Análise de Padrões em Casos Jurídicos com Mineração de Processos e Machine Learning

Pattern Analysis in Legal Cases using Process Mining and Machine Learning

Marcus V. de França[1]
orcid.org/0009-0000-5856-5547

Byron L. D. Bezerra[2]
orcid.org/0000-0002-8327-9734

Luiz F. V. Verçosa[3]
orcid.org/0000-0003-2095-9000

Vinícius Ferreira Silva[4]
orcid.org/0000-0001-9889-2331

Jaqueline K. L. da Cruz[5]
orcid.org/0009-0005-2046-0498

[1,2,3,4,5]Escola Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.

E-mail do autor principal:
Marcus V. de França mvf2@poli.br

DOI: 10.25286/repa.v11i4.3948

RESUMO

A morosidade na tramitação de processos judiciais representa um desafio crítico para a eficiência do sistema de justiça. Este artigo apresenta uma metodologia que integra Mineração de Processos e Machine Learning para analisar o tempo de processamento de casos na Justiça Comum brasileira. A abordagem inicia com a aplicação do algoritmo *K-means*, utilizando a representação *bag of activities*, para agrupar processos com fluxos de trabalho similares. Posteriormente, esses aglomerados são subdivididos com base na duração dos traços, classificando-os em categorias de execução rápida, média e lenta. A etapa final emprega o algoritmo *PrefixSpan* para identificar padrões sequenciais de atividades recorrentes dentro de cada perfil temporal. Os resultados demonstram comportamentos processuais característicos e sequências de atividades predominantes para cada faixa de duração, fornecendo subsídios essenciais para a otimização da gestão judiciária e o aprimoramento dos fluxos de trabalho nos tribunais.

PALAVRAS-CHAVE: *Processos Judiciais; Mineração de Processos; Clusterização.*

ABSTRACT

Delays in judicial case processing represent a critical challenge to justice system efficiency. This article presents a methodology integrating Process Mining and Machine Learning to analyze case processing times in the Brazilian Common Justice system. The approach begins with the application of the K-means algorithm, using a bag-of-activities representation, to group cases with similar workflows. Subsequently, these clusters are subdivided based on trace duration, classifying them into fast, medium, and slow execution categories. The final step employs the PrefixSpan algorithm to identify recurrent sequential activity patterns within each temporal profile. The results demonstrate characteristic procedural behaviors and predominant activity sequences for each duration range, providing essential insights for optimizing judicial management and improving workflows within the courts.

KEY-WORDS: *Judicial Cases; Process Mining; Clustering.*

1 INTRODUÇÃO

O Poder Judiciário brasileiro enfrenta desafios consideráveis relacionados à morosidade e à variabilidade dos processos judiciais. Fatores como a lentidão na tramitação de processos físicos e eletrônicos, bem como a complexidade das normas legais, contribuem para um sistema sobrecarregado e com baixa eficiência. Ao final de 2022, o Brasil registrava mais de 81 milhões de processos em tramitação, com um tempo médio de 5 anos e 8 meses para a conclusão de um processo na Justiça Comum Estadual [1], evidenciando assim, não somente um alto volume de processos e um longo tempo de espera, mas também questões sobre a complexidade referente à sequência processual das atividades do sistema jurídico. A diversidade de trajetórias possíveis para o mesmo tipo de processo cria uma dificuldade para a identificação de padrões eficientes e ineficientes, o que, por sua vez, impede a otimização da gestão judicial e a formulação de políticas públicas eficazes.

Nesse cenário, a mineração de processos (*Process Mining*) surge como uma abordagem inovadora que visa apoiar a melhoria da eficiência processual por meio da análise de grandes volumes de dados gerados nos sistemas processuais utilizados pelos Tribunais de Justiça. Essa técnica combina fundamentos de ciência de dados e gerenciamento de processos de negócio (*BPM*) para extrair modelos representativos do comportamento real das atividades executadas, permitindo a reconstrução e análise de fluxos processuais com base em registros de eventos [2]. Lakshmanan et al [3] enfatizam que a análise de fluxos de cuidado, embora originalmente aplicada ao contexto clínico, é diretamente aplicável à otimização de fluxos de trabalho em sistemas complexos como o judiciário, ao descobrir como as atividades impactam a jornada do processo.

