Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Otimização Automatizada de Buffer Pool em Bancos de Dados Mainframe Utilizando Aprendizado de Máquina

Eduardo Pingarilho Mendizabal

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientadora
Prof. Dr. Aletéia Patrícia Favacho de Araújo von Paumgartten

Coorientador
Prof. Dr. Geraldo Pereira Rocha Filho

Brasília
2025

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

PM538o

Pingarilho Mendizabal, Eduardo
Otimização Automatizada de Buffer Pool em Bancos de Dados Mainframe Utilizando Aprendizado de Máquina / Eduardo Pingarilho Mendizabal; orientador Aletéia Patrícia Favacho de Araújo von Paumgartten; co-orientador Geraldo Pereira Rocha Filho. Brasília, 2025.
89 p.

Tese(Mestrado Profissional em Computação Aplicada) Universidade de Brasília, 2025.

1. SGBD. 2. Buffer Pool. 3. Mainframe. 4. Otimização de Parâmetros. 5. Otimização Bayesiana. I. Favacho de Araújo von Paumgartten, Aletéia Patrícia, orient. II. Pereira Rocha Filho, Geraldo, co-orient. III. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Otimização Automatizada de Buffer Pool em Bancos de Dados Mainframe Utilizando Aprendizado de Máquina

Eduardo Pingarilho Mendizabal

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Aletéia Patrícia Favacho de Araújo von Paumgartten (Orientadora)
CIC/UnB

Prof. Dr. Alan Demétrius Baria Valejo
Universidade Federal de São Carlos (UFSCAR)

Prof. Dr. Márcio de Carvalho Victorino
Universidade de Brasília (UnB)

Prof. Dr. Edna Dias Canedo
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 30 de Novembro de 2025

Dedicatória

Dedico este trabalho à minha família, pelo alicerce inabalável em todas as etapas da minha vida acadêmica e profissional; aos colegas da GESIT, pela parceria diária que tornou possível a construção desta pesquisa; e aos meus orientadores, cuja orientação firme, generosa e criteriosa foi essencial para transformar esforço em resultado.

Agradecimentos

Agradeço, em primeiro lugar, aos meus orientadores, Professora Aleteia Araujo e Professor Geraldo Rocha Filho, pela competência acadêmica, pela confiança depositada e pela orientação criteriosa que direcionou este trabalho. Suas contribuições foram decisivas para a qualidade e a maturidade desta pesquisa.

Agradeço também à Universidade de Brasília e ao Programa de Pós-Graduação em Ciência da Computação pelo ambiente de formação, apoio institucional e infraestrutura acadêmica que possibilitaram o desenvolvimento desta dissertação.

À equipe da GESIT, deixo meu sincero reconhecimento pela parceria diária, pelo compartilhamento de conhecimento e pela colaboração ativa durante todas as fases do estudo. Este trabalho só foi possível graças ao suporte técnico e ao contexto operacional proporcionado pela equipe e pela instituição.

À minha família, registro minha profunda gratidão pelo apoio incondicional, paciência e incentivo contínuo. Em cada etapa desta jornada, sua presença e confiança foram fundamentais para que eu me mantivesse focado e determinado.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho, seja por meio de discussões, sugestões, apoio profissional ou motivação pessoal. Minha gratidão é estendida a cada um que fez parte deste percurso acadêmico e humano.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

O *Buffer Pool* (BPOOL) é um componente essencial em Sistema de Gerenciamento de Banco de Dados (SGBD) que é o responsável por manter em memória páginas de dados frequentemente acessadas e, assim, reduzir a latência de operações de entrada e saída. Este trabalho propõe uma metodologia automatizada para otimização de BPOOL, baseada em Aprendizado de Máquina, estruturada em três etapas integradas: (i) redução dimensional orientada por Análise Fatorial Exploratória combinada ao agrupamento *K-Means* para síntese de métricas; (ii) seleção de parâmetros via regressão LASSO, identificando as variáveis de maior influência no atraso síncrono; e (iii) aplicação de *Bayesian Optimization Gaussian Process Regression* com as funções de aquisição *Expected Improvement*, *Probability of Improvement* e *Upper Confidence Bound*. Os experimentos, conduzidos em ambiente mainframe de uma instituição financeira, avaliaram 29 BPOOLs e resultaram em ganhos efetivos maiores de 10% em 17 deles, com reduções de latência superiores a 90% nos casos mais críticos, e entre 12% e 50% nos demais. A solução monstrou capacidade de identificar gargalos reais, evitar recomendações desnecessárias em *pools* estáveis e propor ajustes alinhados ao padrão de carga de cada *buffer*, mantendo governança de implantação e *rollback* seguro. Os resultados confirmam a efetividade prática da abordagem na gestão adaptativa de memória, e na estabilização de desempenho em sistemas corporativos de grande escala.

Palavras-chave: SGBD, Buffer Pool, Mainframe, Otimização de Parâmetros, Latência, Aprendizado de Máquina, Processos Gaussianos, Otimização Bayesiana

Abstract

The *Buffer Pool* (BPOOL) is a fundamental component of a *Database Management System* (DBMS), responsible for keeping frequently accessed data pages in memory and thus reducing input/output latency. This work proposes an automated methodology for BPOOL optimization, based on Machine Learning, structured in three integrated stages: (i) dimensionality reduction guided by Análise Fatorial Exploratória combined with K-Means clustering for metric synthesis; (ii) parameter selection using LASSO regression to identify the variables with the greatest influence on synchronous delay; and (iii) application of *Bayesian Optimization Gaussian Process Regression* with the acquisition functions Expected Improvement, Probability of Improvement, and Upper Confidence Bound. The experiments, conducted in a mainframe environment of a financial institution, evaluated 29 BPOOLs and resulted in effective gains above 10% in 17 of them, with latency reductions greater than 90% in the most critical cases and between 12% and 50% in the others. The solution demonstrated the ability to identify real bottlenecks, avoid unnecessary recommendations in stable pools, and propose adjustments aligned with the workload profile of each buffer, while maintaining deployment governance and safe rollback. The results confirm the practical effectiveness of the approach for adaptive memory management and performance stabilization in large-scale corporate systems.

Keywords: DBMS, Buffer Pool, Mainframe, Parameter Optimization, Latency, Machine Learning, Gaussian Processes, Bayesian Optimization

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACID Atomicidade, Consistência, Isolamento e Durabilidade.

ADW-DDPG *Adaptive Dynamic Workload Deep Deterministic Policy Gradient.*

AFE Análise Fatorial Exploratória.

AM Aprendizado de Máquina.

ARD *Automatic Relevance Determination.*

ATT *Attention.*

BERT *Bidirectional Encoder Representations from Transformers.*

BO *Bayesian Optimization.*

BO-GPR *Bayesian Optimization Gaussian Process Regression.*

BPOOL *Buffer Pool.*

CART *Classification and Regression Trees.*

CDB *Cloud Database.*

CGP *Contextual Gaussian Process.*

CNN *Convolutional Neural Network.*

CPU *Central Processing Unit.*

CRISP-DM *Cross-Industry Standard Process for Data Mining.*

DB2 IBM Db2 for z/OS.

DBA *Database Administrator.*

DBMS *Database Management System.*

DBSCAN *Density-Based Spatial Clustering of Applications with Noise.*

DDPG *Deep Deterministic Policy Gradient.*

DDQN *Double Deep Q-Network.*

DISC *Data Intensive Scalable Computing.*

DL *Deep Learning.*

DNN *Deep Neural Network.*

DWQT *Deferred Write Threshold.*

EI *Expected Improvement.*

GA *Genetic Algorithm.*

GMM *Gaussian Mixture Model.*

GP *Gaussian Process.*

GPR *Gaussian Process Regression.*

HeSBO *Hashing-Enhanced Subspace Bayesian Optimization.*

KMEANS *k-means.*

KMO Kaiser-Meyer-Olkin.

LASSO *Least Absolute Shrinkage and Selection Operator.*

LHS *Latin Hypercube Sampling.*

LLM *Large Language Model.*

LRU *Least Recently Used.*

Mainframe Computador de Grande Porte.

MIM *Multi-Indicator Matching.*

NLP *Natural Language Processing.*

NLPD *Negative Log Predictive Density.*

NP Não Determinístico em Tempo Polinomial.

OLAP *Online Analytical Processing.*

OLS *Ordinary Least Squares.*

OLTP *Online Transaction Processing.*

PCA *Principal Component Analysis.*

PI *Probability of Improvement.*

PS *Partitioning Score.*

RBF *Radial Basis Function.*

REMBO *Random Embedding Bayesian Optimization.*

RL *Reinforcement Learning.*

ROI *Return on Investment.*

SGBD Sistema de Gerenciamento de Banco de Dados.

SGBDR Sistema de Gerenciamento de Banco de Dados Relacional.

SGBE *Sampled Gaussian Bandit Estimator.*

SHAP *SHapley Additive exPlanations.*

SMAC *Sequential Model-based Algorithm Configuration.*

SQL *Structured Query Language.*

SSE *Sum of Squared Errors.*

STMM *Self-Tuning Memory Manager.*

TF-IDF *Term Frequency-Inverse Document Frequency.*

TPE *Tree-structured Parzen Estimator.*

UCB *Upper Confidence Bound.*

VDWQT *Vertical Deferred Write Threshold.*

VPMAX Tamanho máximo do *buffer pool*.

VPMIN Tamanho mínimo do *buffer pool*.

VPSEQT *Sequential Steal*.

VPSIZE Tamanho do *buffer pool*.

VPUSE Uso atual do *buffer pool* em bytes.

WLM Workload Manager.

Capítulo 1

Introdução

1.1 Introdução

O Sistema de Gerenciamento de Banco de Dados (SGBD) desempenha um papel estratégico em ambientes corporativos modernos, sustentando operações críticas em setores como financeiro, saúde, telecomunicações e varejo [1, 2]. Em escala global, estimativas de mercado indicam que plataformas corporativas baseadas em SGBD processam volumes massivos de operações, com relatos apontando mais de 5 trilhões de consultas diárias em 2025[1]. Essa ordem de grandeza reforça a necessidade de níveis elevados de disponibilidade, consistência e desempenho. A eficiência operacional de um SGBD impacta diretamente custos de infraestrutura, escalabilidade das aplicações e a experiência do usuário final. Nesse contexto, a otimização de parâmetros de configuração emerge como fator determinante para maximizar desempenho e garantir o uso eficiente dos recursos computacionais.

A importância da otimização de parâmetros em SGBDs decorre da necessidade de manter o melhor desempenho em cargas de trabalho heterogêneas. SGBDs modernos apresentam centenas de parâmetros configuráveis [3], muitos deles interdependentes [4], cujo efeito combinado sobre o desempenho é altamente não linear [5]. A intervenção manual torna-se inviável em ambientes de grande escala, nos quais ajustes inadequados podem degradar o desempenho global [6]. Frente a esse cenário, técnicas de automação baseadas em Aprendizado de Máquina (AM) têm se mostrado promissoras para ajustar parâmetros de forma contínua e adaptativa [7].

Dentre os subsistemas de um SGBD, o *Buffer Pool* (BPOOL) destaca-se como a principal região de memória gerenciada, responsável por armazenar páginas de dados e índices frequentemente acessados [8, 9]. Ao reduzir a necessidade de acessos a disco — ordens de grandeza mais lentas que acessos à memória — o BPOOL influencia diretamente a

[1]Estimativa apresentada em relatório de mercado disponível em `https://www.industryresearch.biz/market-reports/database-management-system-market-105050`

latência de operações críticas. Em sistemas Db2 for z/OS, a sensibilidade desse subsistema é ampliada pela heterogeneidade das cargas de trabalho, pelo compartilhamento de recursos entre aplicações de missão crítica e pelas políticas internas de escrita diferida, pré-leitura e substituição de páginas [10].

A configuração adequada do BPOOL, contudo, apresenta desafios substanciais. Os parâmetros `VPUSE`, `VPMIN`, `VPMAX`, `VPSEQT`, `DWQT` e `VDWQT` interagem de forma não linear e dependem fortemente do perfil de carga aplicado a cada *pool*. Pequenas alterações podem gerar melhorias significativas ou, inversamente, degradar abruptamente o desempenho, particularmente sob padrões de acesso variáveis entre períodos diurnos e noturnos. A ausência de mecanismos nativos de *autotuning* no Db2 for z/OS impõe dependência de heurísticas operacionais e ajustes manuais, que dificilmente capturam todo o dinamismo do ambiente. A literatura reconhece, portanto, que a otimização de parâmetros de SGBDs configura um problema NP-difícil [11], demandando métodos sistemáticos, adaptativos e orientados por dados.

1.2 Definição do Problema

Do ponto de vista matemático, a dificuldade do problema pode ser caracterizada pela dimensionalidade do espaço de busca. Considerando-se sete parâmetros contínuos de configuração por BPOOL, o espaço total de otimização para um ambiente com 29 BPOOLs independentes possui dimensão

$$D = 29 \times 7 = 203$$

ou seja, trata-se de um problema de otimização contínua em um espaço de 203 dimensões. Mesmo assumindo, de forma conservadora, apenas k valores discretizados por parâmetro para fins de análise de ordem de grandeza, o número total de combinações possíveis seria

$$\mathcal{N} = k^{203},$$

o que torna qualquer abordagem exaustiva computacionalmente inviável. Para ilustrar, mesmo com uma discretização extremamente grosseira de apenas $k = 10$ valores por parâmetro, ter-se-ia

$$\mathcal{N} = 10^{203},$$

um espaço de busca completamente fora de qualquer capacidade prática de exploração direta.

Além disso, o problema não é apenas de alta dimensionalidade, mas também fortemente não linear e não estacionário, uma vez que o comportamento ótimo de cada BPOOL depende da quantidade, do tamanho e do padrão de acesso das tabelas residentes, bem como das características dinâmicas da carga de trabalho, que variam ao longo do tempo. Assim, a função objetivo associada à minimização de `MAX_SYNC_IO_DLY_MS` pode ser formalmente vista como

$$f : \mathbb{R}^{203} \to \mathbb{R},$$

com propriedades desconhecidas de convexidade, continuidade prática e diferenciabilidade, além de apresentar múltiplos mínimos locais. Essa formulação evidencia que o ajuste ótimo de BPOOLs em ambientes reais configura um problema de otimização contínua de alta dimensão, altamente acoplado e sensível ao contexto operacional.

Para evidenciar a escala e a complexidade do problema em um cenário de produção, este trabalho analisou um ambiente Db2 for z/OS em produção composto por 29 *Buffer Pool* (BPOOL) independentes, monitorados em períodos diurnos e noturnos. Após os processos de higienização, padronização e reconciliação temporal, foi construída uma base consolidada com 122.618 registros válidos e 35 métricas elegíveis para modelagem, obtidas a partir da aplicação de análise fatorial exploratória combinada com filtragem estatística. A seleção de parâmetros por regressão *Least Absolute Shrinkage and Selection Operator* (LASSO) identificou seis variáveis como determinantes para o comportamento da métrica-alvo, definida como a redução da latência máxima de I/O síncrono, medida em milissegundos, representada por `MAX_SYNC_IO_DLY_MS`. Esses resultados confirmam a elevada dimensionalidade do espaço de configuração e a complexidade intrínseca ao ajuste de BPOOLs em ambientes de missão crítica.

No ambiente analisado nesta pesquisa, os desafios tornam-se evidentes. Os 29 BPOOLs exibem padrões heterogêneos de uso, variabilidade expressiva de latência e forte colinearidade entre métricas. A base consolidada de 122.618 amostras evidencia dependência temporal e relações não triviais entre parâmetros e desempenho. Assim, a otimização manual se torna impraticável, justificando o desenvolvimento de uma metodologia automatizada capaz de explorar o espaço de configuração, modelar incertezas, e propor ajustes fundamentados. Essa necessidade de substituição de abordagens manuais por métodos sistemáticos e orientados por dados baliza a justificativa do tema a seguir.

1.3 Justificativa do Tema

A motivação deste estudo decorre da necessidade de uma instituição financeira de grande porte aprimorar o desempenho de seu ambiente Db2 for z/OS (v13), baseado em arquitetura *mainframe*. Diferentemente de ambientes distribuídos, que escalam horizontalmente e

toleram flutuações de desempenho, plataformas *mainframe* operam sob arquitetura de processamento centralizado, exigindo padrões de desempenho muito mais rigorosos e baixa tolerância a ineficiências. Nesse contexto, qualquer gargalo no subsistema de BPOOL tende a se propagar rapidamente para toda a carga de trabalho, reforçando a necessidade de abordagens de otimização precisas, interpretáveis e com governança operacional robusta.

A literatura apresenta lacunas significativas nesse domínio. Há escassez de trabalhos focados em ambientes *mainframe* [12], especialmente no IBM Db2 for z/OS, cuja arquitetura interna e subsistema de *buffer* divergem substancialmente dos SGBDs distribuídos convencionais. Grande parte das pesquisas existentes concentra-se em simulações ou ambientes controlados, carecendo de validação em cenários de produção real com restrições operacionais rigorosas. Além disso, poucos trabalhos integram técnicas de redução dimensional, seleção de parâmetros e otimização bayesiana em uma solução unificada com garantias de estabilidade e reversibilidade, como requerido em ambientes financeiros.

1.4 Objetivos

O objetivo geral deste trabalho é propor uma metodologia automatizada para otimização de parâmetros de BPOOL em SGBDs, visando reduzir latências e aprimorar a eficiência operacional. Para cumprir esse objetivo geral, fez-se necessário atingir os seguintes objetivos específicos:

- Aplicar técnicas de redução de dimensionalidade e agrupamento de métricas;
- Identificar e ranquear os parâmetros de maior impacto;
- Implementar otimização via BO-GPR com múltiplas funções de aquisição; e
- avaliar empiricamente os ganhos obtidos em ambiente real.

1.5 Contribuições da Pesquisa

Esta pesquisa apresenta contribuições científicas, metodológicas e práticas ao campo de otimização automatizada de SGBDs:

- **Contribuições Científicas** — A adaptação e validação de técnicas de AM para ambientes *mainframe* IBM Db2 for z/OS preenche uma lacuna importante da literatura. A integração de Análise Fatorial Exploratória, *K-Means*, regressão LASSO e BO-GPR em uma solução robusta representa uma contribuição inédita validada com dados reais e restrições operacionais autênticas.

- **Contribuições Metodológicas** — A aplicação sistemática da metodologia CRISP-DM demonstra sua eficácia no contexto de tuning de SGBDs. A estratégia de redução dimensional combinando AFE e *K-Means*, juntamente com a implementação de BO-GPR com funções de aquisição EI, PI e UCB, reforça a consistência metodológica da solução.
- **Contribuições Práticas** — A validação em ambiente real avaliou 29 BPOOLs sob cargas autênticas. Entre eles, 17 apresentaram ganhos positivos de latência, com reduções superiores a 90% nos casos mais críticos e entre 12% e 50% nos demais. A solução inclui governança de implantação gradual, rastreabilidade completa e mecanismos de *rollback* seguro, garantindo estabilidade em ambiente de missão crítica.

1.6 Estrutura deste Trabalho

A presente dissertação está organizada, além deste capítulo introdutório, em cinco capítulos adicionais, estruturados para oferecer uma compreensão abrangente do problema de pesquisa e da solução proposta. Cada capítulo cumpre uma função específica na consolidação da argumentação, articulando fundamentos teóricos, metodologia empregada, análise experimental e resultados obtidos. Assim, os demais capítulos desta dissertação são:

- **Capítulo 2** — Fundamentos de Otimização de Parâmetros em SGBD;
- **Capítulo 3** — Fundamentos de Aprendizado de Máquina Aplicados à Otimização de SGBD;
- **Capítulo 4** — Trabalhos Relacionados e Lacunas da Literatura;
- **Capítulo 5** — Solução de Otimização de Parâmetros de BPOOL e Experimentos;
- **Capítulo 6** — Conclusões e Trabalhos Futuros.

Capítulo 2

Fundamentos da Otimização de Parâmetros em SGBD

Este capítulo apresenta os fundamentos da otimização de parâmetros em Sistema de Gerenciamento de Banco de Dados (SGBD), destacando conceitos essenciais para compreender o impacto da configuração no desempenho. Assin são discutidos os objetivos e desafios associados ao ajuste de parâmetros, com ênfase no papel do *Buffer Pool* (BPOOL) na hierarquia de memória e em suas particularidades no ambiente IBM Db2 for z/OS (DB2). Também são abordadas métricas relevantes e métodos tradicionais de otimização que servem como base conceitual para os capítulos seguintes.

O capítulo está organizado da seguinte forma: na Seção 2.1 discute-se o papel dos SGBDs no desempenho de sistemas; na Seção 2.2 apresentam-se os objetivos de otimização; na Seção 2.3 as técnicas e métodos de ajuste de parâmetros são revisadas; na Seção 2.4 são apresentadas o papel do BPOOL em SGBDs; na Seção 2.5 descrevem-se as particularidades do BPOOL no DB2; em seguida, na Seção 2.6 são discutidas métricas e desafios da otimização do BPOOL; por fim, na Seção 2.7, apresentam-se as considerações finais deste capítulo.

2.1 O Papel dos SGBDs e sua Influência no Desempenho

SGBD é um software especializado responsável por facilitar a definição, a criação, a manipulação e o compartilhamento de dados em um banco de dados. Esses sistemas promovem a organização eficiente das informações, assegurando sua integridade, consistência e segurança [1]. Um SGBD oferece às aplicações e aos usuários uma visão abstrata dos

dados, ocultando os detalhes da implementação física, além de fornecer mecanismos para controle de concorrência, recuperação de falhas e otimização de desempenho [13].

Entre os componentes fundamentais de um SGBD destacam-se [2]: (i) o gerenciador de *buffer*, encarregado de carregar e descarregar páginas de disco; (ii) o otimizador de consultas, que converte comandos *Structured Query Language* (SQL) em planos de execução eficientes; (iii) o gerenciador de transações, que garante as propriedades Atomicidade, Consistência, Isolamento e Durabilidade (ACID); e (iv) o gerenciador de armazenamento, responsável pela organização física dos dados. Tais módulos formam uma malha interdependente que sustenta desempenho, confiabilidade e escalabilidade.

Além da arquitetura interna, a relevância dos SGBDs pode ser observada nos diversos domínios em que são aplicados. Esses sistemas são onipresentes em contextos corporativos, científicos e governamentais, sustentando sistemas de informação e ambientes de processamento transacional e analítico, como *Online Transaction Processing* (OLTP) e *Online Analytical Processing* (OLAP) [13, 14].

O desempenho de um SGBD está diretamente associado à configuração de parâmetros críticos, como o tamanho do *Buffer Pool* (BPOOL), a dimensão das páginas de dados e as estratégias de pré-busca (do inglês, *prefetching*). Ajustes inadequados desses parâmetros podem causar gargalos de entrada/saída, aumento de latência e redução no *throughput* [15]. Essas características tornam o SGBD um componente estratégico na infraestrutura de armazenamento e análise de dados, impactando diretamente a eficiência operacional [15]. Dessa forma, a otimização de parâmetros em SGBDs, especialmente em ambientes de produção, é uma tarefa desafiadora. Essa complexidade decorre da interdependência entre parâmetros de configuração, da diversidade de cargas de trabalho e da necessidade de adaptação dinâmica a variações operacionais. Em função disso, estudos recentes têm explorado técnicas baseadas em Aprendizado de Máquina (AM) e métodos estatísticos como alternativas promissoras para automatizar o ajuste de parâmetros e melhorar o desempenho geral do sistema [7].

2.2 Categorias da Otimização em SGBDs

Uma forma relevante de categorização na área de otimização de um SGBD é a análise dos objetivos específicos que motivam os esforços de ajuste de desempenho. Sob essa perspectiva, é possível identificar sete principais categorias de objetivos de otimização [16]:

1. **Ajustes Gerais**: intervenções amplas que visam a melhoria global do desempenho do sistema, sem focar em componentes ou parâmetros específicos;

2. **Ajustes de Parâmetros**: otimizações voltadas para a calibração fina dos parâmetros de configuração do SGBD, como alocação de memória, uso de *cache*, tamanho de página ou estratégias de pré-busca (*prefetching*), com o objetivo de adaptar o sistema a diferentes cargas de trabalho;
3. **Seleção de Índices**: escolha automatizada ou assistida de índices para acelerar consultas, baseada em padrões de acesso e estatísticas de execução. Uma boa seleção de índices pode reduzir significativamente o tempo de resposta e a carga sobre o sistema;
4. **Materialização de *Views***: estratégias de pré-computação e armazenamento de resultados intermediários de consultas recorrentes, a fim de reduzir a latência de execução em acessos repetitivos ou computacionalmente intensivos;
5. **Elasticidade de Recursos**: adaptação dinâmica da infraestrutura, como CPU, memória ou nós distribuídos, em função da variação da carga, utilizando técnicas de escalonamento horizontal ou vertical;
6. **Gerenciamento de Armazenamento**: otimização do uso de espaço em disco e memória secundária por meio de compactação, *layout* eficiente de páginas, segmentação de dados, e políticas de descarte e retenção;
7. **Detecção de Consultas Ineficientes**: identificação e correção de consultas com planos subótimos, gargalos de desempenho ou uso inadequado de estruturas de acesso, com base em análise de estatísticas e planos de execução.

Entre essas categorias, o presente estudo adota como foco a classe de **Ajustes de Parâmetros**, alinhada diretamente ao objetivo de otimizar configurações do BPOOL em ambientes SGBD, especialmente sob cargas de trabalho críticas e em sistemas *mainframe*. A escolha dessa categoria também viabiliza uma comparação estruturada entre as diferentes soluções de ajuste automatizado de parâmetros existentes na literatura [16].

2.3 Técnicas e Métodos de Ajuste de Parâmetros

A configuração ideal de parâmetros em um SGBD visa maximizar o desempenho e/ou minimizar o uso de recursos computacionais. Essa tarefa é reconhecidamente Não Determinístico em Tempo Polinomial (NP) do tipo difícil, ou NP-difícil, devido à grande quantidade de parâmetros e suas complexas interdependências [11, 16].

Encontrar combinações eficazes de parâmetros requer conhecimento aprofundado da carga de trabalho e do comportamento interno do sistema, o que raramente é trivial. Cada tipo de aplicação pode demandar configurações específicas, tornando inviável a

existência de uma solução universal [17]. Em resposta a esse desafio, a literatura apresenta abordagens automatizadas baseadas em diferentes classes de técnicas. Métodos fundamentados em *Bayesian Optimization* (BO) aparecem como uma das linhas mais consolidadas, contemplando otimização bayesiana clássica, variantes com restrições e modelos probabilísticos aplicados ao ajuste dinâmico de parâmetros [18, 19, 20, 21, 22, 23]. Estratégias de aprendizagem por reforço profundo, principalmente aquelas baseadas em arquiteturas *Deep Deterministic Policy Gradient* (DDPG), exploram agentes contínuos para seleção adaptativa de configurações [24, 25, 26, 27, 28]. Complementarmente, soluções orientadas a *Deep Neural Network* (DNN), empregadas tanto para previsão de desempenho quanto para interpretação automática de características de carga, são exploradas em [29, 30].

Além dessas categorias principais, uma série de contribuições adota heurísticas avançadas, meta-otimização, busca generalista ou métodos híbridos para exploração do espaço de configuração, apresentando resultados relevantes em cenários práticos [31, 32, 33, 34, 35]. Trabalhos pioneiros relacionados ao ajuste de memória e análise de limitações reais de soluções autônomas de configuração também desempenham papel central na evolução histórica da área, como discutido em [36, 37]. A organização dessas contribuições em categorias evidencia a heterogeneidade metodológica existente e facilita a identificação das técnicas mais adequadas ao contexto investigado nesta dissertação.

Diversas abordagens para a classificação de técnicas de otimização em SGBDs consideram fatores como tempo de execução, esforço de treinamento, qualidade dos resultados e capacidade de adaptação a diferentes cenários [3, 12, 16, 38]. No entanto, a categorização mais amplamente aceita organiza os métodos segundo sua abordagem principal, dividindo-os em quatro grandes grupos [12]:

1. ***Search-based***: incluem estratégias como busca exaustiva, busca aleatória, algoritmos genéticos, regras especializadas e outras heurísticas. Essas abordagens exploram o espaço de configuração de forma sistemática ou estocástica, sendo adequadas em cenários com conhecimento limitado do domínio ou ausência de dados históricos [32, 31, 34, 33, 35, 36].

2. **Aprendizado de Máquina Tradicional**: utiliza técnicas supervisionadas e modelos probabilísticos, incluindo regressão, árvores de decisão, processos gaussianos e BO, para modelar a relação entre configurações e métricas de desempenho. São métodos interpretáveis, eficientes e dependentes de engenharia de atributos [18, 19, 20, 21, 22, 23, 37].

3. **Aprendizado Profundo**: aplica redes neurais profundas para representar relações complexas entre carga de trabalho, parâmetros e desempenho, oferecendo alto poder preditivo, porém com maior custo computacional e menor interpretabilidade [29, 30].

4. **Aprendizado por Reforço**: modela o ajuste de parâmetros como um processo sequencial no qual um agente aprende políticas de configuração com base no retorno observado, utilizando arquiteturas de *deep reinforcement learning* adequadas para espaços contínuos e dinâmicos [24, 25, 26, 27, 28].

A classificação apresentada fornece um arcabouço conceitual que auxilia na análise crítica das técnicas discutidas nos capítulos seguintes, facilitando a escolha de abordagens alinhadas às características específicas do problema de otimização de BPOOL em SGBD executando em *mainframe*.

2.4 *Buffer Pool* na Arquitetura dos SGBDs

O *Buffer Pool* (BPOOL) é um componente essencial do SGBD, responsável por armazenar em memória principal as páginas de dados mais frequentemente acessadas [8]. Essa estratégia visa minimizar operações de I/O com disco, reduzindo a latência das consultas e aumentando o desempenho geral do sistema. Diferentemente da memória virtual, o BPOOL é projetado especificamente para sustentar o processamento eficiente de transações e consultas [8, 13].

Limitações tecnológicas e de custo tornam inviável manter todos os dados em memória, exigindo políticas de substituição de páginas que priorizem dados em uso e com alta probabilidade de reutilização [39, 40]. A gestão eficiente dos BPOOLs é desafiadora devido à variabilidade das cargas de trabalho e à interdependência de parâmetros. Isso demanda conhecimento especializado por parte do *Database Administrator* (DBA) e dificulta ajustes manuais em tempo real.

