CONFORMIDADE DE PROCESSOS DE NEGOCIO BASEADA EM CLASSIFICAÇÃO DE DOCUMENTOS E MINERAÇÃO DE LOG DE EVENTOS

Rosângela Maria Silva Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro

Setembro de 2018

CONFORMIDADE DE PROCESSOS DE NEGOCIO BASEADA EM CLASSIFICAÇÃO DE DOCUMENTOS E MINERAÇÃO DE LOG DE EVENTOS

Rosângela Maria Silva Oliveira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Jano Moreira de Souza, D.Sc.

Prof.ª Renata Mendes de Araujo, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2018

Oliveira, Rosângela Maria Silva

Conformidade em processos de negócio baseada em classificação de documentos e mineração de log de eventos/ Rosângela Maria Silva Oliveira. – Rio de Janeiro: UFRJ/COPPE, 2018.

XIV, 135 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2018.

Referências Bibliográficas: p. 101-105.

1. *Process mining*. 2. *Conformance checking*. 3. *Discovery*. 4. *Analysis*. 5. *Training data*. I. Xexéo, Geraldo Bonorino II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Agradeço a Deus pela saúde, força e, principalmente, por guiar-me nesta realização.

À Capes pela concessão da bolsa de estudo, concretizando meu sonho.

Aos meus pais, Maria e Nestor, por sempre me incentivarem a estudar e por me mostrarem o valor da educação; às minhas irmãs, Elisangela e Rosilene, por serem exemplos – suas demonstrações de orgulho sempre me incentivaram a não desistir do meu sonho –; às minhas sobrinhas Ana Clara, Eduarda, Giovanna e Vanessa, por fazerem parte da minha vida; e à minha família como um todo, pelo apoio de cada um durante a vida.

A Fernanda e Patrícia, aos grandes mestres da computação que nos presenteiam a cada dia com novas descobertas no mundo da tecnologia: a vocês, meu muito obrigada, de coração.

Aos meus amigos, pelos momentos dedicados a mim durante esta jornada, especialmente a Carolina, Paulinho, Iara, Sergio, Francisco e Inara.

Às pessoas que participaram e me incentivaram de alguma forma durante minha vida, tornando-se parte desta jornada, em especial Maria do Socorro, Holanda, Eudezia e Vera. Vocês demonstraram amizade sincera e me permitiram trilhar os primeiros passos até aqui.

Aos colegas de classe com os quais trabalhei durante este curso, especialmente para Felipe, Julio, Matheus e Max, que me ajudaram muito durante esta pesquisa.

Aos colegas do projeto PNT: Eduardo, Fabricio, Gerusa, Marcelo, Marcus e Saul, que me permitiram participar de um ambiente de desenvolvimento e conhecimento dentro da Coppetec.

Ao mestre Rodrigo, que me ajudou bastante com seu conhecimento e paciência. Muito obrigada pela ajuda e direcionamento que foram de suma importância para trilhar o caminho a ser seguido.

Aos professores com que tive contato na UFRJ: o conhecimento que vocês transmitiram foi, sem dúvidas, um pilar para o meu desenvolvimento. Muito obrigada a Ana Regina, Assis, Jano, Nathália, Zimbrão, a todos os professores que tive durante a vida e, em especial, ao meu orientador Geraldo Xexéo, por ter acreditado em mim, pelo seu incentivo, pelas ideias, paciência e reuniões de muito valor.

Finalmente, meu agradecimento vai para a pessoa que mais me incentivou durante esta jornada: Nara. Seu carinho, amizade, incentivo e crença fazem de mim a cada dia uma pessoa mais forte. Você é o maior presente que Deus colocou na minha vida.

A todas as pessoas que direta ou indiretamente estiveram comigo durante este sonho – amigos, primos e colegas de trabalho –, o meu muito obrigada.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CONFORMIDADE DE PROCESSOS DE NEGOCIO BASEADA EM CLASSIFICAÇÃO DE DOCUMENTOS E MINERAÇÃO DE LOG DE EVENTOS

Rosângela Maria Silva Oliveira

Setembro/2018

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

A progressiva demanda pela qualidade subsidia os sistemas de informação e processos de negócios. Consequentemente, há uma crescente carência para o alinhamento do processo entendido e mapeado nas organizações, além de uma escassez no entendimento da execução do processo no dia a dia. Com sistemas automatizados, a indústria de tecnologia registra eventos e gera dados com facilidade, dados esses que podem produzir valor e *insights*, visando melhoria de desempenho em diversas áreas organizacionais. Pensando nisso, neste trabalho propõe-se um estudo sobre a avaliação da conformidade nos modelos gerados a partir do *log* de eventos de um sistema de informação. Um estudo que contemple os principais algoritmos e métricas utilizados na literatura para mensurar a conformidade em mineração de processos, como algoritmos são definidos e em quais ferramentas as abordagens são testadas. A abordagem proposta avalia o *log* de eventos semiestruturados; para isso, técnicas de classificação de texto são utilizadas na preparação da estrutura requerida do *log* de eventos. O objetivo é avaliar a abordagem da conformidade aplicada à área de mineração de processos para analisar o *log* extraído, contextualizando o valor da abordagem com a definição do processo existente mapeado a partir da visão do gestor. Para apoiar o uso em outros conjuntos de dados, o modelo proposto pretende ser extensivo para modificação e uso em outros cenários.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CONFORMANCE IN BUSINESS PROCESSES BASED ON DOCUMENT CLASSIFICATION AND MINING EVENT LOG

Rosângela Maria Silva Oliveira

September/2018

Advisor: Geraldo Bonorino Xexéo

Department: Systems and Computer Engineering

The growing necessity for quality involves information systems and business processes. As a consequence, there is a growing demand to align the mapped and understood process of organizations with the performance of how the process is actually performed on a daily basis. With automated systems, the technology industry currently records events in information systems and generates data with ease, which are produced to generate value and insights to improve performance in the most diverse areas of organizations. In this work, we propose a study on *conformance checking* for generated models from the event *log* of an information system. A study that considers the main algorithms and metrics used in the literature to measure *conformance checking* in process mining, how the algorithms are defined and in which tools the approaches are tested. The proposed approach evaluates the semi-structured event *log*; for thus, text classification techniques are used to prepare the required structure of the event *log*. The main objective is to evaluate the *conformance checking* applied to the process mining area in order to analyze the extracted *log*, contextualizing the value approach with the definition of the existing process mapped from the manager understanding. To support this use in other data sets, the proposed model intends to be extensive for modification and use in other scenarios.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Fórmulas

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

À medida que a tecnologia avança, os processos de organizações tendem a utilizar e consumir padrões para manter um escopo cada vez mais controlado e homogêneo. A busca por melhor desempenho e ganho de performance e assertividade no fluxo de serviços e gestão de demandas é algo que confere à organização ganhos e competitividade.

A competitividade demanda da organização transparência, tanto no fornecimento dos seus insumos quanto no entendimento do que ela produz e como ela produz, e, nesse contexto, muitas organizações buscam se adequar a normas e padrões de qualidade, como os estabelecidos pela International Standards Organization (ISO), que monitora de forma sistemática os processos e auxilia as organizações a manterem seus padrões e continuidade dos processos.

A norma ISO 9001 (2015)[1] considera importante o desenvolvimento de normas e requisitos de conformidade para o negócio e é uma extensão da versão 9001 (2008), que contribui para o foco que afeta a conformidade dos produtos e serviços da organização.

Outra norma de padronização com foco na conformidade que surgiu recentemente, ainda vinculada à família de normas ISO, é a ISO 19600[2], que foca em implementar medida para designar transparência e conformidade no processo organizacional.

A padronização de processos de negócio na família ISO iniciou-se, de acordo com Ferreira e Gerolamo (2016), em 1987, e, desde então, vem contando com a aderência de um grande número de empresas, o que demonstra um forte valor agregado na gestão dos processos de forma sistemática nas organizações. Um dos principais ganhos é poder trabalhar com a tomada de decisões baseada em fatos, visto que a própria norma sobre os processos requer evidências em relação aos itens que ela estabelece ou recomenda.

[1] ISO :International Organization for Standardization
[2] ISO 19600 – Sistema de Gestão de Compliance

Nesse quesito de tomada de decisões baseada em fatos, iniciamos a nossa abordagem de agregar aos fatos identificados pelos dados a relação que estamos buscando neste estudo inerente à conformidade. A ISO preza sobre o contexto de transparência em relação aos seus processos, seja para determinar uma melhor adequação daquilo que faz e produz –para seus clientes internos e externos –, seja para padronização dos processos.

Com essa visão, a área de mineração de processos tende a apoiar de forma significativa as organizações na busca por padronização e pela melhoria contínua, uma vez que diversas atividades atualmente produzem insumos de evidências em dados e proporcionam o armazenamento de diversos eventos através dos sistemas informatizados.

Vivemos na era de *big data*, em que informações são processadas diariamente de forma estruturada e não estruturada, em que 90% das informações geradas no mundo são de dados não estruturados (ILYASOVA *et al.* 2018) e em que há o desafio de se ter transparência em relação ao que realmente agrega valor e pode trazer mais insumos de indicadores que proporcionem melhoria ou até mesmo de validação do processo, a fim de validar a conformidade, haja vista à proliferação das informações e principalmente à forma como ela se apresenta.

A tecnologia avança e proporciona grandes progressos em relação à análise de dados de todas as formas: texto estruturado e não estruturado e até mesmo imagem, com diversos métodos e algoritmos de classificação, clusterização, aprendizado estatístico, entre outros. Além disso, ferramentas de apoio à análise de dados, como Weka (WITTEN *et al*.2016), Knime (WITTEN *et al*.2016) e Pentaho (SOUSA *et al*.015), e linguagens de programação, como R e Python, são comumente citadas em estudos e análises com o uso dos algoritmos. Todos esses métodos e ferramentas têm por objetivo mensurar os dados de forma assertiva e buscam expor melhores resultados. Para Chen *et al.* (2014), a área de mineração de dados procura extrair informações e conhecimentos ocultos de dados imprecisos. Por outro lado, (KUMAR, VINEET; REINARTZ, 2012) definem que a área de mineração de dados consiste em identificar os clientes certos, o que inclui diversas vertentes do processo de negócio, sendo importantes a especificação do negócio, a seleção de variáveis relevantes, modelos preditivos de treinamento e a seleção de modelos mais adequados com base nos resultados.

Ainda no que diz respeito às técnicas de mineração de dados, a fim de apoiar na avaliação dos dados que um determinado processo pode apoiar ou o processo que pode ser gerado pelos dados, a área de mineração de processos surge para ajudar a agregar

valor às técnicas já existentes de mineração de dados. De acordo com van der Aalst (2016), a área de mineração de processos surgiu para preencher a lacuna entre a análise de processo baseada em modelo tradicional – como a simulação e outras técnicas de gerenciamento de processos e negócios – e técnicas que são focadas em análise de dados, como aprendizado de máquina e mineração de dados. Para Ghawi (2016), a área de mineração de processos é uma disciplina que emerge e vincula análise de dados ao gerenciamento de processos.

Em sua definição, a área de mineração de processos engloba três grupos no contexto de mineração de processos através dos dados de eventos. São eles:

1. Descoberta: Possibilita a geração de modelos de processos através do *log* de eventos.
2. Conformidade: Possibilita a avaliação da conformidade do processo de acordo com o que é entendido *versus* o que é descoberto.
3. Extensão ou melhoria: Possibilita a melhoria do processo.

Para cada grupo, existem vários algoritmos que avaliam cada um dos itens e ferramentas que possibilitam gerar métricas e modelos para entender, avaliar e melhorar o processo existente a partir do *log* de eventos.

A área de mineração de processo utiliza *input* de dados em formato específico para a geração do processo. Neste passo, técnicas e os algoritmos de mineração de dados e tratamento de texto podem ser utilizados para tratar informações e ajudar a definir padrões dentro dos dados existentes. Por outro lado, as técnicas de mineração de processos, principalmente as que mensuram a conformidade, podem auxiliar na melhor tomada de decisão para avaliar um processo.

Nossa proposta de modelo avalia a conformidade de um grande conjunto de *log* de eventos não estruturados. Investigamos um caso da vida real, em que um processo de trabalho entendido é mapeado a partir da visão do usuário gestor do processo, seguindo o modelo do *AS – IS* de como o responsável entende o processo. Em seguida, técnicas de mineração de dados são aplicadas para mapear os dados na forma adequada para o *input* do *log* de eventos para aplicar a técnica de mineração de processos.

Na área de mineração de processos, o *input* requer uma sequência de eventos para definir o processo e possibilitar a criação do modelo. Para esta adequação, utilizar as

técnicas de mineração de texto é uma abordagem necessária, uma vez que, o contexto deste estudo considera trabalhar com um conjunto de dados não estruturado. Em seguida, as técnicas de conformidade definidas na área de mineração de processos são utilizadas para obter as métricas extraídas dos algoritmos e efetuar a verificação na análise do *log*.

Com este modelo, podemos avaliar medidas de conformidade ou comparar o modelo descoberto com o modelo desenhado do processo, o qual foi mapeado como sendo o que a organização segue na prática. A verificação da conformidade será avaliada a partir do *log* de eventos considerando o modelo gerado por parte do *log versus* a avaliação dentro do contexto do *log* completo. Para isso, utilizaremos a técnica estatística de validação cruzada (REFAEILZADEH *et al*. 2008), separando cada classe de processo e em seguida avaliação do modelo é efetuada . Em seguida, cada modelo será criado com uma parte do *log* da classe e comparado com as demais partes da mesma classe; então, utilizaremos as medidas para avaliar os modelos dentro dos parâmetros de conformidade que a área de mineração de processos define. Além disso, será possível avaliar o modelo descoberto e compará-lo com o modelo entendido.

1.2 OBJETIVO

O objetivo deste trabalho é propor um modelo para avaliar a conformidade em mineração de processos utilizando dados não estruturados, para isto, efetuamos uma revisão da literatura relacionada ao tema de conformidade em mineração de processos. Nessa pesquisa nosso objetivo é identificar:

- os métodos propostos para mensurar a conformidade;
- os algoritmos aplicados na área de mineração de processos que se propõem a trabalhar com esta abordagem.