A mineração de processos permite compreender o comportamento real dos casos, revelando fluxos típicos, desvios e padrões de execução. Entre as diversas técnicas utilizadas na área, a clusterização de rastros processuais destaca-se como uma estratégia eficaz para lidar com a alta variabilidade dos dados, permitindo agrupar casos com características semelhantes antes de realizar análises mais detalhadas. Conforme apontado por Suriadi et al [4], a clusterização em logs de casos é essencial para explorar formas alternativas de agrupar coortes baseadas nas características do processo, facilitando o discernimento de padrões de comportamento e duração.

Neste trabalho, essas abordagens foram aplicadas diretamente aos dados da Justiça Comum brasileira, disponibilizados pelo CNJ, com foco específico em processos oriundos dos Tribunais de Justiça Estaduais. A metodologia proposta inicia com a clusterização baseada em *bag of activities* para identificar grupos de casos com perfis semelhantes. Após a formação dos clusters, realiza-se uma subdivisão interna focada na duração dos traços, classificando os processos em categorias de execução rápida, média ou lenta. A etapa final consiste na aplicação do algoritmo *PrefixSpan* para discernir padrões sequenciais de atividades recorrentes dentro de cada uma dessas categorizações temporais. Essa análise detalhada visa revelar comportamentos processuais característicos, oferecendo insights valiosos para aprimorar os fluxos de trabalho no sistema judiciário.

2 TRABALHOS RELACIONADOS

Diversos pesquisadores têm investigado como tecnologias avançadas, especialmente a inteligência artificial, a mineração de processos e a mineração de padrões, podem auxiliar para a melhoria de sistemas robustos. Para fundamentar este estudo, foram analisadas algumas publicações relevantes na área, como:

Wil van der Aalst [2], que na sua obra essencial para a área da mineração de processos, estabelece as bases teóricas da disciplina, juntando fundamentos de ciência de dados e gerenciamento de processos de negócio (BPMN) para a análise de fluxos reais a partir de logs de eventos.

Lakshmanan [3] propõem uma abordagem que combina clusterização, mineração de processos e mineração de padrões frequentes em ambientes hospitalares para analisar "care pathways" (caminhos de cuidado) clínicos correlacionados com os resultados dos pacientes. O estudo destaca a importância de entender como as atividades clínicas impactam os pacientes e como essa análise pode ser usada para otimizar os fluxos de trabalho na área da saúde. A metodologia aborda desafios como a grande quantidade de eventos que ocorrem no mesmo dia e a diversidade de nomes de eventos, propondo soluções para o pré-processamento dos dados.

Suriadi et al [4] aplicaram técnicas de mineração de processos em um contexto hospitalar para mensurar e analisar variações nos fluxos de atendimento a pacientes. A

abordagem incluiu o uso de algoritmos de clusterização, permitindo identificar padrões distintos de comportamento entre diferentes organizações de saúde. Tal metodologia oferece uma base conceitual relevante para investigar a variabilidade em fluxos processuais no domínio jurídico.

Para a identificação de padrões sequenciais, o algoritmo PrefixSpan, proposto por Pei et al. [10], destaca-se por sua eficiência ao evitar a geração explícita de candidatos, comum em métodos tradicionais de mineração sequencial. A técnica baseia-se na projeção de prefixos para restringir o espaço de busca e acelerar a descoberta de subsequências frequentes. Em seus experimentos, os autores demonstraram que o PrefixSpan é capaz de lidar com grandes bases de dados contendo sequências extensas, superando algoritmos anteriores tanto em desempenho quanto em escalabilidade.

Especificamente na análise de processos judiciais, Verçosa [5] propôs uma abordagem baseada na utilização de atributos hierárquicos e uma estratégia de clusterização em duas etapas. Essa metodologia foi desenvolvida para lidar com a ausência de um padrão sequencial fixo nos movimentos processuais, permitindo a identificação de agrupamentos homogêneos com características similares. O estudo contribui significativamente para a compreensão da variabilidade estrutural dos fluxos judiciais.