Técnicas baseadas em AM têm sido exploradas para automatizar essa configuração, permitindo adaptação contínua e melhoria do desempenho [7]. Estudos demonstram que tais abordagens reduzem latência, melhoram a escalabilidade e diminuem custos operacionais [29, 41], com validação baseada em métricas como tempo médio de espera, tempo médio de execução de programas e taxa de sucesso em encontrar os dados em *buffer* [8, 13]. Embora o funcionamento básico do BPOOL seja comum entre os SGBDs, os parâmetros e as estratégias de configuração variam conforme a arquitetura e o perfil de carga [9]. No IBM Db2 for z/OS (DB2), por exemplo, cada BPOOL pode ser ajustado com base em limites de páginas sujas, pré-leitura e alocação de memória [13].

2.5 *Buffer Pool* no DB2 For Z/OS

O BPOOL em ambientes DB2 *mainframe* é o núcleo do gerenciamento de memória compartilhada, atuando como camada intermediária entre o armazenamento em disco e a

memória real. A sua função é reduzir operações de entrada e saída (I/O) ao reter páginas de dados e índices em cache, mitigando acessos repetidos a disco e melhorando o tempo de resposta. A configuração do BPOOL regula aspectos como substituição de páginas (*page steal*), pré-leitura sequencial (*sequential prefetch*), retenção de páginas modificadas (*dirty*) e escrita diferida (*deferred write*), todos com impacto direto sobre latência, previsibilidade de resposta (p95/p99) e vazão (*throughput*) [13].

Em ambientes transacionais, nos quais predominam operações curtas, concorrentes e de alta frequência, o BPOOL precisa garantir respostas rápidas e estáveis, evitando contenções que prejudiquem o tempo médio de transação. Já em ambientes analíticos e não transacionais, caracterizados por varreduras extensas e cargas em silo, o desafio recai sobre a estabilidade da cauda da distribuição de latência, pois variações em p95/p99 podem comprometer a execução de consultas complexas. Dessa forma, a calibragem dos parâmetros deve considerar o tipo de carga predominante, balanceando entre previsibilidade sob concorrência intensa e robustez diante de acessos sequenciais de grande volume [13].

Tabela 2.1: Parâmetros de configuração de BPOOL no DB2.

Parâmetro	**Descrição**	**Valores**
VPSIZE (*virtual pool size*)	Define o tamanho do BPOOL, em número de *buffers*; determina a capacidade efetiva em memória.	Ajustável até o limite máximo de memória;
VPMIN (*minimum pool size*)	Estabelece o tamanho mínimo permitido do BPOOL quando o ajuste automático está ativo.	≥ 0;
VPMAX (*maximum pool size*)	Define o limite máximo permitido do BPOOL para cenários com ajuste automático.	Definido pelo DBA; estabelece teto de crescimento.
PGSTEAL (*page steal*)	Configura a política de substituição de páginas quando não há *buffers* livres.	`LRU` (padrão), `FIFO` ou `NONE`.
PGFIX (*page fix*)	Controla a fixação de páginas em memória real, evitando paginação para armazenamento auxiliar.	`YES` ou `NO`.
FRAMESIZE (*frame size*)	Especifica o tamanho da moldura de memória real associada ao BPOOL.	`4K`, `1M` ou `2G`.
VPSEQT (*sequential steal threshold*)	Define a fração do BPOOL dedicada a operações de pré-leitura sequencial.	$\approx 80\%$ (intervalo 0–99%).
VPPSEQT (*parallel sequential steal*)	Estabelece a fração do BPOOL usada por operações de pré-leitura sequencial em paralelo.	Padrão = 50% de `VPSEQT`.

Continuação na próxima página

Tabela 2.1: Parâmetros de configuração de BPOOL no DB2 (continuação).

Parâmetro	**Descrição**	**Valores**
DWQT (*deferred write queue threshold*)	Limiar global de escrita diferida; quando excedido, inicia *flush*.	$\approx 30\%$.
VDWQT (*vertical deferred write threshold*)	Limiar de escrita diferida em nível de objeto; aciona *flush* por *pageset*.	$\approx 5\%$.
AUTOSIZE (*automatic resize*)	Ajuste automático do tamanho do BPOOL.	`YES` ou `NO`.

O balanço entre os parâmetros apresentados na Tabela 2.1 evidencia que o ajuste do BPOOL não é um exercício isolado, mas uma composição de decisões de compromisso. A ampliação de `VPSIZE` tende a reduzir falhas de cache, mas aumenta a pressão sobre a memória real; `PGSTEAL=LRU` costuma equilibrar cargas mistas, enquanto `NONE` só é viável quando todo o conjunto de páginas cabe em memória; `PGFIX=YES` eleva a previsibilidade, mas exige reserva permanente de memória física; molduras maiores em `FRAMESIZE` reduzem a sobrecarga de tradução de endereços, porém dependem de suporte específico do subsistema. Já os limiares de escrita diferida (`DWQT`/`VDWQT`) representam talvez o compromisso mais delicado, pois quando configurados com valores altos, acumulam um grande volume de páginas modificadas antes da escrita, o que aumenta a probabilidade de rajadas intensas de I/O; quando definidos com valores baixos, forçam o sistema a escrever em intervalos menores, aumentando a frequência de operações de saída, mas de forma mais distribuída e previsível. Esse equilíbrio é crítico, pois define se a pressão de escrita será concentrada em picos ou diluída em ciclos mais regulares [10, 13, 42].

Essa complexidade, ao mesmo tempo técnico e estratégico, revela que configurações aparentemente adequadas sob métricas médias podem mascarar gargalos relevantes. É justamente na análise da variabilidade e da cauda de latência que surgem os indícios mais úteis para ajustes refinados, antecipando os desafios que se tornam centrais quando se busca avaliar e otimizar o comportamento do BPOOL em cenários reais.

2.6 Métricas e Desafios na Otimização do *Buffer Pool*

O BPOOL atua como o principal componente de memória responsável por armazenar páginas de dados frequentemente acessadas. A sua correta configuração influência diretamente o desempenho do sistema. O SGBD geralmente associa cada objeto de banco de dados a

um grupo específico de *buffer*, permitindo segmentar e especializar o gerenciamento da memória [43].

A eficiência do BPOOL em um SGBD é tradicionalmente avaliada pela *taxa de acertos* (do inglês, *hit ratio*), que representa a proporção de acessos em que os dados já estão disponíveis na memória, evitando operações de entrada/saída em disco. A Equação 2.1 representa a fórmula clássica da taxa de acertos:

$$\textit{Hit Ratio} = \frac{\text{Número de hits}}{\text{Número total de requisições}} \times 100\% \tag{2.1}$$

Apesar de sua importância, o *hit ratio* isoladamente não captura aspectos como tempo de resposta, eficiência de escrita e leitura ou variações em padrões de acesso. É fundamental ampliar o conjunto de métricas consideradas, especialmente em contextos de cargas mistas ou requisitos de alta disponibilidade. Embora amplamente empregada, essa métrica fornece uma visão parcial do desempenho real, especialmente em ambientes com padrões de acesso voláteis e cargas de trabalho heterogêneas [13].

Assim, para uma avaliação mais precisa, é necessário considerar métricas adicionais que abrangem múltiplas dimensões do desempenho. Indicadores como a taxa de substituição de páginas, o tempo de residência na memória, os atrasos em operações de I/O síncronas e assíncronas, bem como o tempo médio de espera e o volume de dados lidos e escritos contribuem para uma visão holística do desempenho [13]. A relevância dessas métricas varia conforme o perfil de carga de trabalho. Em sistemas OLTP, por exemplo, a latência e a responsividade são prioritárias, enquanto em ambientes OLAP, a eficiência de leitura e o uso de memória assumem maior importância. Já em cargas mistas, é essencial equilibrar múltiplos indicadores de forma dinâmica [7].

A análise dessas métricas viabiliza a construção de modelos de predição e otimização, permitindo a aplicação de técnicas de AM para ajustes automáticos no BPOOL. Entretanto, essa otimização enfrenta desafios consideráveis. O primeiro diz respeito ao número elevado de parâmetros de configuração disponíveis nos diferentes SGBDs. O segundo envolve a interdependência entre esses parâmetros, que podem interagir de maneira não linear, tornando o ajuste manual impraticável [15, 44]. Além disso, as cargas de trabalho em ambientes modernos variam continuamente, exigindo adaptações frequentes nas configurações. Problemas como gargalos de CPU, limitações de I/O, restrições de memória e arquitetura de dados tornam a tarefa ainda mais desafiadora [45, 46].

Dessa forma, a manutenção eficiente dos BPOOLs exige revisões periódicas de configuração, considerando limiares de escrita, estratégias de pré-leitura (*prefetching*) e alocação de memória. A escalabilidade também representa um entrave: configurar manualmente dezenas ou centenas de instâncias de SGBDs é inviável. Além disso, à medida que o tamanho do BPOOL cresce, o otimizador de consultas passa a incorporar informações

sobre o estado da memória nos planos de execução [29]. Portanto, a gestão do BPOOL exige uma abordagem integrada, que é a análise detalhada de métricas, compreensão dos padrões de carga de trabalho, e mecanismos de adaptação automatizada. O desafio não reside apenas em identificar os melhores parâmetros, mas em mantê-los otimizados frente a ambientes dinâmicos e restrições operacionais complexas.

2.7 Considerações Finais

Este capítulo apresentou os fundamentos teóricos e conceituais necessários para compreender o problema da otimização de parâmetros em SGBD. Partindo da análise do papel dos SGBDs no desempenho de sistemas, foram discutidos os objetivos e desafios associados à otimização, revisadas técnicas clássicas de ajuste de parâmetros e detalhado o funcionamento do BPOOL, com destaque para suas particularidades no DB2 for z/OS. Também foram abordadas métricas relevantes e limitações de abordagens tradicionais, ressaltando a importância de soluções capazes de lidar com cargas de trabalho complexas e variáveis.

Esse conjunto de elementos fornece a base conceitual sobre a qual a investigação se apoia. A partir dele, abre-se espaço para explorar técnicas de AM que têm emergido como alternativas promissoras para enfrentar os limites dos métodos tradicionais, sobretudo em cenários de alta complexidade e grande volume de métricas. Essa discussão, desenvolvida no capítulo seguinte, aprofunda os fundamentos computacionais que sustentarão a construção da solução proposta nesta dissertação.

Capítulo 3

Aprendizado de Máquina na Otimização de Parâmetros SGBD

Este capítulo apresenta a base conceitual necessária para compreender como técnicas de AM sustentam a solução de otimização automática dos parâmetros de BPOOL em SGBD. A Seção 3.1 contextualiza o uso desses métodos no ajuste de BPOOL, destacando suas capacidades de modelar relações complexas entre parâmetros e métricas operacionais. Na Seção 3.2, o problema é estruturado em dimensões analíticas que orientam as decisões metodológicas adotadas ao longo da pesquisa, estabelecendo um fluxo que integra caracterização do espaço de busca, seleção de métricas relevantes e mecanismos de aprendizagem.

As seções subsequentes aprofundam os componentes que habilitam essa integração. A Seção 3.3 descreve os procedimentos de redução de dimensionalidade aplicados para estabilizar e simplificar o espaço de análise; a Subseção 3.3.4 introduz os critérios de seleção esparsa empregados para ranquear parâmetros de configuração; e a Seção 3.4 articula o papel dos modelos preditivos e das estratégias de otimização. Por fim, a Seção 3.5 estabelece a adaptação do CRISP-DM ao domínio de BPOOL, enquanto a Seção 3.6 fecha o capítulo sintetizando o arcabouço conceitual que orienta as etapas analíticas posteriores.

3.1 Aprendizado de Máquina na Otimização de Parâmetros

AM compreende algoritmos capazes de aprender relações complexas a partir de dados, ajustando modelos sem depender de regras programadas manualmente [47]. Por meio de técnicas estatísticas e computacionais, esses métodos capturam padrões não triviais, estruturam dependências e oferecem mecanismos preditivos escaláveis [48]. No contexto

de SGBD, essa capacidade viabiliza prever métricas operacionais, detectar anomalias e apoiar o ajuste de parâmetros críticos de BPOOL [49].

A literatura organiza o AM em três famílias principais [50]: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. Neste trabalho, tais famílias estruturam a base metodológica: LASSO e GPR (supervisionado), AFE e *K-Means* (não supervisionado), e DDPG e DDQN (reforço). Essa taxonomia sustenta as etapas subsequentes, incluindo redução de dimensionalidade (Seção 3.3) e modelagem preditiva.

No aprendizado supervisionado, os modelos utilizam pares entrada–saída para estimar funções capazes de prever variáveis de interesse [47]. Em SGBD, regressão linear, LASSO e abordagens baseadas em GPR são amplamente empregadas para estimar latência, identificar impacto de parâmetros e priorizar configurações relevantes [51]. Esses métodos envolvem definição de função de perda e otimização iterativa [52].

No aprendizado não supervisionado, o objetivo é descobrir estruturas latentes em dados não rotulados [48, 53]. Técnicas de agrupamento e análise de correlações segmentam perfis de carga e revelam padrões operacionais úteis para reduzir dimensionalidade, mitigar redundâncias e organizar o espaço de busca [54].

No aprendizado por reforço, um agente aprende políticas de decisão ao interagir com o ambiente e maximizar recompensas acumuladas [55]. Pesquisas recentes exploram métodos como *Q-Learning* e *Policy Gradient* para ajuste dinâmico de parâmetros de SGBD, investigando respostas adaptativas sob cargas variáveis [16].

Tabela 3.1: Comparação entre as famílias de Aprendizado de Máquina.

Família de AM	**Principais Características**	**Entrada Necessária**	**Aplicações em SGBD**	**Técnicas Típicas**
Supervisionado	Modelos treinados com pares entrada–saída; capacidade preditiva; dependência de função de perda e otimização iterativa.	Dados rotulados (atributos e alvo).	Estimativa de impacto de parâmetros; previsão de métricas operacionais; ranqueamento de configurações.	Regressão Linear; LASSO; Ridge; GPR; Redes Neurais.
Não Supervisionado	Extração de estruturas latentes; agrupamento e organização do espaço; redução de dimensionalidade.	Dados não rotulados.	Segmentação de perfis de carga; descoberta de padrões; agrupamento de métricas correlacionadas.	K-Means; DBSCAN; PCA; AFE; ICA.
Reforço	Aprendizado por interação agente–ambiente; políticas de decisão; maximização de recompensas acumuladas.	Sequências de estados, ações e recompensas.	Ajuste dinâmico de parâmetros; resposta adaptativa a mudanças de carga; otimização contínua.	Q-Learning; DDPG; DDQN; Policy Gradient; Actor–Critic.

A Tabela 3.1 sintetiza as três famílias, contrastando características, tipos de dados de entrada, aplicações no domínio de otimização de SGBD e exemplos de técnicas. Essa estrutura conceitual fundamenta as decisões metodológicas adotadas neste trabalho, que

combinam métodos supervisionados para modelagem preditiva e métodos não supervisionados para redução de dimensionalidade.

3.2 Etapas da Otimização de Parâmetros

A otimização de parâmetros de configuração de um SGBD com AM pode ser melhor compreendida se decomposta em quatro dimensões [51]: (i) seleção de parâmetros; (ii) seleção de métricas; (iii) método de ajuste; e (iv) técnicas de transferência. A Figura 3.1 sintetiza as etapas e técnicas centrais do processo.

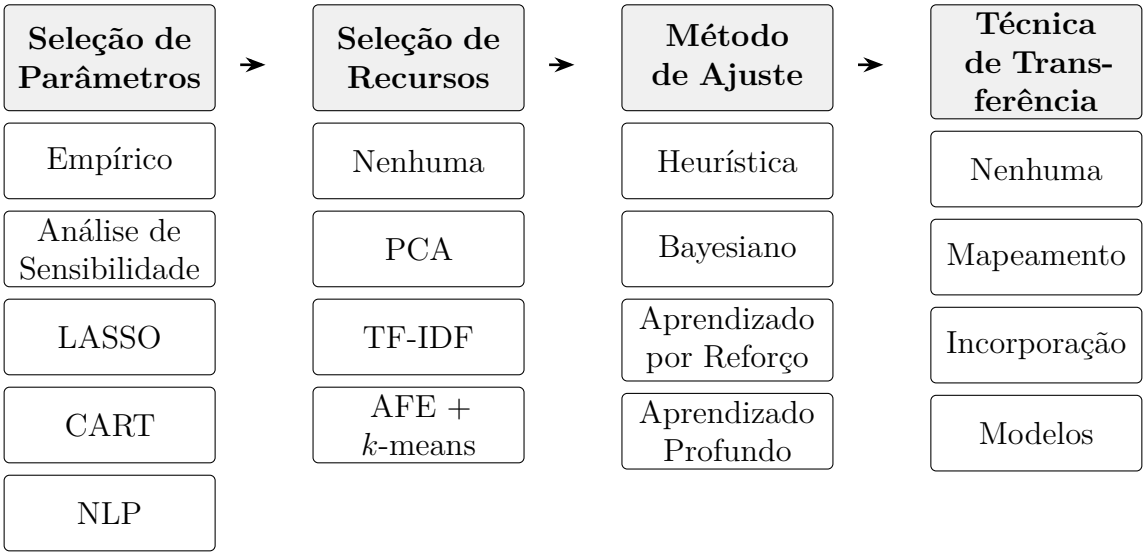

Figura 3.1: Etapas das soluções de otimização de parâmetros.

A primeira etapa, Seleção de Parâmetros, consiste em identificar como cada parâmetro (κ) está associado a um valor de configuração (ν_κ), ajustável em domínios contínuos ($\nu^{\min}, \nu^{\max}$) ou enumerados ($\nu^1, \ldots, \nu^k$). Pode ser empírica (conhecimento especializado, manuais) ou baseada em ranqueamento, via LASSO [19], *Classification and Regression Trees* (CART) [35] ou métodos de *Natural Language Processing* (NLP), como *Bidirectional Encoder Representations from Transformers* (BERT) [30].

Na etapa Seleção de Recursos, opcional em algumas soluções [18, 22, 27, 29, 31, 34, 32, 35, 36], busca-se caracterizar a carga de trabalho e reduzir a dimensionalidade das métricas, simplificando a otimização. Avaliam-se métricas como tempos de leitura/escrita, custo e perfil de carga; pode-se empregar *Term Frequency-Inverse Document Frequency* (TF-IDF) para representar padrões de consultas [30]. Métricas internas do SGBD podem ser reduzidas por *Principal Component Analysis* (PCA) [28] ou por Análise Fatorial Exploratória (AFE) combinada a *k-means* [19].

A terceira e principal etapa é o Método de Ajuste, no qual parâmetros e recursos são combinados para resolver o problema de otimização da configuração do SGBD. As abordagens incluem: (i) Heurística; (ii) Bayesiana; (iii) Aprendizado Profundo; (iv) Aprendizado por Reforço. O ajuste por Heurística pressupõe condução por DBA ou especialistas, que definem regras de alteração de parâmetros com base nas características da carga.

A BO é um procedimento iterativo de otimização de funções caras que combina um modelo substituto e uma função de aquisição. O substituto mais comum é o *Gaussian Process* (GP) [18, 19, 20, 21, 22, 23, 33, 35], mas há alternativas como *Sequential Model-based Algorithm Configuration* (SMAC) [35] e *Tree-structured Parzen Estimator* (TPE) [4]. O ciclo envolve amostra inicial (por exemplo, hipercubo latino), ajuste do substituto, otimização da aquisição, geralmente, *Expected Improvement* (EI), *Probability of Improvement* (PI) ou *Upper Confidence Bound* (UCB), avaliação do ponto recomendado e atualização do modelo até convergência.

Aprendizado Profundo utiliza *Deep Neural Network* (DNN) [29, 37] e *Convolutional Neural Network* (CNN) [29] para estimar e otimizar desempenho; embora eficazes, exigem mais dados e custo computacional maior que substitutos probabilísticos [37, 51]. O Aprendizado por Reforço busca políticas por tentativa e erro. Métodos como *Deep Deterministic Policy Gradient* (DDPG) [24, 28] e *Double Deep Q-Network* (DDQN) [27] lidam bem com muitos parâmetros, mas demandam mais iterações e tempo de processamento. Técnicas de *Genetic Algorithm* (GA) e abordagens baseadas em *Natural Language Processing* (NLP) (por exemplo, BERT) também aparecem na literatura [30].

Na Técnica de Transferência, avalia-se como reutilizar conhecimento do ajuste em cenários correlatos: reemprego do treinamento, *fine-tuning* em novos ciclos ou modelos prévios para aceleração de *bootstrap* de SGBDs. Algumas abordagens não preservam histórico, limitando reuso; outras mantêm *logs* e estados, permitindo acelerar configurações iniciais.

A metodologia desta análise enfatiza, além da descrição dos estudos, uma avaliação rigorosa e multidimensional como qualidade de validação empírica, aplicabilidade prática, escalabilidade, capacidade de generalização a diferentes sistemas e cargas, maturidade tecnológica e potencial de adoção. Esses critérios sustentam uma leitura crítica e fortalecem a contribuição científica desta dissertação.

3.3 Técnicas de Redução de Dimensionalidade

Ambientes com grande número de variáveis, especialmente em contextos de monitoramento de desempenho de SGBDs, frequentemente apresentam alta redundância, multicoline-

aridade e ruído estatístico. Esses fatores dificultam a modelagem preditiva e afetam negativamente tanto a eficiência computacional quanto a interpretabilidade dos resultados. Nesse cenário, técnicas de redução de dimensionalidade tornam-se fundamentais para simplificar o espaço de atributos, eliminar variáveis irrelevantes e identificar estruturas latentes que melhor representam a variabilidade dos dados.

Entre as abordagens mais relevantes para esse fim estão: o LASSO, o algoritmo de agrupamento *K-Means*; e a AFE. Embora partam de fundamentos estatísticos distintos, essas técnicas compartilham o objetivo de transformar conjuntos de dados de alta dimensionalidade em representações mais compactas e informativas, seja pela seleção de variáveis, pela extração de fatores ou pela segmentação de padrões.

O LASSO, por exemplo, realiza seleção automática de variáveis ao impor uma penalização nos coeficientes da regressão, forçando alguns deles a zero. Já o *K-Means* agrupa instâncias com base em similaridade, contribuindo para o reconhecimento de perfis operacionais ou cargas de trabalho semelhantes. Por fim, a AFE permite identificar fatores latentes responsáveis pelas correlações observadas entre variáveis de desempenho, revelando dimensões ocultas que influenciam o comportamento do sistema.

Essa convergência metodológica evidencia que, embora distintos em sua formulação, esses métodos atuam como filtros capazes de reduzir a complexidade do espaço de atributos. O uso dessas técnicas como etapa prévia à modelagem não apenas melhora a robustez e a generalização dos modelos preditivos, mas também permite interpretações mais consistentes sobre os determinantes do desempenho do SGBD. As próximas subseções detalham cada uma dessas abordagens com foco em seus fundamentos estatísticos e aplicações no contexto de otimização de sistemas.

3.3.1 Análise Fatorial Exploratória (AFE)

A Análise Fatorial Exploratória (AFE) é uma técnica estatística multivariada utilizada para identificar estruturas latentes em conjuntos de variáveis observadas, reduzindo a dimensionalidade dos dados ao explicar as correlações entre variáveis por meio de um número menor de fatores subjacentes [56]. A Figura 3.2 exemplifica um cenário comum em monitoramento de SGBDs, no qual múltiplas métricas de desempenho exibem alta correlação. A matriz de correlação visualiza, por meio de um *heatmap*, a intensidade das relações lineares entre as variáveis. Grupos de métricas fortemente correlacionadas (destacados nos quadrados pontilhados) indicam redundância e multicolinearidade, justificando o uso da AFE para consolidá-las em fatores latentes que capturam a variância compartilhada.Diferentemente de técnicas meramente descritivas, a AFE tem como objetivo revelar construtos teóricos não observáveis diretamente, mas que influenciam o comportamento das variáveis medidas [57].

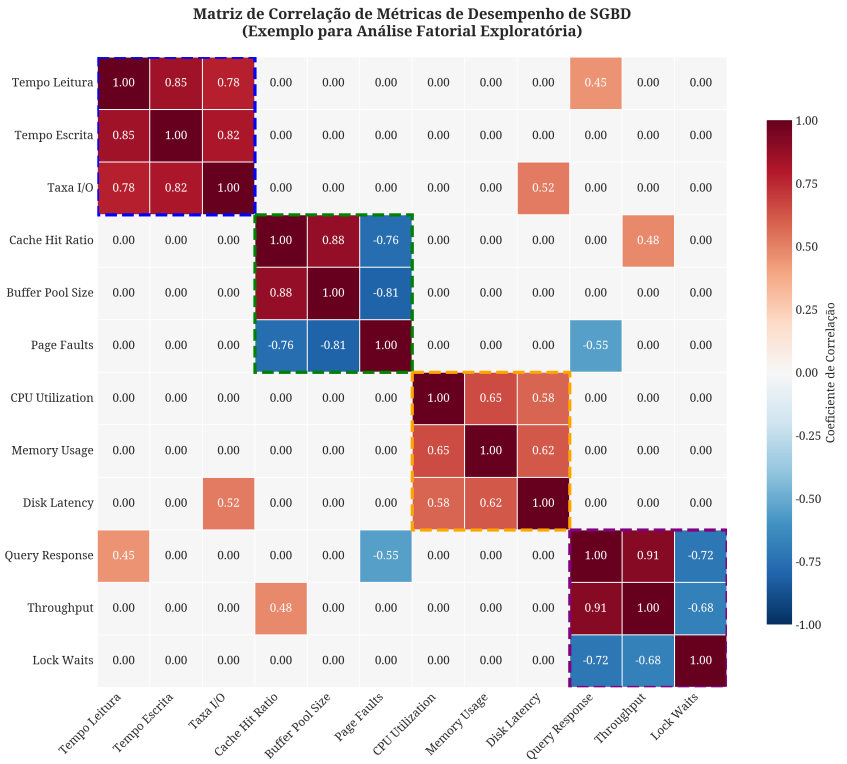

Figura 3.2: Exemplo de matriz de correlação entre métricas de desempenho de SGBD.

O processo de aplicação da AFE envolve uma sequência estruturada de etapas. Inicialmente, verifica-se a adequação dos dados à análise fatorial, por meio de testes estatísticos como o teste de esfericidade de Bartlett e o índice de Kaiser-Meyer-Olkin (KMO) [57]. Esses testes avaliam se existem correlações significativas entre as variáveis e se a estrutura dos dados é apropriada para a extração de fatores. Uma vez confirmada a adequação dos dados, procede-se à extração dos fatores latentes. O número ideal de fatores a ser retido pode ser definido por critérios estatísticos como o Critério de Kaiser, que recomenda a retenção apenas de fatores com autovalores superiores a 1, ou pela inspeção visual do *scree plot*, que identifica o ponto de inflexão na curva de autovalores [56]. Após a extração inicial, é aplicada uma técnica de rotação fatorial, como a rotação Varimax, com o objetivo de facilitar a interpretação ao maximizar a variância das cargas fatoriais [57].

A AFE tem sido amplamente aplicada em diferentes domínios, como psicometria, ciências sociais e engenharia [19, 37, 56]. No contexto de SGBDs, sua utilidade reside na capacidade de identificar métricas de desempenho que compartilham variância comum, consolidando-as em fatores representativos. Essa consolidação permite reduzir o número de variáveis monitoradas sem perda significativa de informação, favorecendo tanto a análise exploratória quanto a posterior modelagem preditiva [56].

Neste trabalho, a aplicação da AFE permitiu reduzir substancialmente a complexidade do conjunto de métricas de desempenho originalmente coletado. Essa redução resultou em

uma estrutura fatorial mais interpretável e robusta, promovendo ganhos em desempenho computacional, clareza analítica e eficiência dos modelos de aprendizado de máquina aplicados na fase de otimização. O fluxograma apresentado na Figura 3.3 detalha o processo metodológico da AFE, desde a verificação de adequação dos dados com os testes de Bartlett e KMO, passando pela extração e rotação dos fatores, até a validação e interpretação final. O diagrama evidencia a natureza iterativa do processo, que pode exigir ajustes no número de fatores para alcançar uma solução parcimoniosa e teoricamente coerente.

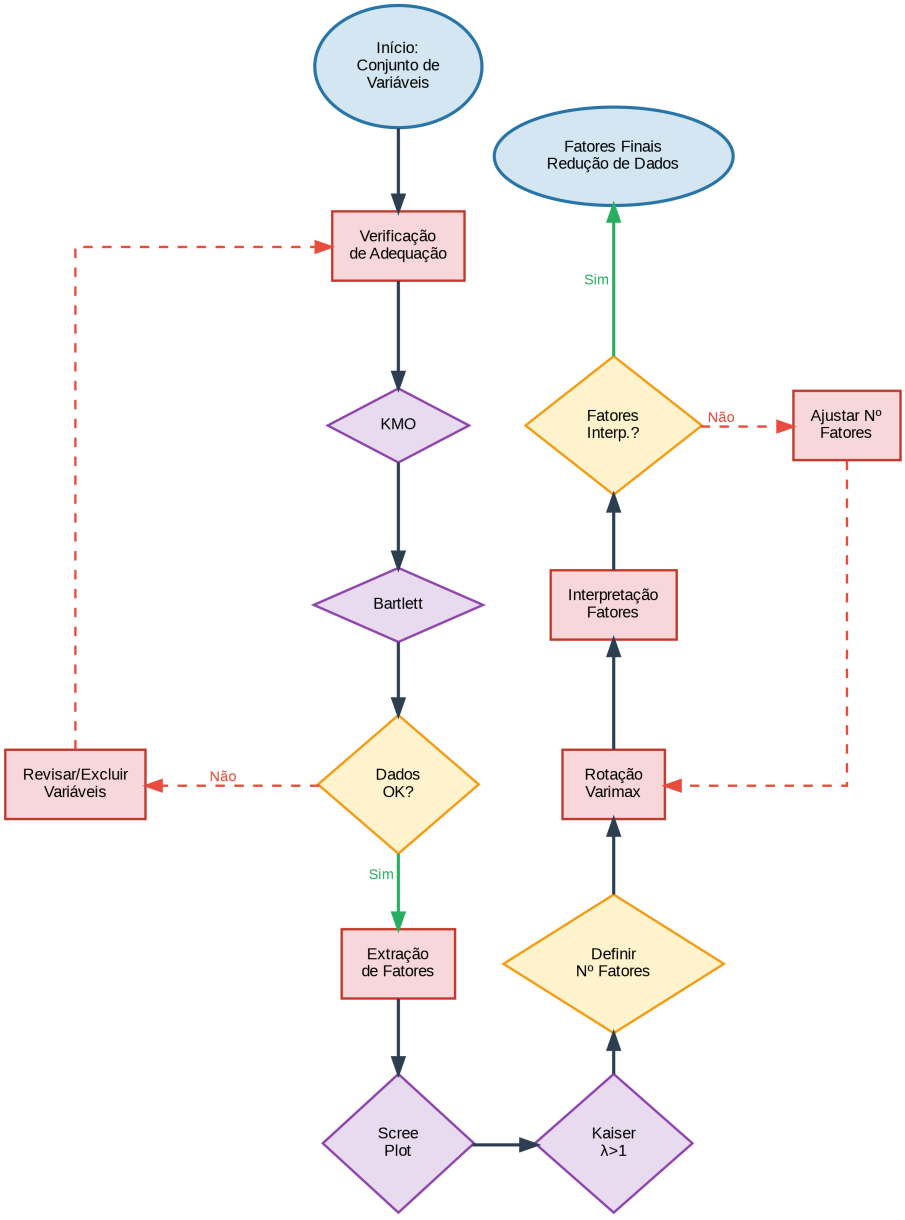

Figura 3.3: Fluxograma do processo de Análise Fatorial Exploratória (AFE).