O modelo proposto considera o contexto de trabalhar com dados não estruturados, um cenário que requer a adequação do *log* de eventos para a estrutura de *input* requerida para a área de mineração de processos. Nessa etapa, propomos utilizar o método estatístico de validação cruzada (REFAEILZADEH et al. 2008) para particionar os dados em *K folders*. Em seguida utilizamos os *K-Folders* de dados aplicados aos algoritmos de avaliação da conformidade que são aplicados na área de mineração de processos.

Antes de particionar os dados utilizamos 3 algoritmos de classificação (KNN, Naive Bayes e árvore de decisão) para classificar os dados. Essa parte do processo está inserida no contexto de geração do *output* do *log* de eventos para a mineração de processos está diretamente ligado à qualidade do evento. Para van der Aalst (2016), se a qualidade dos dados dos eventos não for confiável, o valor agregado aos resultados da mineração de processos declinará no objetivo de mensurar o valor que ele pode realmente agregar. A qualidade dos dados é fortemente necessária no processo de obtenção de qualidade e melhores métricas.

Logo, utilizar técnicas para desenvolver melhores conjuntos de dados para trabalhar com o processo que avalia a qualidade em mineração de processos é uma abordagem que se faz necessária dada a variedade de fontes e formatação de dados que são produzidas na atualidade.

Para avaliar as métricas de conformidade, o modelo proposto utiliza a abordagem de gerar os modelos em *Rede* de *Petri* para efetuar a avaliação do modelo com o *log* de eventos.

Por fim, será possível avaliar o modelo criado com o *log* de eventos classificados e identificar a relação de desvios ou de pontos em comum dos k *folders* que foram avaliados.

1.3 METODOLOGIA

Para desenvolver este trabalho, inicialmente foi efetuada uma pesquisa na literatura relacionada a conformidade aplicada a área de mineração de processos com o objetivo de identificar os métodos utilizados para mensurar a conformidade dentro da área de estudo.

Nessa parte de estudo, a pesquisa feita pode ser definida como um estudo qualitativo. A pesquisa qualitativa orienta o pesquisador fornecendo diretrizes e regras relacionadas à coleta de dados qualitativos. Um dos pontos de decisão da abordagem é a definição da estratégia de coleta de dados e o objetivo final de sua aplicação (PERSSON, 2004). Para RUDIO (2001) essa etapa possibilita que o pesquisador desenvolva o seu raciocionio embasado na observação dos fenômenos por ele identificado.

O segundo passo deste trabalho apesenta um modelo para avaliação da conformidade a ser aplicado a área de mineração de processos uma vez que se utiliza dados não estruturados. O modelo proposto trabalha com a iteração dos dados particionados na abordagem de validação cruzada e avalia a classificação dos modelos e métricas utilizando os algoritmos de avaliação da conformidade aplicados a área de mineração de processos.

Em uma fase antecessora, antes do particionamento dos dados, utilizamos algoritmos de classificação de texto vastamente utilizados na literatura para a classificação dos dados não estruturados, a fim de formatar o *log* de eventos para o padrão requerido de *input* minerar um processo, os dados classificados são avaliados e finalmente aplicamos o modelo proposto que particiona os dados e faz a avaliação cruzada utilizando os algoritmos de mineração de processos, retornando as métricas e os modelos.

Essa parte da pesquisa está inserida no contexto descrito por Marconi e Lakatos (2015), que afirmam que o instrumental metodológico da pesquisa está relacionado com o problema de estudo.

O problema estudado, muitas vezes, requer mais de uma técnica ou se enquadra em mais de uma. O problema estudado neste trabalho está fortemente inserido no contexto da análise quantitativa, mas não excluído da análise qualitativa.

Na definição de Godoy (1995), análise qualitativa oferece três tipos de possibilidades de estudos, entre eles o estudo de caso. Neste trabalho, além da pesquisa qualitativa, usamos o modelo de estudo de caso para mapear um processo de negócio a partir da visão do gestor e, então, verificar e avaliar o modelo a partir do *log* de eventos. Em seguida, é feita uma análise em relação à realidade do que ocorre a partir da visão do *log* de eventos.

Por fim, o estudo de caso é aplicado no modelo de situação real, com dados reais, em que o modelo do processo descoberto a partir do *log* tratado retorna às métricas de conformidade de algoritmo da mineração de processos, possibilitando uma análise e conclusão entre o modelo real e o modelo descoberto.

3.8 ORGANIZAÇÃO

Este trabalho está dividido em sete capítulos. O capítulo 2 contextualiza os trabalhos relacionados à área de mineração de processos com as técnicas utilizadas no preparo do *log* para geração do modelo e a conformidade na área de mineração de processos. Por fim, o capítulo fala sobre a metodologia BPM.

O capítulo 3 descreve as técnicas utilizadas durante o desenvolvimento do modelo. Nesse capítulo, expõem-se as técnicas de mineração de dados e tratamento do *log* de eventos, além das técnicas de mineração de processos utilizadas para a geração dos modelos como o algoritmo IM e Rede de Petri. Para a preparação do *log* de eventos, as técnicas de mineração de dados e os classificadores KNN, Naive Bayes e árvore de decisão são apresentados.

O capítulo 4 trata do modelo proposto para avaliação da conformidade de um *log* de eventos utilizando as técnicas de mineração de processos. A proposta aborda a aplicação da validação cruzada na criação dos modelos e avaliação do *log* de eventos.

Ainda no capitulo 4 apresentamos a avaliação experimental aplicada ao modelo definido a partir de um *dataset* e um estudo de caso do mundo real com os modelos entendidos do processo de solicitação de processos acadêmicos da UFRJ/POLI.

O capítulo 5, finalmente, mostra as conclusões com os resultados derivados do *dataset*, a partir das classes definidas e da geração dos modelos. Esses resultados são discutidos e apontam para possíveis trabalhos futuros.

2 MINERAÇÃO DE PROCESSOS (MP)

A mineração de processos é uma disciplina emergente que fornece *insights* baseados em fatos para apoiar melhorias no processo. A abordagem baseia-se em modelos de processos e na mineração de dados (VAN DER AALST, 2016).

Neste capítulo, descrevemos os trabalhos relacionados à área de mineração de processos com enfoque para os trabalhos relativos à conformidade. Além disso, apresentamos uma breve descrição da literatura sobre o BPM.

2.1 MINERAÇÃO DE PROCESSOS BASEADA NO *LOG*

Como definido por van der Aalst (2011b) e van der Aalst e Verbeek (2014), o *log* de eventos é o ponto de partida para a área de mineração de processo.

Na composição do *log* de eventos, o identificador da instância é definido como caso (do inglês *case*), que pertence a uma instância do processo. As diversas ações percorridas são as atividades identificadas como traços (do inglês *traces*). Cada ação executada para o *trace* de forma manual ou automática é efetivada em uma determinada data e hora, formando a *feature timestamp*, que pode ser composta pela informação de início, de fim ou por ambas. Para exemplificar uma instância de processo, podemos citar um pedido de aprovação de crédito ou uma reserva de hotel, por exemplo. Todas as etapas que compõem a aprovação do crédito para um determinado cliente pertencem a um *case*.

O *log* de eventos é definido em van der Aalst (2013a) como um conjunto de múltiplos *traces*, em que vários casos podem conter o mesmo *trace*. No padrão da mineração de processos, o *log* de eventos requer um *input* padrão seguindo a regra de pelo menos conter [*case*, *trace*, *timestamp*].

É a partir do *log* de eventos de sistemas informatizados que as técnicas de MP são aplicadas para a descoberta de modelos, medição da conformidade, melhoria e extensão do processo. No contexto de MP, a partir dos modelos do processo, é possível segmentar uma visão do processo, seja o atual, que já se conhece, ou o descoberto a partir do *log*. Para Davenport (1993), os negócios devem ser vistos em termos dos principais processos.

Os processos são definidos por um conjunto de tarefas executadas por pessoas, equipamentos ou ambos.

Aproveitando do contexto requerido e da definição de Davenport, a área de mineração de processo se entrelaça com a área de processo. De acordo com Daniel *et al*. (2011), a área de mineração de processos é um tema emergente na área de BPM (do inglês Business Process Management). Falaremos mais sobre definição e características de BPM no item Capítulo 2. No trabalho de Daniel *et al*. (2011), os autores também falam do manifesto da área de mineração de processos, portanto, sempre que citarmos o manifesto documentado pela IEEE da área de mineração de processos, estaremos fazendo referência a esse trabalho.

A partir do *log* de eventos, diversas ferramentas da área de mineração de processos são capazes de mapear o modelo. Para Çela *et al*. (2018), a dependência dos *logs* de eventos limita as técnicas de mineração de processos, no que tange ao contexto de atividades que o *log* não é capaz de mostrar. Uma difícil visão de *input* que não conseguimos mapear e inserir nas ferramentas mais conhecidas da área é o contexto da atividade manual.

No trabalho de Esposito (2012), o autor propõe um método que busca identificar as instâncias de um processo. Uma base de dados padrão é utilizada como ponto de partida para auxiliar no mapeamento, onde os itens que possuem semântica relevante podem ser escolhidos pelo analista.

No entanto, a dificuldade do mapeamento do *log* é definida por van der Aalst e Verbeek (2014) como um desafio à medida que o *log* de eventos e os modelos de processos crescem.

Crescimento de eventos é um fator mais que presente atualmente nos sistemas informatizados, dada a gama de informações colhidas e armazenadas nos sistemas ou até mesmo geradas por eles. Davenport (2014) dá um exemplo da dificuldade de se trabalhar com os eventos do mundo atual, dado o armazenamento e geração de informações: o monitoramento de lâminas das turbinas a gás de produção de energia elétrica da empresa GE (General EleCtric) produz mais de 588 *gigabytes* por dia.

Dado o complexo e volumoso número de eventos, mesmo com o *log* mapeado, o desafio é a leitura do modelo gerado pelo *log* de eventos quando este aponta para um processo gerado em forma de *spaghetti*. O modelo *spaghetti* é definido em van der Aalst (2011a) como um modelo não estruturado, no qual as atividades de pré e pós-condições

são difíceis de definir. Em outro trabalho, Aalst aponta o problema de consumo de tempo computacional e defende a boa prática de buscar equilibrar os tamanhos dos diferentes fragmentos do modelo, obtidos a partir do particionamento das atividades (VAN DER AALST, 2013a).

A evolução da geração de dados e das técnicas para tratá-los impulsionou a área de mineração de processos. Um exemplo é o *framework* ProM, um kit de ferramentas de mineração de processos que oferece várias ferramentas de análise de processos a partir do *log* de eventos. A ferramenta é *open source*[3], então recebe contribuição de diversos pesquisadores da área que, a partir dos pacotes existentes na ferramenta, criam novos modelos ou melhoram os existentes. Mais de 1.500 *plugins* são contabilizados no ProM (VAN DER AALST, 2016) para a versão 6.5.

Algumas técnicas de mineração de processos são aplicadas para resolver o problema do modelo *spaghetti*. Alguns exemplos são:

- Aprendizado supervisionado: Método proposto por Tax *et al.* (2016), no qual os autores descrevem um método para representar em vetores os recursos de *log* de eventos provenientes da extensão XES. Uma métrica é proposta para avaliar os resultados dos eventos supervisionados que mais se enquadram nas tarefas que possibilitam tanto a descoberta do processo quanto a verificação da conformidade.
- Aprendizado não supervisionado: A abstração proposta por Mannhardt e Tax (2017) utiliza a descoberta do modelo de processos locais, definida como sendo os desvios frequentes observados, e em seguida usa esse modelo para alcançar outro nível de abstração. Nesse contexto da elevação do nível, a entrada é um *log* de eventos com um conjunto de padrões de atividades.
- Princípio de Pareto: Uma métrica aplicada para os algoritmos heurísticos. Em Weijters *et al*. (2006), os autores demostram as técnicas aplicadas a um *log* de eventos com mais de 12.000 registros e eventos diferentes.
- Clusterização: Com algoritmo implementado no ProM, uma das ferramentas mais usadas na área de mineração de processos, Veiga e Ferreira (2009) destacam a importância da técnica de clusterização para trabalhar com o entendimento de dados que geram modelos em *spaghetti*. A proposta para a clusterização com

[3] A ferramenta ProM distribuída sob a GNU Lesser General Public License (L-GPL).

particionamento dos casos é de acordo com a sequência de eventos. No trabalho de ESPOSITO *et al.* (2011) os autores utilizam a clusterização para mapear o *log* de eventos e extrair o processo, mesmo assim, o método mostrou-se ineficiente sendo necessário utilizar a técnica de divisão e conquista para preencher as lacunas e padronizar o *log* de eventos.

A padronização do *log* de eventos, entre outros itens necessários para a área de mineração de processos, foi especificada no manifesto aceito pela IEEE *Task Force on Process Mining*. Composto por um grupo de pesquisadores que apoiam e defendem a área de mineração de processos, o manifesto provoca um marco de união no desenvolvimento da pesquisa e na evolução da área (DANIEL *et al*. 2011).

De forma resumida, o manifesto IEEE conclui que o *log* de eventos é o *input* que as ferramentas de mineração de processos recebem como insumo para gerar ou extrair conhecimento.

