para viabilizar a atualização automática e agendada dos dados incluindo a execução do script Python que carrega o modelo .pkl, foi configurado um Gateway de Dados do Power BI, que faz a ponte entre o ambiente online e os arquivos locais.

Com a conclusão do *dashboard* preditivo e o detalhamento das métricas DAX, encerra-se o Capítulo 3 do Desenvolvimento. O resultado é uma solução completa e integrada, capaz não apenas de tratar e visualizar dados históricos para análise de causa-raiz nas seis telas do ambiente de BI, mas também de prever tendências e comportamentos futuros por meio de um modelo de *Machine Learning* validado. Essa combinação transforma o *dashboard* em uma ferramenta estratégica de gestão preditiva e ação proativa.

O próximo capítulo apresentará uma avaliação formal do impacto dessa solução em relação aos objetivos definidos no início do projeto, destacando os ganhos de eficiência e inteligência operacional alcançados.

4 CONSIDERAÇÕES FINAIS

Chega-se à parte final deste trabalho, reunindo tudo o que foi feito ao longo das etapas anteriores. Depois de apresentar a construção da solução em Business Intelligence e o modelo preditivo de *Machine Learning*, o objetivo deste capítulo é refletir sobre os resultados alcançados e avaliar o impacto real da proposta diante dos desafios que foram definidos no início do projeto. Além disso, deseja-se destacar as contribuições práticas do estudo, reconhecer seus limites e indicar caminhos possíveis para aprimoramentos e novas pesquisas no futuro.

O objetivo geral deste trabalho, conforme definido na introdução, foi "desenvolver e avaliar uma solução utilizando *Business Intelligence* e técnicas de *Machine Learning* com o objetivo de otimizar o tempo de atuação das equipes responsáveis pelos processos de certificação digital".

Para alcançar esse objetivo, desenvolveu-se uma solução integrada de ponta-a-ponta, Ela é composta por uma ferramenta de *Business Intelligence* com sete telas analíticas no Power BI, construídas sobre um modelo de dados Floco de Neve resultante de um processo robusto de ETL, onde contém dados de diferentes fontes e possui métricas complexas na linguagem DAX para análise diagnóstica. Junto a isso, criou-se um modelo preditivo de *Machine Learning* usando o classificador *RandomForestClassifier* que atingiu uma acurácia de 85% e com 93% de AUC. Esse modelo foi desenvolvido em Python, utilizando o *Google Colab*, onde foi integrado nativamente ao *Power BI* gerando assim uma gestão proativa de risco de SLA.

A avaliação desta solução demonstra que os objetivos foram atingidos. A ferramenta de BI respondeu diretamente aos gargalos operacionais que identificamos: o Gargalo 1 (Retrabalho) foi endereçado pelas telas 1 ("Análise Solicitações") e 5 ("Tempo Médio por Regional"). Elas agora permitem à gestão identificar as equipes e os motivos que mais geram erros, possibilitando um treinamento focado. O gargalo 2 (Dependência Externa) foi resolvido pela tela 2 ("Análise Concluídas"), que expôs o tempo médio de cada certificadora parceira, fornecendo uma ferramenta de negociação baseada em dados. O Gargalo 3 (Etapa Manual de Conclusão), embora não automatizado pela ferramenta, teve sua falha processual exposta: a Tela 6 de Consulta de solicitações, permite ao gestor cruzar informações da fila nas solicitações ainda ativas com a coluna data de assinatura vinda do *Snowflake*. Isso possibilita a identificação imediata de processos que, na prática, já foram finalizados e já estão com o contrato assinado, mas que a equipe operacional esqueceu de concluir manualmente no

sistema, dando visibilidade e permitindo a auditoria deste erro. Finalmente o gargalo 4 (falta de visibilidade gerencial) foi completamente sanado pelo conjunto das sete telas, que fornecem visão estratégica (KPIs), tática (análise de equipes) e operacional em tempo real.

A solução de Machine Learning foi a contribuição mais significativa do projeto, pois muda o paradigma da equipe, de reativo para proativo. Em vez de apenas usar o BI para analisar o que já atrasou, a Tela 7 ("Gestão Proativa de Risco - ML") permite à equipe focar nas solicitações que o modelo previu com alta probabilidade de atraso. Isso realmente permite uma otimização real do tempo da equipe, que pode agir para evitar a falha no SLA antes que ela ocorra.