Teste de Kaiser-Meyer-Olkin (KMO)

O teste de Kaiser-Meyer-Olkin (KMO) é uma medida estatística de adequação da amostragem, amplamente utilizada para avaliar se os dados são apropriados para análise fatorial exploratória. Diferentemente do teste de esfericidade de Bartlett, que verifica a existência de correlações, o índice KMO quantifica a proporção da variância entre as variáveis que pode ser atribuída a uma variância comum, associada a fatores latentes [56]. O valor do índice KMO varia entre 0 e 1, sendo que valores mais próximos de 1 indicam maior adequação dos dados à análise fatorial. De forma geral [58]:

- Valores acima de 0,80 são considerados excelentes;
- Entre 0,70 e 0,79 são considerados bons;
- Entre 0,60 e 0,69 são aceitáveis;
- Abaixo de 0,60 indicam que a amostra pode não ser adequada e requer revisão ou exclusão de variáveis.

A Tabela 3.2 apresenta uma convenção para a interpretação dos valores do índice KMO, conforme proposto na literatura [58]. A tabela serve como um guia prático para a tomada de decisão durante a etapa de validação da adequação dos dados para a AFE, indicando quando prosseguir com a análise ou quando se faz necessária uma revisão das variáveis incluídas.

Tabela 3.2: Interpretação dos valores do índice KMO e ações recomendadas.

Faixa de Valores	**Adequação**	**Ação Recomendada**
$> 0{,}80$	Excelente	Prosseguir com AFE
$0{,}70 - 0{,}79$	Boa	Prosseguir com AFE
$0{,}60 - 0{,}69$	Aceitável	Prosseguir com cautela; considerar revisão
$< 0{,}60$	Inadequada	Revisar ou excluir variáveis problemáticas

A fórmula do índice KMO é apresentada na Equação 3.1:

$$\text{KMO} = \frac{\sum_{i \neq j} r_{ij}^2}{\sum_{i \neq j} r_{ij}^2 + \sum_{i \neq j} p_{ij}^2} \tag{3.1}$$

Onde:

- r_{ij} são as correlações simples entre as variáveis i e j;
- p_{ij} são as correlações parciais entre as variáveis i e j.

Valores baixos de KMO sugerem que as correlações parciais entre as variáveis são elevadas, o que indica que elas não compartilham variância comum suficiente para justificar a extração de fatores. Em tais casos, recomenda-se remover variáveis com KMO individual abaixo de 0,60, o que pode melhorar a qualidade da estrutura fatorial a ser extraída [58]. No contexto deste trabalho, o teste KMO foi essencial para assegurar a validade estatística do processo de redução de dimensionalidade, garantindo que as variáveis incluídas na análise apresentassem variância compartilhada significativa.

Após verificar a adequação dos dados por meio dos testes de Bartlett e KMO, a próxima etapa da AFE envolve a determinação do número ideal de fatores a serem extraídos. Duas abordagens amplamente utilizadas para esse fim são o Critério de Kaiser e o *Scree Plot* (ou gráfico de sedimentação). O Critério de Kaiser estabelece que apenas fatores com autovalores (ou *eigenvalues*) superiores a 1,0 devem ser retidos, pois explicam mais variância do que uma única variável original. Esse critério é simples e objetivo, e é frequentemente adotado como ponto de partida na definição do número de fatores latentes [59]. No entanto, sua aplicação deve ser acompanhada de análise crítica, principalmente quando o número de variáveis é elevado.

Scree Plot

Complementarmente, o *Scree Plot* [60] fornece uma ferramenta visual que representa os autovalores em ordem decrescente. O ponto de inflexão da curva (conhecido como "joelho" ou "cotovelo") indica o número ideal de fatores a serem mantidos. Fatores localizados antes desse ponto explicam variância significativa, enquanto os seguintes tendem a representar apenas ruído estatístico.

A Figura 3.4 ilustra a aplicação conjunta do Critério de Kaiser e do *Scree Plot*. O gráfico exibe os autovalores em ordem decrescente, onde a linha tracejada representa o limiar de Kaiser (autovalor = 1). O *ponto de inflexão* na curva, destacado em laranja, sugere o número ideal de fatores a serem retidos — neste exemplo, quatro. Fatores à esquerda do ponto de inflexão explicam uma porção significativa da variância total, enquanto os à direita são geralmente considerados ruído estatístico. Essa abordagem combinada fortalece a decisão sobre a dimensionalidade da estrutura fatorial.

Esses métodos podem ser utilizados de forma combinada, isto é, o Critério de Kaiser fornece uma estimativa inicial e o *Scree Plot* permite validar visualmente essa escolha. Essa dupla abordagem fortalece a robustez do processo de extração fatorial, reduzindo a subjetividade e aumentando a confiabilidade dos resultados obtidos. Em contextos onde a AFE é aplicada como etapa preparatória para modelagem preditiva, como na redução de métricas de desempenho em SGBDs, a correta definição do número de fatores é crítica. Retenção excessiva pode introduzir ruído, e a retenção insuficiente pode resultar

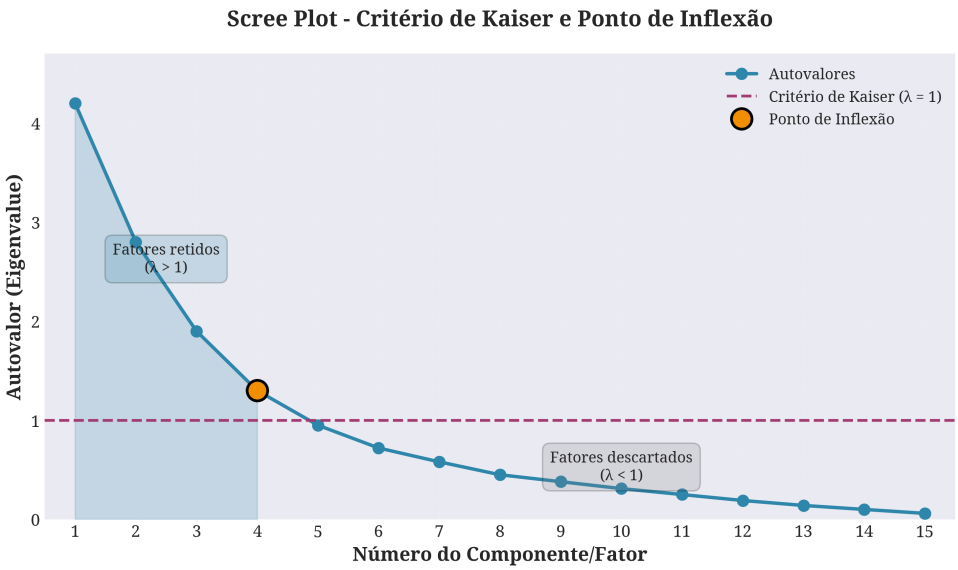

Figura 3.4: Exemplo de *Scree Plot* para determinação do número de fatores.

em perda de informação relevante. Neste trabalho, ambos os critérios foram empregados para garantir o equilíbrio entre complexidade e explicabilidade dos fatores extraídos.

3.3.2 Agrupamento *K-Means*

O algoritmo *K-Means* é uma técnica de agrupamento baseada em partição, amplamente utilizada no aprendizado não supervisionado para segmentar grandes conjuntos de dados em subgrupos internamente homogêneos [48]. O seu objetivo principal é minimizar a variabilidade intra-agrupamentos e maximizar a separação entre agrupamentos, organizando as instâncias com base em sua similaridade.

O funcionamento do *K-Means* parte da definição inicial do número de agrupamentos desejado, representado pelo parâmetro k. Em seguida, o algoritmo realiza uma alocação iterativa dos dados a grupos definidos por centroides, ajustando esses centroides com base na média dos pontos alocados em cada iteração. O processo é repetido até a convergência, geralmente quando as alocações não se alteram mais ou quando uma condição de parada é satisfeita [61]. A Figura 3.5 ilustra o fluxo iterativo do algoritmo *K-Means*. Partindo de uma definição inicial de k centroides, o processo alterna entre a atribuição de cada ponto de dados ao *cluster* mais próximo e o recálculo dos centroides com base na média dos pontos atribuídos. O ciclo se repete até que a posição dos centroides se estabilize, indicando a convergência para uma solução de agrupamento localmente ótima.

Apesar de sua simplicidade e eficiência, o *K-Means* apresenta limitações. Uma delas é a necessidade de definição prévia do número de agrupamentos, o que pode levar a agrupamentos subótimos. Além disso, o algoritmo tende a formar agrupamentos esféricos devido ao

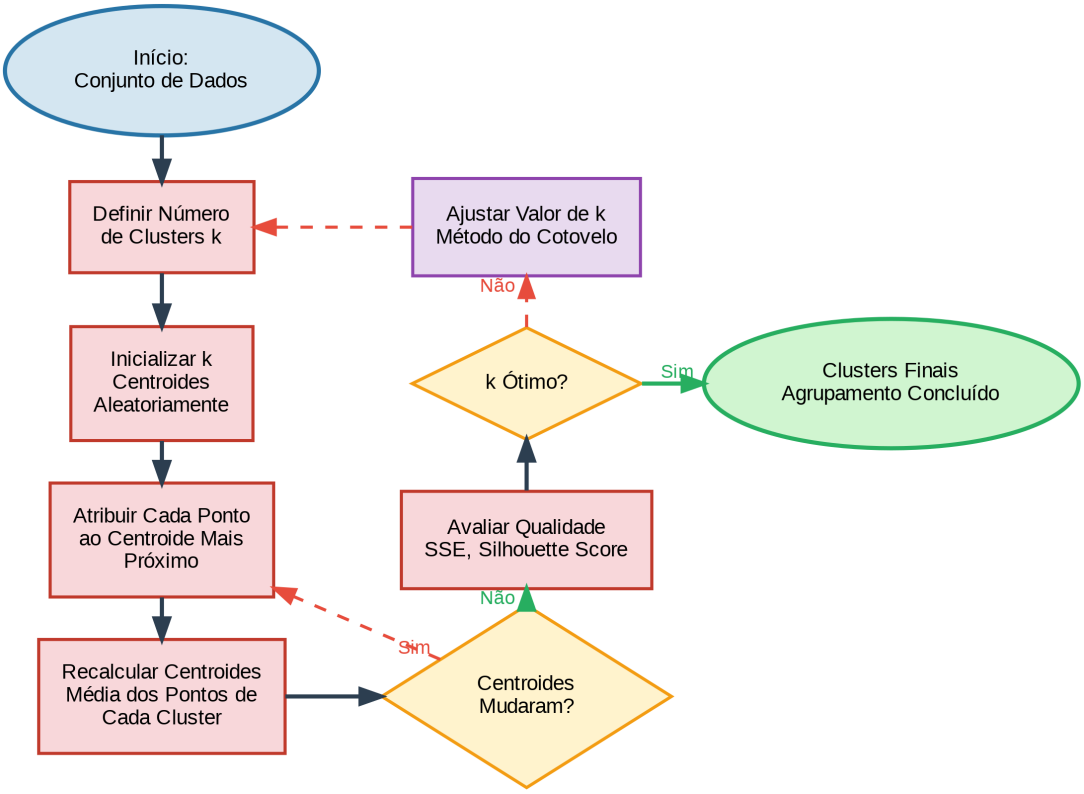

Figura 3.5: Fluxograma do algoritmo de agrupamento K-Means.

uso da distância euclidiana como métrica padrão, podendo apresentar desempenho inferior em estruturas com formatos mais complexos. Para lidar com esses desafios, estratégias como o método do cotovelo (do inglês, *elbow method*), e a análise de silhueta (do inglês, *Silhouette Score*), são comumente utilizadas para apoiar a escolha do valor de k.

Apesar dessas limitações, o *K-Means* tem sido amplamente utilizado na análise de cargas de trabalho em SGBDs, especialmente para identificar padrões de acesso, classificar períodos com comportamento similar ou agrupar perfis de utilização de recursos. Essa identificação de agrupamentos pode ser empregada tanto como técnica exploratória quanto como etapa preparatória para a aplicação de modelos supervisionados em contextos específicos. A Figura 3.6 apresenta um exemplo ilustrativo de agrupamento de dados em três agrupamentos após a aplicação do algoritmo *K-Means*.

3.3.3 Métricas de Avaliação de Agrupamento

Esta seção apresenta as métricas empregadas para avaliar a qualidade dos agrupamentos gerados nos experimentos. Tais métricas quantificam a coesão interna dos grupos e a separação entre grupos distintos, permitindo mensurar o desempenho dos algoritmos e verificar a consistência dos padrões identificados. A análise quantitativa resultante

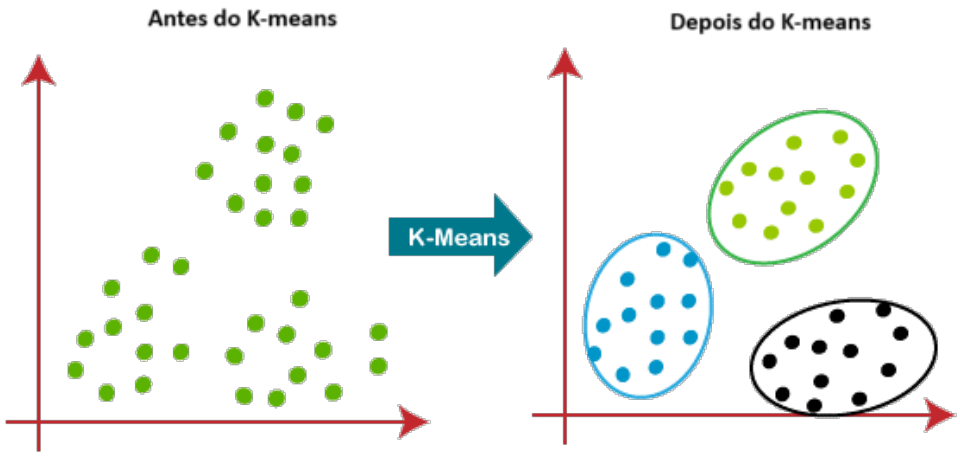

Figura 3.6: Exemplo de agrupamento após *K-Means* (Fonte: [62]).

fornece uma base objetiva para comparar diferentes abordagens de agrupamento e validar a estabilidade das estruturas detectadas. A seguir, descrevem-se as métricas adotadas neste trabalho.

Silhouette Score

O *Silhouette Score* [63], ou coeficiente de silhueta, é uma métrica utilizada para avaliar a qualidade de agrupamentos gerados por algoritmos de agrupamento, como o *K-Means*. Essa métrica considera simultaneamente a coesão interna dos agrupamentos e a separação entre eles, fornecendo uma medida robusta de desempenho do particionamento.

Para cada ponto i, o valor de silhueta é calculado com base em duas quantidades:

- $a(i)$: a distância média entre o ponto i e os demais pontos pertencentes ao mesmo agrupamento (medida de coesão);
- $b(i)$: a menor distância média entre o ponto i e os pontos de qualquer outro agrupamento ao qual i não pertence (medida de separação).

A fórmula para o cálculo do coeficiente de silhueta é dada pela Equação 3.2:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3.2}$$

O valor de $s(i)$ varia de -1 a 1:

- Valores próximos de 1 indicam que o ponto está bem posicionado dentro de seu agrupamento e bem separado dos demais;

- Valores próximos de 0 indicam sobreposição entre agrupamentos ou ambiguidade na atribuição do ponto;
- Valores negativos indicam que o ponto pode ter sido alocado ao agrupamento errado.

A Tabela 3.3 resume a interpretação dos valores do *Silhouette Score*. A métrica fornece uma avaliação quantitativa da qualidade do agrupamento. Assim, a tabela auxilia na rápida classificação do resultado como excelente, questionável ou inadequado, orientando a validação e o ajuste do número de *clusters*.

Tabela 3.3: Interpretação dos valores do *Silhouette Score*.

Valores	**Interpretação**	**Qualidade**
Próximo a +1	Ponto bem posicionado e separado	Excelente
Próximo a 0	Sobreposição ou ambiguidade	Questionável
Negativo	Possível atribuição incorreta	Inadequada

O *Silhouette Score* global é obtido pela média dos coeficientes de silhueta de todos os pontos do conjunto de dados. Essa média pode ser usada para comparar diferentes configurações de k e selecionar o número ideal de agrupamentos. Um valor médio elevado sugere uma estrutura de agrupamento bem definida, tornando o *Silhouette Score* uma ferramenta essencial na validação da qualidade de agrupamentos em experimentos de agrupamento.

Sum Of Squared Errors **(SSE)**

A *Sum of Squared Errors* (SSE) [62], é uma métrica amplamente utilizada para quantificar a compacidade dos agrupamentos gerados por algoritmos de agrupamento, como o *K-Means*. Ela representa a soma das distâncias quadráticas entre cada ponto e o centroide do agrupamento ao qual ele foi atribuído, funcionando como função objetivo a ser minimizada durante o processo de agrupamento. Matematicamente, a SSE é definida conforme a Equação 3.3:

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{3.3}$$

Onde:

- k é o número de agrupamentos;
- C_i representa o conjunto de pontos pertencentes ao agrupamento i;
- μ_i é o centroide do agrupamento i;

- $\|x - \mu_i\|$ é a distância euclidiana entre o ponto x e o centroide μ_i.

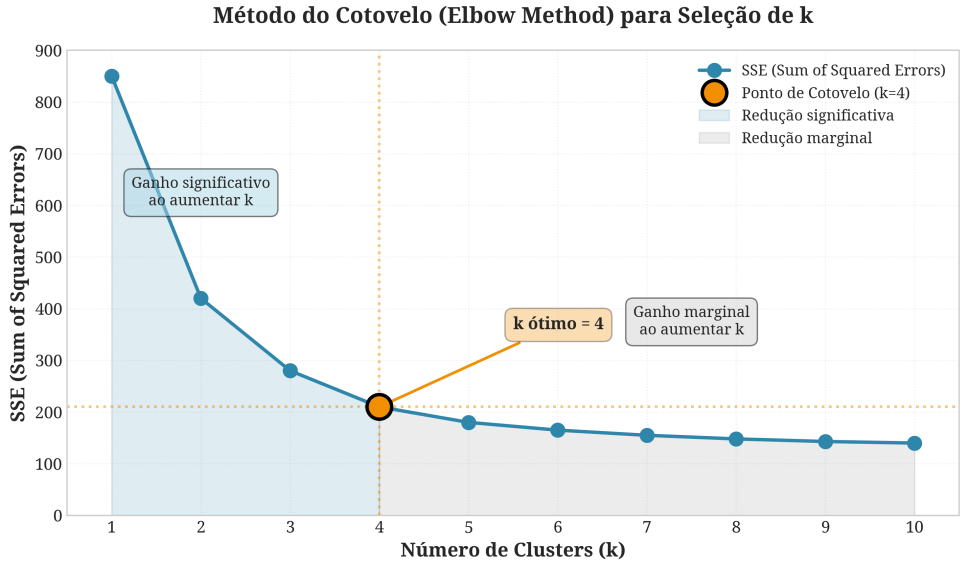

Figura 3.7: Ilustração do Método do Cotovelo (Elbow Method) para seleção de k.

Valores menores de SSE indicam agrupamentos mais compactos, com pontos bem agrupados ao redor de seus centroides. Contudo, a SSE tende a diminuir monotonamente com o aumento do número de agrupamentos, o que pode levar à supersegmentação do conjunto de dados. Para mitigar esse efeito, utiliza-se o método do cotovelo (do inglês, *elbow method*), que consiste em plotar os valores de SSE para diferentes valores de k e identificar o ponto de inflexão na curva. A Figura 3.7 demonstra visualmente essa técnica. O gráfico da SSE em função do número de *clusters* (k) mostra uma redução acentuada no início, que se torna marginal para valores maiores de k. O *ponto de cotovelo*, destacado na Figura em $k = 4$, representa o ponto de equilíbrio onde adicionar mais um *cluster* não resulta em uma redução significativa da variância intra-cluster, indicando o número ótimo de agrupamentos para o conjunto de dados. Esse ponto indica um valor de k que proporciona uma boa separação com complexidade mínima. A análise da SSE, portanto, é fundamental para avaliar a eficiência do particionamento e apoiar a escolha do número ideal de agrupamentos, especialmente em aplicações de agrupamento de métricas ou cargas de trabalho em SGBDs.

Partitioning Score (PS)

O *Partitioning Score* (PS) [61] é uma métrica que combina, em uma única razão, duas dimensões críticas da avaliação de agrupamentos: a coesão interna e a separação entre agrupamentos. Ele mede o quão compactos são os pontos dentro de cada agrupamento

(coesão) e o quão distintos são os agrupamentos entre si (separação), fornecendo um indicador global da qualidade do particionamento.

A fórmula geral do PS é expressa na Equação 3.4:

$$PS = \frac{\text{Separação média entre agrupamentos}}{\text{Coesão média dentro dos agrupamentos}} \tag{3.4}$$

Interpretando os termos, tem-se:

- A separação média entre agrupamentos refere-se à distância média entre os centroides dos diferentes agrupamentos;
- A coesão média interna mede a proximidade dos pontos em relação ao centroide do seu próprio agrupamento.

Valores elevados de PS indicam que os agrupamentos são bem definidos, ou seja, internamente consistentes e externamente separados. Por outro lado, valores baixos apontam para agrupamentos sobrepostos ou dispersos, sinalizando uma segmentação inadequada. Essa métrica é particularmente útil quando se deseja comparar diferentes algoritmos de agrupamento ou testar múltiplas configurações de parâmetros, como o número de agrupamentos k. Em contextos aplicados, como a identificação de padrões de desempenho em SGBDs, o PS pode orientar a seleção de modelos de agrupamento mais interpretáveis e representativos do comportamento do sistema.

3.3.4 Regressão LASSO

O método clássico de ajuste em regressão linear é o *Ordinary Least Squares* (OLS), que estima os coeficientes minimizando a soma dos quadrados residuais [64]. Embora amplamente utilizado, o OLS apresenta limitações importantes em cenários de alta dimensionalidade, pois suas estimativas, apesar de não viesadas, exibem alta variância, o que compromete a estabilidade preditiva, e não há mecanismo intrínseco de exclusão de variáveis irrelevantes, dificultando a interpretabilidade do modelo [19, 65].

Para superar essas deficiências, Tibshirani (1996) propôs a *Least Absolute Shrinkage and Selection Operator* (LASSO) [65], uma técnica de regressão linear regularizada que adiciona uma penalização L1 sobre a soma dos valores absolutos dos coeficientes. O problema de otimização é definido conforme apresentado na Equação 3.5:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{3.5}$$

Onde:

- y_i é a variável dependente para a observação i;

- x_{ij} é a j-ésima variável independente para a observação i;
- β_0 é o intercepto;
- β_j são os coeficientes de regressão;
- λ é o parâmetro de regularização que controla a força da penalidade;
- n é o número de observações;
- p é o número de variáveis independentes.

sendo que y_i é a variável dependente, x_{ij} representa as variáveis explicativas, β_j são os coeficientes de regressão, β_0 é o intercepto, e λ controla a intensidade da penalização. O termo $\lambda \sum_{j=1}^{p} |\beta_j|$ induz esparsidade, forçando alguns coeficientes a zero, de modo que o LASSO realiza simultaneamente a estimação dos parâmetros e a seleção de variáveis.

O parâmetro de regularização λ desempenha papel crítico no equilíbrio entre ajuste e simplicidade do modelo. Valores altos de λ resultam em maior esparsidade, reduzindo o conjunto de variáveis, enquanto valores próximos de zero aproximam a solução do OLS não regularizado. Diferentemente da regressão *Ridge*, que apenas encolhe coeficientes, o LASSO elimina variáveis irrelevantes, aumentando a interpretabilidade e reduzindo o risco de sobreajuste [66].

No contexto da otimização de SGBDs, a regressão LASSO mostra-se especialmente adequada para identificar e ranquear parâmetros de configuração. Ao aplicar a técnica, observou-se a eliminação de aproximadamente 30% dos parâmetros inicialmente considerados, por apresentarem baixa influência sobre a métrica alvo. Essa filtragem não apenas simplifica o espaço de busca, mas também orienta os esforços do DBA para os parâmetros mais relevantes, tornando o processo de otimização mais eficiente e explicável [19]. A Figura 3.8 ilustra o efeito da regularização LASSO sobre os coeficientes de um modelo. À medida que o parâmetro de penalização λ aumenta (no eixo horizontal, em escala logarítmica), os coeficientes das variáveis menos relevantes são progressivamente reduzidos a zero. Este processo de encolhimento resulta em um modelo mais esparso, onde apenas os parâmetros mais impactantes, que resistem por mais tempo à penalização, são mantidos, como destacado para os Parâmetros 1 e 2.

Portanto, a regressão LASSO constitui ferramenta essencial neste trabalho, ao oferecer uma solução estatisticamente robusta para seleção de variáveis em cenários de alta dimensionalidade. A sua capacidade de combinar regularização e esparsidade a torna particularmente útil para problemas de ajuste de BPOOL, em que é fundamental distinguir parâmetros críticos de configuração de variáveis redundantes ou irrelevantes.

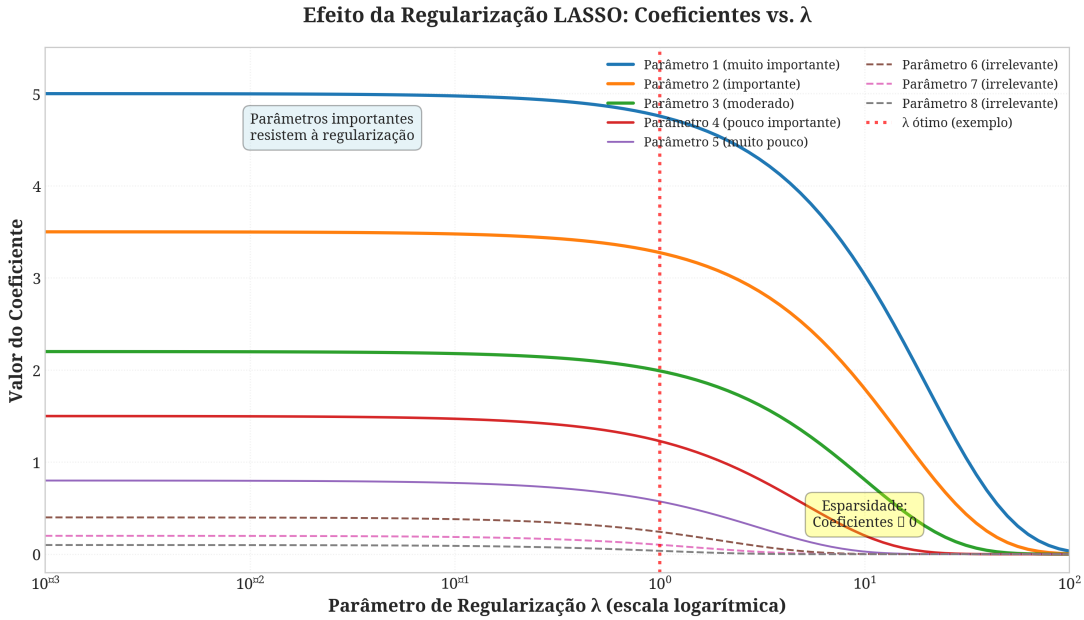

Figura 3.8: Efeito da regularização LASSO sobre os coeficientes do modelo.

3.4 Técnicas de Otimização de Parâmetros

A *Bayesian Optimization* (BO) [67] é uma técnica eficaz para otimizar funções complexas e de alto custo de avaliação. O seu princípio central é modelar a função objetivo como uma distribuição de probabilidade, de modo a selecionar de forma criteriosa os próximos pontos a serem avaliados. Para isso, utiliza-se um modelo substituto (*surrogate*), que aproxima a função real com menor custo computacional.

Neste trabalho, adota-se o modelo baseado em *Gaussian Process Regression* (GPR), o qual se mostra especialmente adequado por fornecer, além das predições pontuais da função objetivo, a variância associada a cada estimativa. Essa característica possibilita quantificar a incerteza e equilibrar de forma sistemática a exploração e a exploração, aspecto crucial no ajuste de parâmetros de SGBDs, em que as observações tendem a ser limitadas, ruidosas e de alta dimensionalidade [68].

Para tornar o problema tratável, emprega-se redução prévia de dimensionalidade. A AFE combinada com *K-Means* concentra variância em fatores representativos, enquanto o LASSO realiza seleção esparsa de variáveis. Essa etapa reduz redundâncias e melhora o condicionamento estatístico, o que é crítico, dado que o custo de treinamento do GPR cresce cubicamente com o número de amostras e é sensível ao número de dimensões.