2.2 CONFORMIDADE EM MINERAÇÃO DE PROCESSOS

Neste item, descreveremos os principais trabalhos identificados na literatura em relação ao tratamento da conformidade no contexto de mineração de processos. A abordagem pretende identificar os principais pontos tratados para avaliar a conformidade do processo obtido através das técnicas de mineração de processos. Essa revisão foi efetuada com o objetivo de atender aos seguintes tópicos:

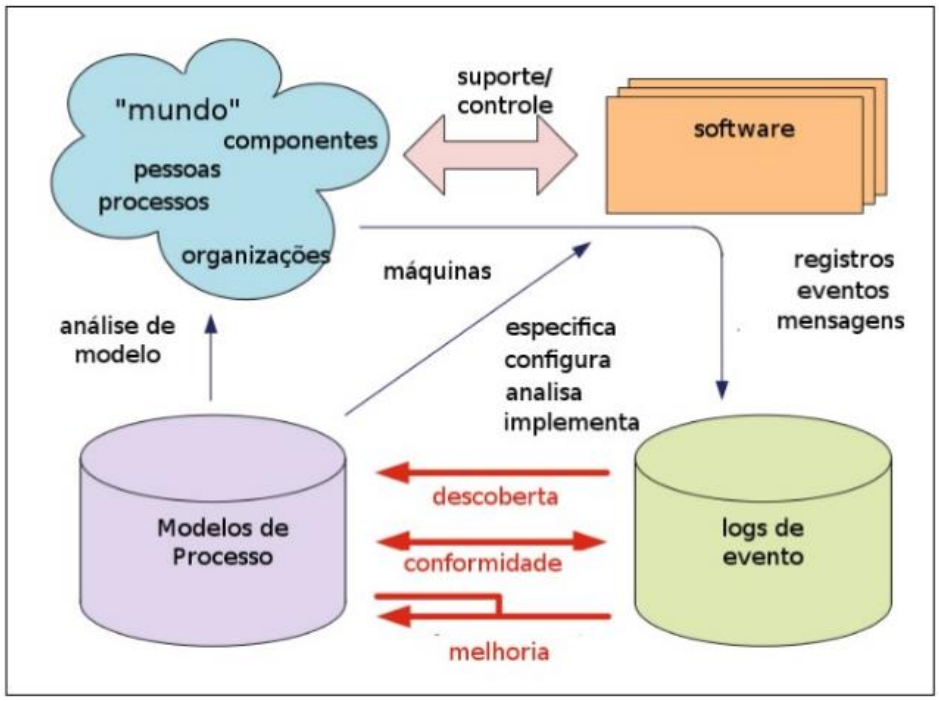
- Definição de conformidade ou *conformance checking* na literatura;
- problemas relacionados a área e visões futuras;
- algoritmos definidos na literatura que abrangem o contexto;
- as ferramentas utilizadas para verificar a conformidade;
- as principais características relacionadas aos dados que são usados para verificar a conformidade; e
- verificar se as técnicas focam em modelos existentes.

Sabendo que os principais insumos para a análise da conformidade em *process mining* são um *log* de eventos e o desejo ou necessidade do que se deseja verificar, é importante que os objetivos esperados sobre a avaliação sejam definidos.

2.2.1 Conformidade em PM

Basicamente todos os trabalhos relacionados à mineração de processos apresentam uma definição simplificada de conformidade. Isso se justifica devido aos três objetivos da mineração de processos demonstrados na Figura 1. O presente estudo tem como foco o objetivo do manifesto IEEE definido em Daniel *et.al* (2011) referente ao item c) verificação da conformidade, que é aplicado para validar um modelo de processo existente confrontado com a realidade, que pode ser identificada através do *log* de execução referente ao processo.

Figura 1 – Manifesto 2011 – O posicionamento dos três tipos principais de *Process Mining*: (a) descoberta, (b) verificação de conformidade e (c) extensão



Fonte: Manifesto, 2011

Para (VAN DER AALST, WIL MP; ADRIANSYAH; VAN DONGEN, 2012), a reprodução do modelo usando o histórico do *log* de eventos possibilita estabelecer uma

relação precisa entre os eventos e os elementos do modelo. O alinhamento do *log* é importante para a análise, pois estabelece uma relação de caminho para avaliar a conformidade, em que a realidade identificada pelo *log* de eventos pode ser comparada com o modelo existente e vice-versa.

Porém, identificar os eventos relevantes no *log* e mapear a realidade nem sempre é um trabalho simples, dadas as características das bases de dados atuais, que possuem diversas características. Vários trabalhos descrevem sobre as perspectivas de se obterem características da conformidade. As abordagens propõem formas de aplicar a avaliação.

Em um trabalho recente, de Leoni *et al.* (2016) definem que a conformidade na mineração de processos pode ser requerida a partir de normativas existentes na organização, em que atividades são requeridas. O trabalho afirma que a ordem em que as atividades devem ser executadas é importante para o processo. Outras questões relacionadas ao contexto do gerenciamento da conformidade, auditoria, segurança, entre outras, estão diretamente relacionadas à checagem da conformidade a partir do *log* de eventos.

No mesmo trabalho, os autores propuseram um *framework* que busca unificar as abordagens e as propostas de análise identificadas na literatura. A abordagem combina técnicas de mineração de dados e mineração de processos, onde é destacada a importância de correlacionar diferentes características do processo, o que resultará em uma análise mais refinada e, consequentemente, acarretará na melhor identificação da conformidade do processo. Abordagens que utilizam das técnicas de Redes de Petri e Lógica Linear Temporal para avaliar o comportamento do modelo gerado são destacadas neste trabalho. O motivo pelo qual ocorrem desbalanceamento na carga de trabalho e gargalo em um determinado ponto deve ser alvo de questionamento e investigação. Avaliar a influência do tempo quanto ao processo não seguido é importante. Nessas análises, de acordo com o estudo, para avaliar a conformidade, códigos ou *scripts* complexos podem ser necessários.

Em outro trabalho, van der Aalst *et al.* (2012) descrevem que os padrões tradicionais de descoberta de mineração de processos não representam o processo de ponta a ponta. Características importantes, como transações simultâneas e compartilhamento de recursos, não são levadas em consideração. No estudo, os autores estabeleceram quatro dimensões para tratar a qualidade e comparar o modelo existente com o *log* de eventos. As dimensões são: (1) aptidão, (2) simplicidade, (3) precisão e (4)

generalização. Para mensurar as dimensões, são definidas métricas que tratam o alinhamento em relação ao comportamento do *log*. Uma delas é o total de *traces* completos considerando os movimentos efetuados no *log*.

Mais tarde, van der Aalst (2014), em um trabalho que consolida abordagens existentes sobre as características de conformidade na área de mineração de processo, destaca a conformidade como um dos principais problemas a serem tratados, pois, apesar de haver um número de trabalhos considerável sobre o tema e uma alta maturidade sobre o assunto, as técnicas existentes deixam muito a desejar, principalmente quando se referem a um grande e complexo volume de dados. Para analisar a conformidade, o autor usa técnica de decompor o modelo do processo em pequenas partes: se todos os *traces* se encaixam no modelo decomposto, também se encaixarão no modelo total e gerarão um modelo adequado. Na análise, a exemplificação é feita apenas com *log* sintético de forma explicativa. Não há exemplificação da aplicação da técnica em *framework*.

Como já citado anteriormente, uma forma de avaliar conformidade em mineração de processo é a análise do modelo existente *versus* os desvios identificados no modelo gerado através do *log* de eventos. Diversos trabalhos encontrados na literatura descrevem esta análise. De acordo com van der Aalst e van Dongen (2013), o primeiro trabalho que aborda a análise da conformidade em mineração de processos foi desenvolvido por (ROZINAT; VAN DER AALST, 2008a). Nesse primeiro trabalho, além de concordarem com a forma de analisar a conformidade avaliando os modelos, os autores destacam que a conformidade aplicada à mineração de processos também pode ser usada para avaliar o desempenho das técnicas aplicadas na descoberta do processo. Isso foi reforçado por van der Aalst (2014), que destaca ainda cinco pontos em que os desvios podem ocorrer:

- Fraude (comportamento não conforme).
- Ineficiência (descuido ou negligência podendo acarretar em custo ou atraso).
- Exceções (casos selecionados são tratados *ad hoc* por não se encaixarem no modelo).
- Procedimentos malconcebidos (para executar o trabalho, as pessoas precisam desviar continuamente do modelo do processo).
- Procedimentos desatualizados (o processo evoluiu e a descrição do processo não mais corresponde à realidade).

Para avaliar os itens, o trabalho aborda técnicas de decomposição tanto para a avaliação da conformidade quanto para a descoberta do modelo. Quando avaliamos a perspectiva da conformidade em mineração de processos, podemos ver que há uma clara preocupação em descobrir um modelo ideal para se estar adequado a uma melhor conformidade. Nesse contexto, duas grandes frentes nos direcionam a dois pontos importantes: 1) a conformidade, que está fortemente ligada à descoberta do modelo, e 2) a perspectiva de cada negócio, cujos parâmetros específicos devem ser mapeados para o contexto da avaliação que está fortemente atrelada ao negócio.

Pérez-Castillo *et al*. (2014) propõem outra abordagem que foca, em primeiro lugar, na extração de eventos do *log* do sistema a partir dos eventos que constam no modelo existente para, então, analisar a conformidade. Nessa abordagem, a conformidade é tratada com ênfase no tratamento da busca do evento e sua extração, e, nesse contexto, os autores definem três diretrizes para extração dos eventos:

- Especificar as atividades que representam uma operação do negócio na vida real e que estejam no modelo do processo de negócio.
- Se uma atividade é relevante, ao recuperá-la, se houver uma atividade que a anteceda, esta deve existir no *log*, seja ela ligada direta ou indiretamente àquela recuperada.
- Assegurar que todas as atividades subsequentes sejam recuperadas de forma direta ou indireta.

Após a extração, a conformidade é avaliada de acordo com os modelos (o modelo existente e o modelo gerado pelo *log*). Percebe-se, mais uma vez, que o tratamento feito na busca pelo evento correspondente interpola as necessidades de mineração de dados, o entendimento do processo atual juntamente com o modelo existente e finalmente, a descoberta de um modelo gerado pelo dado através da mineração de processo. O modelo exemplificado após a extração dos eventos é gerado pela ferramenta de mineração ProM e os algoritmos propostos focam na extração dos eventos.

Em uma abordagem mais concentrada em modelo do que em conteúdo do *log*, Kalenkova *et al.* (2015) apresentam uma nova técnica para gerar o modelo na notação BPMN a partir do *log* de eventos. Características para tratar a conformidade são apontadas como consequência do modelo gerado a partir da notação BPMN. A nova abordagem usa as técnicas implementadas no ProM que geram os modelos em Rede de

Petri, *causal net* e *process trees*, e possibilita a mudança dos modelos para a notação BPMN. A implementação, apesar de ser testada com modelos existentes na ferramenta ProM, não apresenta resultados estatísticos da conformidade.

Outmazgin e Soffer (2016), em uma perspectiva diferente, apresentam uma lista genérica dos caminhos alternativos (*work-around*) a serem identificados no *log* de eventos. São definidas seis características para medir a conformidade no processo. A lista ressalta os desvios provenientes de usuários do sistema, que são delineados como intencionais. O objetivo é ajudar a identificar os itens da lista a fim de melhorar a conformidade. A capacidade da análise dos itens através das técnicas de mineração de processo apresentou dois itens que não podem ser identificados no *log* de eventos gerados, construindo uma perspectiva de que outros parâmetros devem ser avaliados para uma abrangência completa. A Tabela 1 descreve as características.

Dos seis itens citados na Tabela 1, dois são apontados como não podendo ser identificados no radar das técnicas de mineração de processos a partir do *log* de eventos. Isso reflete o alto grau de dependência em relação ao conhecimento do processo e às atividades nele inseridas.

Outro problema apontado na utilização das técnicas de mineração de processos, desta vez referente ao grande volume de dados e à variedade de atividades distintas, é tratado por van der Aalst (2013), que propõe uma abordagem genérica de divisão e conquista aplicada à Rede de Petri para decompor problemas apontados na área de mineração de processos. A promessa é aplicar essa divisão em diferentes técnicas existentes de descoberta do processo e verificação da conformidade: divide-se o modelo do processo em fragmentos menores e o *log* de eventos em sub-*logs*. Para trabalhar com um grande volume de dados, essa técnica pode ser válida. No entanto, definir as características da base de dados (por exemplo, o domínio) é necessário antes de separar o *log* de eventos; caso contrário, um novo gargalo poderá criar um viés na verificação da conformidade.

Ainda sobre o tratamento da conformidade na área de mineração de processos, Bolt *et al.* (2016) apresentam um novo *framework* contemplando seis categorias inerentes ao *log* de eventos, com especial preocupação com as características de extração dos dados e do modelo. As categorias são propostas para apoiar a análise de fluxos de trabalho denominados de *workflow* científico. Uma nova ferramenta definida como rapidProM[4]

[4] Disponível em: <http://www.rapidprom.org>.

foi desenvolvida com extensões de algoritmos da ferramenta ProM[5]. Para reparar problemas de conformidade, duas perspectivas são abordadas; estas foram definidas como problemas estruturais ou comportamentais.

Tabela 1 – Seis características definidas por Outmazgin e Soffer (2016)

Item	Definição	Exemplo
Ignorar partes do processo	Sequência das atividades.	Instâncias que possuem atividades executadas antes de atividades requeridas
Seleção da instância de uma entidade pelo caminho preferível	Adequação da instância a limite tolerado que não precise de aprovação	Um pedido que requer aprovação a partir de um determinado limite, mas o usuário particiona em mais de uma instância, ficando dentro de um limite que não precise de aprovação
Alterações de informações (*update*) Alteração não pode ser identificada no *log*	Modificar uma operação que foi usada para uma decisão, provocando inconsistência na decisão atual e deixando de refletir a alteração na atividade seguinte	Introduzir informações falsas que permitem ao processo passar para a próxima atividade ou averiguar que a quantidade de um pedido aprovado é superior ao limite estabelecido, modificando-se a etapa em vez de voltar o processo para a fase de aprovação.
Função incompatível com a definição	Identificar instância com aprovador cuja responsabilidade não lhe compete	Um processo com cotação aprovada e deve ser tratado por um funcionário de compras, mas é tratado por outro funcionário. como solução alternativa é transferido para o departamento de compras para prosseguir
Instâncias de entidades fictícias	Instâncias de entidades fictícias que são criadas para determinado monitoramento e documentação, mas que **NÃO** são suportadas pelo sistema	Formalizar entrevista com aluno em um processo de admissão requer que seja feito o cadastro inicial e que este seja alocado em uma sala de aula fictícia
Separar o processo real do relatado Alteração não pode ser identificada no *log*	Instâncias que apresentam situações que possuem tratamento manual e sofrem execução *ad hoc* fora do prazo	Uma requisição de compra precisa de tempo para aprovação de um gerente. Porém por ter pequena possibilidade de ser reprovada, é movida pelo participante para outra etapa, possibilitando avançar sem esperar o tempo estipulado/entendido.