Para validar o impacto prático e a eficácia da solução, foram coletados feedbacks. O depoimento da gestora da área, Luana Vendramim, apresentado abaixo, foca na visão estratégica e gerencial:

O BI de certificados me permitiu ter uma visão estratégica de como a área operacional está performando, o que está sendo entregue e o que não está sendo entregue. Quais são as divisões por área solicitante dos casos além de permitir criar planos de ação específicos para cada necessidade de acordo com o momento de cada área que solicita. Me permitiu ter insights de como poderíamos melhorar o tempo médio da execução das tarefas, foi através dele que a gente identificou um gargalo grande de uma prestadora de serviço específica e corremos pra ter uma segunda opção para que a gente não sobrecarregasse a nossa operação por conta da ineficiência da prestadora de serviço. Conseguimos desctrinchar quais quais áreas precisam de treinamento, quais são os solicitantes que talvez estejam menos engajados com com quais são as práticas da área quem que está enviando casos errados, casos que que faltam detalhes de documentação e critérios específicos de acordo com cada necessidade. Então foi muito bom poder ter essa visão ampla e automatizada não apenas via planilhas, o que fez toda a diferença é que eu consigo diariamente entrar no BI e já ter uma prévia do que que tá acontecendo do que que eu já posso fazer um plano de ação mais imediato e não esperar um dia da semana específico pra baixar um relatório e acompanhar isso de maneira manual.

Já na perspectiva comercial, o feedback de um consultor, Alisson Goulart, validou a melhoria no fluxo do processo de certificado digital:

O processo está muito melhor, bem mais rápido do que era antes. Agora temos também a visão para os consultores, assim conseguimos acompanhar em tempo real as emissões de certificado e os status, facilitando muito o processo para não ter que ficar perguntando e gastando tempo operacional com isto.

Estes relatos, apresentados integralmente, comprovam que a solução atingiu os objetivos propostos, entregando valor tanto para a tomada de decisão estratégica quanto para a eficiência operacional do dia a dia.

Adicionalmente, o objetivo de "explorar a escalabilidade da solução... em outros setores da empresa" também foi validado na prática. A metodologia e a arquitetura desenvolvidas neste trabalho provaram ser adaptáveis, não se limitando apenas ao processo de certificação digital. Prova disso é que uma solução baseada neste mesmo modelo (ETL, BI e ML) já está em fase de implantação em outra área da empresa, sendo aplicada na análise de antifraude de um produto específico. Isso comprova o potencial de replicação e o valor estratégico da ferramenta para a organização como um todo.

Apesar dos resultados positivos e da solução já estar em uso na empresa, é importante observar as limitações da versão atual e os pontos que precisam ser aprimorados a longo prazo. A principal limitação ainda é a dependência de processos manuais de atualização, como a exportação de planilhas do Pipefy, o que impede que a análise ocorra em tempo totalmente real. Mas já há trabalho em cima disso para puxar uma API do Pipefy para o *Snowflake*, mas conforme o tempo curto da realização deste trabalho não foi possível a finalização da mesma. Além disso, o modelo de *Machine Learning* foi treinado apenas uma vez e, para manter sua precisão ao longo do tempo e evitar a redução gradual na precisão, ainda é necessário realizar treinamentos periódicos de forma manual, pois o processo ainda não foi automatizado.

Diante dessas limitações, naturalmente surgem oportunidades claras para próximos trabalhos no futuro. Uma das principais propostas é a criação de um pipeline de MLOps (*Machine Learning Operations*) em ambiente de nuvem, que permita automatizar o retreinamento do modelo de forma contínua. Também seria muito interessante substituir a importação manual de arquivos por conexões diretas via API e permitir que o *script* de *Machine Learning* leia os dados diretamente do *Snowflake*. Isso tornaria todo o processo mais ágil e automatizado.

Por fim, uma melhoria relevante seria criar um "mecanismo de *feedback*", no qual os gestores pudessem registrar as ações tomadas em cada solicitação de risco. Isso possibilitaria que o modelo aprendesse com os resultados reais das decisões humanas e tornando-se cada vez mais assertivo.

REFERÊNCIAS

AMORIM, Victor Hugo Oliveira de. Análise e otimização de lead time em processos ágeis: estudo de caso em empresa de marketplace. 2023. Acesso em: 02 Mar. 2025. Disponível em: <http://repositorio.ifba.edu.br/jspui/handle/123456789/587>

ANGELONI, Maria Terezinha. Elementos intervenientes na tomada de decisão. 2003. Acesso em: 17 Mar. 2025. Disponível em: <https://www.scielo.br/j/ci/a/3RVhpdpmsgkwCxtCC6sXkt/?format=pdf&lang=pt>

ARAÚJO, Rayanne Cristina et al. Tomada de decisão na administração pública: uma revisão sistemática. REA-Revista Eletrônica de Administração, v. 18, n. 1, p. 32-49, 2020. Acesso em: 02 Abril. 2025 Disponível em: <http://periodicos.unifacef.com.br/rea/article/view/1406/1420>