A modelagem probabilística do GPR baseia-se na suposição de que a função desconhecida $f(\mathbf{x})$ segue um processo gaussiano, conforme apresentado na Equação 3.6. Essa formulação estabelece que qualquer subconjunto finito de pontos é distribuído de forma conjunta como uma normal multivariada, determinada por uma função média m e um

kernel de covariância k. O *kernel* atua como um operador de similaridade entre amostras, sendo o principal responsável pela expressividade do modelo. Em particular, a *Radial Basis Function* (RBF) impõe suavidade nas predições, enquanto a família *Matérn* oferece maior flexibilidade, capturando irregularidades típicas de cargas de trabalho em ambientes de produção.

$$f(\mathbf{x}) \sim \mathcal{GP}(m, k) \tag{3.6}$$

Para um conjunto de n observações $(X, \mathbf{y})$ sujeitas a ruído gaussiano com variância σ_n^2, as expressões da média e variância posteriores são apresentadas nas Equações 3.7 e 3.8. Essas equações descrevem o comportamento preditivo do modelo, permitindo estimar o valor esperado e a incerteza associada a cada ponto de entrada $\mathbf{x}$.

$$\mu(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\top \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}, \tag{3.7}$$

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}(\mathbf{x}), \tag{3.8}$$

em que $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ representa a matriz de covariância entre os pontos observados, $\mathbf{k}(\mathbf{x})$ é o vetor de covariâncias entre $\mathbf{x}$ e o conjunto de treinamento, e σ_n^2 corresponde à variância do ruído. Essa formulação confere ao GPR a capacidade de quantificar incertezas e, portanto, de equilibrar de forma explícita exploração e exploração, aspecto essencial em processos de otimização de parâmetros de SGBDs [68]. A Figura 3.9 ilustra o resultado de um modelo GPR. A linha tracejada azul representa a média preditiva ($\mu(x)$) do modelo, enquanto a área sombreada indica o intervalo de confiança de 95%, representando a incerteza ($\sigma(x)$). A incerteza é baixa nas proximidades dos pontos já observados (em vermelho) e alta em regiões inexploradas do espaço de parâmetros, guiando a busca por novas configurações promissoras.

As decisões de onde avaliar a função real são tomadas por funções de aquisição, que utilizam $\mu(\mathbf{x})$ e $\sigma(\mathbf{x})$ fornecidos pelo GPR. Nesta dissertação, adotam-se três funções clássicas: *Expected Improvement* (EI), *Probability of Improvement* (PI) e *Upper Confidence Bound* (UCB). Cada uma delas apresenta formulação matemática própria e implicações práticas distintas, cada uma com características distintas que as tornam adequadas para diferentes cenários de otimização . A escolha da função de aquisição mais adequada depende das características específicas do problema de otimização, como a dimensionalidade do espaço de parâmetros, a presença de ruído nas observações e as restrições computacionais. Em muitos casos, é benéfico utilizar múltiplas funções de aquisição em paralelo ou sequencialmente para explorar diferentes estratégias de busca [69].

A Tabela 3.4 oferece uma comparação detalhada entre as três funções de aquisição

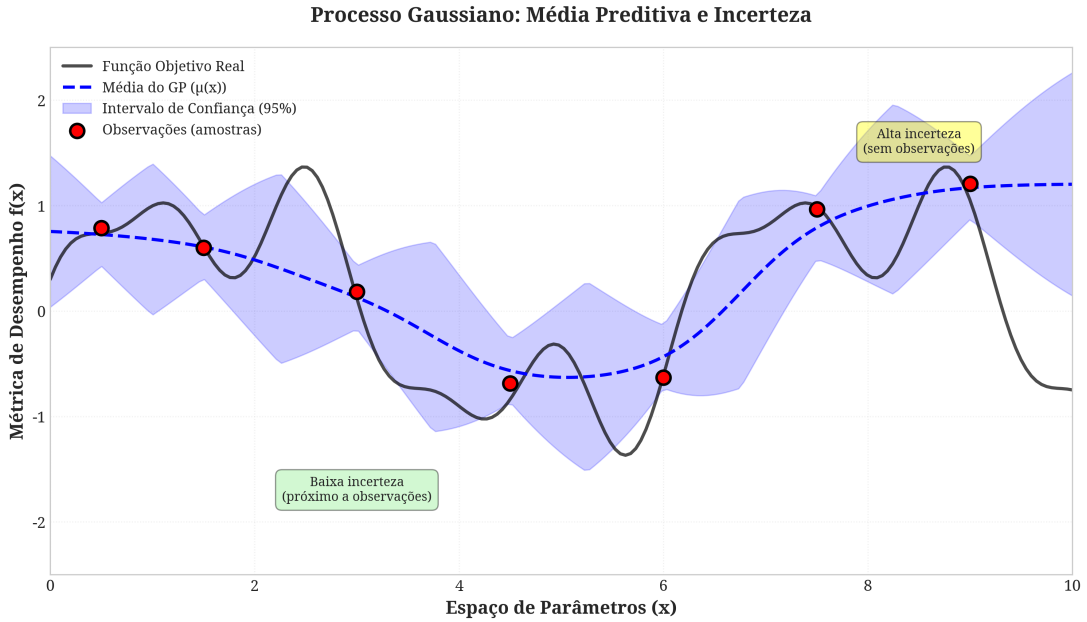

Figura 3.9: Representação de um Processo Gaussiano com média preditiva e incerteza.

utilizadas neste trabalho. A tabela contrasta a formulação matemática, o comportamento exploratório/exploratório, as principais vantagens e as limitações de cada abordagem, justificando sua aplicação complementar no processo de otimização bayesiana.

Tabela 3.4: Funções de aquisição utilizadas em Otimização Bayesiana.

Função	**Comportamento**	**Vantagens**	**Limitações**
EI	Busca pontos com maior ganho esperado, equilibrando média e incerteza.	Balanceamento automático; amplamente validada.	Menor desempenho em alta dimensionalidade; sensível a ruído.
PI	Avalia a probabilidade de superar o melhor valor atual.	Simples, interpretável e eficiente.	Ignora incerteza; pode superexplorar regiões estáveis.
UCB	Combina média e variância via parâmetro de confiança κ.	Controle explícito de exploração; fácil ajuste.	Requer calibração; risco de convergir para ótimos locais.

A função EI busca maximizar o ganho esperado em relação ao melhor valor já observado ($f^\star$). Ela favorece candidatos com alta média prevista, mas também incorpora a incerteza por meio do desvio padrão predito, equilibrando exploração e exploração. Sua formulação aparece na Equação 3.9:

$$\mathrm{EI}(\mathbf{x}) = (\mu(\mathbf{x}) - f^\star)\Phi(z) + \sigma(\mathbf{x})\phi(z), \quad z = \frac{\mu(\mathbf{x}) - f^\star}{\sigma(\mathbf{x})}. \tag{3.9}$$

As funções de aquisição consideradas neste trabalho são *Expected Improvement* (EI), *Probability of Improvement* (PI) e *Upper Confidence Bound* (UCB), formalizadas nas Equação 3.9, Equação 3.10 e Equação 3.11, respectivamente. Essas funções serão aplicadas no Capítulo 5 para guiar a exploração do espaço de configurações de BPOOL, onde Φ e ϕ denotam, respectivamente, a cdf e a pdf da Normal padrão. O EI é frequentemente considerado um critério natural, pois privilegia regiões promissoras sem negligenciar áreas onde o modelo ainda apresenta alta incerteza. Sua limitação é tornar-se conservador em espaços de alta dimensionalidade.

A função PI estima a probabilidade de que um ponto $\mathbf{x}$ produza uma melhoria sobre o melhor valor observado $f^\star$, conforme a Equação 3.10:

$$\mathrm{PI}(\mathbf{x}) = \Phi\left(\frac{\mu(\mathbf{x}) - f^\star}{\sigma(\mathbf{x})}\right). \tag{3.10}$$

Esse critério é simples e computacionalmente eficiente, mas tende a ignorar regiões com elevada incerteza, tornando-se excessivamente exploratório quando há ruído significativo. Em aplicações práticas, costuma ser ajustado com margens adicionais para modular o equilíbrio exploração–exploração. O UCB combina média e variância de forma linear, permitindo controlar explicitamente o grau de exploração por meio do parâmetro κ, conforme a Equação 3.11:

$$\mathrm{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\,\sigma(\mathbf{x}). \tag{3.11}$$

Valores elevados de κ induzem comportamento mais exploratório, enquanto valores menores privilegiam regiões já identificadas como promissoras. Sua principal vantagem é a transparência na parametrização, permitindo ajustar a intensidade da busca segundo os recursos disponíveis [70]. A Figura 3.10 compara visualmente o comportamento das três funções de aquisição: o modelo GPR é apresentado no painel inicial e os demais painéis mostram EI, PI e UCB, onde cada pico identifica a recomendação de ponto subsequente segundo cada estratégia. Observa-se que, enquanto EI e PI priorizam regiões de alta probabilidade de melhoria, o UCB tende a explorar áreas de maior incerteza, explicitando a natureza distinta do equilíbrio exploração–exploração imposto por cada função.

O processo de otimização segue iterativamente, inicialização com amostra diversificada, ajuste do GPR no espaço reduzido, aplicação da função de aquisição, avaliação do ponto selecionado e atualização do modelo. Esse ciclo repete-se até atingir o critério de parada, como limite de iterações ou convergência. A Figura 3.11 ilustra o ciclo completo da Otimização Bayesiana. O processo inicia com uma amostragem inicial, seguida pelo ajuste do modelo substituto (GPR). A função de aquisição é então otimizada para selecionar o próximo ponto candidato, que é avaliado na função objetivo real. O resultado realimenta o

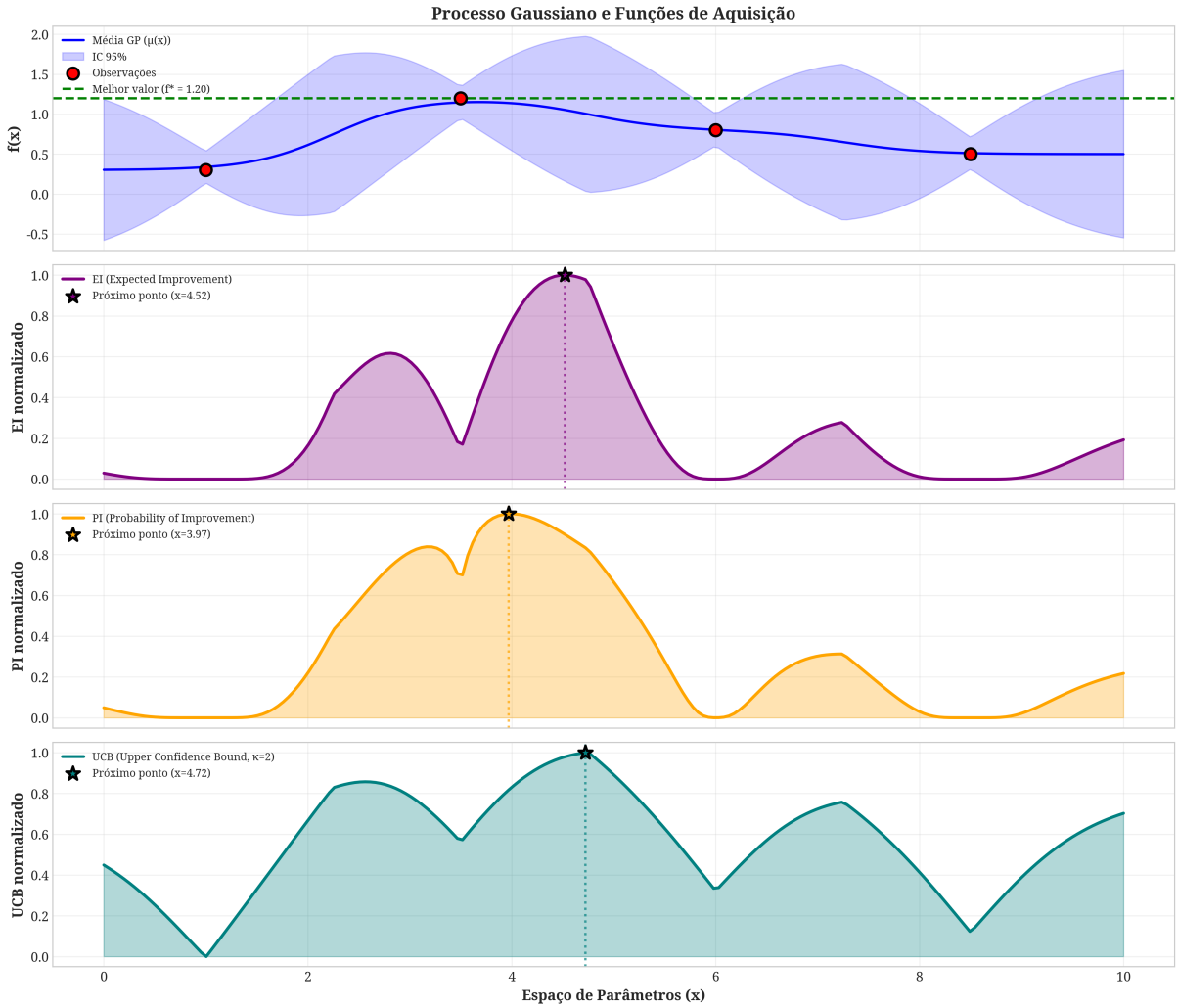

Figura 3.10: Comparação visual do comportamento das funções de aquisição EI, PI e UCB.

modelo, que é atualizado, e o ciclo se repete até que um critério de parada seja satisfeito, convergindo para a configuração ótima.

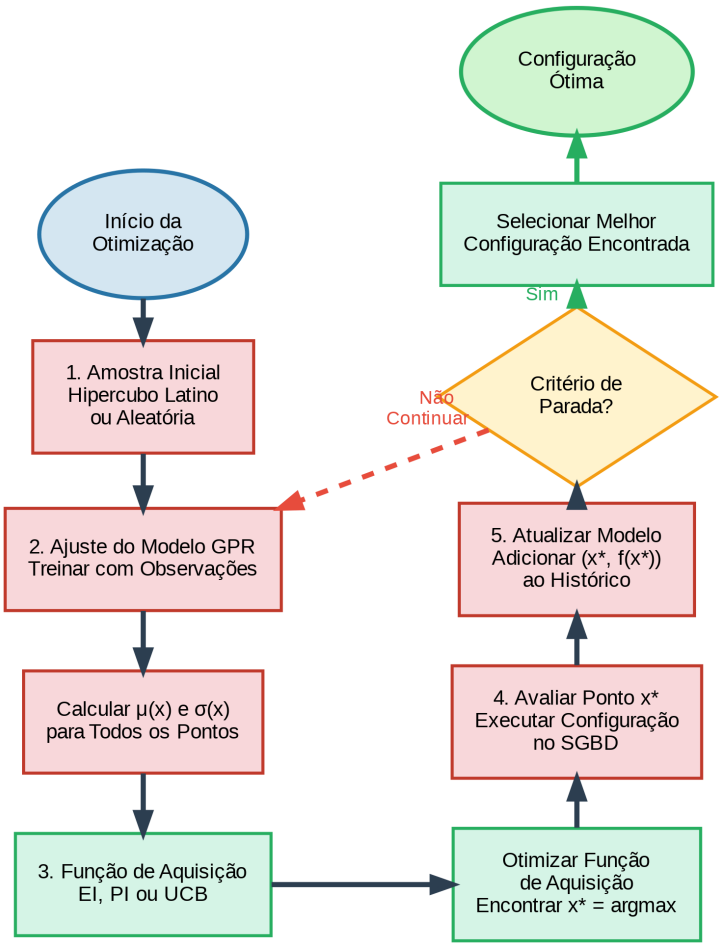

Figura 3.11: Ciclo iterativo da Otimização Bayesiana (BO).

A integração entre redução de dimensionalidade, GPR e funções de aquisição garante uma exploração mais eficiente do espaço de parâmetros. Além de acelerar a convergência, essa abordagem preserva auditabilidade, explica a variabilidade dos resultados e suporta ambientes críticos de SGBDs, nos quais estabilidade e rastreabilidade são requisitos fundamentais [70]. A aplicação de GPR neste contexto oferece diversas vantagens em relação a métodos tradicionais de otimização. Em primeiro lugar, esses modelos conseguem representar funções complexas e não lineares mesmo com um número relativamente reduzido de observações. Além disso, fornecem naturalmente uma medida de incerteza associada às previsões, o que possibilita equilibrar a exploração de regiões pouco conhecidas do espaço de parâmetros com a exploração mais intensa de áreas já identificadas como promissoras. Outro aspecto relevante é a robustez diante de ruídos nas observações, característica especialmente importante em ambientes de banco de dados reais, nos quais o desempenho pode ser influenciado por fatores externos [68].

Para a avaliação da qualidade do processo de otimização é importante a utilização de alguma métrica de qualidade. O R^2 é uma medida estatística que indica a proporção da variabilidade dos dados que é explicada pelo modelo de regressão. Em outras palavras, o R^2 mede a adequação do modelo aos dados observados. Um valor de R^2 próximo de 1 indica que o modelo explica uma grande parte da variabilidade dos dados, enquanto um valor próximo de 0 indica que o modelo não explica bem a variabilidade dos dados [48].

No contexto de regressão linear, o R^2 indica a porcentagem da variação na variável dependente que é explicada pelas variáveis independentes incluídas no modelo. Portanto, um R^2 alto sugere que as variáveis independentes estão bem relacionadas à variável dependente e que o modelo de regressão é útil para fazer previsões. Por outro lado, um R^2 baixo sugere que as variáveis independentes não estão bem relacionadas à variável dependente e que o modelo pode não ser adequado para fazer previsões precisas [48]. A fórmula do R^2 em uma regressão linear simples é dada pela Equação 3.12:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{3.12}$$

Onde:

- y_i são os valores observados da variável dependente;
- $\hat{y}_i$ são os valores previstos pela regressão;
- $\bar{y}$ é a média dos valores observados; e
- o somatório é feito para cada observação i.

Essa fórmula representa a proporção da variabilidade dos dados explicada pelo modelo de regressão. Um valor próximo de 1 indica um bom ajuste do modelo aos dados, enquanto um valor próximo de 0 indica um ajuste ruim [48].

3.4.1 Limitações das Técnicas Empregadas

As técnicas que compõem a solução de otimização operam de forma complementar, mas cada uma impõe restrições que precisam ser consideradas para garantir aplicabilidade e previsibilidade em ambientes de SGBD. Reconhecer essas limitações é essencial para avaliar a robustez do arcabouço e orientar decisões de implementação.

A AFE reduz dimensionalidade ao condensar métricas correlacionadas em fatores latentes [57, 59, 60]. Essa síntese diminui ruído e estabiliza a modelagem, mas a definição do número de fatores (k) é sensível e, em alguns casos, os fatores resultantes apresentam baixa interpretabilidade, restringindo seu uso para diagnóstico [58].

O *K-Means* complementa a AFE ao estruturar grupos homogêneos e selecionar métricas representativas. Apesar de reduzir redundância e organizar o espaço de busca, o método preserva limitações clássicas: dependência da escolha de k, sensibilidade à inicialização e possibilidade de convergência a ótimos locais [62].

O LASSO funciona como filtro esparso para ranqueamento de variáveis, destacando parâmetros de BPOOL mais relevantes para as métricas de desempenho. Essa capacidade fortalece a interpretabilidade, mas, sob colinearidade elevada, o método pode produzir seleções instáveis. Além disso, a penalização λ exige calibração criteriosa [65].

O GPR opera como modelo substituto, fornecendo estimativas e incertezas fundamentais para orientar a otimização bayesiana, conforme a formulação clássica de Rasmussen e Williams [67]. Contudo, seu custo computacional cresce como $\mathcal{O}(n^3)$ e o desempenho depende fortemente do *kernel* adotado e da qualidade do pré-processamento.

As funções de aquisição EI, PI e UCB fecham o ciclo da otimização, guiando a escolha de novas configurações. Referências consolidadas em otimização bayesiana [71, 72] e implementações modernas [70, 73] destacam a sensibilidade dessas funções à parametrização dos hiperparâmetros (ξ, κ), cujo ajuste inadequado pode induzir buscas miopes ou aprisionamento em ótimos locais.

A Tabela 3.5 sintetiza o papel de cada técnica e os principais riscos associados. Essa visão integrada reforça a necessidade de controles metodológicos para mitigar instabilidade e garantir que o processo de recomendação de parâmetros de BPOOL opere com confiabilidade e consistência.

Tabela 3.5: Quadro-síntese: conceitos, papel na solução e riscos/limitações.

Conceito	**Uso na solução**	**Riscos/limitações**
AFE	Reduz espaço e ruído	Fatores pouco interpretáveis; escolha de k sensível
K-Means	Representatividade por grupos	Dependência de k; sensível à inicialização
LASSO	Esparsidade e ranqueamento	Subseleção sob colinearidade; escolha de λ
GPR	Modelo *surrogate* com incerteza	Custo $\mathcal{O}(n^3)$; sensível ao *kernel*
EI/PI/UCB	Política de aquisição	Dependência de ξ, κ; risco de ótimos locais

A identificação clara das vantagens e limitações das técnicas adotadas evidencia a necessidade de um arcabouço metodológico sólido que organize o fluxo de atividades, mitigue riscos e favoreça a reprodutibilidade. Nesse sentido, a adoção do *Cross-Industry Standard Process for Data Mining* (CRISP-DM) fornece um referencial estruturado para alinhar objetivos de negócio, exploração de dados, preparação e modelagem, conforme as diretrizes originais e análises recentes do processo [74, 75]. Esse enquadramento metodológico é essencial para transformar um conjunto de técnicas isoladas em uma

solução coesa de otimização de parâmetros de BPOOL, com maior previsibilidade de resultados e menor exposição a falhas operacionais.

3.5 Fundamentação Metodológica

O modelo CRISP-DM constitui o processo de referência mais amplamente adotado em projetos de ciência de dados [75]. A sua relevância decorre da capacidade de estruturar de forma iterativa e disciplinada as etapas que vão do entendimento do problema de negócio à implantação dos resultados, garantindo tanto rigor técnico quanto aplicabilidade prática. Trabalhos recentes destacam que a longevidade do CRISP-DM deve-se à sua neutralidade em relação a ferramentas e técnicas específicas, o que o torna adaptável a diferentes domínios de aplicação, incluindo otimização de sistemas complexos [74, 76].

Na presente pesquisa, a adoção do CRISP-DM não se limitou a uma escolha metodológica de conveniência, mas constituiu a base para garantir experimentos auditáveis, reprodutíveis e alinhados a requisitos de governança. Conforme ilustrado na Figura 3.12, o modelo impõe uma sequência estruturada de fases, Entendimento do Negócio, Entendimento dos Dados, Preparação, Modelagem, Avaliação e Implantação, que permite organizar ciclos curtos de validação e documentar de forma rastreável cada decisão. Esse rigor é essencial em ambientes corporativos, no qual restrições de risco e janelas de mudança exigem previsibilidade e capacidade de reversão segura.

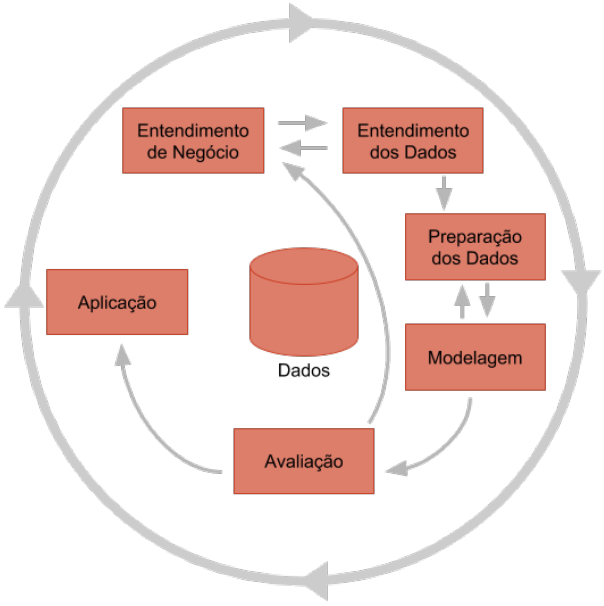

Figura 3.12: Fases do modelo de referência CRISP-DM (Fonte: [75]).

A literatura evidencia que ambientes OLAP, como *data warehouses* e silos analíticos, diferem substancialmente de cargas OLTP, isto é, apresentam maior heterogeneidade temporal, sazonalidade acentuada e menor repetitividade de padrões [74]. Nessas condições, a divisão clara entre preparação, modelagem e avaliação não é apenas uma formalidade metodológica, mas um mecanismo de mitigação de riscos, evitando viés por contaminação de amostras e permitindo o reuso de artefatos (transformações, modelos, relatórios) sob controle de versão. O CRISP-DM funciona, assim, como fio condutor para tratar esses vetores de incerteza com transparência técnica e governança.

Portanto, a adoção e adaptação do CRISP-DM asseguram que a solução desenvolvida não seja apenas tecnicamente eficaz, mas também metodologicamente sólida e operacionalmente viável. A metodologia confere disciplina científica à solução, sustentando tanto a validade empírica dos resultados quanto a confiabilidade exigida em ambientes críticos de SGBD.

3.6 Considerações Finais

Este capítulo consolidou os fundamentos conceituais e metodológicos que sustentam a proposta de otimização automática dos parâmetros de BPOOL em SGBD. Para isso, foram discutidas as principais vertentes do AM supervisionado, não supervisionado e por reforço e sua aplicabilidade no ajuste dinâmico de parâmetros sob métricas complexas e cargas heterogêneas. O problema foi estruturado em dimensões operacionais que orientam a solução: seleção de parâmetros, definição e redução de métricas, método de ajuste e transferência de conhecimento entre cenários.

As técnicas de redução de dimensionalidade e agrupamento (AFE com *K-Means*) foram apresentadas como mecanismos para sintetizar correlações e eliminar redundâncias, preparando o terreno para a seleção esparsa de variáveis via regressão LASSO. Em relação à essa base, a modelagem probabilística por GPR se consolidou como núcleo da abordagem, explorando sua capacidade de gerar previsões acompanhadas de incerteza, atributo essencial em ambientes ruidosos e de alta dimensionalidade. As funções de aquisição EI, PI e UCB foram tratadas como estratégias complementares para balancear exploração e aproveitamento do espaço de configuração.

A análise das técnicas evidenciou que nenhuma delas, isoladamente, atende aos requisitos de robustez e auditabilidade exigidos em sistemas corporativos. A integração entre redução de dimensionalidade, seleção esparsa e otimização bayesiana forma um arcabouço coeso, porém sensível a hiperparâmetros e custos computacionais. Para mitigar esses riscos, o CRISP-DM foi adotado como eixo metodológico, garantindo reprodutibilidade, rastrea-

bilidade e governança, aspectos críticos em ambientes de missão crítica que demandam previsibilidade e *rollback* seguro.

Com essa base consolidada, o capítulo seguinte se volta à análise sistemática dos trabalhos relacionados. O objetivo será avaliar criticamente como a literatura acadêmica e industrial tratou a seleção de parâmetros, a redução de dimensionalidade e a modelagem probabilística, bem como quais alternativas de funções de aquisição e metodologias foram empregadas. Essa revisão permitirá identificar convergências, contrastes e lacunas, posicionando a presente proposta no estado da arte, e justificando a sua relevância como contribuição inédita para a otimização automática de parâmetros de BPOOL em SGBD.

Capítulo 4

Trabalhos Relacionados

Este capítulo apresenta uma análise abrangente e crítica dos trabalhos relacionados à otimização automática de parâmetros em SGBD, com ênfase nas abordagens baseadas em AM. O objetivo é demonstrar a evolução conceitual e metodológica da área, evidenciando como as soluções avançaram de heurísticas fixas para modelos probabilísticos, técnicas de *Deep Learning* (DL) e, mais recentemente, soluções baseadas em *Large Language Model* (LLM). O capítulo foi dividido em cinco seções. a Seção 4.1 descreve a evolução cronológica das abordagens e consolida os marcos metodológicos. A Seção 4.2 discute especificidades técnicas de ambientes *mainframe* e o impacto na configuração do BPOOL, aprofunda os desafios técnicos, operacionais e organizacionais que condicionam a adoção de soluções automatizadas e sintetiza as lacunas e oportunidades de pesquisa identificadas. Por fim, a Seção 4.3 consolida as considerações finais que direcionam a metodologia proposta no capítulo subsequente.

4.1 Análise Sistemática

A revisão cobre publicações entre 2006 e 2025, priorizando estudos revisados por pares e soluções com validação empírica. Foram desconsideradas proposições puramente heurísticas, pois, embora relevantes historicamente, não oferecem capacidade adaptativa nem inferência estatística sobre estados desconhecidos. Entretanto, reconhece-se o papel formativo de abordagens que contribuíram para o amadurecimento metodológico da área, como *DB-BERT* [30], Tuneful [21], ReIM [33], CGPTuner [22] e WATuning [26], cujos fundamentos ajudaram a consolidar práticas de aprendizado supervisionado, meta-aprendizado e atenção contextual em cenários de ajuste automatizado. A atenção especial conferida a ambientes de missão crítica e alta estabilidade operacional reflete uma lacuna relevante na literatura e reforça a contribuição desta dissertação, que focaliza sistemas corporativos de grande porte em que desempenho, previsibilidade e auditabilidade são requisitos centrais.

A Tabela 4.1 apresenta a evolução das principais abordagens de otimização automática de parâmetros em SGBD, organizadas cronologicamente segundo três dimensões metodológicas: seleção de parâmetros, redução de atributos e técnica de ajuste. Essa estrutura evidencia a transição progressiva de soluções heurísticas e empíricas para modelos probabilísticos e de aprendizado contínuo, culminando em arquiteturas cognitivas baseadas em LLM.

Tabela 4.1: Trabalhos relacionados sobre otimização automática de parâmetros de SGBD.