[5] Disponível em: <http://www.promtools.org>.

2.2.2 Por que medir a conformidade em PM

A tabela a seguir apresenta itens que justificam a importância de verificar a conformidade da mineração de processos. Tais itens foram mapeados a partir dos artigos lidos para a formulação do item 2 deste trabalho e também a partir de leitura sobre o tópico.

Tabela 2 – Motivos para investigar a conformidade em mineração de processos

Por que usar as técnicas de verificação da conformidade em mineração de processos:
1. Ajudam a apontar desvios no processo. (PREMCHAISWADI; POROUHAN, 2015)
2. Ajudam a identificar fraudes. (YANG; HWANG, 2006)
3. Ajudam a identificar procedimentos malconcebidos. (VAN DER AALST, 2014)
4. Ajudam a identificar procedimentos ultrapassados. (ROZINAT; VAN DER AALST, 2008b)
5. Ajudam a avaliar uma técnica de descoberta de processo. (PÉREZ-CASTILLO *et al.* 2014)
6. Ajudam a selecionar possíveis candidatos para usar em um próximo modelo. (PÉREZ-CASTILLO *et al.* 2014)
7. Ajudam a julgar um modelo de processo descoberto automaticamente a partir de um *log* de eventos. (LI *et al.* 2015)
8. Ajudam a medir a severidade de um desvio. (VAN DER AALST, WIL MP; ADRIANSYAH; VAN DONGEN, 2012)
9. Ajudam a verificar a conformidade de um processo estabelecido. (PREMCHAISWADI; POROUHAN, 2015)

2.2.3 Evolução dos artigos analisados

Para avaliar os artigos e estudar o tema de conformidade em mineração de processos, inicialmente utilizamos a metodologia de busca de artigo por assunto. O objetivo não era fazer uma revisão sistemática, mas entender a evolução dos trabalhos relacionados à conformidade.

Dos artigos da busca geramos a visualização em gráfico demonstrada na Figura 2, que mostra como a verificação da conformidade na área de mineração de processos tem evoluído. Esse gráfico é proveniente dos artigos consultados nas bases Periódicos Capes, Scopus e Springer, das quais, após uma análise prévia, foram escolhidos 32 artigos. O

primeiro critério de avaliação estava pautado na busca por artigos que descrevessem sobre a verificação da conformidade em mineração de processos.

Dos 32 artigos analisados, 12 apresentam proposta para trabalhos futuros, ou seja, apenas 37,5%. Somente 17 artigos têm como destaque a conformidade. Foram excluídos 15 artigos, que apenas definem a conformidade na área ou nem a citam.

A consulta realizada para busca dos artigos utilizou os termos "*process mining*" e "*conformace checking*" combinados com "*investigate*", "*identify*" e "*analyze*".

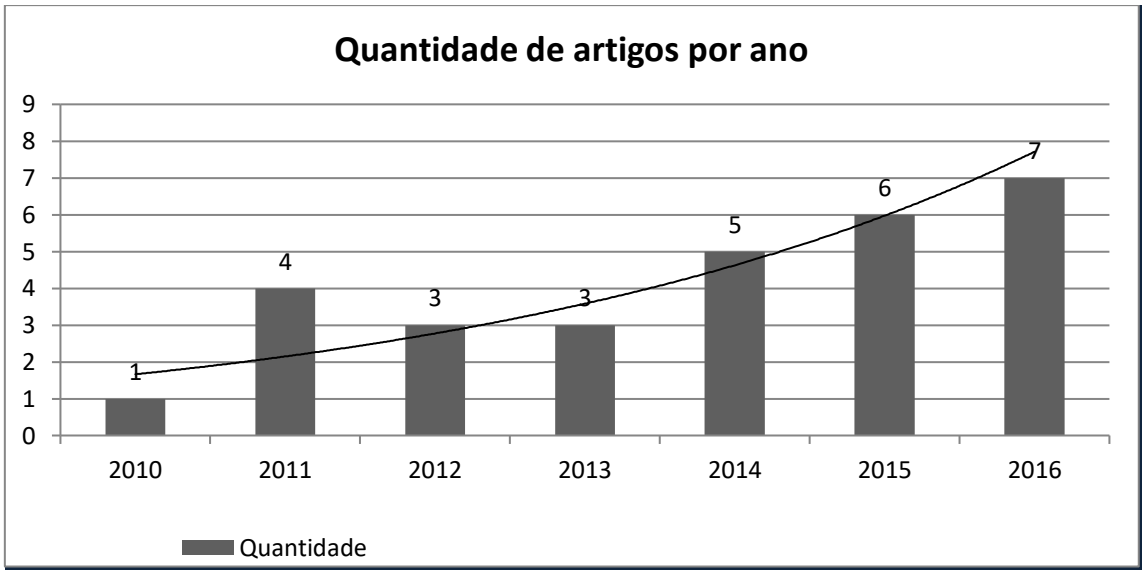


Figura 2 – Total de artigos avaliados por ano, de acordo com consulta realizada

Durante a análise dos artigos dessa consulta, observou-se que muitos artigos citados em trabalhos com foco na análise de conformidade em mineração de processos não foram identificados na consulta gerada. Por isso, para melhorar a análise nos trabalhos da referência, uma nova consulta foi efetuada em março de 2017, considerando os artigos publicados no período de 2011 a 2016. Nessa consulta, foram usados apenas os termos "*process mining*" e "*conformance checking"*, e, dessa vez, 75 artigos retornaram na consulta. Da nova consulta, os artigos referentes aos anos de 2015 e 2016 foram usados para referência neste trabalho.

Sob a perspectiva da conformidade, diversos trabalhos utilizam a abordagem que considera o modelo de processo existente. Nem sempre a perspectiva do dado projeta o que de fato se entende. Nesse sentido, os trabalhos não apontam direcionamento de junção entre a avaliação da conformidade dos modelos através da ótica do comportamento do dado, se preocupando com a ótica do tratamento aplicado para gerar o *output* desejado.

2.3 BPM

A sigla BPM deriva do inglês *Business Process Management*. Extensivamente utilizada para modelagem de processos de negócio, a área necessita do apoio de sistemas. Em definição ampla, BPM é um conjunto de tecnologias que auxilia no suporte e apoio ao gerenciamento de processos. Toda organização está apta a implementar o BPM em seu contexto de trabalho. A tecnologia aplicada ao BPM pode proporcionar impacto positivo dentro de um grupo, instituição ou coletivo, trazendo melhorias na forma de trabalho. Para Daniel *et al*. (2011), o BPM se segmenta em duas vertentes: a que avalia o conjunto de itens atrelado ao uso de tecnologias e a que avalia as tecnologias envolvidas no uso do BPM.

A disciplina de BPM condiciona o processo a seguir padrões e possibilita a medição de desempenho. Na implementação, o BPM traz como característica a mudança de paradigma de trabalho. Sustentado pela tecnologia, como é a maioria dos processos do mundo atual, o BPM se torna uma ferramenta de tecnologia de valor dentro das organizações. Há um crescimento na busca por agregar tecnologia à disciplina que poderá perdurar por vários anos (ABPMP, 2013).

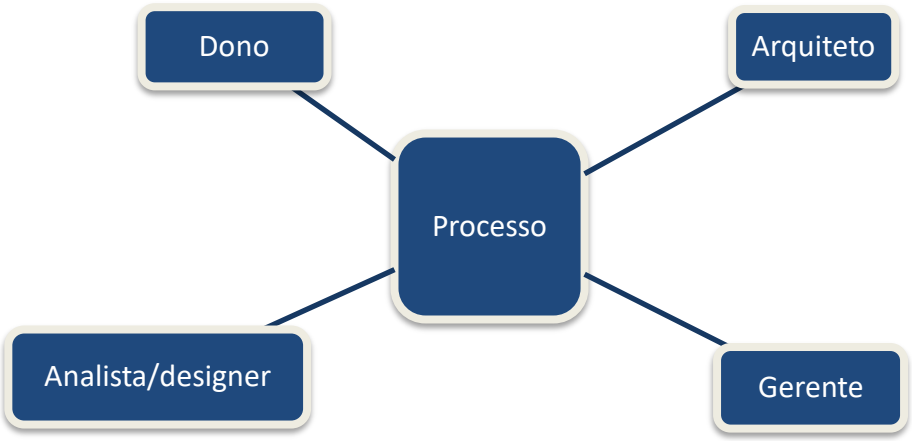


Figura 3 – Visão das quatro posições do BPMN definidas em ABPMP (2013)

O BPM define papéis e responsabilidades na sua implementação a Figura 3 apresenta quatro papéis e sugere o foco de todos no processo. Nesse contexto, o escopo de BPM intercala com o BPO (Orientação do Processo de Negócio) na busca pela melhoria contínua. Uma das características do BPO é a busca pela maturidade com base

na orientação da cultura e da estrutura da organização. Com foco nas pessoas e no processo, as características para o BPM são entendidas e definidas pela organização, fazendo o ciclo evoluir para mitigar os papéis e responsabilidades. A Figura 4 demonstra as principais etapas de cada ciclo, definindo os papéis para o BPO e para o BPM. Na imagem, podemos destacar o papel do BPM como um ciclo evolutivo de desenvolvimento continuado, enquanto o BPO se preocupa em inserir no ciclo a cultura e a estrutura da organização.

Os papéis bem definidos são de suma importância para o bom andamento e orquestração do processo. São formas de mensurar a qualidade e buscar pela melhoria contínua, apoiando as organizações no direcionamento para melhor conduzir o dia a dia de trabalho. As ações aumentam a produtividade e mitigam pontos de gargalos, o que aprimora o processo como um todo, além de conduzi-lo para o objetivo de garantir maior qualidade daquilo que se produz.

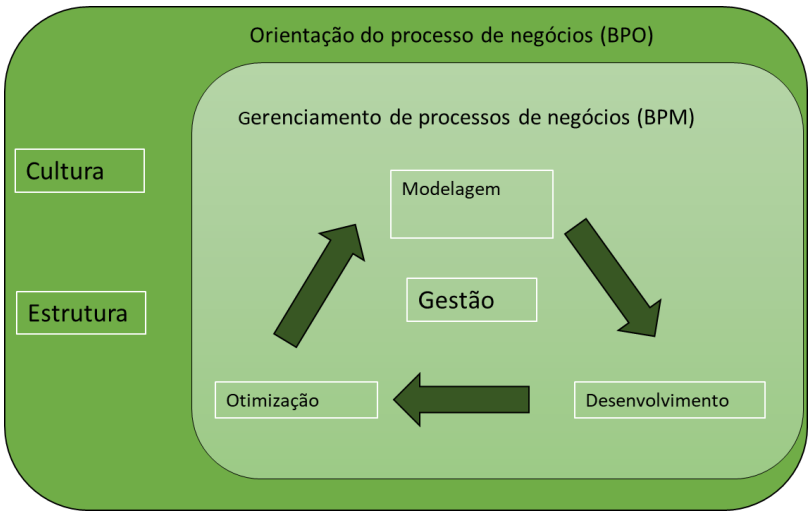


Figura 4 – Estrutura do BPM e BPO
Fonte: VAN LOOY, 2014 (adaptado).

A busca pela equalização e conformidade do produto ou processo, assim como a garantia de adequação do processo à cultura e estrutura da organização, unem o BPM a normas e modelos de gestão que auxiliam nesse direcionamento. Van Looy (2014) define o compartilhamento de diversas metodologias que auxiliam a implementação e gestão do BPM, como normas de qualidade que mensuram a maturidade do processo. Alguns exemplos: CMMI, ISSO/IEC 15504, FAA-ICMM e OMG-BPMN. Além disso, mapeia 69 modelos genéricos que são praticados na busca pela maturidade do processo.

Há uma vasta gama de ferramentas e metodologias para apoiar o BPM. Sinur (2003) destaca a existência de mais de 70 produtos disponíveis no mercado relativos ao BPMS (sigla para Suíte de Gerenciamento de Processos de Negócio). Nos últimos 10 anos, com a globalização e, principalmente, com a inovação tecnológica, as abordagens de BPM passaram por avanços e foram conduzidas para as ferramentas de TI (Tecnologia da Informação), que possibilitam o uso de notações como EPC, BPMN e UML que auxiliam na geração de métricas baseadas na pré e pós-execução do modelo. Para Wasilewski (2016), os fornecedores das ferramentas mais modernas agregaram um pouco de inteligência, trazendo para a evolução o iBPMS (Suíte de Gerenciamento de Processos de Negócios Inteligente).

As notações citadas, entre outras, estão disponíveis em algumas ferramentas de mineração de processo.

2.3.1 Outras verificações sobre o BPM

Nos últimos anos, o BPM evoluiu e ganhou destaque. Diversas organizações seguem processos com metodologias distintas para agregar valor ao produto, garantir qualidade ou assegurar a competitividade. Como destaque dessa evolução, temos:

- A própria metodologia do BPM (WASILEWSKI, 2016).
- O guia de gerenciamento de processos de negócio (BPM CBOK V3.0; ABPMP, 2013).
- A evolução das ferramentas de modelagem e notações, incluindo as ferramentas *open source* – com destaque para o *Bizagi*, a única ferramenta que, além de suportar integrações, trabalha com diferentes linguagens (SOUSA *et al.* 2018).

Por fim, anualmente, o grupo Gartner divulga um relatório em que as soluções de controle e automatização de processos são avaliadas e as principais ferramentas são exemplificadas no quadrante mágico de Gartner [6] . A Figura 5, adaptada de (WASILEWSKI, 2016) e Gartner[7], mostra os três principais fornecedores dos últimos anos (2012, 2014, 2015 e 2017). O quadrante dos desafiadores mostra que organizações

[6] Empresa americana de consultoria. Disponível em: <https://www.gartner.com/technology/about.jsp>.
[7] Disponível em: <https://www.gartner.com>.

só aparecem no relatório do ano de 2017. Ainda no ano de 2017, a gigante da área de tecnologia, Oracle, é movida do quadrante dos visionários para o quadrante dos desafiadores. Por fim, o último ano aponta quatro novas organizações (Axon Ivy, AuraPortal, Genpact e Newgen Software) e mantém outras três no mesmo patamar (Whitestein, Software AG e Tibco).