Embora a literatura já conte com um número considerável de pesquisas focadas na melhoria e análise do sistema judiciário, esse campo ainda está em expansão. A complexidade dos dados jurídicos oferece inúmeras oportunidades para extração de informações valiosas, capazes de transformar significativamente esse ecossistema. Especificamente, a aplicação de técnicas de mineração de processos ainda é limitada, mas apresenta grande potencial de impacto.

3 METODOLOGIA

3.1 Base de Dados

Para a elaboração deste artigo, foi utilizada a base de dados pública disponibilizada pelo Conselho Nacional de Justiça (CNJ), extraída do Banco de Dados do Poder Judiciário Nacional (DataJud) [6].

O conjunto de dados mencionado contém informações relativas tanto à Justiça Comum quanto à Justiça Especial. Entretanto, neste trabalho, analisaremos exclusivamente a Justiça Comum. Mais precisamente, a análise se aterá aos Tribunais de Justiça, os quais se organizam em jurisdições correspondentes às unidades federativas do país.

No contexto brasileiro, cada Tribunal de Justiça está vinculado a um estado. Como o Brasil possui 26 estados e o Distrito Federal, existem, atualmente, 27 Tribunais de Justiça em vigor, conforme detalhado no Quadro 1.

Quadro 1: Relação dos Tribunais de Justiça brasileiros por unidade federativa.

UF	Tribunal de Justiça
AC	Tribunal de Justiça do Estado do Acre (TJAC)
AL	Tribunal de Justiça do Estado de Alagoas (TJAL)
AM	Tribunal de Justiça do Estado do Amazonas (TJAM)
AP	Tribunal de Justiça do Estado do Amapá (TJAP)
BA	Tribunal de Justiça do Estado da Bahia (TJBA)
CE	Tribunal de Justiça do Estado do Ceará (TJCE)
DF	Tribunal de Justiça do Distrito Federal e dos Territórios (TJDFT)
ES	Tribunal de Justiça do Estado do Espírito Santo (TJES)
GO	Tribunal de Justiça do Estado de Goiás (TJGO)
MA	Tribunal de Justiça do Estado do Maranhão (TJMA)
MG	Tribunal de Justiça do Estado de Minas Gerais (TJMG)
MS	Tribunal de Justiça do Estado de Mato Grosso do Sul (TJMS)

DOI: 10.25286/repa.v11i4.3948

MT	Tribunal de Justiça do Estado de Mato Grosso (TJMT)
PA	Tribunal de Justiça do Estado do Pará (TJPA)
PB	Tribunal de Justiça do Estado da Paraíba (TJPB)
PE	Tribunal de Justiça do Estado de Pernambuco (TJPE)
PI	Tribunal de Justiça do Estado do Piauí (TJPI)
PR	Tribunal de Justiça do Estado do Paraná (TJPR)
RJ	Tribunal de Justiça do Estado do Rio de Janeiro (TJRJ)
RN	Tribunal de Justiça do Estado do Rio Grande do Norte (TJRN)
RO	Tribunal de Justiça do Estado de Rondônia (TJRO)
RR	Tribunal de Justiça do Estado de Roraima (TJRR)
RS	Tribunal de Justiça do Estado do Rio Grande do Sul (TJRS)
SC	Tribunal de Justiça do Estado de Santa Catarina (TJSC)
SE	Tribunal de Justiça do Estado de Sergipe (TJSE)
SP	Tribunal de Justiça do Estado de São Paulo (TJSP)
TO	Tribunal de Justiça do Estado do Tocantins (TJTO)

3.2 Criação do Log de Eventos

O log de eventos é a principal fonte de dados na mineração de processos, uma vez que contém registros ordenados de atividades associadas a casos específicos. Cada evento inclui, no mínimo, a atividade executada, o identificador do caso e um carimbo de tempo, podendo ainda incluir atributos adicionais como o recurso responsável e dados contextuais [2].