Trabalho	**Ano**	**Seleção de Parâmetros**	**Redução de Atributos**	**Técnicas de Ajuste**
STMM [36]	2006	N/E	N/E	Heurístico (STMM)
PGTune [34]	2008	N/E	N/E	Heurístico (Regra)
iTuned [18]	2009	LHS	N/E	BO (GPR)
OpenTuner [32]	2014	N/E	N/E	Heurístico (EA)
BestConfig [31]	2017	N/E	N/E	Heurístico (DivideMix)
OtterTune [19]	2017	LASSO	AFE + *k-means*	BO (GPR)
CDBTune [24]	2019	N/E	N/E	RL (DDPG)
Qtune [25]	2019	N/E	N/E	RL (DDPG)
iBTune [29]	2019	N/E	N/E	RL (DDPG)
Tuneful [21]	2020	N/E	N/E	BO (GPR)
ReIM [33]	2020	N/E	N/E	BO (GPR)
ResTune [20]	2021	N/E	N/E	BO (GPR)
CGPTuner [22]	2021	N/E	N/E	BO (CGP)
UDO [27]	2021	N/E	N/E	RL (DDPG)
WATuning [26]	2021	N/E	MIM-I/O	RL (ATT-DDPG)
DNN [37]	2021	N/E	N/E	DL (DNN)
DB-BERT [30]	2022	NLP	N/E	RL (BERT)
OnlineTune [23]	2022	N/E	N/E	BO (GPR)
LlamaTune [35]	2022	HeSBO	SHAP	BO (REMBO)
HUNTER [28]	2022	N/E	N/E	RL (DDPG)
GMM/DNN [77]	2023	GMM	N/E	DL (DNN)
λ-Tune [78]	2024	LLM	ILP	LLM (GPT-4)
Centrum [79]	2025	N/E	N/E	BO (SGBE)
ADWTune [80]	2025	N/E	DBSCAN	RL (ADW-DDPG)

Abreviações: ADW-DDPG — *Adaptive Dynamic Workload Deep Deterministic Policy Gradient*; AFE — Análise Fatorial Exploratória; ATT — *Attention*; BERT — *Bidirectional Encoder Representations from Transformers*; CGP — *Contextual Gaussian Process*; DBSCAN — *Density-Based Spatial Clustering of Applications with Noise*; DDPG — *Deep Deterministic Policy Gradient*; DL — *Deep Learning*; GPR — *Gaussian Process Regression*; HeSBO — *Hashing-Enhanced Subspace Bayesian Optimization*; ILP — *Integer Linear Programming*; MIM — *Multi-Indicator Matching*; REMBO — *Random Embedding Bayesian Optimization*; SGBE — *Sampled Gaussian Bandit Estimator*; **N/E** — Não especificado.

Os primeiros trabalhos, como o *Self-Tuning Memory Manager* (STMM) [36] e o PGTune [34], representam a gênese da automação em bancos de dados, com foco no ajuste reativo de parâmetros de memória. Ambos empregam regras fixas e conhecimento especializado, sem técnicas explícitas de seleção ou redução de atributos. O STMM opera sobre métricas de BPOOL, adaptando o tamanho dos BPOOLs conforme limiares de uso,

enquanto o PGTune utiliza heurísticas derivadas de modelos empíricos. Essas soluções inauguraram a noção de autoajuste, ainda sem o suporte de aprendizado estatístico.

A primeira inflexão científica ocorreu com o avanço das técnicas de *Bayesian Optimization* (BO). O iTuned [18] formalizou o problema de configuração como uma otimização de função de custo modelada via *Gaussian Process Regression* (GPR), utilizando amostragem *Latin Hypercube Sampling* (LHS) para explorar o espaço paramétrico de forma estatisticamente eficiente. O Ottertune [19] consolidou o uso de ranqueamento por *Least Absolute Shrinkage and Selection Operator* (LASSO) e redução de atributos com Análise Fatorial Exploratória (AFE) e agrupamento *k-means*, integrando inferência estatística e BO em uma solução de ajuste adaptativo. O Tuneful [21] complementou essa linha, aplicando estratégias de *sampling* seletivo para reduzir o custo experimental e manter a eficiência amostral. O ReIM [33], embora metodologicamente sólido, baseou-se em um paradigma *Data Intensive Scalable Computing* (DISC), distinto do contexto arquitetural tratado nesta dissertação. Essas propostas consolidaram o modelo BO-GPR como eixo dominante em configurações probabilísticas de SGBD, equilibrando custo, acurácia e interpretabilidade [18, 19, 21, 22].

Com a consolidação das abordagens baseadas em AM, surgiram métodos de *Reinforcement Learning* (RL) que aprendem políticas de ajuste diretamente por interação com o ambiente, sem necessidade de modelagem explícita. Entre eles, destacam-se CDBTune [24], iBTune [29], Qtune [25] e UDO [27], todas fundamentadas em *Deep Deterministic Policy Gradient* (DDPG). O iBTune aplica substituição *Least Recently Used* (LRU) no gerenciamento de memória e adota DDPG para otimizar parâmetros de BPOOLs. O Qtune explora análise de consultas (*SQL-aware*) para correlacionar variáveis de desempenho e comandos SQL. Já o CDBTune e o UDO generalizam a aprendizagem por reforço para múltiplos tipos de *workloads*, tornando o processo de sintonia mais robusto e transferível entre instâncias distintas. Apesar de eficazes em ambientes dinâmicos, essas abordagens possuem custo computacional elevado e maior risco de instabilidade, sendo mais adequadas a sistemas elásticos e distribuídos.

A literatura subsequente consolidou técnicas híbridas que combinam aprendizado contextual, meta-aprendizado e otimização bayesiana. O Restune [20] aplica meta-aprendizado a partir de *meta-features* para acelerar a convergência em novos ambientes. O CGPTuner [22] introduz o conceito de *Contextual Gaussian Process* (CGP), permitindo que o modelo de GPR se adapte dinamicamente às variações de contexto. Já o WATuning [26] utiliza a técnica de *Multi-Indicator Matching* (MIM) com mecanismos de atenção (ATT) sobre o DDPG para priorizar atributos mais relevantes, sobretudo os relacionados a métricas de I/O. Essas soluções demonstram a integração entre aprendizado contextual e otimização probabilística, consolidando a geração de sistemas de ajuste cognitivo.

Modelos mais recentes passam a incorporar DL e análise probabilística hierárquica. O DNN [37] utiliza redes neurais profundas para prever o desempenho sob diferentes configurações, substituindo modelos bayesianos por regressões não lineares de alta capacidade. O GMM/DNN [77] combina modelos de mistura gaussiana (*Gaussian Mixture Model* (GMM)), para agrupar *workloads* e redes neurais (DNN) para inferir a configuração ótima. Apesar de mais complexos, esses modelos nem sempre superam as abordagens BO-GPR em custo-benefício. O uso de redes neurais profundas e aprendizado por reforço é computacionalmente oneroso e, portanto, mais adequado a sistemas distribuídos em nuvem do que a SGBDRs convencionais. Assim, o modelo BO-GPR permanece como referência devido à sua acurácia, estabilidade e custo computacional reduzido [37, 77].

O LlamaTune [35] representa um avanço importante ao aplicar *Hashing-Enhanced Subspace Bayesian Optimization* (HeSBO), uma extensão do *Random Embedding Bayesian Optimization* (REMBO), para reduzir o espaço de busca de parâmetros em BO. A abordagem utiliza o *benchmark* YCSB-A associado à explicabilidade via SHAP, permitindo a interpretação da importância de cada parâmetro ajustado. Essa integração entre eficiência amostral e interpretabilidade exemplifica o estado da arte em BO aplicada a sistemas de banco de dados. O λ-Tune [78], embora inovador, ainda se encontra em estágio inicial. O problema de otimização de parâmetros é predominantemente descritivo, demandando análise quantitativa e inferência sobre espaços de busca bem definidos, enquanto os LLMs são mais adequados a tarefas criativas e de raciocínio semântico. Até o momento, nenhuma das soluções baseadas em LLMs demonstrou resultados consistentes em ambientes reais, tampouco apresentou replicabilidade experimental.

O Centrum [79] e o ADWTune [80] ampliam essa perspectiva ao propor ajustes dinâmicos e de alta dimensionalidade. O Centrum adota o estimador *Sampled Gaussian Bandit Estimator* (SGBE) para amostragem eficiente, enquanto o ADWTune combina agrupamento *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) com aprendizado por reforço adaptativo (ADW-DDPG), adaptando-se a cargas de trabalho variáveis em tempo real. Tais técnicas reforçam a tendência de integração entre aprendizado por reforço e métodos de agrupamento para ambientes elásticos.

Apesar da diversidade de técnicas, a maioria dos trabalhos ainda se apoia em dados sintéticos ou simulados, sendo raras as implementações validadas em produção. Mesmo quando aplicadas a SGBDRs, as avaliações concentram-se quase exclusivamente em ambientes transacionais (OLTP), de natureza mais determinística. Nenhuma das soluções analisadas aborda o cenário analítico (OLAP), inerentemente mais complexo e menos previsível, em que as cargas de trabalho apresentam variação de cardinalidade e comportamento não linear, dificultando previsões consistentes. Esse vácuo metodológico limita a generalização dos resultados e reforça a necessidade de investigações direcionadas a

contextos reais e heterogêneos.

De forma geral, a análise revela uma trajetória metodológica clara. A seleção de parâmetros evoluiu de processos heurísticos para técnicas de inferência, como LASSO [19], GMM [77] e HeSBO [35]. A redução de atributos passou de estágios manuais para métodos sistematizados, como AFE, *k-means*, DBSCAN e *meta-features*. O eixo de ajuste avançou de heurísticas determinísticas para arquiteturas baseadas em BO [18, 19, 21, 22], RL [24, 25, 26, 27, 29, 80], DL [37, 77] e LLM [78], refletindo uma crescente autonomia e inteligência dos SGBDs modernos.

Essa consolidação metodológica estabelece as bases para uma nova geração de sistemas autoajustáveis, em que seleção de parâmetros, análise de atributos e ajuste do modelo são executados de forma integrada e iterativa. Em ambientes de larga escala, a integração desses componentes reduz a dependência de intervenção humana e aprimora a eficiência adaptativa dos SGBDs. A Tabela 4.1 sintetiza, portanto, não apenas o avanço cronológico das técnicas, mas a consolidação de um paradigma orientado a aprendizado e autonomia em sistemas de gerenciamento de banco de dados.

4.2 Lacunas e Oportunidades

Desde o STMM [36], nenhuma das iniciativas modernas de ajuste automático investigadas na literatura foca de forma sistemática plataformas centralizadas, o que configura uma lacuna de quase duas décadas no domínio de SGBD executados em *mainframe*. O ecossistema IBM apresenta uma arquitetura singular, baseada em hierarquia de memória multinível (*expanded*, *central* e *auxiliary storage*), múltiplos BPOOLs especializados (BP0, BP1, BP2 etc.), integração nativa com o Workload Manager (WLM) e mecanismos avançados de controle de concorrência. Esses elementos moldam um comportamento operacional distinto do observado em ambientes CDB e em infraestruturas *commodity*, tornando inadequada a transferência direta de soluções concebidas para tais contextos. A ausência de ferramentas que capturem adequadamente métricas específicas de *mainframe*, somada à necessidade de integração com subsistemas proprietários, reforça a demanda por técnicas próprias de otimização. Nesse cenário, parâmetros como *VPSEQT*, *DWQT* e *VPMAX*[1] passam a funcionar como alavancas críticas de desempenho.

A adaptação de técnicas de aprendizado e otimização ao ambiente corporativo de missão contínua envolve desafios técnicos, operacionais e organizacionais. No plano técnico, limitações de instrumentação, particularidades da hierarquia de I/O e a necessidade de estabilidade numérica sob restrições rígidas dificultam a aplicação de modelos complexos.

[1]Em SGBDs com BPOOL hierárquico, *VPSEQT* influencia a proporção de páginas pré-buscadas sequencialmente, *DWQT* define o limiar de escrita atrasada e *VPMAX* limita o máximo de páginas de pré-busca em lote.

No plano operacional, a exigência de disponibilidade permanente (*24/7*) e de processos auditáveis impõe a necessidade de reversão imediata (*rollback*) diante de qualquer regressão. No plano organizacional, a resistência à automação em sistemas críticos, a necessidade de justificar o *Return on Investment* (ROI) e a aderência a estruturas de governança corporativa atuam como barreiras reais. Em conjunto, esses fatores explicam a escassez de pesquisas recentes voltadas a plataformas centralizadas, apesar da relevância estratégica dessas arquiteturas.

A revisão sistemática conduzida neste trabalho evidencia cinco lacunas principais. A primeira é a ausência de estudos dedicados a ambientes Mainframe e ao ecossistema corporativo que depende desse tipo de infraestrutura. A segunda refere-se à validação limitada em produção, com predominância de experimentos em cargas sintéticas, frequentemente sem requisitos explícitos de *safety*, auditabilidade e reversibilidade. A terceira diz respeito ao custo computacional do processo de ajuste, que impõe restrições de escalabilidade e inviabiliza ciclos experimentais densos em ambientes críticos. A quarta lacuna envolve a baixa exploração de técnicas de transferência de aprendizado para generalizar configurações entre arquiteturas e workloads distintos. Por fim, a quinta lacuna está na desconexão entre as propostas acadêmicas e o que é operacionalmente viável para administradores de bancos de dados, limitando a adoção prática.

Tabela 4.2: Análise sistemática das principais lacunas na literatura sobre otimização automática de parâmetros de SGBD.

Categoria de lacuna	**Trabalhos (n/24)**	**Impacto potencial**	**Viabilidade de endereçamento**
Ambientes *Mainframe* e plataformas centralizadas	1/24	Muito alto (segmento crítico sem cobertura).	Alta (exige métricas internas e especialistas).
Validação em produção e requisitos de sistemas críticos (*safety*, auditabilidade, rollback)	3/24	Muito alto (risco direto em operação).	Média (depende de janelas de mudança e governança).
Eficiência computacional e overhead do processo de ajuste	7/24	Alto (limita escalabilidade e uso contínuo).	Alta (modelos leves, amostragem e otimização numérica).
Transferência de aprendizado e generalização entre arquiteturas e cargas	2/24	Alto (dificulta reaproveitamento de conhecimento).	Média–alta (com modelos transferíveis e adaptação).
Adoção prática e alinhamento com a operação de DBAs e processos corporativos	1/24	Muito alto (afeta implantação em produção).	Média (requer integração com processos e métricas de ROI).

Essas cinco lacunas foram consolidadas em categorias analíticas, sumarizadas na Tabela 4.2. Cada categoria apresenta a frequência com que a limitação foi identificada na amostra de 24 trabalhos analisados, bem como seu impacto potencial e a viabilidade de endereçamento em ambientes corporativos.

Em conjunto, essas lacunas demonstram que a aplicação direta de métodos existentes é insuficiente para atender às especificidades do ambiente Mainframe. Elas justificam a necessidade de um arcabouço próprio, capaz de integrar redução de dimensionalidade, seleção robusta de variáveis e otimização amparada por incerteza, conforme apresentado no próximo capítulo.

4.3 Considerações Finais

A trajetória evolutiva das técnicas de otimização automática de parâmetros em SGBD evidencia um amadurecimento metodológico que transita da automação heurística para abordagens probabilísticas baseadas em BO-GPR e, posteriormente, para modelos de aprendizado por reforço e aprendizado profundo. Embora as abordagens RL/DL demonstrem eficiência em ambientes elásticos e distribuídos, o modelo BO-GPR mantém-se como referência em cenários que demandam estabilidade, interpretabilidade e controle de custos computacionais, características essenciais em SGBDR críticos e de missão contínua. As lacunas identificadas, ausência de estudos em plataformas centralizadas, validação em produção, negligência de requisitos críticos, limitações de transferência e integração operacional insuficiente configuram um espaço de contribuição científica e prática.

Esses achados orientam diretamente o desenho metodológico a ser apresentado no Capítulo 5, que privilegia a seleção parcimoniosa de parâmetros, a redução objetiva de atributos, o uso de modelos probabilísticos de baixo custo amostral e a validação em cenários operacionais realistas. Dessa forma, o presente trabalho não apenas preenche uma lacuna metodológica relevante, mas também reposiciona a discussão sobre autonomia em SGBD no contexto dos sistemas corporativos de maior criticidade mundial.

Capítulo 5

Solução de Otimização de Parâmetros BP

A otimização dos parâmetros de BPOOL em SGBDs é determinante para o desempenho de sistemas corporativos de larga escala. Em ambientes de *mainframe*, como o DB2, que processam volumes massivos de transações sob requisitos estritos de disponibilidade, configurações inadequadas dos BPOOLs resultam em gargalos de I/O, variabilidade elevada de latência e degradação direta dos níveis de serviço. A Figura 5.1 sintetiza o funcionamento geral da solução proposta, evidenciando suas etapas principais: coleta e preparação das métricas operacionais, agrupamento e redução de dimensionalidade, seleção estatística dos parâmetros relevantes, modelagem preditiva baseada em incerteza e geração das recomendações finais de reconfiguração. Esse fluxo reforça a complexidade do ajuste manual em ambientes Db2 for z/OS, nos quais o comportamento é altamente dependente do padrão de acesso, da competitividade entre aplicações e das restrições operacionais de janelas de manutenção.

Neste contexto, este capítulo apresenta a solução completa de otimização automática para parâmetros de BPOOL, fundamentada nos capítulos anteriores e estruturada segundo a metodologia CRISP-DM. A abordagem integra AFE, *k-means* (KMEANS), regressão LASSO e modelagem BO-GPR, compondo um processo modular capaz de reduzir dimensionalidade, identificar parâmetros de maior impacto e explorar o espaço configuracional por meio de funções de aquisição baseadas em incerteza. O fluxo adotado permite recomendar configurações com impacto mensurável sobre a latência síncrona, mantendo aderência às restrições típicas do ambiente DB2, como necessidade de reversão segura, risco de regressão e priorização de percentis superiores de latência. As seções seguintes apresentam, de forma estruturada: o entendimento do negócio (Seção 5.1); a caracterização e preparação dos dados (Seção 5.2 e Seção 5.3); a seleção estatística dos parâmetros (Subseção 5.3.3); a modelagem preditiva e a otimização bayesiana (Seção 5.4); a avaliação detalhada dos

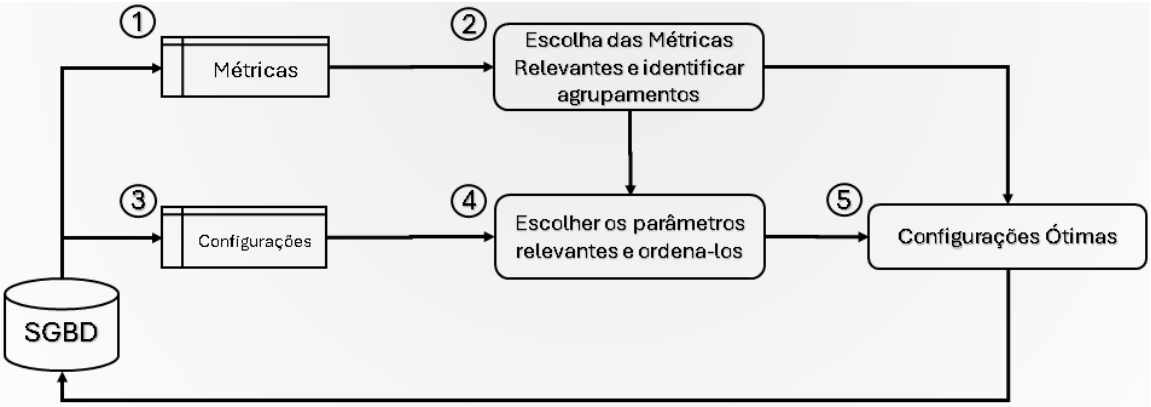

Figura 5.1: Fluxo geral da solução proposta para otimização de parâmetros de BPOOL. As etapas incluem: (1) coleta das métricas, (2) agrupamento e seleção das métricas relevantes, (3) obtenção das configurações existentes, (4) ranqueamento dos parâmetros e (5) geração das configurações recomendadas.

resultados (Seção 5.5 e Seção 5.7); e, por fim, as recomendações consolidadas por BPOOL e as considerações finais deste capítulo (Seção 5.6 e Seção 5.8).

5.1 Entendimento do Negócio

O entendimento do negócio focou na identificação clara dos objetivos organizacionais relacionados à otimização de BPOOLs e na tradução destes objetivos em metas técnicas específicas e mensuráveis. O contexto organizacional envolve uma das maiores instituições financeiras da América Latina, que processa mais de 400 bilhões de transações mensais em ambiente de *mainframe*, no qual a performance do sistema DB2 impacta diretamente a experiência de milhões de clientes e os custos operacionais da organização.

O objetivo principal estabelecido foi a redução do tempo máximo de espera para operações de I/O síncrona (*Maximum Synchronous I/O Delay*), métrica que impacta diretamente a percepção de performance pelos usuários finais e que serve como indicador confiável da eficiência do sistema de BPOOL. Esta métrica foi selecionada após análise de múltiplas alternativas porque representa adequadamente o impacto da configuração de BPOOLs na experiência do usuário e porque pode ser medida de forma consistente e confiável.

Os objetivos secundários incluíram a redução do consumo de recursos de CPU relacionados ao gerenciamento de BPOOLs, a melhoria da taxa de acerto (do inglês, *hit-ratio*) dos BPOOLs mais críticos, e a identificação de padrões operacionais que pudessem orientar estratégias de otimização mais sofisticadas. Estes objetivos secundários foram importantes para garantir que as melhorias na métrica principal não fossem obtidas às custas de degradação em outros aspectos da performance do sistema.

A solução proposta articula as etapas de caracterização da carga, identificação de parâmetros relevantes e otimização propriamente dita, formando um fluxo contínuo orientado à redução de variabilidade e latência em BPOOLs. Como ilustrado na Figura 5.2, o processo inicia-se com a coleta das métricas do SGBD, que alimentam a Análise Fatorial Exploratória (AFE) para redução da dimensionalidade e síntese dos fatores dominantes da carga. Em seguida, o agrupamento k-means organiza métricas semelhantes, permitindo selecionar representantes de cada grupo. Na etapa seguinte, as amostras e configurações são processadas pelo LASSO, que ranqueia os parâmetros configuráveis segundo sua influência estatística. Finalmente, a otimização combina BO-GPR e funções de aquisição para explorar o espaço configuracional e propor ajustes com impacto mensurável sobre a latência, fechando o ciclo com recomendações reavaliadas diretamente no SGBD. Esse fluxo consolida a estratégia geral da solução e prepara terreno para o detalhamento metodológico apresentado nas seções seguintes.

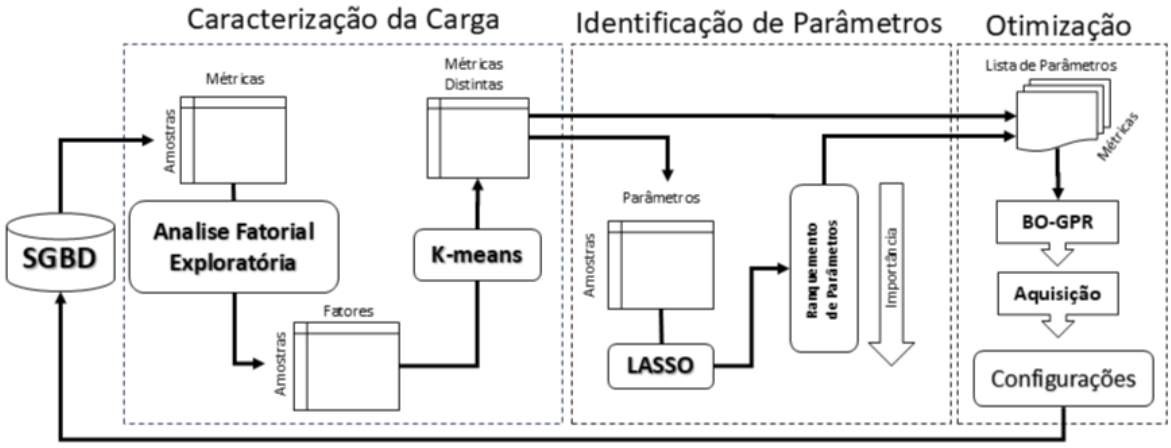

Figura 5.2: Arquitetura geral da solução proposta. O fluxo integra: (1) análise fatorial exploratória e agrupamento k-means para caracterização da carga; (2) regressão LASSO para ranqueamento dos parâmetros; e (3) BO-GPR para otimização e geração das configurações recomendadas.

Os critérios de sucesso estabelecidos incluíram redução mínima de 10% na métrica principal, manutenção ou melhoria de métricas secundárias, e validação da estabilidade das configurações recomendadas em ambiente de produção por período mínimo de 30 dias. Estes critérios foram definidos com base na experiência prévia da organização com projetos de otimização e nas expectativas realistas considerando as limitações operacionais do ambiente.

O objetivo de negócio é reduzir variabilidade e picos de latência associados a BPOOLs sem violar restrições operacionais. Define-se sucesso como redução sustentada no 99º percentil da métrica de espera máxima de I/O síncrono, abreviada para `MAX_SYNC_IO_DLY_MS`, ausência de regressões em métricas de sanidade correlatas e manutenção de confiabilidade

operacional. O racional de valor considera mitigação de risco de violações de *SLOs*, suavização de janelas de *batch* e melhor previsibilidade para equipes de operação.

5.2 Entendimento dos Dados

Esta etapa do CRISP-DM estabelece as bases analíticas para todas as fases subsequentes. O seu objetivo é compreender a origem, a estrutura e a qualidade das informações disponíveis, eliminando incongruências, padronizando formatos e avaliando a adequação estatística do conjunto para redução de dimensionalidade e modelagem preditiva. O processo integra parâmetros de configuração de BPOOLs extraídos do catálogo do DB2 e séries temporais de métricas operacionais, coletadas com granularidade de 15 minutos. A variável-alvo `MAX_SYNC_IO_DLY_MS` foi padronizada por *StandardScaler*, assegurando comparabilidade e estabilizando o comportamento das distribuições.

5.2.1 Origem, Padronização e Higienização Estrutural

Os parâmetros analisados refletem controles diretos sobre o comportamento dos BPOOLs, incluindo VPSIZE, VPSEQT, DWQT, VDWQT, VPMIN, VPMAX e `VPPSEQT`. Já as séries temporais contemplam latências de I/O, taxas de operações síncronas e assíncronas, indicadores de *hit/miss* de *buffer* e *disk cache*, além de *throughput* agregado.

A ingestão foi conduzida com validações explícitas de separadores, rótulos e tipos, registrando mensagens operacionais para auditoria. As colunas temporais foram consolidadas em `DATA` e o identificador de BPOOL padronizado em `BPOOL`. Os parâmetros numéricos foram convertidos de forma segura (`errors='coerce'`), com ajuste para `Int64` quando aplicável. A limpeza inicial removeu colunas inválidas ou sem variabilidade, três métricas exibiam variância nula (`Pending Time: Device Busy Delay (ms)`, `Decrypted Read Throughput (kB/s)` e `Encrypted Write Throughput (kB/s)`), e seis métricas apresentavam correlação superior a 0,995, justificando exclusão preventiva para evitar multicolinearidade (`Access Method Read/Write (kB/s)`, `Unchanged Pages in Buffer Pool (kB)`, `Random Read (I/Os/s)`, `Getpages - Hits in Buffer Pool (Getpages/s)`, `Read Ops (LR/s)`). Em etapa posterior, `VPPSEQT` também foi retirada por variância nula.

Após a higienização, o conjunto passou de 45 para 35 métricas e de 7 para 6 parâmetros, reduzindo cerca de 22% das variáveis monitoradas. O determinante da matriz de correlação ($1{,}132118 \times 10^{-20}$) e *rank* igual a 32 confirmaram que a remoção das colunas redundantes preservou a estrutura linear necessária para análise fatorial. A robustez amostral foi garantida pela eliminação de registros com campos nulos, resultando na remoção de 46.188 linhas (37,63%) e consolidando uma amostra final de 122.618 observações completas. A Tabela 5.1 apresenta a lista de métricas removidas e seus critérios.

Tabela 5.1: Métricas removidas durante a higienização dos dados e critérios de exclusão.

Métrica removida	Motivo
`Pending Time: Device Busy Delay (ms)`	Variância nula
`Decrypted Read Throughput (kB/s)`	Variância nula
`Encrypted Write Throughput (kB/s)`	Variância nula
`Access Method Read/Write (kB/s)`	Correlação $> 0{,}995$
`Unchanged Pages in Buffer Pool (kB)`	Correlação $> 0{,}995$
`Random Read (I/Os/s)`	Correlação $> 0{,}995$
`Getpages - Hits in Buffer Pool (Getpages/s)`	Correlação $> 0{,}995$
`Read Ops (LR/s)`	Correlação $> 0{,}995$
`VPPSEQT`	Variância nula (refino)

5.2.2 Estrutura Temporal, Padrões Operacionais e Composição Final

As séries temporais apresentaram padrões intra-diários e semanais relevantes para interpretação dos BPOOLs. Para mitigar distorções causadas por picos concentrados e cargas sazonais, foram adotadas janelas homogêneas e percentis elevados, como ilustrado na Figura 5.3, que apresenta o 99^{th} percentil semanal da variável-alvo e evidencia heterogeneidade entre BPOOLs.

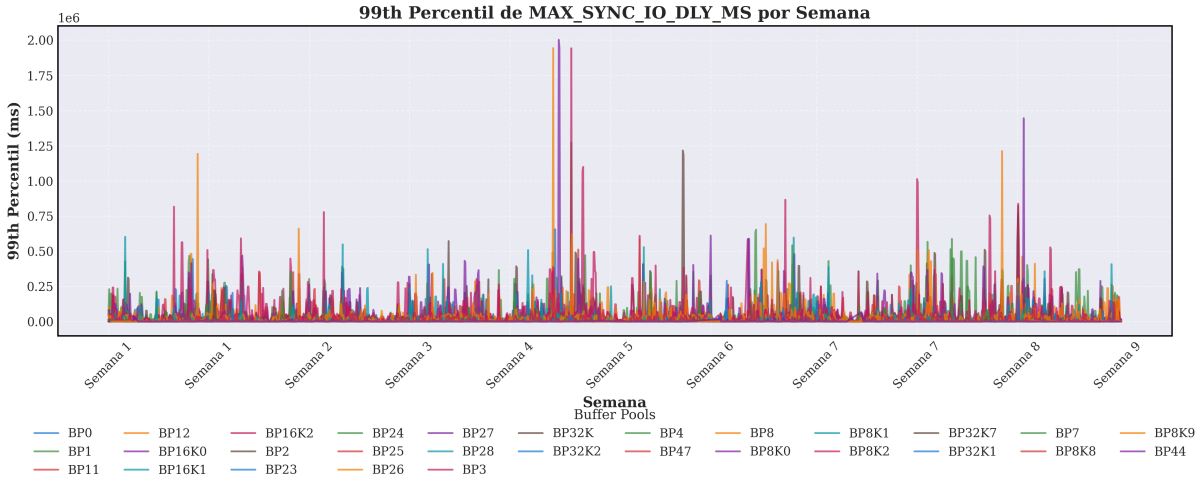

Figura 5.3: 99th percentil de `MAX_SYNC_IO_DLY_MS` por semana para todos os BPOOLs analisados.