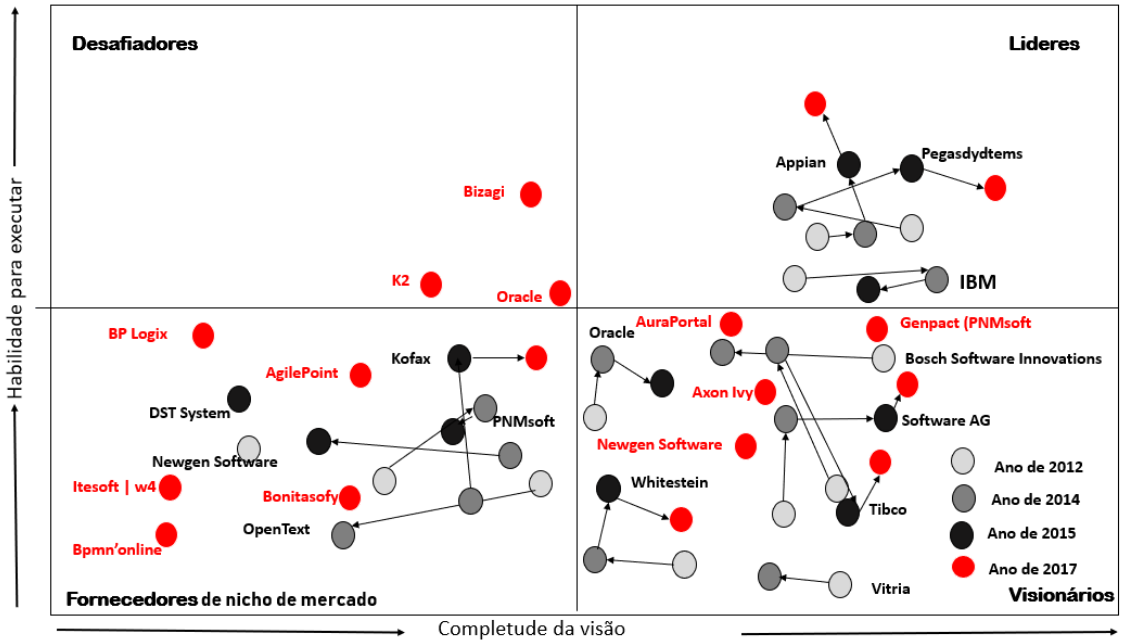


Figura 5 – Quadrante de Gartner dos anos de 2012, 2014, 2015 e 2017
Fonte: (WASILEWSKI, 2016); GARTNER, [s.d]. (adaptado).

3 MINERAÇÃO DE PROCESSOS – COMPONENTES

Neste capítulo, descreveremos os principais conceitos relacionados à área de mineração de processos. Como foco, consideramos os conceitos relevantes para o entendimento do objeto da pesquisa. Além disso, concluímos contemplando os conceitos relacionados às técnicas de mineração de texto, necessários para complementar o estudo em questão.

Temos como objetivo apresentar os conceitos principais para a compreensão do modelo proposto neste trabalho.

3.1 FUNDAMENTOS

A essência da mineração de processos é a geração de modelos de processos através de *log* de eventos derivados de sistemas informatizados. A MP é capaz de gerar o modelo de processo sem o conhecimento embasado do que o *log* possui, ou seja, nesse contexto o *log* é capaz de demonstrar um processo utilizando apenas o contexto que ele mesmo possui.

A premissa é que esse *log* possua uma estrutura requerida de sequência de atividades dentro de um mesmo item.

Os sistemas de informação que armazenam esses *logs* nem sempre possuem a adequação que ele exige de forma direta, sendo necessária a padronização com utilização de técnicas de mineração de dados e transformação de texto.

3.2 TIPOS DE MINERAÇÃO DE PROCESSOS

O particionamento dos três objetivos da mineração de processos foi definido pelo grupo IEEE *Task on Force Process Mining* (VAN DER AALST, WIL *et al.*, 2011) como:

- Descoberta: é a parte da MP responsável pela geração do modelo baseado no *log* de eventos. Nessa parte, a MP é capaz de mostrar as características que o modelo possui como o fluxo do processo e o modelo que a organização possui inputado nos *logs*, ou seja, o modelo que o *log* externaliza.
- Verificação da conformidade ou auditoria: esse item trabalha com a avaliação do modelo descoberto através da mineração com o uso do *log* de eventos em relação ao modelo entendido, ou seja, o modelo pré-existente, uma relação que visa identificar na análise possíveis pontos de melhorias, desvios ou inconsistências.
- Aprimoramento: o objetivo dessa etapa é melhorar o processo de forma a estender o modelo existente ou melhorá-lo com os insumos que forem produzidos pela MP através do *log* de eventos.

Além das técnicas de análise citadas, Buijs (2010) descreveu outras cinco:

- Visualização do desempenho do processo: mensura o desempenho do processo através da análise qualitativa do *log* de eventos. Indicadores de tempo e espera de uma atividade, tempo de processamento de um caso e tempo entre início e fim de uma atividade são alguns exemplos citados por Hornix (2007) como indicadores de desempenho.
- Análise de redes sociais: nos *logs* de eventos, é comum encontrar os registros de locais do evento, responsável pelo registro, responsável pela atividade, entre outros, o que possibilita identificar a densidade da rede, a distância entre os recursos, os recursos ligados a outros ou recursos isolados (VAN DER AALST *et al.* 2005). Em outra definição, van der Aalst (2011) conclui que se o *log* de evento possui as informações de recursos, é possível descobrir modelos sobre eles, como uma rede social que mostra a forma como as pessoas trabalham juntas.
- Previsão de casos com dados históricos: baseado no *log* de eventos e seu histórico, possibilita prever o que pode acontecer em cenários futuros.
- Verificação de regras: cenário que avalia se as regras definidas como premissas no processo estão sendo seguidas. Auxilia diretamente na avaliação e validação da conformidade.
- Detecção de regras de negócio: cenário que detecta regras não definidas no processo, mas que são de fato seguidas, possibilitando uma melhoria ou

adequação do processo. Auxilia a avaliação da conformidade, criando cenários de melhoria contínua e garantia da qualidade.

Em trabalhos recentes, há ênfase na conformidade e na análise de gargalos juntamente com a descoberta do processo, o que auxilia na tratativa de conformidade e desempenho (DE LEONI *et al*. 2016).

3.3 CONCEITOS-CHAVE DE MINERAÇÃO DE PROCESSOS

Os principais requisitos da área de mineração de processos foram definidos no Manifesto IEEE (VAN DER AALST *et al.* 2011), encontram-se reforçados em VAN DER AALST *et al.* (2016) e são aplicados aos escopos de MP. São eles:

- Atividade: para o BPM, o conjunto de atividades executadas em sincronização pela organização compõe o processo do negócio (SOUSA *et al.* 2018); para MP (WEIJTERS *et al.* 2006), cada evento se refere a uma atividade. Na definição do Manifesto IEEE (VAN DER AALST, WIL *et al.*, 2011), a atividade pode ser estendida para refletir cenários como uso de recursos, custos ou frequência de *input*. De forma simplificada, a atividade é uma tarefa executada por um recurso ou por um processo automatizado, o qual é definido em uma sequência de eventos anteriormente inicializada por um indivíduo. No exemplo da Tabela 3, cada campo que compõe a coluna "atividade" pode ser visto como uma tarefa, mas cada tarefa abrange o contexto de "atividade" no sentido de que cada uma delas possui no seu contexto outras unidades. Para abrir um chamado, o usuário precisa logar no sistema, digitar informação de busca, inserir as informações necessárias e, depois, clicar em "cadastrar". Todo esse contexto que não aparece nesse *log* compõe a atividade.

- Caso: é definido com a instância do processo; cada *case* tem uma identidade única e cada evento se refere a um *case*. Na Tabela 3, o *case* de id=1 possui quatro atividades em seu *log*.
- Eventos: são as informações inseridas nos sistemas, seja por humanos, seja por máquinas. Refere-se a cada unidade lógica registrada, por exemplo: a data, o início de uma atividade etc.
- Recurso: pessoa ou o equipamento responsável pela execução da atividade.
- *Timestamp*: registro da data e hora em que o evento foi executado pelo recurso ou pelo equipamento. Por exemplo: a solicitação da abertura do processo mostraria a data inicial do evento, com a atividade de abertura. Cada nova ação no processo deve conter ao menos uma data e seu andamento, podendo exibir também o responsável pela execução.

Tabela 3 – Exemplo de *log* com 4 atividades

Case id	Atividade	Timestamp	Responsável
1	Abrir chamado	2015-01-06 11:01:58.937	Alan
1	Encaminhado	2015-01-06 12:01:58.937	Danny
1	Resolvido	2015-01-06 13:01:58.937	Duda
1	Finalizado	2015-02-06 11:01:58.937	Alan

Outros trabalhos estendem esses requisitos para além dos eventos de dados, porém são definições feitas a partir deles. Por exemplo:

- Processo de negócio: explorar as múltiplas perspectivas do processo a partir do *log* de eventos (MANNHARD *et al.* 2015). Casos em que há grande volume de eventos e variedade de atributos, por exemplo, podem evidenciar características em torno do processo não exatamente explícitas no *log.*
- Tarefa: é a unidade de trabalho que uma pessoa ou um grupo de pessoas pode realizar. Para ABPMP (2013), a tarefa é a decomposição de atividades em unidades de trabalho menores. Para *MP*, esse contexto se torna valor agregado no *log.* No exemplo da
- Tabela 3, a tarefa é cada item descrito no campo “atividade”.

3.4 XES OU XML

O formato do *input* de *log* de eventos adotado pela IEEE *Task Force*[8] em mineração de processos é o XES. O formato foi padronizado em 2010 e adotado por ferramentas de mineração de processos como o ProM por ser suportado pela biblioteca OpenXES[9].

O formato XES (eXtensible Event Stream) é o padrão esperado para armazenar e trabalhar com os *logs* de eventos no ProM. O XES é sucessor do MXML (Mining eXtensible Markup Language). A extensão XES suporta conversão de diferentes tipos de fontes de dados para o formato XML (MS Access, CSV, CPN Tools, Cognos etc). De acordo com van der Aalst (2016), a organização de Padrões do IEEE está avaliando o XES com o objetivo de transformá-lo em um padrão IEEE oficial.

Tabela 4 – Fragmento de *log* de evento

Case id	Atividade	Timestamp	Responsável
1234	a	2016-03-10 14:00:53.893	Paulo
1234	b	2016-04-10 14:00:53.894	Iara
1234	c	2016-05-10 14:00:53.895	Giovanna
1234	d	2016-06-10 14:00:53.896	Carolina
1234	e	2016-07-10 14:00:53.897	Nara
1234	f	2016-08-10 14:00:53.898	Sergio
1234	g	2016-03-10 14:00:53.893	Paulo

[8] Disponível em: <http://www.xes-standard.org/>.
[9] Disponível em: <www.openxes.org>.

```
<?xml version="1.0" encoding="UTF-8" ?>
...
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7">
    <extension name="Lifecycle" prefix="lifecycle" uri="http:// .../lifecycle.xesext"/>
    <extension name="Time" prefix="time" uri="http://.../time.xesext"/>
    <extension name="Concept" prefix="concept" uri="http://.../concept.xesext"/>
    <classifier name="Event Name" keys="concept:name"/>
    <classifier name="(Event Name AND Lifecycle transition)" keys="concept:name
lifecycle:transition"/>
    <string key="concept:name" value="XES Event Log"/>
     <trace>
         <string key="concept:name" value="1"/>
         <event>
             <string key="lifecycle:transition" value="start"/>
             <string key="concept:name" value="a"/>
             <date key="time:timestamp" value="2016-03-10T14:00:53.893-03:00"/>
             <string key="Responsável" value="Paulo"/>
         </event>
         <event>
             <string key="lifecycle:transition" value="start"/>
             <string key="concept:name" value="b"/>
             <date key="time:timestamp" value="2016-04-10T14:00:53.894-03:00"/>
             <string key="Responsável" value="Iara"/>
         </event>
         <event>
             <string key="lifecycle:transition" value="start"/>
             <string key="concept:name" value="c"/>
             <date key="time:timestamp" value="2016-05-10T14:00:53.895-03:00"/>
             <string key="Responsável" value="Giovanna"/>
         </event>
         <event>
             <string key="lifecycle:transition" value="start"/>
             <string key="concept:name" value="d"/>
             <date key="time:timestamp" value="2016-06-10T14:00:53.896-03:00"/>
             <string key="Responsável" value="Carolina"/>
         </event>
         <event>
             <string key="lifecycle:transition" value="start"/>
             <string key="concept:name" value="e"/>
             <date key="time:timestamp" value="2016-07-10T14:00:53.897-03:00"/>
             <string key="Responsável" value="Nara"/>
         </event>
         <event>
             <string key="lifecycle:transition" value="start"/>
             <string key="concept:name" value="g"/>
             <date key="time:timestamp" value="2016-08-10T14:00:53.898-03:00"/>
             <string key="Responsável" value="Sergio"/>
         </event>
     </trace>
</log>
```

Figura 6 – Fragmento de *log* no formato .XES

O *plugin* do ProM (*Convert .csv to .XES*) gera o arquivo “XES” exemplificado na Figura 6, proveniente do *log* de eventos da Tabela 4. Por padrão, o XES não descreve um conjunto fixo de atributos. Identificadores do *log* são definidos como as extensões (VAN DER AALST, 2016) . No exemplo “XES”, são identificadas três extensões: *Lifecycle*, *Time* e *Concept*, e dois classificadores: *Event name* e *Lifecycle transition*. O classificador “Atividade” é evidenciado pelo evento classificador *concept:name*, o classificador

“Data_hora” é evidenciado pelo evento classificador t*ime:timestamp* e o classificador “Responsável” evidenciado pelo evento classificador “responsável”.