De maneira análoga ao supracitado, para obtenção do log de eventos desejado, foi necessário o uso de algumas etapas de pré-processamento da nossa base de dados, tais como:

Quadro 2: Etapas de pré-processamento para construção do log de eventos

Etapa	Descrição
Limpeza de dados	Remoção de linhas com informações nulas ou inconsistentes
Validação com glossário	Eliminação de registros em que classe, assunto ou movimento não estavam mapeados no glossário oficial do CNJ
Filtro temporal	Exclusão de processos iniciados antes de 1970
Ordenação cronológica	Ordenação das atividades com base na data de ocorrência
Filtragem por encerramento	Remoção de processos sem a atividade final de arquivamento definitivo (código 246)
Filtragem por volume de eventos	Eliminação de processos com apenas uma atividade registrada
Generalização de atividades	Agrupamento de atividades com base em suas hierarquias

Além dessas etapas, com base nos critérios apresentados, realizou-se a filtragem dos processos pertencentes exclusivamente à classe de execução fiscal (código 1116). Esse recorte se justifica pelo fato de que processos nessa fase impactam de forma significativa a morosidade do Poder Judiciário, permitindo, assim, uma análise mais direcionada ao cerne do problema.

Uma característica notável da base de dados é a estrutura hierárquica em forma de árvore observada em atributos como movimentos, classes e assuntos. Essas hierarquias podem ser exploradas com mais profundidade por meio da plataforma pública de consulta do CNJ. No caso específico dos movimentos processuais, há duas principais ramificações: uma referente às ações tomadas pelos magistrados (juízes) e outra às ações

executadas pelos servidores (serventuários), ambas contendo subdivisões próprias.

Após a construção do log de eventos, foram registrados 40.471 rastros de processos. Desses, 31.072 correspondem a sequências únicas de atividades — ou seja, aproximadamente 76% da base apresenta rastros não repetidos, o que evidencia uma alta heterogeneidade nos fluxos de execução processual.

3.3 Clusterização

A técnica de clusterização exerce um papel fundamental na mineração de processos, pois organiza logs de eventos complexos em grupos significativos, auxiliando na identificação de padrões comuns, outliers e ineficiências [7]. Como mencionado na seção anterior, 76% dos processos presentes na base de dados apresentam uma sequência única de movimentos. Técnicas de clusterização podem auxiliar na identificação de padrões entre esses processos, facilitando a análise comportamental em relação ao tempo de execução. Para os experimentos realizados, foi utilizado o algoritmo K-means, baseado na representação bag of activities. A escolha do K-means se deu por sua simplicidade e ampla utilização como técnica básica de clusterização no campo da ciência de dados.

3.3.1 K-means com Representação Bag of Activities

A representação bag of activities transforma cada rastro de processo em um vetor que contabiliza a frequência de cada atividade, ignorando a ordem de execução. Por exemplo, a sequência [A, A, B, C, C, D] é representada como:

v = [fA, fB, fC, fD] = [2, 1, 2, 1]

Essa estrutura vetorial permite a aplicação de algoritmos de aprendizado de máquina, como o K-means, que operam sobre espaços numéricos. A implementação foi realizada com a biblioteca scikit-learn [8], utilizando o método k-means++ para inicializar os centróides. Diversas configurações foram testadas com base na métrica R^2 aplicada à predição do tempo de execução dos processos. Os melhores resultados foram obtidos com 70 clusters e o algoritmo de Elkan. A análise dos clusters revelou padrões comportamentais nos fluxos processuais, como grupos com alta repetição de atividades ou trajetórias mais enxutas.

Uma limitação conhecida do K-means é sua sensibilidade a outliers, uma vez que todo registro é forçado a pertencer a um cluster. Para mitigar esse efeito, foi aplicada a técnica de Winsorização. Em vez de remover os registros, os valores de tempo de execução situados acima do percentil 95 e abaixo do percentil 5 foram ajustados para os seus respectivos limites (teto e piso). Essa abordagem preserva o tamanho da amostra, reduzindo a influência de extremos estatísticos sem a perda de dados.