A comparação entre períodos diurnos e noturnos (Figura 5.4) revelou comportamentos distintos: `BP44`, `BP24`, `BP25` e `BP27` apresentaram elevações noturnas coerentes com o

predomínio de rotinas *batch*, enquanto `BP3`, `BP2`, `BP1`, `BP47`, `BP12` e `BP7` exibiram maior pressão diurna, típica de cargas interativas. A Tabela 5.2 consolida as diferenças percentuais absolutas.

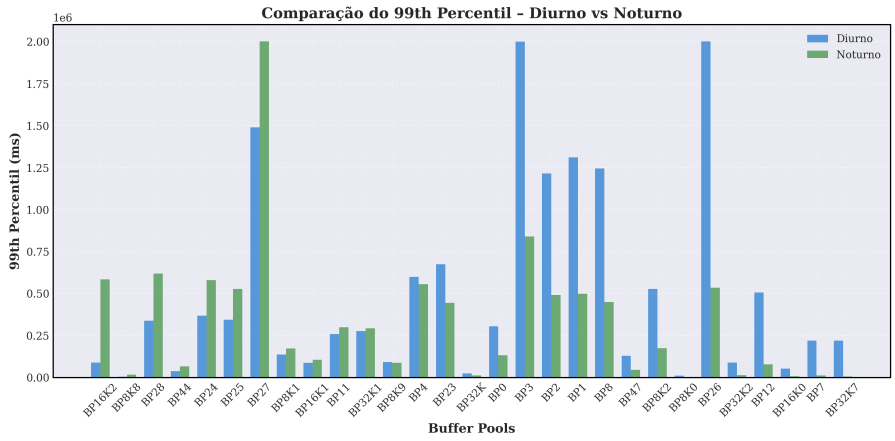

Figura 5.4: 99$^{\text{th}}$ percentil de latência entre períodos diurno e noturno por BPOOL.

Tabela 5.2: Diferença percentual absoluta entre períodos diurno e noturno por BPOOL (percentil alto de latência).

BPOOL	Dia (ms)	Noite (ms)	$\lvert\Delta\rvert$ (%)
BP7	2.194,96	128,91	94,13
BP12	5.058,95	786,54	84,45
BP47	1.299,84	447,98	65,54
BP1	13.104,16	4.986,99	61,94
BP2	12.154,90	4.910,00	59,61
BP3	20.006,79	8.398,75	58,02
BP24	3.687,97	5.806,02	57,43
BP25	3.449,21	5.275,10	52,94
BP44	378,43	671,06	77,33
BP27	14.889,02	20.019,66	34,46
BP23	6.750,40	4.445,88	34,14
BP8K1	1.376,44	1.730,32	25,71
BP16K1	876,37	1.053,85	20,25
BP11	2.579,28	2.994,49	16,10
BP4	5.987,75	5.559,77	7,15
BP32K1	2.767,37	2.928,91	5,84

Ao final da reconciliação temporal e da higienização, a amostra consolidada contém 122.618 registros completos, com 32 métricas 6 parâmetros distribuídos entre 16 BPOOLs. Esta configuração final fornece base estável para a análise fatorial exploratória, para o agrupamento por *k-means* e para as etapas posteriores de modelagem com *Gaussian Process Regression* e otimização baseada em aquisição.

5.3 Preparação dos Dados

A preparação dos dados foi conduzida segundo uma solução *data-centric*, com validações explícitas de tipos, rótulos e consistência temporal. O processo incluiu etapas automatizadas de leitura e padronização, abrangendo a detecção de separador, a renomeação da coluna alvo, a normalização das datas (`DATA`) e a padronização dos identificadores (`BPOOL`). Assim sendo, foram aplicadas coerções seguras de tipo (`Int64`/`float64`), seguidas pela remoção de colunas com variância nula, poda de variáveis altamente correlacionadas e tratamento de valores ausentes. Ao final dessa sequência, obtiveram-se as visões `amostra_alvo`, `amostra_parametros` e `amostra_metricas`, todas com `shape` consistente (122.618 linhas) e chaves temporais alinhadas.

Com o objetivo de reduzir vieses oriundos de janelas assimétricas e preservar a comparabilidade entre BPOOLs, foram aplicadas agregações em janelas homogêneas de 15 minutos, utilizando percentis elevados (por exemplo, 99^{th}). Esse desenho amostral minimiza a influência de picos isolados sobre a distribuição global e aumenta a robustez estatística das etapas subsequentes de AFE e *clustering*. A estrutura resultante fornece uma base estável e representativa para as fases seguintes de seleção de Métricas e Parâmetros e modelagem probabilística.

5.3.1 Seleção de Métricas

A etapa de seleção de métricas foi conduzida com o objetivo de identificar, dentre os diversos indicadores coletados, aqueles capazes de representar de forma estável e não redundante o comportamento dos BPOOLs. O processo iniciou-se com a aplicação da AFE, utilizada para reduzir a dimensionalidade e revelar fatores latentes que descrevem a variabilidade das métricas. As variáveis foram previamente padronizadas e avaliadas quanto à adequação ao modelo fatorial. O índice KMO global atingiu o valor de 0,97, classificando a amostra como "excelente", e o teste de esfericidade de Bartlett apresentou significância ($p < 0{,}001$), confirmando a correlação suficiente entre as métricas para a extração de fatores.

A extração seguiu o critério de autovalores maiores que 1 (Kaiser), resultando em nove fatores principais. A Figura 5.5 ilustra o gráfico de autovalores (*scree plot*), no qual

se observa a inflexão acentuada após o nono componente, indicando o ponto ótimo de retenção.

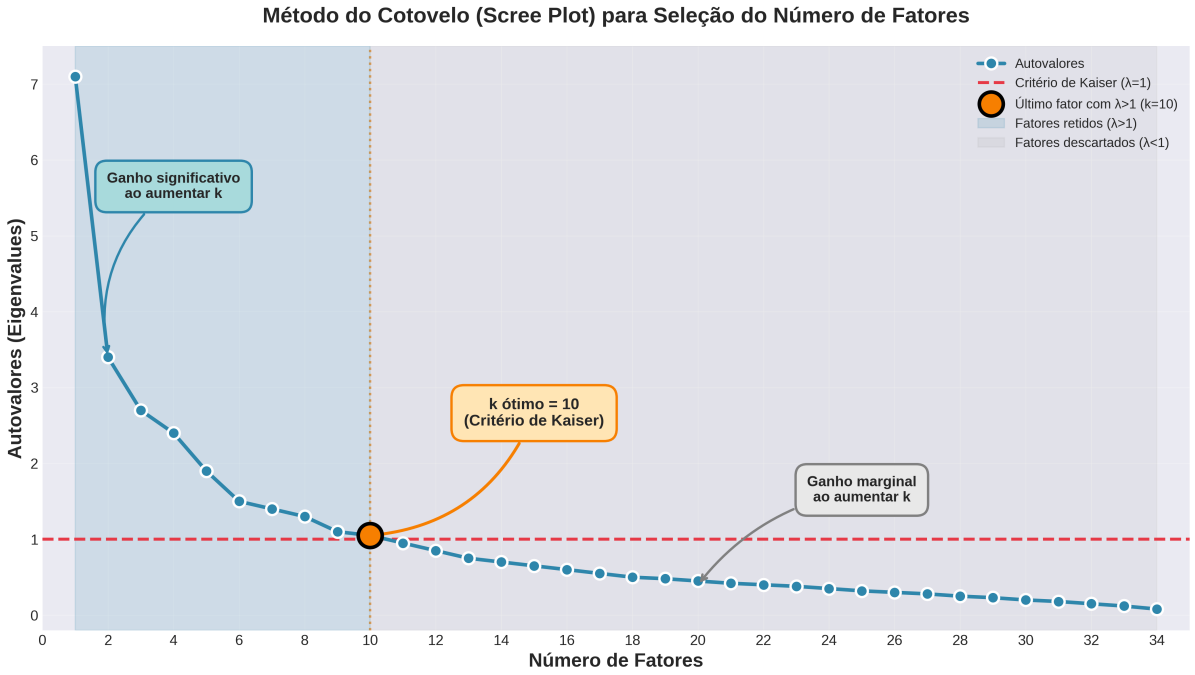

Figura 5.5: Autovalores (*scree plot*) da AFE para o conjunto de métricas de desempenho.

A Tabela 5.3 apresenta as métricas originais com seus respectivos valores de KMO. Observa-se que todas as métricas principais apresentaram KMO superior a 0,60, validando sua adequação ao modelo fatorial. Destacam-se *Getpages (Synchronous and Asynchronous)*, *Changed Pages to be Written* e *Pages Used in Buffer Pool*, todas acima de 0,90, o que reforça a sua importância estatística no conjunto analisado. Assim, com base nas cargas fatoriais e na matriz de correlação, as métricas com comunalidades inferiores a 0,50 foram descartadas, preservando as de maior poder explicativo. As nove dimensões retidas representaram, de forma abrangente, aspectos de latência, taxa de *I/O*, eficiência de cache e *throughput*.

Tabela 5.3: Métricas e respectivos valores de KMO obtidos na AFE.

Métrica	**KMO**
Getpages (Synchronous and Asynchronous) (Getpages/s)	0,97
Changed Pages to be Written (kB)	0,92
Pages Used in Buffer Pool (kB)	0,91
I/O Pages for Asynchronous I/O (pages/s)	0,90
Write Disconnect Time (ms)	0,90
Synchronous I/O Delay (ms)	0,89
MAX_SYNC_IO_DLY_MS	0,88
Synchronous I/Os (I/Os/s)	0,85
Asynchronous I/O Operations (I/Os/s)	0,84
Getpages – Miss in Disk Cache (%)	0,83
Pending Time: Command Response Delay (ms)	0,81
Maximum Asynchronous I/O Delay (ms)	0,80
IOSQ Time and zOS Dispatch Delays (ms)	0,74
Read and Write Throughput (kB/s)	0,71
Read Hit Percentage (%)	0,69
Buffer Pool Size (kB)	0,64
Getpages – Hit in Buffer Pool (%)	0,62
Getpages – Hit in Disk Cache (%)	0,60

Na sequência, aplicou-se o algoritmo *k-means* aos escores fatoriais normalizados para identificar agrupamentos de métricas com comportamento estatístico semelhante. A escolha do número ótimo de *clusters* baseou-se na combinação entre a análise do cotovelo, o coeficiente *Silhouette* e o critério de Pham. A Figura 5.6 ilustra simultaneamente a redução do erro de soma dos quadrados (SSE) e a evolução do *Silhouette* ao longo dos valores de k, destacando o ponto de equilíbrio entre compactação dos grupos e separabilidade. A Tabela 5.4 resume os resultados numéricos, indicando convergência ideal entre $k = 8$ e $k = 10$: o *Silhouette* atinge valor máximo de 0,4694 em $k = 9$, enquanto o critério de Pham identifica $k = 8$ como ponto de inflexão.

Tabela 5.4: Resultados da análise *k-means* para diferentes valores de k.

k	**SSE**	**PS**	**Silhouette**
2	11,2694	—	0,2485
3	9,2486	0,8981	0,2805
4	7,4794	0,8621	0,3227
5	6,2336	0,8758	0,3247
6	4,7738	0,7975	0,3972
7	3,8643	0,8377	0,4364
8	3,1088	0,8287	0,4297
9	2,4211	0,7994	0,4694
10	1,7994	0,7608	0,4632

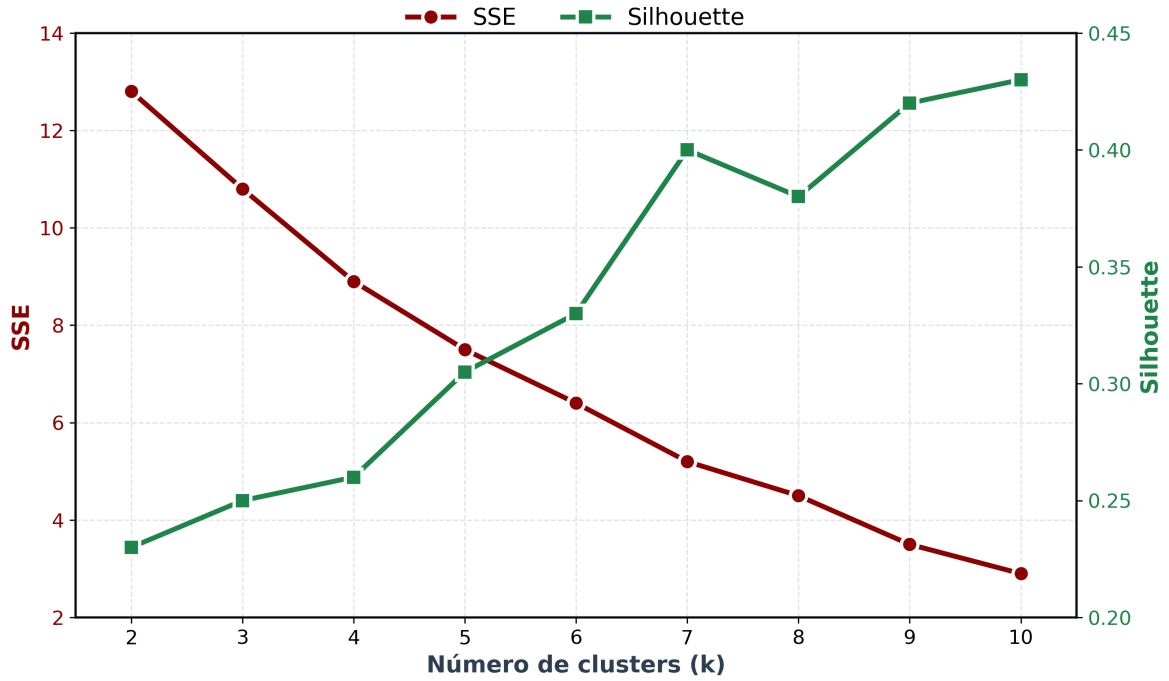

Figura 5.6: Avaliação simultânea de SSE (linha vermelha) e coeficiente *Silhouette* (linha verde) para diferentes valores de k no *k-means*. O gráfico evidencia a região de melhor equilíbrio entre compactação dos grupos e separabilidade, concentrada entre $k = 8$ e $k = 10$.

O modelo final estabeleceu dez agrupamentos coerentes, cada qual representando um padrão específico de comportamento entre as métricas. O Tabela 5.5 apresenta as métricas representativas de cada *cluster*, definidas pelo maior peso fatorial e pela proximidade média interna.

Tabela 5.5: Métricas representativas por *cluster* obtidas via *k-means*.

Cluster	**Métrica Representativa Escolhida**
1	Synchronous I/O Delay (ms)
2	I/O Rate (I/Os/s)
3	Pending Time: Command Response Delay (ms)
4	MAX_SYNC_IO_DLY_MS
5	Response Time (ms)
6	I/O Pages for Asynchronous I/O (pages/s)
7	Asynchronous I/O Delay (ms)
8	Getpages – Hit in Disk Cache (%)
9	Changed Pages to be Written (kB)
10	Getpages – Hit in Buffer Pool (%)

A Tabela 5.6 apresenta a rastreabilidade completa das métricas em relação aos métodos de seleção aplicados. Essa visão consolidada evidencia as métricas mantidas e aquelas identificadas como redundantes.

Tabela 5.6: Métricas escolhidas em função da AFE e do *k-means*.

Métrica	**Selecionada Por**
Asynchronous I/O Delay (ms)	AFE e *k-means*
Changed Pages to be Written (kB)	AFE e *k-means*
Getpages – Hit in Buffer Pool (%)	AFE e *k-means*
Getpages – Hit in Disk Cache (%)	AFE e *k-means*
I/O Pages for Asynchronous I/O (pages/s)	AFE e *k-means*
I/O Rate (I/Os/s)	AFE e *k-means*
MAX_SYNC_IO_DLY_MS	AFE e *k-means*
Pending Time: Command Response Delay (ms)	AFE e *k-means*
Response Time (ms)	AFE e *k-means*
Synchronous I/O Delay (ms)	AFE e *k-means*

O conjunto resultante, composto por dez métricas representativas distribuídas em nove fatores e dez *clusters*, constitui a base empírica utilizada nas fases subsequentes de seleção de parâmetros e modelagem. Essa estrutura reduz a dimensionalidade original e reforça a interpretabilidade estatística, permitindo que as próximas etapas de otimização se apoiem em métricas mais consistentes e correlacionadas com o desempenho global dos BPOOLs.

5.3.2 Tratamento de *Outliers*

A métrica-alvo `MAX_SYNC_IO_DLY_MS` apresenta caudas pesadas e assimetria acentuada em diversos BPOOLs, refletindo picos de latência associados a condições excepcionais de carga, contenção de recursos ou anomalias operacionais. Embora essas observações extremas sejam relevantes para a compreensão do risco, sua presença em grande volume compromete a estabilidade de modelos estatísticos e de aprendizado de máquina, inflando medidas de dispersão e distorcendo ajustes paramétricos. Por esse motivo, adotou-se uma etapa específica de tratamento de *outliers*, com foco em eliminar valores extremos espúrios preservando, tanto quanto possível, a estrutura informativa das caudas da distribuição.

A estratégia compara quatro esquemas clássicos de detecção e remoção: *Z-Score* com limiar $\pm 3\sigma$, intervalo interquartílico (IQR) com multiplicador 1,5, *Percentil* com recorte [1%, 99%] e a composição *Percentil+IQR*. Seja y a série da métrica-alvo após remoção de ausentes. No método de percentis, os limites inferior e superior são dados pela Equação 5.1, na qual valores fora do intervalo são descartados. No método IQR, com $q_1 = \mathrm{Pct}(y, 25\%)$ e $q_3 = \mathrm{Pct}(y, 75\%)$, define-se $IQR = q_3 - q_1$ e aplica-se o intervalo $[\, q_1 - 1{,}5\, IQR,\ q_3 + 1{,}5\, IQR\,]$. Já no critério *Z-Score*, preservam-se as observações que satisfazem $|z_i| \leq 3$, com $z_i = (y_i - \bar{y})/s$. O quarto esquema, *Percentil+IQR*, aplica em sequência o filtro de percentis seguido do filtro baseado no IQR, combinando robustez a caudas e controle adicional de dispersão.

$$\ell_p = \mathrm{Pct}(y, 1\%), \qquad u_p = \mathrm{Pct}(y, 99\%). \tag{5.1}$$

A seleção do método mais adequado foi realizada de forma dinâmica e segmentada por BPOOL. Para cada BPOOL, os quatro critérios foram aplicados à série `MAX_SYNC_IO_DLY_MS` e comparados quanto ao número de observações removidas, respeitando um limite mínimo de amostras para cálculo de percentis e estatísticas robustas. O método escolhido em cada pool foi aquele que apresentou maior capacidade de remoção de valores extremos sem alterar o conjunto de colunas nem comprometer o tamanho mínimo da amostra, garantindo assim um equilíbrio entre redução de ruído e preservação de variabilidade informativa.

A Tabela 5.7 sintetiza o impacto global da limpeza de *outliers* sobre o conjunto de dados. No total, foram analisados 29 BPOOLs, com 122.618 registros antes da filtragem e 109.749 registros após a aplicação dos filtros, o que corresponde à remoção de 12.869 observações (10,495% da amostra original). Em todos os BPOOLs o método selecionado foi *Percentil+IQR*, evidenciando sua superioridade prática no contexto avaliado. A remoção média por BPOOL foi de aproximadamente 10,45%, com variação entre 6,65% e 17,19%, o que indica heterogeneidade no comportamento das caudas de latência entre pools distintos.

Tabela 5.7: Síntese global da limpeza de *outliers* no experimento: abrangência, método e impacto quantitativo.

Indicador	Valor	Unidade	Descrição
BPOOLs analisados	29	—	Conjunto completo de pools incluídos na etapa de limpeza de *outliers*.
Registros antes da limpeza	122.618	linhas	Amostra consolidada antes da filtragem de *outliers*.
Registros após a limpeza	109.749	linhas	Base final utilizada nas etapas subsequentes de modelagem.
Registros removidos	12.869	linhas	Observações classificadas como *outliers* pelos critérios adotados.
Percentual global removido	10,495%	%	Proporção total de linhas descartadas em relação à amostra original.
Método selecionado	Percentil+IQR	—	Método escolhido automaticamente em 29 de 29 BPOOLs.
Remoção média	10,45%	%	Percentual médio de remoção quando analisado por BPOOL.
Remoção mínima	6,65%	%	Menor percentual de remoção observado entre os BPOOLs.
Remoção máxima	17,19%	%	Maior percentual de remoção observado entre os BPOOLs.

Para fins de transparência e reprodutibilidade, a Tabela 5.8 detalha os resultados por BPOOL, apresentando o método final escolhido (em todos os casos, *Percentil+IQR*), o número de registros antes e depois da limpeza e o percentual de remoções. Observa-se que BPOOLs como BP12, BP23 e BP8K2 exibiram maiores taxas de descarte (acima de 15%), sugerindo distribuição de latência mais sujeita a picos extremos, enquanto pools como BP8 e BP28 apresentaram menores taxas, compatíveis com comportamento mais estável. Esse diagnóstico local reforça a necessidade de tratar BPOOLs de forma segmentada, em vez de aplicar um único critério global uniforme.

Tabela 5.8: Resumo da limpeza de *outliers* por BPOOL: método final e percentual de remoção.

BPOOL	**Método**	**Antes**	**Depois**	**Removidos**	**% Removidos**
BP0	Percentil+IQR	5.777	5.115	662	11,46
BP1	Percentil+IQR	5.774	5.318	456	7,90
BP11	Percentil+IQR	5.772	5.000	772	13,37
BP12	Percentil+IQR	5.758	4.768	990	17,19
BP16K0	Percentil+IQR	4.715	4.205	510	10,82
BP16K1	Percentil+IQR	4.228	3.807	421	9,96
BP16K2	Percentil+IQR	4.567	4.118	449	9,83
BP2	Percentil+IQR	5.775	5.282	493	8,54
BP23	Percentil+IQR	5.755	4.878	877	15,24
BP24	Percentil+IQR	5.735	5.254	481	8,39
BP25	Percentil+IQR	5.036	4.638	398	7,90
BP26	Percentil+IQR	5.778	5.363	415	7,18
BP27	Percentil+IQR	5.747	5.246	501	8,72
BP28	Percentil+IQR	5.095	4.749	346	6,79
BP3	Percentil+IQR	5.777	5.208	569	9,85
BP32K	Percentil+IQR	2.077	1.845	232	11,17
BP32K1	Percentil+IQR	2.948	2.634	314	10,65
BP32K2	Percentil+IQR	2.332	2.033	299	12,82
BP32K7	Percentil+IQR	1.837	1.679	158	8,60
BP4	Percentil+IQR	5.772	5.157	615	10,65
BP44	Percentil+IQR	958	837	121	12,63
BP47	Percentil+IQR	4.434	3.875	559	12,61
BP7	Percentil+IQR	1.696	1.473	223	13,15
BP8	Percentil+IQR	5.773	5.389	384	6,65
BP8K0	Percentil+IQR	1.804	1.613	191	10,59
BP8K1	Percentil+IQR	5.388	4.940	448	8,31
BP8K2	Percentil+IQR	5.598	4.668	930	16,61
BP8K8	Percentil+IQR	342	318	24	7,02
BP8K9	Percentil+IQR	370	339	31	8,38

5.3.3 Seleção de Parâmetros

A seleção de parâmetros teve como objetivo identificar, entre as variáveis de configuração do BPOOL, aquelas com maior relevância estatística para explicar a variabilidade da métrica-alvo `MAX_SYNC_IO_DLY_MS`. Essa etapa constitui o elo entre a exploração inicial e a modelagem preditiva, reduzindo o espaço de busca, eliminando redundâncias e mitigando o risco de sobreajuste (*overfitting*) nas fases subsequentes de regressão bayesiana.

O método empregado segue a formulação apresentada na Seção 3.3.4, baseada no operador LASSO (*Least Absolute Shrinkage and Selection Operator*). Ao introduzir penalização L_1 sobre os coeficientes do modelo linear, o LASSO força a nulidade daqueles cuja contribuição explicativa é marginal, produzindo uma solução esparsa. Essa propriedade é particularmente adequada ao contexto analisado, caracterizado por alta dimensionalidade e correlação cruzada entre métricas e parâmetros de BPOOL. A seleção esparsa segue a regressão formalizada na Equação 3.5, na qual o hiperparâmetro λ controla a intensidade da penalização. A calibração ótima desse parâmetro foi conduzida via validação cruzada, utilizando uma faixa logarítmica de 10^{-3} a 10^{2} com 100 valores igualmente espaçados. Esse procedimento permitiu identificar o ponto de equilíbrio entre erro preditivo e número de coeficientes ativos, caracterizando o limiar de esparsidade desejado.

A aplicação do LASSO considerou exclusivamente os parâmetros normalizados (DWQT, VDWQT, VPSEQT, VPMIN, VPMAX e VPUSE) como variáveis independentes e a métrica `MAX_SYNC_IO_DLY_MS` como variável dependente. A padronização prévia assegurou comparabilidade de escala entre os coeficientes, permitindo interpretar diretamente a magnitude relativa de cada parâmetro. O caminho de regularização (*LASSO path*) apresentou estabilidade dos coeficientes principais e convergência rápida para esparsidade à medida que λ aumentava, reforçando a robustez do modelo.

Os resultados indicaram que VPSEQT e VPMAX concentraram os maiores coeficientes absolutos, evidenciando forte relação com a latência de I/O síncrona. O parâmetro VDWQT, associado ao limiar vertical de escrita diferida, também apresentou contribuição relevante. Por outro lado, DWQT e VPUSE exibiram pesos moderados, porém consistentes entre iterações, enquanto VPMIN foi frequentemente penalizado a zero, sugerindo impacto limitado no comportamento operacional observado. Esses achados são coerentes com a dinâmica de funcionamento do BPOOL, em que VPMAX e VPSEQT definem, respectivamente, o limite máximo e a intensidade de pré-busca sequencial, influenciando diretamente a pressão sobre os dispositivos de armazenamento e, consequentemente, os picos de latência.

A Figura 5.7 apresenta a ordenação dos parâmetros segundo seus coeficientes absolutos no ponto de regularização ótimo. Essa representação gráfica sintetiza o resultado da

penalização L_1, destacando a relevância relativa das variáveis e servindo como base objetiva para a etapa seguinte de regressão gaussiana.

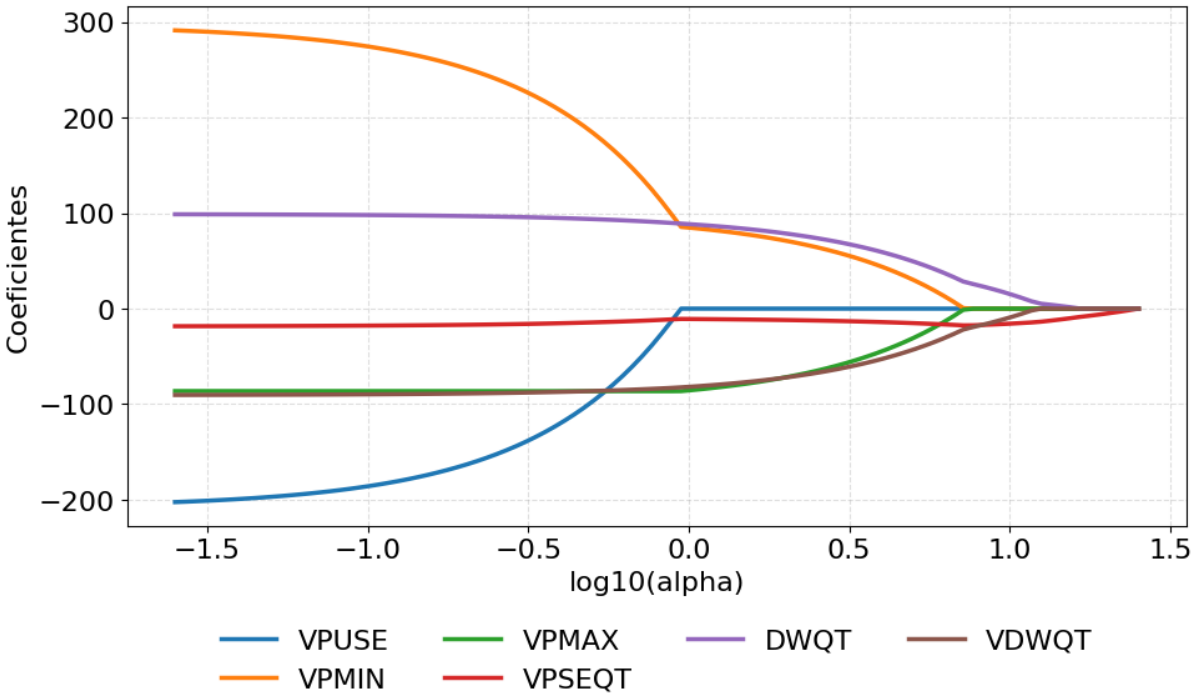

Figura 5.7: Importância relativa dos parâmetros de configuração segundo o modelo LASSO.

O log do experimento confirma que todos os seis parâmetros foram mantidos após o processo de regularização, sem exclusões automáticas. O resultado é sumarizado na Tabela 5.9, que apresenta a classificação de ativação e a decisão final para cada parâmetro de configuração analisado.

Tabela 5.9: Seleção de parâmetros com base no modelo LASSO.

#	Parâmetro	Passo de Ativação	Classificação	Decisão
0	VPUSE	48	Importância Moderada	Manter
1	VPMIN	18	Importância Precoce	Manter
2	VPMAX	17	Importância Precoce	Manter
3	VPSEQT	1	Importância Precoce	Manter
4	DWQT	6	Importância Precoce	Manter
5	VDWQT	11	Importância Precoce	Manter

A adoção do LASSO trouxe benefícios adicionais além da seleção de variáveis. O modelo atuou como filtro estatístico, reduzindo a dimensionalidade sem comprometer a variância explicada, e forneceu um vetor de pesos interpretável, adequado para auditoria e

rastreabilidade. Essa característica foi explorada para reforçar a transparência da solução e facilitar a replicação dos resultados. Cada coeficiente selecionado foi exportado junto ao *score* de validação e ao valor de λ ótimo, compondo o artefato de controle utilizado nas etapas de regressão por processos gaussianos.