Em um *log* de eventos, para cada *trace* existe um evento global que é obrigatório. A primeira linha que contém a descrição *Concept:name* é o item que define esse evento no exemplo da Figura 6. Um classificador (*Classifier*) é definido por qualquer evento $e \in \varepsilon$; *e* é o nome do evento, ou seja, considerando que os eventos são identificados pelo nome da atividade, então e = #atividade(e), que representa {(1,a), (1,b), (1,c), (1,d), (1,e), (1,g)}.

Outras ferramentas de mineração de processos, como Disco (GÜNTHER; ROZINAT, 2012), Celonis, Minit, Rialto Process (VAN DER AALST, 2016), suportam o modelo de arquivo “.XES”. O site <xes-standard.org> dispõe de mais informações sobre a sintaxe “.XES”.

3.5 TIPOS DE PROCESSOS: LASANHA E *SPAGHETTI*

Os modelos de processo em *MP* podem ser definidos como “lasanha*”* ou “*spaghetti”*, derivados dos modelos estruturados, não estruturados ou semiestruturados.

A análise do modelo lasanha (ou estruturado) produz resultados mais consistentes, sendo, portanto, o mais apropriado para demonstrar gargalos de forma assertiva e produzindo modelos mais fáceis de serem compreendidos. A maioria das técnicas de mineração de processos focam na estrutura dos modelos lasanha.

Já os modelos do tipo *spaghetti* são derivados de *logs* que possuem um grande número de atividades e diversos caminhos possíveis para se executar um *trace*. São modelos mais complexos e difíceis de serem lidos.

De acordo com Kumar *et al.* (2017), devido à complexidade intrínseca no modelo *spaghetti*, a maioria das técnicas de mineração de processos não consegue ser aplicada sem que seja necessária uma intervenção na formatação para simplificar o modelo *spaghetti* produzido.

Na definição de Aalst (2016), um processo é consistente com o modelo lasanha se pelo menos 80% dos eventos ocorrem de acordo com o planejado.

Com o grande volume de dados oriundos de diversas fontes e contextos, produzir informações que resultem em modelos de processos torna-se um desafio. No entanto, a

diversificação do contexto pode trazer descoberta de conhecimento para as organizações. O desafio é minerar o processo proveniente de contexto de diversas atividades e caminhos aglomerados. As técnicas de mineração de processos precisam evoluir nesse sentido. Um exemplo dessa necessidade de evolução é a Figura 7, que mostra um modelo de processo do tipo *spaghetti* que, apesar de ter sido gerado com dados estruturados, apresenta um grande conjunto de dados e possui diversos processos, o que o torna muito difícil de ser analisado.

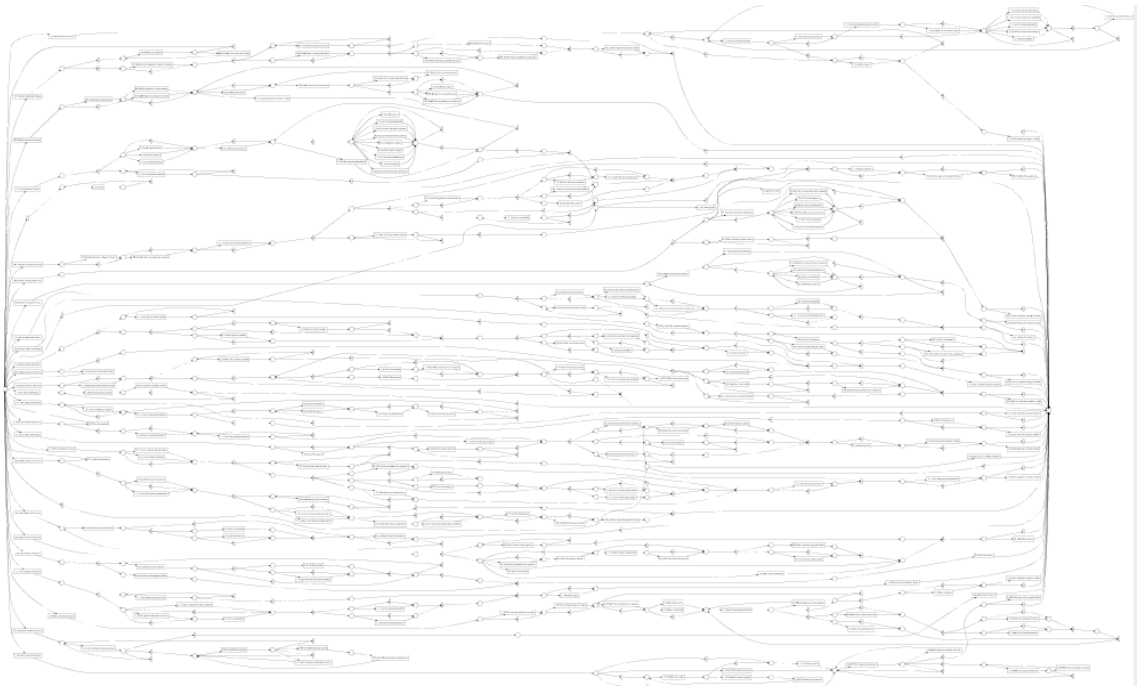


Figura 7 – Modelos *spaghetti* podem ser gerados por informações estruturadas

Muitos modelos *spaghetti* são provenientes do cenário de acúmulo de dados. De acordo com Anne (2011), a mineração de processos está mais relacionada a modelos estruturados, mas pode ser usada com técnicas de mineração de dados não estruturados como mineração de texto. Diversas técnicas podem auxiliar na mineração de textos não estruturados; algoritmos já consolidados e modelos treinados estão incorporados em ferramentas como Weka, Rapdminer, Knime e Pentaho (WITTEN *et al*. 2016, BERTHOLD *et al.* 2016, NAIK; SAMANT, 2016, SOUSA *et al.* 2015). Essas ferramentas podem auxiliar na limpeza e tratamento de dados e possuem algoritmos de treinamento e teste para melhorar modelos, aplicar métricas etc. Falaremos mais dessas ferramentas e algoritmos na sequência.

3.6 ALGORITMOS DE MINERAÇÃO DE PROCESSOS NO PROM

Diversos algoritmos para a descoberta do processo em mineração de processos são propostos na literatura. De acordo com van der Aalst (2016), a mineração indutiva tem como objetivo dividir o *log* de eventos em *sub-logs*, e a forma como se dará depende do operador. Basicamente, o algoritmo *inductive miner* trabalha de forma a separar o *log* de eventos e introduzir os operadores necessários para formular o modelo. A iteração de separação do *log* de eventos é feita repetidamente. Um exemplo de *trace* é mostrado na parte 1 da Figura 8; em seguida, na parte 2, define-se a ilustração da separação de cada item agrupado pelo operador; e, na parte 3, é gerado o modelo de Rede de Petri.

O resultado do modelo gerado pela técnica de mineração indutiva pode ser convertido para modelos nas notações BPMN e Rede de Petri de forma direta dentro da própria ferramenta do ProM.

O algoritmo *inductive miner* é um dos poucos implementados na ferramenta que garantem a solidez do processo.

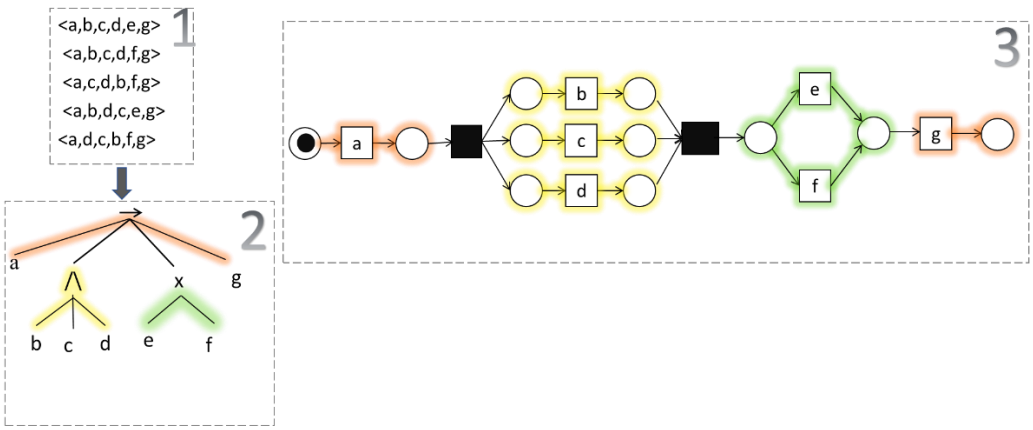


Figura 8 – Processo de *split* do *log* e geração do modelo em Rede de Petri

A Figura 8, na parte 1, contém 5 linhas de um *log* de eventos denominados *traces* ou casos. A árvore representada na parte 2 separa as atividades mais comuns, **a** e **g**, sendo o início e o fim de cada *trace*, seguidas de **e** ou **f**; posteriormente cada *trace* contém as

atividades **b, c** e **d** em ordens distintas, havendo um paralelismo entre as atividades. Os 5 casos do trecho do *log* e os 30 eventos são apresentados na Fórmula 1.

$$L = [\langle a, b, c, d, e, g\rangle^1 , \langle a, b, c, d, f, g\rangle^1 , \langle a, c, d, b, f, g\rangle^1 , \langle a, b, d, c, e, g\rangle^1 , \langle a, d, c, b, f, g\rangle^1]$$

Fórmula 1 – Representação do *log* da Figura 8 na parte 1

Além do modelo em Rede de Petri (Petri Net), há possibilidade de geração do modelo usando a notação BPMN.

$$\beta_1 = Im(L1) = (a, \wedge(b, b, d), x(e, f), g)$$

Fórmula 2 – Equação base para geração da árvore

Vários trabalhos apresentam resultados de análises utilizando o *plugin inductive miner* e a ferramenta ProM na mineração do processo. Em Feeley *et al*. (2017), os autores usaram-no para criar o modelo de *log* de eventos de solicitação de crédito de clientes no segmento de banco. Para descobrir o modelo de um *dataset* treinado (Ghawi, 2016), usa o algoritmo *inductive miner* do ProM.

A implementação no *framework* ProM com os pacotes que usam o método de indução junto com Rede de Petri para gerar o modelo foi descrita por Leemans *et al*. (2013).

3.6.1 Redes de Petri

Uma Rede de Petri é um grafo bipartido com **transições, posições e arcos**. A abordagem de Redes de Petri é descrita por van der Aalst (2016) como a mais antiga linguagem usada para modelos de processo. Os arcos interligam as posições e transições. A transição age sobre os símbolos (*tokens*) se o número de símbolos que foi requisitado aparecer em cada posição de entrada.

A distribuição dos símbolos sobre as posições define o estado da Rede de Petri. Na Figura 9, existe apenas um símbolo, que está marcado como estado inicial.

uma topologia formada por três elementos básicos representados por N= (C, T, G), sendo C um conjunto finito de posições; T um conjunto finito de transições de forma

que C∩T = ∅; e G o conjunto de arcos, o qual representa as relações do fluxo, sendo que G ⊆ (C x T)∪(T x C) compõe o conjunto dos arcos, ou seja, a relação do fluxo. Diz-se que (A, B) compõe um conjunto marcador das Redes de Petri onde A= (C,T,G) e B⊆ $\mathbb{N}(C)$ é um multiconjunto de C definindo a marcação da rede. A Rede de Petri da Figura 9 tem os estados [*C*], as ações [*T*] e as relações do fluxo (*G*) apresentadas na Fórmula 3.

$C = \{Início, c1, c2, c3, c4, c5, fim\ )$
$T = \{a, b, c, d, e, f, g, h\}$
$G = \{Inicio, a), (a, c1), (a, c2), (c1, b), (c1, c), (c2, d), (b, c3), (c, c3), (d, c4),$
$(c3, e), (c4, e), (e, c5), (c5, f), f, c1), (f, c2), (c5, g), (c5, h), (g, fim), (h, fim)\}$

Fórmula 3 – Representação de transições e sistemas

A transição é habilitada se cada uma das posições de entrada possui um símbolo. Na Figura 9, a posição de entrada é marcada pelo "**Inicio**", neste caso, um multiconjunto que contém apenas um símbolo. Cada transição da Rede de Petri tem o comportamento de consumir símbolos e produzir posições, fazendo com que uma posição seja marcada e ou disparada. Dito isso, na representação da imagem citada, temos que **a** é habilitado a partir da marcação "**Inicio**". Em seguida, após disparar o **a**, (c1, c2) são marcados, neste caso havendo **a** o consumo de um símbolo apenas e a produção de dois. Depois da marcação em (c1, c2), a transição **a** é desabilitada. Já as transições **b**, **c** e **d** ficam habilitadas nas marcações de (c1, c2), disparando **b** em (c2, c3), que são marcados, o que faz com que **d** continue ativo, mas **b** e **c** inativos. A atividade **f** no modelo apresenta um *loop* que possibilita uma infinita sequência de disparos para as atividades [início] e [fim].

Conforme já citado, uma das características da Rede de Petri é o fato de trabalhar com marcações através do consumo e da produção de símbolos, que podem pertencer a um multiconjunto. Na definição de van der Aalst (2016), o multiconjunto corresponde a um conjunto em que cada elemento pode ocorrer n vezes. Se considerarmos a marcação inicial da Figura 9 como $\{Início^5\}$, e o disparo for feito por **a**, a marcação muda para $\{Início^4, c1, c2\ \}$, produzindo a marcação de **a**; desta vez, se fizer o disparo de a, uma vez que ele está marcado, o conjunto de marcação mudará para $\{Início^3, c1, c2\ \}$, o que fará com que a transição **a** seja disparada 5 vezes $\{c1^5, c2^5\}$. Como consequência, **a**, **b**, **c** e **d** ficam ativas e podem ser disparadas simultaneamente.

A Fórmula 3 contempla a definição de transição de sistemas, uma abordagem básica utilizada na modelagem de processo. Esses modelos são comumente criados a partir de multiconjuntos nos quais os elementos podem ocorrer várias vezes. Por exemplo, em um multiconjunto com 11 elementos em que os 3 conjuntos são idênticos $\{a, b^2, c^2, d^3, e^2, g\}, \{a, b, b, c^2, d, d, d, e^2, g\}, \{g, b^2, c^2, d^3, e^2, a\}$, a ordem dos conjuntos não importa, somente o número de ocorrências de cada um.