Adicionalmente, implementou-se um mecanismo de pós-processamento para tratar agrupamentos pouco representativos. Foi calculado um limiar de frequência (threshold) definido pela fórmula:

infreq = (1 / n_clusters) / 2

Com base nesse cálculo, o algoritmo verifica quais grupos não atingem a quantidade mínima de processos estipulada pelo threshold e realiza a aglutinação destes em um único grupo genérico. Como resultado prático dessa filtragem, a configuração inicial de 70 clusters foi reduzida para 43 grupos efetivos. Isso garante que a análise subsequente foque nos padrões de comportamento mais robustos e frequentes, eliminando a redundância de microagrupamentos pouco significativos.

3.4 Mineração de Padrões Sequenciais

A mineração de padrões sequenciais (ou *Sequential Pattern Mining* — SPM) busca descobrir subsequências que ocorrem com frequência em um conjunto de sequências ordenadas, onde cada sequência representa uma lista de eventos ou itens com uma ordem temporal definida [9]. Essa técnica é amplamente utilizada em diversos domínios, como marketing, bioinformática e, neste caso, análise de processos judiciais. O objetivo é identificar padrões que se repetem

DOI: 10.25286/repa.v11i4.3948

em diferentes casos, o que pode auxiliar na compreensão dos fluxos processuais mais comuns ou críticos.

3.4.1 Definições Formais

Considere um conjunto de itens $I = \{i_1, i_2, ..., i_n\}$. Uma *sequência* é uma lista ordenada de elementos (ou itens), onde cada elemento é um subconjunto de I. Por exemplo, a sequência $s = \langle \{a\}, \{b, c\}, \{d\} \rangle$ indica que o item *a* ocorreu primeiro, seguido pelos itens *b* e *c* simultaneamente, e depois pelo item *d*.

Dado um conjunto de sequências *D*, um padrão sequencial α é considerado *frequente* se ocorre em pelo menos σ sequências de *D*, onde σ representa o limiar mínimo de suporte (*minsup*). O suporte de um padrão α, denotado por sup(α), corresponde à proporção de sequências em *D* que contêm α.

$$sup(\alpha) = |\{ s \in D \mid \alpha \sqsubseteq s \}| / |D|$$

Exemplo: Suponha $D = \{ \langle A, B, C \rangle, \langle A, C \rangle, \langle A, B \rangle \}$. A sequência $\langle A, B \rangle$ aparece em duas das três sequências, portanto $sup(\langle A, B \rangle) = 2/3$.

3.4.2 O Algoritmo PrefixSpan

Dentre os diversos algoritmos existentes para mineração de padrões sequenciais (SPM), o PrefixSpan é amplamente reconhecido por sua eficiência. De acordo com Pei et al. [10]. Ele explora o conceito de projeções prefixadas para evitar a geração de candidatos, reduzindo significativamente o espaço de busca.

A abordagem baseia-se na ideia de projetar o conjunto de dados original em subconjuntos com base em prefixos comuns e, em seguida, minerar padrões nesses subconjuntos de maneira recursiva.

- **Passo 1:** Identificar todos os itens frequentes de uma única etapa.
- **Passo 2:** Para cada item frequente, construir uma base projetada com prefixo correspondente.
- **Passo 3:** Repetir o processo recursivamente nas bases projetadas.

O Quadro 3 apresenta um exemplo simplificado da aplicação do algoritmo em um conjunto reduzido de sequências.

Quadro 3: Exemplo ilustrativo da aplicação do PrefixSpan

Sequência Original	Projeção com prefixo A
⟨ A, B, C ⟩	⟨ B, C ⟩
⟨ A, C ⟩	⟨ C ⟩
⟨ B, A, C ⟩	—

O PrefixSpan demonstrou-se eficaz especialmente em bases com grandes quantidades de dados e baixa repetição de padrões, como é o caso da presente análise.