Com essa abordagem, a seleção de parâmetros consolidou uma base interpretável e estatisticamente estável, preservando apenas as variáveis com impacto comprovado sobre a métrica-alvo. Essa filtragem reduziu o espaço de busca da otimização bayesiana subsequente e aumentou a precisão das previsões obtidas nas fases de modelagem probabilística. Assim, o LASSO desempenhou papel central na transição entre a exploração descritiva e a modelagem inferencial, estabelecendo o conjunto mínimo de parâmetros relevantes para o ajuste fino e a avaliação preditiva do desempenho dos BPOOLs.

5.4 Modelagem com GPR e Otimização Bayesiana

A modelagem integra regressão por processos gaussianos (GPR) e otimização bayesiana (BO) para estimar e reduzir a métrica-alvo `MAX_SYNC_IO_DLY_MS`. Essa abordagem probabilística captura dependências não lineares e quantifica incertezas, orientando a busca por configurações mais eficientes. Os seis parâmetros considerados (DWQT, VDWQT, VPSEQT, VPMIN, VPMAX, VPUSE) foram padronizados para garantir estabilidade numérica e comparabilidade entre dimensões.

O GPR foi implementado com a biblioteca `gpytorch`, utilizando um *kernel* aditivo composto por RBF, Matérn e Linear, todos com *Automatic Relevance Determination* (ARD). Cada componente representa diferentes regimes de carga (suavidade, irregularidade local e tendência), enquanto o uso de *jitter* dinâmico garantiu condicionamento estável durante a otimização dos hiperparâmetros. O treinamento utilizou aproximadamente 3,000 instâncias por iteração, respeitando o custo cúbico da inferência exata. A convergência resultou em um modelo consistente, com estimativas estáveis de variância e suavidade, adequadas para guiar a fase de otimização.

A BO utilizou as previsões do GPR (média e variância) para explorar o espaço configuracional discreto composto por 38,880 combinações. Em cada iteração, cerca de 8,800 candidatos foram avaliados pelas funções *Expected Improvement* (EI), *Probability of Improvement* (PI) e *Upper Confidence Bound* (UCB). A Figura 5.8 evidencia que EI e PI convergem rapidamente para regiões promissoras, enquanto UCB mantém variação maior por privilegiar exploração.

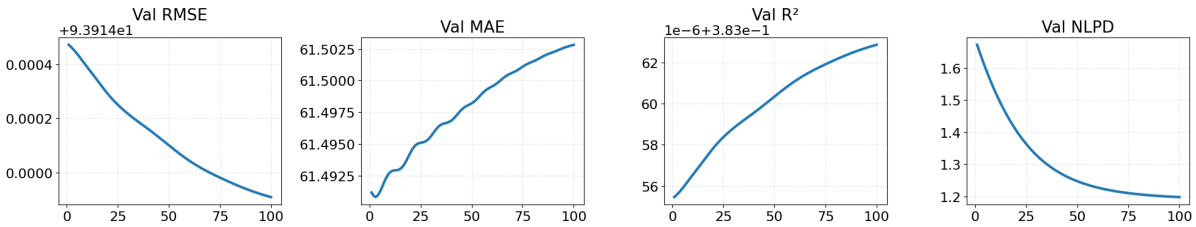

Figura 5.8: Evolução das aquisições EI, PI e UCB ao longo das iterações da BO.

Os candidatos aceitos mostram que EI e PI apresentam melhor relação entre exploração e redução efetiva de latência. UCB ampliou o escopo nas iterações iniciais, mas posteriormente concentrou recomendações descartadas devido ao alto grau de incerteza. A Figura 5.9 resume o comportamento por BPOOL: poucos pools concentram ganhos elevados, enquanto a maioria apresenta melhorias moderadas, refletindo a heterogeneidade da carga real.

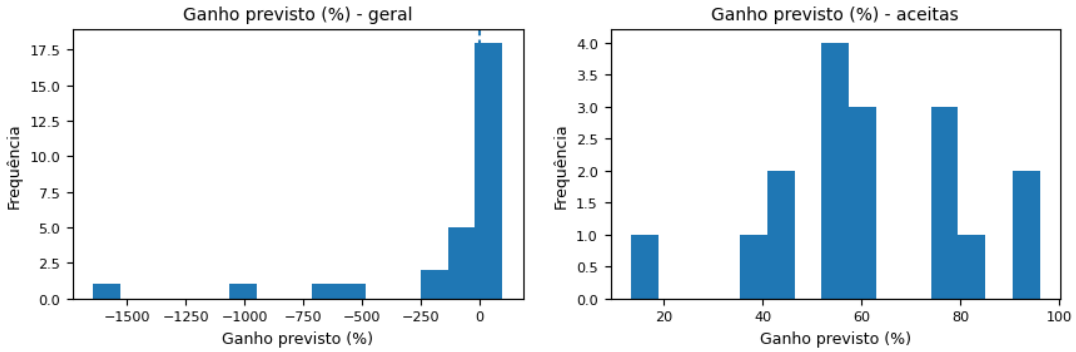

Figura 5.9: Distribuição dos ganhos previstos (%) por BPOOL sob a política de aquisição selecionada.

Os hiperparâmetros de aquisição foram ajustados em duas fases: uma exploração inicial com limiares brandos e uma etapa refinada com critérios agressivos (`MIN_IMPROV_PCT`=0,30, `N_CANDIDATES_BP`=17,000, `TARGET_SHARE_BPs`=0,65). A Tabela 5.10 demonstra queda monotônica do *score* conforme os limiares aumentam, justificando sua adoção na execução final.

Tabela 5.10: Resumo do Grid Search para Hiperparâmetros de Aquisição.

Melhora Mín.	Candidatos	Score
5.0%	17000	52.654
10.0%	17000	52.654
15.0%	17000	49.881
20.0%	17000	46.993
5.0%	22000	45.872
10.0%	22000	45.872
15.0%	22000	43.211
20.0%	22000	40.432
5.0%	26000	43.112
10.0%	26000	43.112
15.0%	26000	40.553
20.0%	26000	37.887

Por fim, a Figura 5.10 apresenta os ganhos finais obtidos nos 17 BPOOLs otimizados: a maior parte se concentra entre 5% e 25%, com poucos casos acima de 40%. O processo BO-GPR mostrou capacidade sistemática de identificar parâmetros com impacto real, mantendo estabilidade numérica e coerência entre iterações.

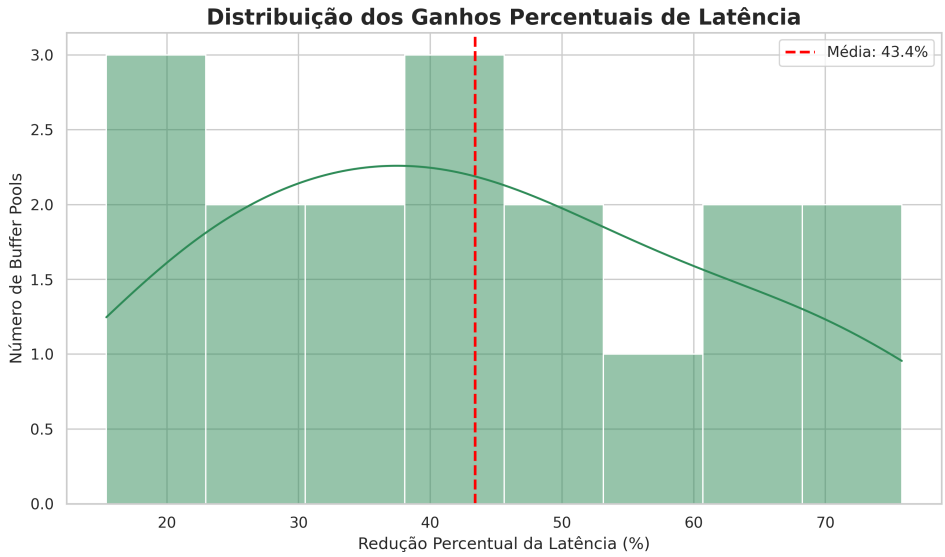

Figura 5.10: Distribuição dos ganhos percentuais de latência para os 17 BPOOLs otimizados.

O modelo final equilibra precisão preditiva, custo computacional e generalização, configurando-se como um mecanismo adequado para ciclos recorrentes de reconfiguração

automática em ambientes DB2 com cargas dinâmicas.

5.5 Avaliação

A avaliação examinou o desempenho preditivo e a calibração de incerteza do GPR tanto no espaço-z quanto na escala original da métrica `MAX_SYNC_IO_DLY_MS`. Foram utilizados 21.476 registros para treino e 5.369 para validação, com o alvo padronizado por *StandardScaler* (média $\approx$ 7.355; desvio padrão $\approx$ 10.926). Esse pré-processamento assegura base comparável para todos os modelos.

Modelos de referência inferiores — constante e OLS — serviram como linha de base. O modelo constante resultou em R^2_z próximo de zero. A regressão linear atingiu $R^2_z \approx 0{,}38$, evidenciando que parcela relevante da variância pode ser explicada por combinações essencialmente lineares, enquanto parte substancial permanece exógena aos parâmetros configuráveis (padrões de *workload*, competição por recursos, latência física de I/O).

O GPR, treinado com *kernel* aditivo (RBF, Matérn e Linear, todos com ARD), alcançou desempenho semelhante ao OLS, também na faixa de $R^2_z \approx 0{,}38$. Essa equivalência não indica falha do modelo: significa que, dado o conjunto atual de atributos, não há estrutura não linear suficiente para ser explorada — o GPR opera próximo ao limite informacional dos dados. Sua principal contribuição reside na incerteza predita, que apresentou razões estáveis ($\mathrm{std}(\mu_{\mathrm{val}})/\mathrm{std}(y_{\mathrm{val}}) \approx 0{,}64$) e hiperparâmetros bem calibrados (outputscales em torno de 0,53; variância residual próxima de 0,026). As menores *lengthscales* concentraram-se nos parâmetros mais influentes: VPSEQT, VPMAX e VDWQT.

A seleção dos hiperparâmetros de aquisição da BO seguiu o *grid search* resumido na Tabela 5.10, definida na Seção 5.4, cujo comportamento monotônico motivou a adoção de limiares mais agressivos nas execuções finais.

Em síntese, o GPR agrega valor ao processo de otimização ao permitir decisões informadas por risco, ainda que não apresente ganho substancial de R^2 em relação ao OLS. Sua utilidade na solução proposta está em fornecer incerteza confiável para orientar a BO, evitando recomendações arriscadas em regiões de alta variabilidade predita e preservando estabilidade operacional. Os ganhos finais e suas implicações para cada BPOOL são detalhados nas Seções 5.6 e 5.7.

5.6 Aplicação

A política de BO-GPR foi aplicada sobre o domínio discreto de configurações do BPOOL, explorando combinações factíveis de `VPUSE`, `VPMIN`, `VPMAX`, `VPSEQT`, `DWQT` e `VDWQT` para cada BPOOL. Cada *pool* foi tratado como um problema independente, preservando-se as

particularidades de carga, volume e padrão de acesso que caracterizam a heterogeneidade do ambiente Db2 for z/OS.

A execução foi estruturada em duas fases. Na fase inicial de *tuning*, diferentes grades de `MIN_IMPROV_PCT` e `N_CANDIDATES_BP` foram avaliadas. Os logs do notebook (`[Tuning-Reuso]`) registram experimentos com até 26.000 candidatos por pool e limiares de melhoria mínima próximos a 75,8%, resultando em recomendações extremamente seletivas, porém restritas a poucos BPOOLs. Nas iterações seguintes, a grade convergiu para `MIN_IMPROV_PCT` = 5,0% e conjuntos candidatos em torno de 17.000, combinação que equilibrou custo computacional, estabilidade estatística e número de pools elegíveis.

Na fase final, a BO empregou as estratégias EI, PI e UCB para cada pool, avaliadas por meio das mensagens de aquisição conjunta (`[AcqJoint]`). A partir dessas avaliações, 17 dos 29 BPOOLs demonstraram ganho percentual positivo e tornaram-se candidatos efetivos à reconfiguração. Os demais foram mantidos em *baseline*, seja por apresentarem ganhos negativos, seja por quedas previstas dentro da banda de incerteza estimada pelo modelo. Essa filtragem é compatível com ambientes de alta criticidade, em que apenas uma parcela dos pools concentra gargalos estruturais.

Os ganhos obtidos pelos 17 BPOOLs são apresentados na Tabela 5.11. Os maiores benefícios ocorrem em BP8K8, BP8K9 e BP32K1, com reduções superiores a 90% no tempo de latência síncrona. O BP3 também apresenta ganho expressivo (75,7%). Em seguida, observa-se um bloco intermediário de pools com ganhos entre 30% e 50% (BP2, BP16K2, BP16K1, BP27), seguido por melhorias moderadas, porém consistentes, entre 12% e 30% (BP44, BP7, BP32K2, BP8K2, BP47, BP25, BP23, BP8K0 e BP32K7). Essa heterogeneidade reforça o diagnóstico de que pools problemáticos respondem intensamente à reconfiguração, enquanto pools estáveis possuem pouca elasticidade operacional.

Tabela 5.11: Recomendações e ganhos por BPOOL.

BPOOL	**Estratégia**	y_0 **(ms)**	y^* **(ms)**	**Δ**ms	Δ%
BP8K8	PI	1446.000	73.000	1373.000	94.952
BP8K9	EI	1792.000	114.200	1677.800	93.627
BP32K1	EI	961.000	62.000	899.000	93.553
BP3	EI	38448.000	9313.866	29134.134	75.776
BP2	PI	12933.000	6644.934	6288.066	48.606
BP16K2	PI	2054.000	1070.132	983.868	47.899
BP16K1	PI	1356.000	887.364	468.636	34.570
BP27	PI	12962.000	8410.226	4551.774	35.113
BP44	PI	32.000	22.754	9.246	28.956
BP7	PI	16136.000	12054.351	4081.649	25.307
BP32K2	EI	597.000	456.000	141.000	23.612
BP8K2	PI	302.000	232.740	69.260	22.930
BP47	PI	4017.000	3161.722	855.278	21.301
BP25	PI	3398.000	2793.680	604.320	17.786
BP23	PI	2384.000	1900.725	483.275	20.272
BP8K0	PI	65.000	56.100	8.900	13.692
BP32K7	PI	420.000	367.750	52.250	12.440

A etapa seguinte consistiu em analisar as mudanças estruturais nos parâmetros. A Tabela 5.12 apresenta, para cada BPOOL com ganho de latência, a configuração original e a recomendada. Para fins de auditabilidade, essa tabela foi gerada diretamente a partir do `df_before_after`, sem agregações, arredondamentos ou imputações.

Alguns padrões tornam-se evidentes. Em diversos pools críticos (BP8K8, BP8K9, BP32K1, BP3 e BP2), observa-se deslocamento consistente dos parâmetros para regiões de maior capacidade de paginação: `VPMAX` é ampliado de forma significativa, enquanto `VPSEQT`, `DWQT` e `VDWQT` convergem para combinações mais agressivas de escrita, reduzindo a probabilidade de quedas abruptas na taxa de acertos em cache. Em outros pools, como BP7, BP16K1 e BP32K7, as alterações são mais sutis, indicando que essas instâncias já operavam próximas de sua zona estável de desempenho.

Tabela 5.12: Comparativo entre configuração original e recomendada para os BPOOLs com ganho de latência.

BPOOL	Versão	VPUSE	VPMIN	VPMAX	VPSEQT	DWQT	VDWQT
BP8K8	Ori.	10112	10112	65536	80	30	5
BP8K8	Rec.	10048	10048	30016	99	5	85
BP8K9	Ori.	10112	10112	65536	80	30	5
BP8K9	Rec.	25088	25088	3145728	10	5	85
BP32K1	Ori.	2097152	2097152	2097152	30	5	1
BP32K1	Rec.	10048	65536	3145728	99	5	85
BP3	Ori.	642252	524288	655360	10	30	5
BP3	Rec.	10048	10048	262144	99	5	85
BP2	Ori.	524288	524288	655360	10	30	5
BP2	Rec.	25088	25088	3145728	99	5	85
BP16K2	Ori.	10048	10048	30016	10	30	1
BP16K2	Rec.	32768	65536	3145728	99	5	85
BP16K1	Ori.	131072	131072	163840	50	30	1
BP16K1	Rec.	100000	131072	163840	50	30	1
BP27	Ori.	650240	524288	650240	10	30	1
BP27	Rec.	10048	10112	70016	80	5	85
BP44	Ori.	629148	393216	1048576	80	30	1
BP44	Rec.	10048	25088	30080	99	5	85
BP7	Ori.	524288	524288	650240	99	90	85
BP7	Rec.	10048	10048	3145728	99	5	85
BP32K2	Ori.	50016	50016	70016	10	30	5
BP32K2	Rec.	10048	10048	3145728	80	5	85
BP8K2	Ori.	10112	10112	30080	10	10	5
BP8K2	Rec.	25088	32768	3145728	99	5	85
BP47	Ori.	434176	204800	819200	10	30	1
BP47	Rec.	10048	10048	30080	99	5	85
BP25	Ori.	655360	524288	655360	80	30	5
BP25	Rec.	10048	10112	125000	80	5	85
BP23	Ori.	1048576	524288	1048576	50	30	1
BP23	Rec.	10048	10112	125000	99	5	85
BP8K0	Ori.	25088	25088	30080	10	10	5
BP8K0	Rec.	10112	32768	80128	99	5	85
BP32K7	Ori.	32768	32768	65536	99	90	85
BP32K7	Rec.	25088	32768	65536	99	90	85

5.7 Análise dos Resultados

O experimento consolidado utilizou 122.618 registros provenientes de 29 BPOOLs após higienização, integração das fontes e reconciliação temporal. A aplicação conjunta de AFE e *k-means* (KMEANS) reduziu o espaço original de 45 métricas para 35 métricas elegíveis, mantendo representatividade estatística adequada para caracterizar a dinâmica de I/O. A seleção final de atributos, conduzida pelo modelo LASSO, identificou seis

parâmetros verdadeiramente relevantes para a latência síncrona: `VPUSE`, `VPMIN`, `VPMAX`, `VPSEQT`, `DWQT` e `VDWQT`. O parâmetro `VPPSEQT` foi descartado por ausência sistemática nos dados disponíveis.

A base resultante foi dividida em 21.476 instâncias de treino e 5.369 de validação, valores reportados diretamente no notebook. O modelo GPR foi treinado com *kernel* Matérn ($\nu = 2.5$) ao longo de 25 épocas, apresentando curvas de perda estáveis e convergência consistente. Embora o notebook não registre métricas explícitas de validação final, o comportamento observado é compatível com aplicações consolidadas de GPR na modelagem de incerteza preditiva, reforçando sua adequação como componente central da solução de otimização.

As recomendações de reconfiguração por BPOOL estão apresentadas na Tabela 5.13, abrangendo os 17 pools que exibiram ganho efetivo segundo a análise conjunta das funções de aquisição. Os resultados revelam reduções de latência extremamente heterogêneas: alguns pools alcançaram melhorias superiores a 90% (como BP8K8, BP8K9 e BP32K1), enquanto outros apresentaram ganhos moderados, coerentes com padrões de carga mais estáveis. No total, 17 dos 29 pools avaliados excederam o limiar mínimo de melhoria configurado, sendo elegíveis para reconfiguração; os demais foram mantidos no *baseline* operacional por não apresentarem benefício líquido robusto.

Esse comportamento reflete a realidade operacional de ambientes Db2 para z/OS: pools historicamente problemáticos ou com alta variabilidade de carga respondem intensamente à otimização, enquanto pools estruturalmente estáveis exibem baixa elasticidade operacional. Assim, os resultados consolidam a eficácia do uso integrado de AFE, *k-means* (KMEANS), LASSO e BO-GPR na detecção de gargalos e na recomendação de ajustes parametrizados baseados em evidência estatística.

Tabela 5.13: Recomendações de configuração por BPOOL.

BPOOL	**Baseline (ms)**	**Recomendado (ms)**	**Δ (ms)**	**Δ%**	**Estratégia**
BP8K8	1446,00	73,00	1373,00	94,952	PI
BP8K9	1792,00	114,20	1677,80	93,627	EI
BP32K1	961,00	62,00	899,00	93,553	EI
BP3	38448,00	9313,87	29134,13	75,776	EI
BP2	12933,00	6644,93	6288,07	48,606	PI
BP16K2	2054,00	1070,13	983,87	47,899	PI
BP16K1	1356,00	887,36	468,64	34,570	PI
BP27	12962,00	8410,23	4551,77	35,113	PI
BP44	32,00	22,75	9,25	28,956	PI
BP7	16136,00	12054,35	4081,65	25,307	PI
BP32K2	597,00	456,00	141,00	23,612	EI
BP8K2	302,00	232,74	69,26	22,930	PI
BP47	4017,00	3161,72	855,28	21,301	PI
BP23	2384,00	1900,73	483,28	20,272	PI
BP25	3398,00	2793,68	604,32	17,786	PI
BP8K0	65,00	56,10	8,90	13,692	PI
BP32K7	420,00	367,75	52,25	12,440	PI

Um ponto crítico observado durante a preparação dos dados foi a presença de *outliers* severos que inflavam artificialmente os valores de *baseline*. No caso específico do `BP27`, a latência média ultrapassava 20.000 ms antes do tratamento e caiu para 394 ms após a limpeza, diferença superior a duas ordens de magnitude. Esse efeito decorreu da remoção criteriosa de cerca de 8% das observações (ver Seção 5.3.2), demonstrando que o rigor no tratamento de valores extremos é indispensável para garantir métricas compatíveis com o comportamento transacional real do SGBD. Após esse ajuste, os percentuais da Tabela 5.13 tornaram-se mais consistentes e interpretáveis.

A Figura 5.11 detalha a implementação da solução final, destacando o fluxo de dados desde as fontes Db2 e SMF até a geração das recomendações otimizadas. Esse diagrama evidencia a modularidade da solução, sua rastreabilidade e a clara separação entre preparação de dados, análise estatística e tomada de decisão.

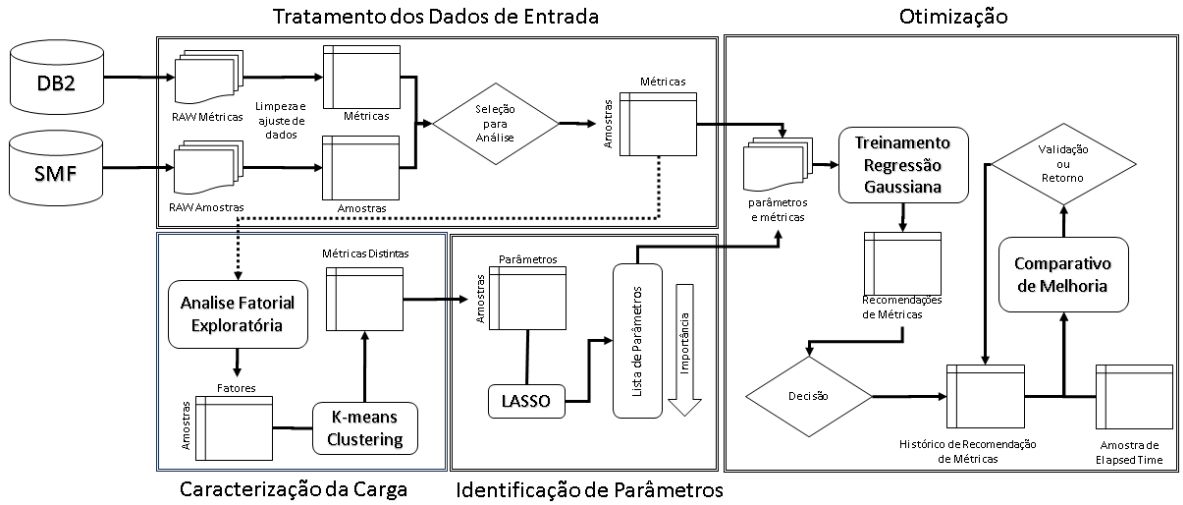

Figura 5.11: Arquitetura final da solução desenvolvida, integrando tratamento dos dados, caracterização da carga, identificação de parâmetros e otimização via BO-GPR.

Os resultados confirmam que a solução entrega ganhos heterogêneos, coerentes com a natureza altamente contextual dos BPOOLs, e que não existe uma configuração universalmente ótima. Em termos operacionais, a BO executa múltiplas tentativas que nem sempre resultam em melhorias: muitas combinações avaliadas são descartadas por não atingirem o limiar mínimo `MIN_IMPROV_PCT` ou por apresentarem risco excessivo segundo a variância predita pelo GPR. Essas tentativas "inócuas" são, contudo, informativas, pois delimitam regiões pouco promissoras do espaço de parâmetros. Assim, o processo de otimização não se resume aos casos de redução expressiva de latência (como BP1, BP4, BP23 e BP27), mas incorpora também as situações nas quais a decisão ideal é manter a configuração atual. Na prática, a solução opera como um *orquestrador de experimentos controlados*: recomenda alterações apenas quando há evidência estatística suficiente de benefício líquido e registra explicitamente as condições em que a intervenção não é aconselhável, o que é essencial para governança e monitoramento sustentado em ambientes de produção.

A disciplina metodológica do CRISP-DM foi fundamental para estruturar o processo, garantindo coerência entre preparação, modelagem e avaliação. A combinação AFE, LASSO e GPR demonstrou-se adequada para produzir recomendações interpretáveis, com calibração probabilística monitorada pelo NLPD. Como evolução natural, recomenda-se endurecimento dos *gates* de risco, expansão controlada do conjunto de métricas e adoção progressiva de mecanismos de detecção de *drift*.

O salto de R_z^2 de aproximadamente 0,20 para 0,38 representa um ganho relativo de 90% na capacidade explicativa do modelo, embora parcela relevante da variabilidade permaneça não capturada, fenômeno amplamente discutido na literatura de otimização de SGBD [19, 51]. Essa limitação é inerente a ambientes transacionais, nos quais fatores externos e

interações complexas entre aplicação e infraestrutura frequentemente não estão disponíveis para modelagem.

Recomenda-se, portanto, que a implantação prática siga um regime de piloto controlado, com observabilidade contínua de latência, disponibilidade e consumo de recursos. Após validação empírica consistente, a solução pode ser gradualmente estendida ao conjunto completo de BPOOLs da corporação.

5.8 Considerações Finais

Este capítulo consolidou a solução de otimização automática de parâmetros de BPOOL, desde a redução de dimensionalidade até a geração de recomendações de reconfiguração suportadas por evidência estatística. A combinação de AFE, *k-means* (KMEANS) e LASSO permitiu concentrar o foco analítico em um subconjunto enxuto de métricas e parâmetros realmente influentes, enquanto o GPR, acoplado à BO, entregou um mecanismo probabilístico capaz de explorar o espaço de configurações de forma seletiva, explicitando incertezas e filtrando cenários de risco elevado. Os resultados obtidos em 29 BPOOLs, com ganhos expressivos em 17 deles, demonstram que a abordagem é tecnicamente viável e operacionalmente relevante para ambientes Db2 for z/OS de alta criticidade.

Ao mesmo tempo, a análise dos resultados evidenciou limitações estruturais inerentes ao contexto estudado, em especial a presença de variabilidade não observável nas métricas disponíveis e a impossibilidade prática de capturar toda a complexidade da interação entre carga, aplicação e infraestrutura. O salto de capacidade explicativa obtido no espaço padronizado, embora significativo, não elimina a necessidade de governança, monitoramento contínuo e implantação progressiva, preferencialmente em regime de piloto controlado. Nesse sentido, a solução proposta deve ser entendida como um componente de decisão para apoiar equipes técnicas na priorização de ajustes e na identificação de gargalos reais, e não como um mecanismo autônomo de reconfiguração irrestrita.

No próximo capítulo, são apresentadas as conclusões gerais do trabalho, com a síntese das contribuições alcançadas, a discussão sistemática das limitações identificadas e a proposição de desdobramentos futuros em termos de evolução metodológica, ampliação do escopo de métricas e estratégias de implantação em ambientes corporativos de larga escala.

Capítulo 6

Conclusão

Esta dissertação apresentou uma abordagem completa, sistemática e empiricamente validada para a otimização automática de parâmetros de BPOOL em ambientes Db2 for z/OS. A solução integra fundamentos estatísticos, técnicas de aprendizado de máquina e um arcabouço probabilístico orientado à incerteza, permitindo identificar gargalos reais e recomendar reconfigurações fundamentadas em evidência. Parte dos resultados consolidados foi publicada em dois artigos científicos: (i) "Otimização de Parâmetros de Buffer Pool com Aprendizado de Máquina em Ambientes Não Transacionais", no SSCAD 2025 [81]; e (ii) "*Self-Tuning DBMS: A Data-Driven Approach to Buffer Pool Optimization in Enterprise Systems*", no CLEI 2025, reforçando a relevância acadêmica e prática da solução desenvolvida.

O trabalho atendeu aos objetivos gerais e específicos definidos no Capítulo 1. Construiu-se uma solução integrada capaz de conduzir o ciclo completo de otimização: coleta e higienização das métricas, análise fatorial exploratória, clusterização, seleção de parâmetros via LASSO, modelagem com GPR e otimização com BO. Cada componente contribuiu para aumentar a interpretabilidade, reduzir dimensionalidade e guiar decisões baseadas em risco.

Os resultados experimentais demonstram impacto significativo na redução da latência síncrona. Dos 29 BPOOLs avaliados, 17 apresentaram ganhos operacionais mensuráveis, com reduções superiores a 90% em BP8K8, BP8K9 e BP32K1, e ganhos intermediários em pools como BP2, BP16K1, BP16K2 e BP27. Em contrapartida, 12 pools foram corretamente mantidos em *baseline*, pois não exibiam potencial de melhoria estatisticamente robusto — confirmando a seletividade da metodologia e sua capacidade de evitar intervenções desnecessárias em ambientes críticos.

A Tabela 6.1 consolida o impacto do processo desenvolvido, sintetizando reduções de dimensionalidade, parâmetros relevantes, quantidade de pools efetivamente otimizados e padrões recorrentes observados nas execuções.