Se X = $\{a, b^2, c^2, d^3, e^2, g\}$, então X ($b$)=2 e X ($f$) = 0. Logo, o somatório de (X⊎Y), a diferença (X/Y), qualquer elemento contido em um multiconjunto faz com que esta pertença a X o subconjunto sendo (X≤Y), estendendo para diferentes domínios se necessário. Os operadores são aplicados nos conjuntos, garantindo que cada elemento ocorrerá somente uma vez. Por exemplo: $\{a, b^2\} \uplus \{b, C, g\} = \{a, b^3, c, g\}$.

Outras características das Redes de Petri, como efeito de ocorrência, redes elementares, definição de sequência, definição de conflitos etc., são descritas em Marranghello (2005), o trabalho estente uma visão aplicada em mineração de processos e identifica os trabalhos que contribuíram para a formulação do *framework* de Redes Petri no ProM. Já no trabalho de van der Aalst (2016), há definições sobre o uso de multiconjuntos, marcação dos estados, regras de ativação da marcação, além de gráfico de alcance.

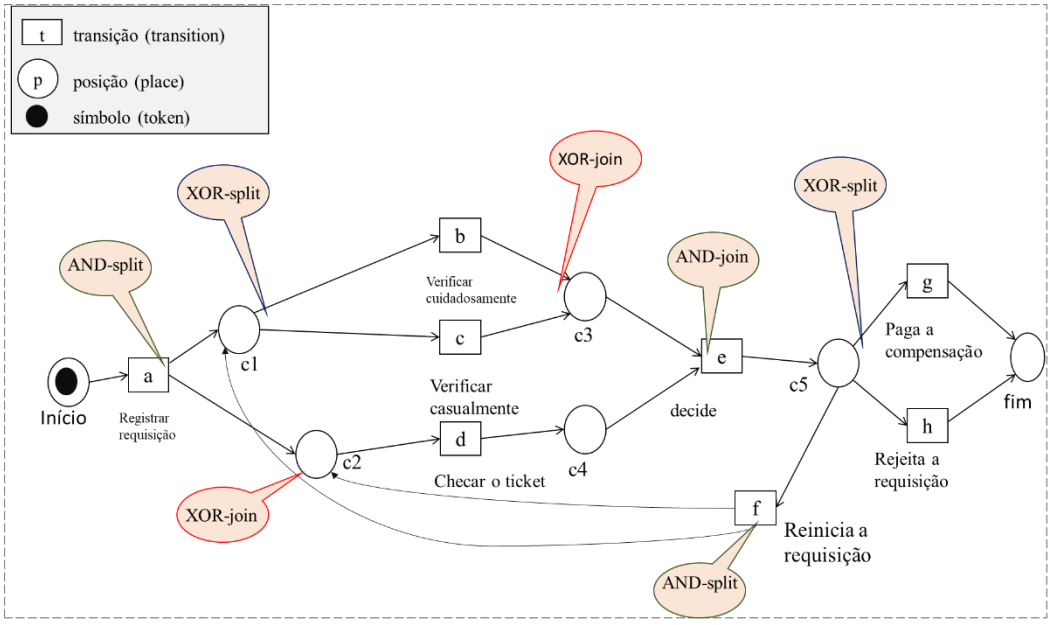


Figura 9 – Representação gráfica de uma Rede de Petri

Os conceitos fundamentais de Redes de Petri são comumente usados nos modelos de notação BPMN, EPCs, UML, entre outros, pois estes usam a semântica baseada em *tokens*. Isso ocorre devido ao fato de haver muitas ferramentas que possibilitam a transformação de Rede de Petri nesses modelos e vice-versa (VAN DER AALST, 2015).

Na Figura 10, tem-se a representação de três modelos (Modelo A, Modelo B e Modelo C), os quais foram gerados pela ferramenta ProM a partir do arquivo transformado para ". XES". Para criá-los, três *plugins* foram utilizados: Redes de Petri (Modelo A), BPMN (Modelo B) e EPC (Modelo C). Primeiramente, o Modelo A, que usa o *plugin* de Redes de Petri, foi criado e, em seguida, usado para transformar o Modelo A nos modelos: Modelo B e Modelo C.

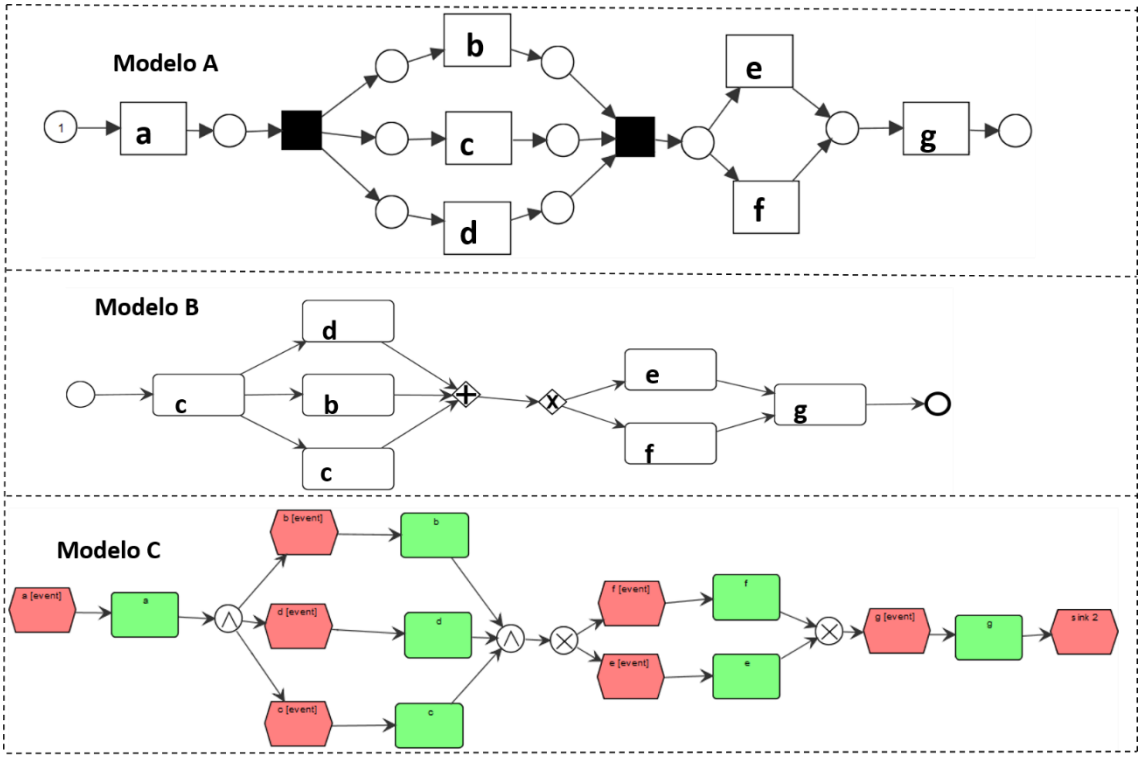


Figura 10 – Transformação Petri Net usando o ProM

3.6.2 Algoritmo Inductive Miner (IM)

Nesta subseção, introduziremos o algoritmo Inductive Miner (IM), um dos principais algoritmos de descoberta de processos. O IM usa a abordagem de divisão e

conquista, dividindo o *log* de eventos de forma recursiva em *sublogs* (LEEMANS *et al.* 2013).

Sendo assim, supondo um cenário em que existam dois grupos de atividades χ e γ, sendo o grupo χ precedido pelo grupo γ, mas nunca ao contrário, diz-se que esse grupo está em uma relação sequencial. O *log* de eventos será dividido em dois grupos e novas escolhas serão feitas considerando concorrências e laços, até que reste apenas uma atividade.

O IM é um dos poucos algoritmos de descoberta de processo que garantem a solidez e a adequação, além de produzir o modelo do processo considerando todo o *log* de eventos (LEEMANS *et al.* 2013). O algoritmo IM trabalha de forma a fazer o *split* no *log* de eventos; em seguida, detecta o operador que descreve o *split* e continua o processo no *sublog*.

Na literatura, existem outros algoritmos de descoberta do processo. Um exemplo é o *α-algorithm* (VAN DER AALST *et al*. 2004), um dos primeiros algoritmos de descoberta, que impulsionou os demais. De acordo com LEEMANS *et al.* (2013), o IM foi capaz de identificar transições no modelo onde o α-algorithm falhou.

O IM consiste na abordagem que considera o sequencial das atividades do processo em que o sequencial retorna uma árvore específica, além de produzir o modelo do processo considerando todo o *log* de eventos.

De acordo com van der Aalst (2016), as técnicas cuja abordagem trabalha com árvores são mais robustas e auxiliam na resolução de anomalias como:

- *Deadlock*: travamentos em determinados estados em que outras ações ou grupos ficam esperando uma resposta de um processo que foi bloqueado.
- *Livelock:* similar ao *Deadlock,* com a diferença de que o estado do travamento muda constantemente.

Deadlock e *Livelock* são problemas comuns em modelos de Redes de Petri, WF-nets, BPMN, EPCs, YAWL e diagrama de atividades da UML. Uma característica do Inductive Miner é o fato de conseguir produzir um modelo de processo com características de todo o *log* de eventos. Isso ocorre devido ao IM trabalhar com modelo estruturado em blocos; além disso, os modelos tendem a ser mais simples (LEEMANS *et al.* 2013). Assumindo que o algoritmo é determinístico, uma árvore $\boldsymbol{P}$ é avaliada como

redescoberta se segue o *log* de eventos ***E*** tal que ***P***, ***E*** **(IM(E)) = M(P)**, se para cada atividade de ***P*** apresentada no *log* com atividade inicial e final em ***P*** existir um *trace* iniciando e finalizando.

Nas árvores de decisão, além dos operadores de concorrência $(\wedge)$ e loop $(\circlearrowleft)$, os operadores de sequência $(\rightarrow)$ e exclusividade (x) podem ser inseridos.

A Figura 11 demonstra os dois processos, os quais são representados por: $Zrd = \circlearrowleft\left(\wedge(a,b),\wedge(c,d)\right)$ e $Zpar = \wedge(\circlearrowleft(a,c),\circlearrowleft(b,d)$, que representam comportamentos distintos, mas o direcionamento e a sequência para os gráficos são idênticos tanto para o loop $(\circlearrowleft,\{a,b\},\{c,d\})$ quanto para o paralelismo $(\wedge,\{a,b\},\{c,d\})$. Esse tipo de abordagem não permite distinção dos dois processos, mas van der Aalst *et al.* (2016) descrevem que, para a maioria dos casos, o gráfico de segmento direto é suficiente e garante a aptidão. A aptidão é um dos indicadores altamente representativos na literatura como forma de avaliar um algoritmo de descoberta e modelagem de processo.

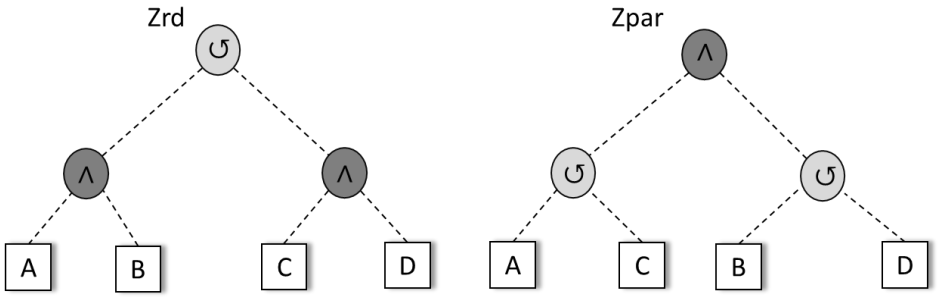


Figura 11 – Árvores de processo

Nesse exemplo, o IM seleciona o operador paralelo da expressão $(\wedge,\{a,b\},\{c,d\})$ e retorna o Zpar, que representa os *traces* [a,b,c,a] e [a,c,b,d,b], os quais não são possíveis em Zrd.

Um ponto negativo do IM é que ele pode gerar modelos com *underfitting* se o *log* de eventos não possuir atividades repetidas. O problema de *underfitting* generaliza modelos, criando mais comportamentos do que o *log* efetivamente possui (VAN DER AALST *et al.* 2010).

Além do *underfitting*, outros problemas, como *overfitting*, podem afetar o modelo gerado. Algumas variações acopladas à técnica de IM, vastamente estudada na literatura, são apresentadas na Tabela 5.

O IM é uma das três técnicas definidas em Leemans *et al*. (2014) que trabalham com a descoberta de processos; as outras duas são definidas como IMi-C e ILP-c e estão inseridas no *plugin* de mineração de processos ILP (Integer Linear Programming) (VAN DER AALST, 2016). A abordagem que consiste na otimização de problema é baseada diretamente em frequência (directly-follows). Cada uma dessas técnicas possui um conjunto de características as quais são definidas como:

- **IM :** caracterizado como rápido, garante a solidez e a aptidão do processo, além de permitir filtragem de ruídos.
- **ILP:** permite semântica, disponibiliza ótima adequação e precisão, mas não garante solidez do modelo.
- **DF:** não fornecem semântica e não trabalham com paralelismo, são rápidos e permitem filtragem.

Tabela 5 – Variações do IM

Algoritmo variação	Características
IMF: *frequency*	Considera a frequência das atividades do *log* e tem por objetivo demonstrar o comportamento principal.
IMC: *incompleteness*	Trata o comportamento excepcional de completude do *log* através das relações de atividade probabilística.
IMD: *directly-follows based*	Variações para o IM, IMF e IMC para melhorar a acurácia.
IMFD: *infrequent – directly-follows based*	Variações para o IM, IMF e IMC para melhorar a acurácia.
IMCD: *incompleteness – directly-follows based*	Variações para o IM, IMF e IMC para melhorar a acurácia.

3.6.3 Petri Net With Inductive Miner

Internamente, o IM trabalha com árvore de decisão e não com Rede de Petri. A Rede de Petri é formada a partir de representação de blocos, e, a partir dos operadores, os blocos são definidos. Na Figura 12, o *trace* de {1...7} é representado no topo, como árvore, e os operadores concorrência (∧) e exclusividade (x) são representados nos blocos da Rede de Petri, na parte inferior da figura.