BRAGHITTINI, Ronaldo. Business Intelligence: Implementar do jeito certo e a custo zero. 2017. Acesso em: 01 Abril. 2025. Disponivel em: https://www.academia.edu/50919879/_BI_Business_Intelligence_Implementar_do_jeito_certo_e_a_custo_zero

CARVALHO, Leandro da Silva. Uma nova abordagem para a criação de perfis de clientes rentáveis utilizando machine learning em ambiente de cloud computing. 2018. Acesso em: 31 Maio 2025. Disponível em: <https://pantheon.ufrj.br/bitstream/11422/13422/1/LeandroDaSilvaCarvalho.pdf>

CASTELLS, Manuel; CARDOSO, Gustavo. A Sociedade em Rede Do Conhecimento à Acção Política. 2005 Acesso em: 08 Mar. 2025. Disponivel em: <https://diegodelpasso.com/wp-content/uploads/2016/05/manuel-castells-a-sociedade-em-rede.pdf>

CRUZ, Bruno Campanella; Miranda, Bruno Gabriel Correa; Turchette, Fellipe Barretto. Conceitos de business intelligence por meio de estudos de caso: ferramentas pentaho e qlikview. 2014. Acesso em: 01 Abril. 2025 Disponível em: <https://lyceumonline.usf.edu.br/salavirtual/documentos/2704.pdf>

CRUZ, Josemar Prates da. LightGBM: Uma Potente Solução em Machine Learning. 2025, Acesso em: 07 Junho 2025. Disponível em: <https://cienciaedados.com.br/lightgbm-uma-potente-solucao-em-machine-learning/>

DA SILVA LEÃO, A. P.; GOMES, B. R. A.; CRUZ , J. C. S.; DA SILVA, V. V.; SENA, C. da C.; OLIVEIRA JÚNIOR, F. A. V. Power bi para tomada de decisões estratégicas: análise de indicadores-chave de desempenho (kpis). Revista foco, [s. L.], V. 16, n. 7, p. E2472, 2023. Acesso em: 02 Mar. 2025. Disponível em: <https://ojs.focopublicacoes.com.br/foco/article/view/2472>

DOS SANTOS, V. L.; GIBERTONI, D. OS IMPACTOS DO BUSINESS INTELLIGENCE PARA TOMADA DE DECISÕES. Revista Interface Tecnológica, Taquaritinga, SP, v. 19, n. 2, p. 258–269, 2022. DOI: 10.31510/infa.v19i2.1524. Acesso em: 17 Mar. 2025. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/1524>.

FARIA, Fabricio Tavares de. Business Intelligence Como Apoio À Tomada De Decisão: Um Estudo De Campo Aplicado. 2022. Acesso em: 17 Mar. 2025. Disponível em:
<https://app.uff.br/riuff/handle/1/28478>

FERREIRA, João et al. O processo etl em sistemas data warehouse. In: INForum. sn, 2010. p. 2010. Acesso em: 30 Maio. 2025. Disponível em:
https://www.researchgate.net/profile/Jose-Machado-23/publication/265195317_O_Processo_ETL_em_Sistemas_Data_Warehouse/links/5580380a08aea3d7096e442e/O-Processo-ETL-em-Sistemas-Data-Warehouse.pdf

FIGUEIREDO, Lucas Oliveira de. Estudo de viabilidade para implementação em hardware de técnica de aprendizado de máquina. 2020. Acesso em: 31 Maio 2025. Disponível em:
<http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/19127>

FREITAS, Marcelo Tavares de. Business Intelligence para gestão de indicadores de desempenho e suporte a tomada de decisão no setor de compras de uma organização. 2021. Acesso em: 01 Abril. 2025 Disponível em:
<https://monografias.ufop.br/handle/35400000/3560>

LIGHITGBM Documentação Oficial [S.l.: s.n.], 2025, Acesso em: 07 Junho 2025. Disponível em: <https://lightgbm.readthedocs.io/en/latest/Features.html>

LOPES, Gesiel Rios et al. Introdução à análise exploratória de dados com python. Escola Regional de Computação Aplicada à Saúde (ERCAS 2019), 2019. Acesso em: 31 Maio 2025. Disponível em:
https://www.researchgate.net/profile/Gesiel-Lopes/publication/336778766_Introducao_a_Ana_lise_Exploratoria_de_Dados_com_Python/links/5db225d2a6fdccc99d9426f2/Introducao-a-A_nalise-Exploratoria-de-Dados-com-Python.pdf

LUCENA, Walter Lucas Alves de. Power BI–uma visão da utilização de uma ferramenta Business Intelligence no ambiente organizacional. 2022. Acesso em: 28 Abril. 2025 Disponível em:
<https://repositorio.ufersa.edu.br/server/api/core/bitstreams/1fc4ac70-59ef-4fcc-ac8a-18c8e3f6f49e/content>