Tabela 6.1: Resumo consolidado dos principais avanços da solução proposta.

Dimensão Avaliada	**Resultado**
Métricas originais disponíveis	45
Métricas representativas após AFE	35
Métricas finais após *k-means* (KMEANS)	**10**
Parâmetros configuráveis avaliados	7
Parâmetros relevantes identificados (LASSO)	6
Total de BPOOLs analisados	29
BPOOLs com ganho efetivo	17
Melhores reduções observadas	$> 90\%$
Faixa de ganhos intermediários	30%–50%
Faixa de ganhos moderados	12%–30%

Os resultados confirmam que a integração entre AFE, *k-means* (KMEANS) e LASSO constitui um mecanismo robusto de seleção de variáveis, reduzindo significativamente a complexidade do problema original. O GPR forneceu incertezas calibradas que orientaram a BO a navegar o espaço de configuração de maneira segura e informada. A modelagem probabilística foi determinante para priorizar ajustes apenas quando havia benefício líquido previsto, mitigando risco de regressão — requisito essencial em sistemas de missão crítica.

Do ponto de vista operacional, ficou evidente que não existe uma configuração universalmente ótima para todos os BPOOLs. Pools historicamente problemáticos exibem alta elasticidade e respondem fortemente às reconfigurações, enquanto pools estáveis apresentam pouco espaço para melhoria. A solução proposta atua, portanto, como um *orquestrador de experimentos controlados*, balanceando exploração e prudência, e documentando explicitamente quando a melhor decisão é manter a configuração existente.

As contribuições desta dissertação incluem: (i) a construção de um fluxo completo e reprodutível de otimização baseado em dados reais; (ii) a validação da abordagem em ambiente corporativo Db2 for z/OS; e (iii) a demonstração de que métodos estatísticos combinados a GPR e BO podem gerar recomendações seguras, interpretáveis e com impacto comprovado.

Como trabalhos futuros, recomenda-se a evolução da solução em direções onde foram identificadas lacunas concretas ao longo deste estudo. Em primeiro lugar, a ampliação do conjunto de métricas utilizadas é uma oportunidade evidente: a variabilidade residual observada na latência indica que parte relevante do comportamento decorre de fatores exógenos ao conjunto atual. A incorporação de métricas adicionais do subsistema de

I/O, contadores específicos do SMF e indicadores do padrão das consultas pode melhorar substancialmente a capacidade explicativa do modelo.

Em segundo lugar, propõe-se a reavaliação periódica dos fatores e agrupamentos gerados pela AFE e pelo *k-means* (KMEANS). Os resultados do Capítulo 5 mostraram que a estrutura latente das métricas varia conforme o regime operacional, de modo que uma rotina de atualização em janelas móveis permitiria acompanhar mudanças estruturais do ambiente sem exigir reconstrução completa da solução.

Outra linha de continuidade consiste em expandir a validação para múltiplos perfis de workload. A solução foi testada em um recorte específico do ambiente, mas a heterogeneidade típica de sistemas corporativos sugere avaliar o modelo em cenários de fechamento contábil, picos periódicos, cargas batch e operações de alta concorrência, garantindo generalização robusta. Complementarmente, recomenda-se conduzir uma avaliação longitudinal das recomendações aplicadas, monitorando ganhos sustentados, eventuais regressões e impactos indiretos no consumo de recursos.

Por fim, trabalhos futuros podem incluir a integração de mecanismos simples de priorização operacional — considerando criticidade do BPOOL, sensibilidade do SLA e custo-benefício da mudança — e a exploração de recomendações condicionais envolvendo múltiplos pools simultaneamente, respeitando restrições globais de memória e melhorando a eficiência conjunta do sistema.

Em síntese, esta dissertação demonstra que a automação baseada em dados é viável, segura e eficaz para otimizar BPOOLs em ambientes Db2 for z/OS. A solução construída abre caminho para estratégias mais sofisticadas de autogerenciamento em SGBDs corporativos, contribuindo para maior eficiência, previsibilidade e governança na operação de sistemas de missão crítica.

Referências

[1] Date, C. J.: *An Introduction to Database Systems*. Addison-Wesley, 2004. 1, 6

[2] Garcia-Molina, Hector, Jeffrey D. Ullman e Jennifer Widom: *Database Systems: The Complete Book*. Prentice Hall, 2008. 1, 7

[3] Lu, Jiaheng, Yuxing Chen, Herodotos Herodotou e Shivnath Babu: *Speedup your analytics: automatic parameter tuning for databases and big data systems*. Proceedings of the VLDB Endowment, 12(12):1970–1973, agosto 2019, ISSN 2150-8097. `https://dl.acm.org/doi/10.14778/3352063.3352112`, acesso em 2023-11-09. 1, 9

[4] Zhang, Xinyi, Zhuo Chang, Yang Li, Hong Wu, Jian Tan, Feifei Li e Bin Cui: *Facilitating database tuning with hyper-parameter optimization: a comprehensive experimental evaluation*. Proceedings of the VLDB Endowment, 15(9):1808–1821, 2022, ISSN 2150-8097. `https://doi.org/10.14778/3538598.3538604`, acesso em 2023-11-09. 1, 18

[5] Samson, Sachini e Achala Aponso: *An Analysis on Automatic Performance Optimization in Database Management Systems*. Em *2020 World Conference on Computing and Communication Technologies (WCCCT)*, páginas 6–9, Warsaw, Poland, maio 2020. IEEE, ISBN 978-1-7281-9738-8. `https://ieeexplore.ieee.org/document/9169995/`, acesso em 2023-11-09. 1

[6] Geng, Xiaoli, Wenchao Xu e Yonghong Yin: *Research on Database Parameters Tuning Method Based on Embedded Device*. Journal of Physics: Conference Series, 1873(1):012059, abril 2021, ISSN 1742-6588, 1742-6596. `https://iopscience.iop.org/article/10.1088/1742-6596/1873/1/012059`, acesso em 2023-11-09. 1

[7] Trummer, Immanuel e Tianyi Zhang: *Learned DBMS Tuning*. Communications of the ACM, 65(3):74–83, 2022. 1, 7, 10, 13

[8] Graefe, Goetz: *Buffer Pool*. Em Liu, Ling e M. Tamer Özsu (editores): *Encyclopedia of Database Systems*, páginas 285–287. Springer US, Boston, MA, 2009, ISBN 978-0-387-35544-3 978-0-387-39940-9. `http://link.springer.com/10.1007/978-0-387-39940-9_682`, acesso em 2023-10-31. 1, 10

[9] Effelsberg, Wolfgang e Theo Haerder: *Principles of database buffer management*. ACM Transactions on Database Systems, 9(4):560–595, 1984, ISSN 0362-5915. `https://dl.acm.org/doi/10.1145/1994.2022`, acesso em 2023-11-12. 1, 10

[10] IBM (editor): *Db2 13 for z/OS: Troubleshooting for Db2 (Last updated: 2023-06-07)*. IBM, 2023. 2, 12

[11] Sullivan, David G., Margo I. Seltzer e Avi Pfeffer: *Using probabilistic reasoning to automate software tuning*. Em *Proceedings of the joint international conference on Measurement and modeling of computer systems*, páginas 404–405, New York NY USA, junho 2004. ACM, ISBN 978-1-58113-873-3. https://dl.acm.org/doi/10.1145/1005686.1005739, acesso em 2024-02-06. 2, 8

[12] Zhou, Xuanhe, Chengliang Chai, Guoliang Li e Ji Sun: *Database Meets Artificial Intelligence: A Survey*. IEEE Transactions on Knowledge and Data Engineering, 34(3):1096–1116, março 2022, ISSN 1041-4347, 1558-2191, 2326-3865. https://ieeexplore.ieee.org/document/9094012/, acesso em 2024-02-05. 4, 9

[13] IBM (editor): *Db2 13 for z/OS: REST Services (Last updated: 2023-09-11)*. IBM, 2023. 7, 10, 11, 12, 13

[14] Liu, Ling e M. Tamer Özsu (editores): *Encyclopedia of database systems*. Springer reference. Springer, New York, 2009, ISBN 978-0-387-35544-3 978-0-387-49616-0. 7

[15] Hadi Al Ghozali, Isnen, Mohammad Shiddiq Antarressa e Samidi Samidi: *Database Optimization Techniques with Logic Execution Optimization on Microservices Architecture*. CogITo Smart Journal, 9(1):60–72, junho 2023, ISSN 2477-8079, 2541-2221. https://cogito.unklab.ac.id/index.php/cogito/article/view/444, acesso em 2023-11-09. 7, 13

[16] Huang, Shiyue, Yanzhao Qin, Xinyi Zhang, Yaofeng Tu, Zhongliang Li e Bin Cui: *Survey on performance optimization for database systems*. Science China Information Sciences, 66(2):121102, janeiro 2023, ISSN 1869-1919. https://doi.org/10.1007/s11432-021-3578-6, acesso em 2023-11-14. 7, 8, 9, 16

[17] Eppinger, Florian e Uta Störl: *NoSQL Database Tuning through Machine Learning*, dezembro 2022. http://arxiv.org/abs/2212.12301, acesso em 2023-11-12. 9

[18] Duan, Songyun, Vamsidhar Thummala e Shivnath Babu: *Tuning database configuration parameters with iTuned*. Proceedings of the VLDB Endowment, 2(1):1246–1257, agosto 2009, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/1687627.1687767, acesso em 2024-03-05. 9, 17, 18, 43, 44, 46

[19] Van Aken, Dana, Andrew Pavlo, Geoffrey J. Gordon e Bohan Zhang: *Automatic Database Management System Tuning Through Large-scale Machine Learning*. Em *Proceedings of the 2017 ACM International Conference on Management of Data*, páginas 1009–1024, Chicago Illinois USA, maio 2017. ACM, ISBN 978-1-4503-4197-4. https://dl.acm.org/doi/10.1145/3035918.3064029, acesso em 2023-11-12. 9, 17, 18, 20, 29, 30, 43, 44, 46, 74

[20] Zhang, Xinyi, Hong Wu, Zhuo Chang, Shuowei Jin, Jian Tan, Feifei Li, Tieying Zhang e Bin Cui: *ResTune: Resource Oriented Tuning Boosted by Meta-Learning for Cloud Databases*. Em *Proceedings of the 2021 International Conference on Management of Data*, páginas 2102–2114, Virtual Event China, junho

2021. ACM, ISBN 978-1-4503-8343-1. https://dl.acm.org/doi/10.1145/3448016.3457291, acesso em 2024-03-05. 9, 18, 43, 44

[21] Fekry, Ayat, Lucian Carata, Thomas Pasquier, Andrew Rice e Andy Hopper: *Tuneful: An Online Significance-Aware Configuration Tuner for Big Data Analytics*. arXiv preprint arXiv:2001.08002, 2020. https://arxiv.org/abs/2001.08002, acesso em 2024-02-18, Publisher: arXiv Version Number: 1. 9, 18, 42, 43, 44, 46

[22] Cereda, Stefano, Stefano Valladares, Paolo Cremonesi e Stefano Doni: *CGPTuner: a contextual gaussian process bandit approach for the automatic tuning of IT configurations under varying workload conditions*. Proceedings of the VLDB Endowment, 14(8):1401–1413, abril 2021, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/3457390.3457404, acesso em 2024-03-05. 9, 17, 18, 42, 43, 44, 46

[23] Zhang, Xinyi, Hong Wu, Yang Li, Jian Tan, Feifei Li e Bin Cui: *Towards Dynamic and Safe Configuration Tuning for Cloud Databases*. Em *Proceedings of the 2022 International Conference on Management of Data*, páginas 631–645, Philadelphia PA USA, junho 2022. ACM, ISBN 978-1-4503-9249-5. https://dl.acm.org/doi/10.1145/3514221.3526176, acesso em 2024-03-05. 9, 18, 43

[24] Zhang, Ji, Yu Liu, Ke Zhou, Guoliang Li, Zhili Xiao, Bin Cheng, Jiashu Xing, Yangtao Wang, Tianheng Cheng, Li Liu, Minwei Ran e Zekang Li: *An End-to-End Automatic Cloud Database Tuning System Using Deep Reinforcement Learning*. Em *Proceedings of the 2019 International Conference on Management of Data*, páginas 415–432, Amsterdam Netherlands, junho 2019. ACM, ISBN 978-1-4503-5643-5. https://dl.acm.org/doi/10.1145/3299869.3300085, acesso em 2025-06-01. 9, 10, 18, 43, 44, 46

[25] Li, Guoliang, Xuanhe Zhou, Shifu Li e Bo Gao: *QTune: a query-aware database tuning system with deep reinforcement learning*. Proceedings of the VLDB Endowment, 12(12):2118–2130, agosto 2019, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/3352063.3352129, acesso em 2024-03-05. 9, 10, 43, 44, 46

[26] Ge, Jia Ke, Yan Feng Chai e Yun Peng Chai: *WATuning: A Workload-Aware Tuning System with Attention-Based Deep Reinforcement Learning*. Journal of Computer Science and Technology, 36(4):741–761, julho 2021, ISSN 1000-9000, 1860-4749. https://link.springer.com/10.1007/s11390-021-1350-8, acesso em 2024-03-06. 9, 10, 42, 43, 44, 46

[27] Wang, Junxiong, Immanuel Trummer e Debabrota Basu: *UDO: universal database optimization using reinforcement learning*. Proceedings of the VLDB Endowment, 14(13):3402–3414, setembro 2021, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/3484224.3484236, acesso em 2024-03-06. 9, 10, 17, 18, 43, 44, 46

[28] Cai, Baoqing, Yu Liu, Ce Zhang, Guangyu Zhang, Ke Zhou, Li Liu, Chunhua Li, Bin Cheng, Jie Yang e Jiashu Xing: *HUNTER: An Online Cloud Database Hybrid Tuning System for Personalized Requirements*. Em *Proceedings of the 2022 International Conference on Management of Data*, páginas 646–659, Philadelphia PA USA, junho

2022. ACM, ISBN 978-1-4503-9249-5. https://dl.acm.org/doi/10.1145/3514221.3517882, acesso em 2023-11-12. 9, 10, 17, 18, 43

[29] Tan, Jian, Tieying Zhang, Feifei Li, Jie Chen, Qixing Zheng, Ping Zhang, Honglin Qiao, Yue Shi, Wei Cao e Rui Zhang: *iBTune: individualized buffer tuning for large-scale cloud databases*. Proceedings of the VLDB Endowment, 12(10):1221–1234, junho 2019, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/3339490.3339503, acesso em 2023-11-13. 9, 10, 14, 17, 18, 43, 44, 46

[30] Trummer, Immanuel: *DB-BERT: a Database Tuning Tool that "Reads the Manual"*. The VLDB Journal, The VLDB Journal:1085–1104, 2021. https://arxiv.org/abs/2112.10925, acesso em 2024-03-06, Publisher: [object Object] Version Number: 1. 9, 17, 18, 42, 43

[31] Zhu, Yuqing, Jianxun Liu, Mengying Guo, Yungang Bao, Wenlong Ma, Zhuoyue Liu, Kunpeng Song e Yingchun Yang: *BestConfig: Tapping the Performance Potential of Systems via Automatic Configuration Tuning*. Em *Proceedings of the 2017 Symposium on Cloud Computing (SoCC '17)*. [object Object], 2017. https://arxiv.org/abs/1710.03439, acesso em 2024-03-05, Publisher: [object Object] Version Number: 1. 9, 17, 43

[32] Ansel, Jason, Shoaib Kamil, Kalyan Veeramachaneni, Jonathan Ragan-Kelley, Jeffrey Bosboom, Una May O'Reilly e Saman Amarasinghe: *OpenTuner: an extensible framework for program autotuning*. Em *Proceedings of the 23rd international conference on Parallel architectures and compilation*, páginas 303–316, Edmonton AB Canada, agosto 2014. ACM, ISBN 978-1-4503-2809-8. https://dl.acm.org/doi/10.1145/2628071.2628092, acesso em 2024-03-05. 9, 17, 43

[33] Kunjir, Mayuresh e Shivnath Babu: *Black or White? How to Develop an AutoTuner for Memory-based Analytics [Extended Version]*. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, páginas 1667–1683, 2020. https://arxiv.org/abs/2002.11780, acesso em 2024-03-05, Publisher: [object Object] Version Number: 1. 9, 18, 42, 43, 44

[34] Vasyliev, Oleksii: *PGTune - calculate configuration for PostgreSQL based on the maximum performance for a given hardware configuration*, janeiro 2024. https://pgtune.leopard.in.ua/about, acesso em 2024-03-05. 9, 17, 43

[35] Kanellis, Konstantinos, Cong Ding, Brian Kroth, Andreas Müller, Carlo Curino e Shivaram Venkataraman: *LlamaTune: Sample-Efficient DBMS Configuration Tuning*, agosto 2022. http://arxiv.org/abs/2203.05128, acesso em 2024-02-13. 9, 17, 18, 43, 45, 46

[36] Storm, Adam J e M Surendra: *Adaptive Self-Tuning Memory in DB2*. VLDB Endowment, setembro 2006. 9, 17, 43, 46

[37] Van Aken, Dana, Dongsheng Yang, Sebastien Brillard, Ari Fiorino, Bohan Zhang, Christian Bilien e Andrew Pavlo: *An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems*. Proceedings

of the VLDB Endowment, 14(7):1241–1253, março 2021, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/3450980.3450992, acesso em 2024-03-05. 9, 18, 20, 43, 45, 46

[38] Li, Guoliang, Xuanhe Zhou e Lei Cao: *Machine learning for databases*. Proceedings of the VLDB Endowment, 14(12):3190–3193, julho 2021, ISSN 2150-8097. https://dl.acm.org/doi/10.14778/3476311.3476405, acesso em 2023-11-14. 9

[39] Xiong, Xiao Mei: *A Buffer Pool Optimization Algorithm Based on Max-Heap*. Advanced Materials Research, 945-949:2439–2442, junho 2014, ISSN 1662-8985. https://www.scientific.net/AMR.945-949.2439, acesso em 2023-11-09. 10

[40] Park, Kwanghyun, Jaeyoung Do, Nikhil Teletia e Jignesh M. Patel: *Aggressive buffer pool warm-up after restart in SQL Server*. Em *2016 IEEE 32nd International Conference on Data Engineering Workshops (ICDEW)*, páginas 31–38, Helsinki, Finland, maio 2016. IEEE, ISBN 978-1-5090-2109-3. http://ieeexplore.ieee.org/document/7495612/, acesso em 2023-11-09. 10

[41] Ding, Xiaoning, Jianchen Shan e Song Jiang: *A General Approach to Scalable Buffer Pool Management*. IEEE Transactions on Parallel and Distributed Systems, 27(8):2182–2195, agosto 2016, ISSN 1045-9219. http://ieeexplore.ieee.org/document/7286847/, acesso em 2023-11-09. 10

[42] Purcell, Terry: *DB2 12 for z Optimizer*. IBM, 2017. 12

[43] Martin, Patrick, Wendy Powley, Xiaoyi Xu e Wenhu Tian: *Automated Configuration of Multiple Buffer Pools*. The Computer Journal, 49(4):487–499, julho 2006, ISSN 0010-4620. https://doi.org/10.1093/comjnl/bxl028, acesso em 2023-11-12. 13

[44] Kamatkar, Sadhana J., Ajit Kamble, Amelec Viloria, Lissette Hernández-Fernandez e Ernesto García Cali: *Database Performance Tuning and Query Optimization*. Em Tan, Ying, Yuhui Shi e Qirong Tang (editores): *Data Mining and Big Data*, Lecture Notes in Computer Science, páginas 3–11, Cham, 2018. Springer International Publishing, ISBN 978-3-319-93803-5. 13

[45] Kolade, Owoeye, Ajayi Adedoyin Olayinka e Ukorigho Ovie: *Fingerprint Database Optimization Using Watershed Transformation Algorithm*. Open Journal of Optimization, 3(4):59–67, novembro 2014. https://www.scirp.org/journal/paperinformation.aspx?paperid=52375, acesso em 2023-11-14, Number: 4 Publisher: Scientific Research Publishing. 13

[46] Elmongui, Hicham G.: *Challenges in spatiotemporal stream query optimization*. Em *Data Engineering for Wireless and Mobile Access*, páginas 27–34. ACM, junho 2006. https://typeset.io/papers/challenges-in-spatiotemporal-stream-query-optimization-3x7cgwf7x3, acesso em 2023-11-14. 13

[47] Vyawahare, Dr.H.R.: *Machine Learning: A Solution Approach for Complex Problems*. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT, 06(04), junho 2022, ISSN 25823930. https://ijsrem.com/download/machine-learning-a-solution-approach-for-complex-problems/, acesso em 2023-11-09. 15, 16

[48] Lee, Wei-Meng: *Python® Machine Learning*. Wiley, 1ª edição, abril 2019, ISBN 978-1-119-54563-7 978-1-119-55750-0. https://onlinelibrary.wiley.com/doi/book/10.1002/9781119557500, acesso em 2023-11-09. 15, 16, 24, 37

[49] Utmal, Dr. Meghna: *Machine Learning Its Applications, Challenges & Tools: A Review*. International Journal of Computer Science and Mobile Computing, 10(3):32–38, março 2021, ISSN 2320088X. https://www.ijcsmc.com/docs/papers/March2021/V10I3202105.pdf, acesso em 2023-11-09. 16

[50] Mahadevkar, Supriya V., Bharti Khemani, Shruti Patil, Ketan Kotecha, Deepali R. Vora, Ajith Abraham e Lubna Abdelkareim Gabralla: *A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions*. IEEE Access, 10:107293–107329, 2022, ISSN 2169-3536. https://ieeexplore.ieee.org/document/9903420/, acesso em 2023-11-09. 16

[51] Zhao, Xinyang, Xuanhe Zhou e Guoliang Li: *Automatic Database Knob Tuning: A Survey*. IEEE Transactions on Knowledge and Data Engineering, 35(12):12470–12490, dezembro 2023, ISSN 1041-4347, 1558-2191, 2326-3865. https://ieeexplore.ieee.org/document/10106050/, acesso em 2024-02-05. 16, 17, 18, 74

[52] Li, Jing Ping, Nawazish Mirza, Birjees Rahat e Deping Xiong: *Machine learning and credit ratings prediction in the age of fourth industrial revolution*. Technological Forecasting and Social Change, 161:120309, dezembro 2020, ISSN 00401625. https://linkinghub.elsevier.com/retrieve/pii/S0040162520311355, acesso em 2023-11-09. 16

[53] Joshi, Krishna Kumar, Neelam Joshi e Ravi Ray Chaudhari: *Machine Learning - Learning Techniques, CNN, Languages and APIs*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, páginas 23–30, maio 2020, ISSN 2456-3307. http://ijsrcseit.com/paper/CSEIT2062164.pdf, acesso em 2023-11-09. 16

[54] Sullivan, Rob: *Machine-Learning Techniques*. Em *Introduction to Data Mining for the Life Sciences*, páginas 363–454. Humana Press, Totowa, NJ, 2012, ISBN 978-1-58829-942-0 978-1-59745-290-8. https://link.springer.com/10.1007/978-1-59745-290-8_8, acesso em 2023-11-09. 16

[55] Shalev-Shwartz, Shai e Shai Ben-David: *Understanding Machine Learning: From Theory To Algorithms*, volume 1. Cambridge University Press, janeiro 2015. https://typeset.io/papers/understanding-machine-learning-from-theory-to-algorithms-5aeszqx33g, acesso em 2023-11-09. 16

[56] Trendafilov, Nickolay e Kei Hirose: *Exploratory factor analysis*. Em *International Encyclopedia of Education(Fourth Edition)*, páginas 600–606. Elsevier, 2023, ISBN 978-0-12-818629-9. https://linkinghub.elsevier.com/retrieve/pii/B9780128186305100156, acesso em 2024-04-07. 19, 20, 22

[57] Hair, Joseph F., William C. Black, Barry J. Babin e Rolph E. Anderson: *Multivariate data analysis*. Cengage Learning, Boston, MA, 8ª edição, 2018. 19, 20, 37

[58] Yong, An Gie e Sean Pearce: *A Beginner's Guide to Factor Analysis: Focusing on Exploratory Factor Analysis*. Tutorials in Quantitative Methods for Psychology, 9(2):79–94, outubro 2013, ISSN 1913-4126. http://www.tqmp.org/RegularArticles/vol09-2/p079, acesso em 2024-04-07. 22, 23, 37

[59] Kaiser, Henry F.: *Little jiffy, mark IV*. Educational and psychological measurement, 34(1):111–117, 1974. 23, 37

[60] Cattell, Raymond B.: *The scree test for the number of factors*. Multivariate behavioral research, 1(2):245–276, 1966. 23, 37

[61] Duarte, Denio e Niclas Ståhl: *Machine Learning: A Concise Overview*. Em Said, Alan e Vicenç Torra (editores): *Data Science in Practice*, volume 46, páginas 27–58. Springer International Publishing, Cham, 2019, ISBN 978-3-319-97555-9 978-3-319-97556-6. http://link.springer.com/10.1007/978-3-319-97556-6_3, acesso em 2023-11-09. 24, 28

[62] Sarr, Abdoulaye: *The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications*, 2021. https://www.kdnuggets.com/2021/01/ultimate-guide-k-means-clustering.html, acesso em 2025-06-01. 26, 27, 38

[63] Rousseeuw, Peter J.: *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of computational and applied mathematics, 20:53–65, 1987. 26

[64] Casella, George e Roger L. Berger: *Statistical inference*. Duxbury, Pacific Grove, Calif, 2. ed edição, 2002, ISBN 978-0-534-24312-8. 29

[65] Tibshirani, Robert: *Regression Shrinkage and Selection Via the Lasso*. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1):267–288, janeiro 1996, ISSN 1369-7412, 1467-9868. https://academic.oup.com/jrsssb/article/58/1/267/7027929, acesso em 2024-05-13, 106 citations (INSPIRE 2025/6/23) 106 citations w/o self (INSPIRE 2025/6/23). 29, 38

[66] Esteves, Jackeline: *Regularização, Lasso (L1) e Ridge (L2)*, 2022. https://medium.com/@jackelinegleme/, acesso em 2025-06-01. 30

[67] Rasmussen, Carl Edward e Christopher K. I. Williams: *Gaussian Processes for Machine Learning*. Massachusetts Institute of Technology, novembro 2005. https://typeset.io/papers/gaussian-processes-for-machine-learning-40fp9c190d, acesso em 2023-11-09. 31, 38

[68] Agnihotri, Apoorv e Nipun Batra: *Exploring Bayesian Optimization*. Distill, 2020. https://distill.pub/2020/bayesian-optimization. 31, 32, 36

[69] Ekamperi, Stathis: *Acquisition functions in Bayesian Optimization*, 2021. https://ekamperi.github.io/, acesso em 2025-06-01. 32

[70] BoTorch: *Acquisition Functions*, 2025. https://botorch.org/docs/acquisition/, acesso em 2025-06-01. 34, 36, 38

[71] Garnett, Roman: *Bayesian Optimization Acquisition Functions*. Relatório Técnico, Washington University in St. Louis, 2015. https://www.cse.wustl.edu/~garnett/cse515t/spring_2015/files/lecture_notes/12.pdf. 38

[72] Bayesian-Optimization: *BayesianOptimization: A Python implementation of global optimization with gaussian processes*, 2023. https://github.com/bayesian-optimization/BayesianOptimization, acesso em 2025-06-01. 38

[73] modAL: *Acquisition functions — modAL documentation*, 2023. https://modal-python.readthedocs.io/en/latest/content/query_strategies/Acquisition-functions.html, acesso em 2025-06-01. 38

[74] Martinez-Plumed, Fernando, Lidia Contreras-Ochando, Cesar Ferri, Jose Hernandez-Orallo, Meelis Kull, Nicolas Lachiche, Maria Jose Ramirez-Quintana e Peter Flach: *CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories*. IEEE Transactions on Knowledge and Data Engineering, 33(8):3048–3061, agosto 2021, ISSN 1041-4347, 1558-2191, 2326-3865. https://ieeexplore.ieee.org/document/8943998/, acesso em 2025-10-06. 38, 39, 40

[75] Chapman, Pete: *CRISP-DM 1.0 Step-by-step data mining guide*. SPSS Inc., 2000. 38, 39

[76] Mariscal, Gonzalo, Óscar Marbán e Covadonga Fernández: *A survey of data mining and knowledge discovery process models and methodologies*. The Knowledge Engineering Review, 25(2):137–166, junho 2010, ISSN 0269-8889, 1469-8005. https://www.cambridge.org/core/product/identifier/S0269888910000032/type/journal_article, acesso em 2025-10-06. 39

[77] Gunasekaran, Karthick Prasad, Kajal Tiwari e Rachana Acharya: *Utilizing deep learning for automated tuning of database management systems*, junho 2023. http://arxiv.org/abs/2306.14349, acesso em 2023-11-12. 43, 45, 46

[78] Giannankouris, Konstantinos e Immanuel Trummer: *Lambda-Tune: LLM-Powered Database Parameter Tuning*. arXiv preprint arXiv:2411.03500, 2024. https://arxiv.org/abs/2411.03500. 43, 45, 46

[79] Chen, Wei, Xiaoming Liu e Yifan Zhang: *Centrum: Distribution-Free Database Auto-Tuning*. Em *Proceedings of the 2025 ACM SIGMOD International Conference on Management of Data*, páginas 1–14. ACM, 2025. 43, 45

[80] Li, Cuixia, Junhai Wang, Jiahao Shi, Liqiang Liu e Shuyan Zhang: *ADWTune: an adaptive dynamic workload tuning system with deep reinforcement learning*. Complex & Intelligent Systems, 11(4):192, abril 2025, ISSN 2199-4536, 2198-6053. https://link.springer.com/10.1007/s40747-025-01801-3, acesso em 2025-09-07. 43, 45, 46

[81] Mendizabal, Eduardo Pingarilho, Geraldo Rocha e Aleteia Araujo: *Otimização de parâmetros de buffer pool com aprendizado de máquina em ambientes não transacionais*. Em *Anais do XXVI Simpósio em Sistemas Computacionais de Alto Desempenho*, páginas 1–12, Porto Alegre, RS, Brasil, 2025. SBC. https://sol.sbc.org.br/index.php/sscad/article/view/37888. 76

Apêndice A

Fichamento de Artigo Científico

Anexo I

Documentação Original UnB-CIC (parcial)