3.5 Integração com os Grupos Temporais

Após a clusterização inicial utilizando o algoritmo K-means, foi realizada uma etapa complementar de agrupamento com base no tempo total de execução dos processos. Essa segmentação temporal, dividida em três faixas — processos rápidos (25% mais curtos), médios (50% centrais) e lentos (25% mais longos) — teve como objetivo permitir uma análise mais refinada dos comportamentos processuais segundo sua duração.

Essa divisão temporal foi diretamente integrada à aplicação do algoritmo PrefixSpan. Em vez de executar a mineração sequencial sobre toda a base de dados de forma agregada, optou-se por aplicar o PrefixSpan de forma independente em cada um dos três grupos temporais. Essa abordagem possibilita identificar padrões recorrentes de atividades característicos de cada faixa de tempo, revelando, por exemplo, se processos mais longos seguem trajetórias processuais significativamente diferentes daqueles concluídos em menor tempo.

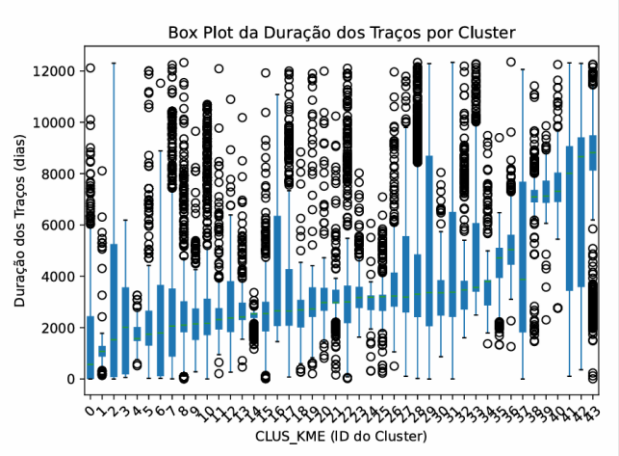
A aplicação segmentada do PrefixSpan contribui para a compreensão de como a duração de um processo está relacionada ao seu fluxo de atividades, permitindo identificar

estruturas sequenciais que potencialmente impactam a morosidade ou a agilidade processual. Os padrões extraídos para cada grupo são posteriormente comparados e discutidos na seção de Resultados.

Além dessas etapas, com base nos critérios apresentados, realizou-se a filtragem dos processos pertencentes exclusivamente à classe de execução fiscal (código 1116). Esse recorte é estratégico e se justifica pelo fato de que processos de execução fiscal representam historicamente um dos maiores gargalos do Poder Judiciário brasileiro, respondendo por uma taxa de congestionamento elevada e impactando significativamente a morosidade geral apontam os relatórios anuais do CNJ [1]. Dessa forma, o foco nesta classe permite uma análise mais precisa dos gargalos de execução, viabilizando a proposição de soluções direcionadas para este segmento crítico.

4 RESULTADOS E DISCUSSÕES

Figura 1 – Boxplot da distribuição da duração dos traços por cluster identificado pelo algoritmo K-Means.



Fonte: O Autor.

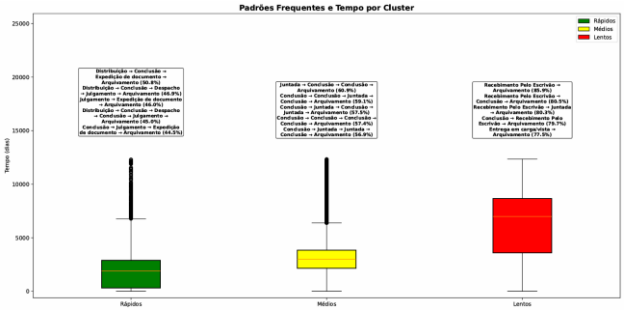
A Figura 1 apresenta a distribuição da variável *duration_days* para os diferentes clusters identificados pelo algoritmo K-Means, representada em formato de boxplot. Observa-se uma separação clara entre os grupos, com distintas medianas e amplitudes. Os clusters classificados como *lentos* exibem medianas mais elevadas e maior variabilidade, enquanto os *rápidos* apresentam durações mais concentradas e valores medianos significativamente menores. Os resultados do teste de Mann-Whitney indicam que há diferença estatisticamente significativa entre os grupos, reforçando que a segmentação por tempo foi eficaz.