MICROSOFT Power BI: Uma plataforma unificada e escalonável para autoatendimento e BI empresarial. [S.l.: s.n.], 2025. Acesso em: 20 Abril. 2025. Disponível em:
https://www.microsoft.com/pt-br/power-platform/products/power-bi#tabs-oc2a1c22m1_tab2

MICROSOFT, Azure: O que é o Azure Machine Learning? [S.l.: s.n.], 2025. Acesso em: 31 Maio 2025. Disponível em:
<https://learn.microsoft.com/pt-br/azure/machine-learning/overview-what-is-azure-machine-learning?view=azureml-api-2>

NASCIMENTO, Bruna Laís Campos do; SILVA, Edilene Maria da. Lei Geral de Proteção de Dados (LGPD) e repositórios institucionais: reflexões e adequações. Em Questão, v. 29, p. e-127314, 2023. Acesso em: 03 Mar. 2025. Disponível em:
<https://www.scielo.br/j/emquestao/a/w3xQNy4bnytwK6MxzgyKgsy/>

NETO, Ascendino Elpidio de Farias. Aplicação do Power BI para o tratamento de dados acadêmicos na universidade federal da paraíba. 2021. Acesso em: 31 Maio 2025. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/27730>

PEREIRA, Alonso Luiz Luiz. A racionalidade no processo decisório nas organizações. NEGÓCIOS EM PROJEÇÃO, v. 10, n. 2, p. 124-139, 2019. Acesso em: 09 Abril. 2025 Disponível em: <https://projecaociencia.com.br/index.php/Projecao1/article/view/1481>

PHYTON Phyton: História e licença. [S.l.: s.n.], 2025. Acesso em: 06 Junho 2025. Disponível em: <https://docs.python.org/pt-br/3.13/license.html>

PIPEFY, Como o Pipefy funciona? Acesso em: 30 Maio. 2025. [S.l.: s.n.], 2025. Disponível em: <https://www.pipefy.com/pt-br/como-pipefy-funciona/>

PRÉVE, Altamiro Damian; MORITZ, Gilberto de Oliveira; PEREIRA, Maurício Fernandes. Organização, processos e tomada de decisão. Florianópolis, SC: Departamento de Ciências da Administração/UFSC, 2010. Acesso em: 09 Abril. 2025. Disponível em:
https://d1wqtxts1xzle7.cloudfront.net/52024051/PREVE_MORITZ_PEREIRA.Organizacao_Processos_e_Tomada_de_Decisao-libre.pdf?1488649601=&response-content-disposition=inline%3B+filename%3DOrganizacao_Processos_e_Tomada_de_Decisa.pdf&Expires=1747958656&Signature=Rfqj8NSDOpXbQoPhopnymXu4qEiaN5xWCgapkiU2OMUv84JGy~68c~hWRD0Q1mOBOJ90EBRliTANI7nGrGn7ex49hVyH-p7~KkhFJaD49E9VEvbKChKHz~3vHUqSrow00fpo-yBTdYviqvzv51kkm-ES~tj8GSn8L0yR11Rqezy5nM2rohWPBBICyWgn7Bx15Sj7lTePiaxq~btbK4WfW8So~uIK0HepBFZKwfI-kV2OC4A40c06CHysbBxLw1G4FTbG~mNy9bx575aatpzpXx6LA~DPYhymNUBr89IWPL85c-OEKFrFTfdv5v3cQbu1bwML~2PmjDoh~m-oRCzFg_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

RIBEIRO, Manuelle Parreira. A tomada de decisão através do Power BI. 2024. Acesso em: 23 Abril. 2025 Disponível em:
<https://adelpha-api.mackenzie.br/server/api/core/bitstreams/82785355-9956-4118-a1f4-252f5c1c3716/content>

RITTMANN, Júlia Del Pino. Análise de três dimensões de qualidade de dados em tabelas de compras de um ambiente de Data Warehouse. 2023. Acesso em: 03 Maio. 2025. Disponível em: <https://lume.ufrgs.br/handle/10183/272039>

SCIKIT-LEARN Documentação Oficial [S.l.: s.n.], 2025. Acesso em: 07 Junho 2025. Disponível em: <https://scikit-learn.org/stable/about.html#history>

SNOWFLAKE, Banco de dados SNOWFLAKE [S.l.: s.n.], 2025. Acesso em: 30 Maio. 2025. Disponível em: <https://docs.snowflake.com/en/sql-reference/snowflake-db>

VELAGANA, Anji. Why Azure Machine Learning?. 2023. Acesso em: 06 Junho 2025. Disponível em: <https://mindmajix.com/why-azure-machine-learning>