O Quadro 4 resume a quantidade de traços (casos) e a respectiva quantidade de eventos (atividades) presentes em cada grupo de cluster:

Cluster	Traços	Eventos
Rápidos	11.488	152.734
Médios	21.004	386.921
Lentos	7.979	256.890

Essa distribuição reflete a diversidade na complexidade dos processos. A maior concentração de traços nos clusters médios sugere uma predominância de casos com tempo intermediário de tramitação, enquanto os extremos representam situações mais específicas.

Figura 2 – Padrões sequenciais frequentes e distribuição temporal para os grupos Rápidos, Médios e Lentos.



Fonte: O Autor.

Na Figura 2, são exibidos os cinco padrões mais representativos de cada grupo (*rápidos*, *médios* e *lentos*), juntamente com os respectivos tempos de duração dos traços. A visualização permite correlacionar sequências de atividades mais frequentes com a categoria de tempo correspondente. Padrões exclusivos também foram identificados por

DOI: 10.25286/repa.v11i4.3948

meio da análise de suporte relativo entre os grupos. Clusters rápidos apresentaram padrões com fluxos mais diretos e objetivos, enquanto, nos clusters lentos, é possível observar sequências mais longas e complexas, indicando maior burocracia ou variação de procedimentos ao longo do tempo.

Essas observações são fundamentais para entender como a dinâmica processual varia entre os grupos. Além disso, podem oferecer suporte para intervenções voltadas à melhoria de desempenho e eficiência na tramitação dos processos judiciais.

Com base na extração de padrões frequentes em cada cluster, foram identificadas sequências recorrentes de atividades que caracterizam os diferentes comportamentos processuais. A seguir, apresentam-se os cinco padrões mais frequentes de cada grupo:

Rápidos (tempo médio: 2143,43 dias):

- *Distribuição → Conclusão → Expedição de documento → Arquivamento* (50,8%)
- *Distribuição → Conclusão → Despacho → Julgamento → Arquivamento* (46,9%)
- *Julgamento → Expedição de documento → Arquivamento* (46,0%)

Médios (tempo médio: 3527,68 dias):

- *Juntada → Conclusão → Conclusão → Arquivamento* (60,9%)
- *Conclusão → Conclusão → Juntada → Conclusão → Arquivamento* (59,1%)
- *Conclusão → Juntada → Conclusão → Arquivamento* (57,5%)

Lentos (tempo médio: 6363,10 dias):

- *Recebimento Pelo Escrivão → Arquivamento* (85,9%)
- *Recebimento Pelo Escrivão → Conclusão → Arquivamento* (80,5%)
- *Recebimento Pelo Escrivão → Juntada → Arquivamento* (80,3%)

5 CONCLUSÃO

Este estudo demonstrou a aplicabilidade e a eficácia da combinação de técnicas de Mineração de Processos e Machine Learning para a análise da tramitação de processos judiciais no Brasil, com foco estratégico na classe de execução fiscal. A metodologia proposta, que integrou a clusterização baseada em bag of activities, a segmentação temporal e a mineração de padrões sequenciais com o algoritmo PrefixSpan, permitiu não apenas agrupar processos similares, mas também revelar as dinâmicas operacionais que influenciam diretamente a duração dos casos.

A análise dos resultados evidenciou uma distinção estrutural clara entre os perfis temporais, correlacionando a complexidade das sequências à morosidade. O grupo classificado como rápido apresentou fluxos lineares, diretos e com alta previsibilidade. A forte presença de atividades como "Expedição de documento" e "Conclusão" sugere uma tramitação simplificada, típica de ações de rotina ou com decisão monocrática clara, indicando uma gestão eficiente e possivelmente padronizada.

Em contraste, os processos de duração média revelaram uma complexidade intermediária, marcada por ciclos repetitivos de "Conclusão" e "Juntada". Esse padrão aponta para uma maior interlocução processual entre as partes e manifestações pendentes, refletindo ciclos de conferência interna que, embora necessários, indicam gargalos operacionais e falta de fluidez nos procedimentos de análise.

Por sua vez, o grupo de processos lentos expôs a natureza crítica dos entraves que afetam a celeridade judicial. A predominância de atos cartorários e administrativos — como "Recebimento pelo Escrivão" e "Entrega em carga/vista" — em detrimento de atos decisórios, sugere que a lentidão nestes casos não decorre necessariamente de uma alta complexidade jurídica, mas sim de ineficiências sistêmicas, logísticas ou de desorganização processual nas secretarias. Esses achados corroboram a hipótese de que problemas de fluxo de trabalho são fatores determinantes para o prolongamento excessivo da lide.

Portanto, esta pesquisa contribui para a compreensão aprofundada dos fluxos de trabalho no Poder Judiciário, fornecendo evidências empíricas que podem subsidiar a formulação de políticas públicas e estratégias de gestão. A identificação desses padrões

sequenciais oferece uma base sólida para o desenvolvimento de intervenções focadas — como a automação de atos cartorários nos grupos lentos ou a padronização de despachos nos grupos médios — visando otimizar recursos e, consequentemente, aprimorar a eficiência e a percepção de justiça no país.

Para trabalhos futuros, sugere-se a aplicação desta metodologia em outras classes processuais e a investigação de atributos adicionais, como o tempo de permanência em cada atividade, para refinar ainda mais o diagnóstico dos gargalos processuais e permitir uma visão mais granular da eficiência judiciária.

REFERÊNCIAS

[1] CONSELHO NACIONAL DE JUSTIÇA (CNJ). Justiça em Números 2024: ano-base 2023. Brasília: CNJ, 2024. Disponível em:https://www.cnj.jus.br/pesquisas-judiciarias/justica-em-numeros/. Acesso em: 07 dez. 2025.

[2] VAN DER AALST, W. M. P. Process Mining: Data Science in Action. 2. ed. Berlin: Springer, 2016.

[3] LAKSHMANAN, G. T.; ROZSNYAI, S.; WANG, F. Extracting clinical pathways from medical process logs. In: BUSINESS PROCESS MANAGEMENT WORKSHOPS. 2013, Beijing. Proceedings... Berlin: Springer, 2013. p. 398-409.

[4] SURIADI, S. et al. Measuring Patient Flow Variations: A Cross-Organisational Process Mining Approach. In: ASIA PACIFIC BUSINESS PROCESS MANAGEMENT CONFERENCE, 2., 2014, Brisbane. Proceedings... Cham: Springer, 2014. p. 43-58. (Lecture Notes in Business Information Processing, v. 181).

[5] VERÇOSA, L. F. V.; BASTOS-FILHO, C. J. A.; BEZERRA, B. L. D. An Approach for Analysing Law Processes based on Hierarchical Activities and Clustering. In: IEEE LATIN AMERICAN CONFERENCE ON COMPUTATIONAL INTELLIGENCE (LA-CCI), 2023, Recife. Proceedings... IEEE, 2023. p. 1-6.

[6] CONSELHO NACIONAL DE JUSTIÇA (CNJ). Base de Dados do Poder Judiciário Nacional (DataJud). Disponível em: https://www.cnj.jus.br/sistemas/datajud/. Acesso em: 07 dez. 2025.

[7] DOGAN, O.; OZTAYSIT, K. O. Process Mining and Clustering Analysis in Judicial Systems. Expert Systems with Applications, [S.l.], v. 202, 15 Sept. 2022, p. 117216.

[8] PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011.

[9] FOURNIER-VIGER, P.; LIN, J. C.-W.; KIRAN, R. U.; KOH, Y. S.; THOMAS, R. A Survey of Sequential Pattern Mining. Data Science and Pattern Recognition, v. 1, n. 1, p. 54-77, 2017.

[10] PEI, J. et al. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), 17., 2001, Heidelberg. Proceedings... Washington, DC: IEEE Computer Society, 2001. p. 215-224.

DOI: 10.25286/repa.v11i4.3948