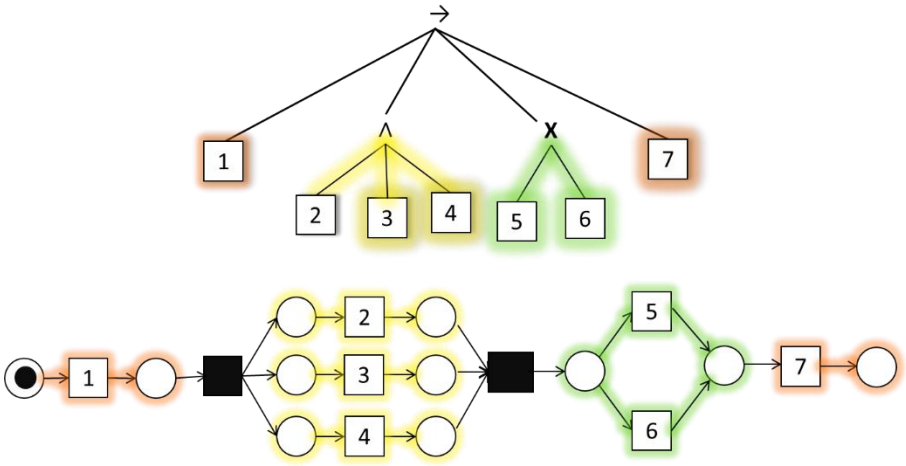


Figura 12 – Árvore de decisão e Petri Net com o mesmo *trace*

3.7 REPLAY A LOG ON PETRI NET FOR CONFORMANCE ANALYSIS

Neste item, falaremos um pouco sobre o *plugin* usado para avaliar a conformidade em mineração de processos através do modelo Petri Net.

A verificação de conformidade é uma das subdivisões da área de mineração de processos que avalia formas de medir a conformidade entre o *log* de eventos e o modelo do processo, sendo importante para o alinhamento do negócio e auditoria. Existem quatro dimensões em que a área de mineração de processo contribui para mensurar a qualidade; elas são representadas na Figura 13. A aptidão mede os desvios no *log* de eventos de acordo com o modelo, logo, é uma das principais métricas para verificação de conformidade (VAN DER AALST *et al.* 2016).

A aptidão trabalha de forma a verificar o alinhamento entre o *log* e o modelo. Diz-se que o modelo está perfeitamente apto se o modelo e o *trace* do *log* estiverem em perfeita equalização.

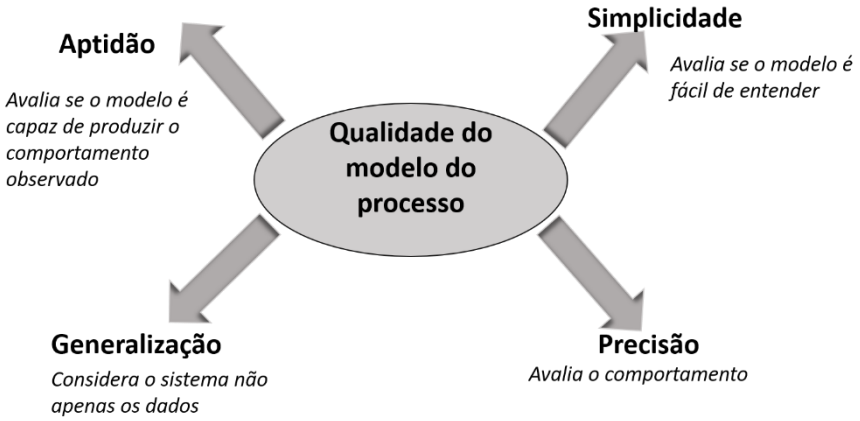


Figura 13 – Dimensões de qualidade

Para mensurar o alinhamento entre o modelo e o *log* em mineração de processos, os desvios são capturados para que seja calculada a aptidão entre ambos. Os movimentos são medidos a fim de capturar os desvios. Para ilustrar, considere o modelo da Figura 8, parte 3, e os seguintes *traces*: t1={ a,b,c,d,e,g }; t2= {a,b,c,e,g} e t3 ={a,b,c,d,e,f,g}. Isso produz o resultado da Tabela 6, em que **P1** possui um alinhamento perfeito entre o *log* e o modelo, ou seja, as atividades de t1estão de acordo tanto com o *trace* quanto com o *log*. Diferentemente de **P2** e **P3,** que apresentam desvios de acordo com t2 e t3. Em **P2**, a atividade "d" só existe no *trace*, enquanto em **P3** a atividade "f" só existe no modelo. Em resumo, **P1**, na Tabela 6, representa 100% de aptidão entre o modelo e o *trace*, enquanto **P2** e **P3** contêm desvios.

Tabela 6 – Alinhamento entre trace e modelo

P1	**Trace**	a	b	c	d	e	g	
	Modelo	a	b	c	d	e	g	
P2	**Trace**	a	b	c	>>	e	g	
	Modelo	a	b	c	d	e	g	
P3	**Trace**	a	b	c	d	e	>>	g
	Modelo	a	b	c	d	e	f	g

Os dois tipos de desvios observados na Tabela 6 são penalizados, sendo calculados custos para ambos. Considere-se apenas o *trace* {a,b,c,d,e,g} para fins de exemplificação.

Esse *trace*, com o modelo da Figura 14, gera o resultado dos *traces* e do modelo na Tabela 7. Todos os caminhos foram representados; dependendo do caminho a ser seguido, a comparação é feita entre o modelo e o *log*, e os eventos são descritos. Quando este não está de acordo, ou seja, não bate, o símbolo ">>" é inserido para esse evento, seja para o modelo, seja para o *trace*. Feito isso, verifica-se que, no *log*, temos as atividades "a" e "g" para todos os P, enquanto a atividade "c" somente aparece ou no *log* ou no *trace*. Isso depende do caminho que seguirá no modelo, sendo possível, após o "a", seguir três opções que se encotram ilustradas na Figura 14 , as quais geram os eventos que são descritos na Tabela 7.

Tabela 7 – *Log* e *trace*

P11	**Trace**	a	b	c	d	>>	e	>>	g	Custo total =4	Replay fitness = 0.67
	Modelo	a	b	>>	d	c	>>	f	g		1- $\frac{4}{6+6}$ = 0.67
P22	**Trace**	a	>>	b	c	d	e	>>	g	Custo total =4	Replay fitness
	Modelo	a	c	b	>>	d	>>	f	g		1 - $\frac{4}{6+6}$ = 0.67
P33	**Trace**	a	>>	b	c	d	e	g		Custo total =2	Replay fitness
	Modelo	a	c	b	>>	d	e	g			1 - $\frac{2}{6+6}$ = 0.83

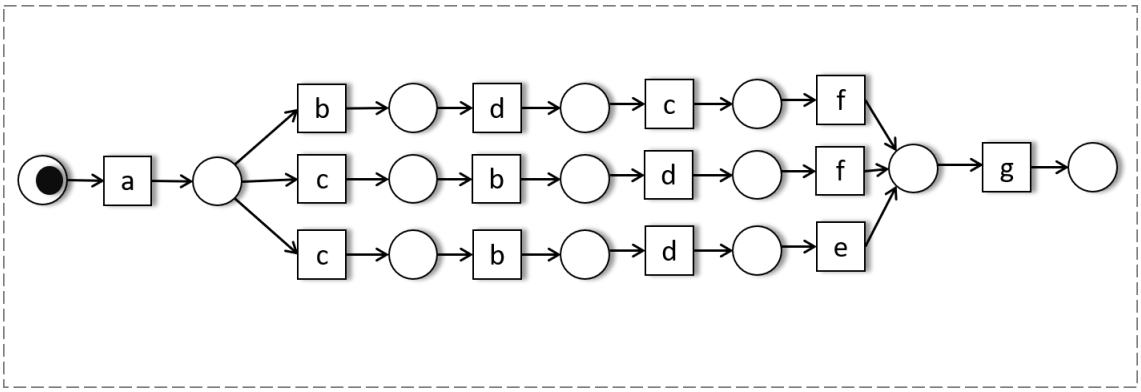


Figura 14 – Modelo Petri Net com seis atividades

Na Tabela 7, em **P11**, o custo do movimento do *log* é igual a 6 para o *trace* {a,b,c,d,e,g} e o custo do modelo também é 6 para esse exemplo, totalizando um custo de movimento igual a 12. O procedimento ocorre para **P22** e **P33**, sendo o valor total de movimento igual para os três.

Porém, o custo total entre o modelo e o *log* não permite avaliar o sincronismo entre ambos. Para isso, considera-se o alinhamento entre o *log* e o modelo, o qual é calculado de acordo com custo do movimento entre eles (log e modelo). No exemplo dado, o custo calculado entre o modelo e os *traces* de *log* foi dado por a1=4; b1=4; c1=2.

Para calcular a aptidão, é necessário normalizar o custo do alinhamento para valores entre 0 (zero) e 1 (um), em que 1 é o melhor resultado e 0 é o pior resultado. O pior resultado é dado quando não há nenhum sincronismo entre o modelo e o *log*. Na primeira tabela, temos o custo = 4, então 1-4 / 6+6 = 0.67; na segunda tabela, temos o custo = 4, então 1-4 / 6+6 = 0.67; na terceira tabela temos o custo = 2, então 1-2 / 6+6 = 0.83.

Na definição de van der Aalst et at. (2016), o cálculo da aptidão (*fitness*) do *log* com o modelo considera os *tokens* consumidos e os *tokens* perdidos na execução do *log* de eventos. Para exemplificar, considere p_N, σ como o número de *tokens* da execução do *log* σ em *N*, sendo $c_N, \sigma, m_N, \sigma, r_N, \sigma$ representados por m_N, σ e que representa o número de *tokens* perdidos quando se executa σ em N. A definição da aptidão do *log* de eventos *L* no WF-NET *N* é definida, na Fórmula 4, por:

$$fitness(L, N) = \frac{1}{2}\left(1 - \frac{\Sigma\sigma\epsilon L^{\,L(\sigma)} \times m_N, \sigma}{\Sigma\sigma\epsilon L^{\,L(\sigma)} \times c_N, \sigma}\right) + \frac{1}{2}\left(1 - \frac{\Sigma\sigma\epsilon L^{\,L(\sigma)} \times r_N, \sigma}{\Sigma\sigma\epsilon L^{\,L(\sigma)} \times p_N, \sigma}\right)$$

Fórmula 4 – Cálculo da aptidão no WF-NET

Onde $\Sigma\sigma\epsilon L^{\,L(\sigma)} \times m_N, \sigma$ = Total de *tokens* perdidos ao reproduzir o *log* de eventos calculado na entrada da leitura, sendo $L(\sigma)$ a frequência do *trace* σ e m_N, σ o número de *tokens* perdidos para cada instância de σ. O valor da aptidão está entre 0 e 1, em que 0 é o pior resultado, ou seja, em caso de valor zero, nenhum *token* é consumido e todos os *tokens* produzidos foram perdidos. Por outro lado, o valor 1 indica a aptidão perfeita, em que todos os *tokens* podem ser reproduzidos. A aptidão (L, N) é focada nos *tokens* locais, e, por conveniência será considerada como a medida dos eventos.

A medida de 80% de aptidão é considerada por van der Aalst (2011a) uma boa métrica. Nessa definição, se considerarmos o exemplo da Tabela 7, apenas o *trace* P33 está com uma métrica boa e, na média de (P11+ P22+ P33), soma menos de 73%, indicando uma métrica de aptidão do modelo com o *log* abaixo do esperado na definição.

3.8 MINERAÇÃO/TRATAMENTO DE TEXTO

Neste trabalho, usamos técnicas de tratamento de texto para padronização do *log* de eventos, colocando-o no formato adequado ou requerido para a mineração de processos. Consideramos o mapeamento das categorias de acordo com o processo levantado.

Neste item, descreveremos os métodos e ferramentas usadas para tratamento de texto, as quais foram utilizadas neste trabalho.

3.8.1 Transformação e classificação do texto

O objetivo da mineração de processos é descobrir o que de fato está sendo executado na visão do dado. A descoberta do modelo deriva do *log* de eventos. Para possibilitar a geração do modelo a partir do *log*, uma estrutura de padronização é requerida. O desafio aumenta quando há necessidade de tratar o texto a fim de mapear e descobrir as atividades, categorizando-as para modelar os caminhos e, assim, permitir a geração do modelo quando o *log* de dados não é estruturado ou semiestruturado.

Nesse contexto, as técnicas de classificação de texto auxiliam na estrutura necessária. Com o uso das técnicas de classificação de documentos de linguagem natural, é possível categorizar e rotular as tarefas requeridas para padronização do *log* de dados. Para Lan *et al*. (2009), o processo de classificação de texto é crucial para permitir o reconhecimento pelo computador possibilitando que a máquina classifique o texto.

Um documento de texto é composto por uma coleção de palavras que podem ser representadas por um modelo de espaço vetorial ou um saco de palavras (*bag of words*). Nos experimentos executados neste trabalho, primeiramente efetuou-se o tratamento do texto para melhorar a adequação do tageamento das palavras; em seguida, o texto foi transferido para um vetor de palavras.

3.8.2 Extração de informação

A decomposição do texto é uma tarefa que possibilita a extração de conteúdo para formatar os atributos em vetores. Uma das formas mais comuns de decomposição do texto é em palavras (WITTEN *et al.* 2016). Parágrafos, sentenças, frases e atributos numéricos são outras possibilidades.

A extração de texto requer algumas ações que auxiliam os métodos a trabalharem com o texto de forma equalizada. Por exemplo:

- Tokenização: transforma o texto em termos.
    - Exemplos de termos => [“cancelamento”, “de”, “matricula”,” fora”, “do”,” prazo”]
- Normalização: remoção de acentos, pontos, números, adequação do texto em letras minúsculas etc.
- Remoção de *stop words*: remover as preposições, os pronomes e as palavras sem significado relevante no contexto.
- *Stemming*: elimina o plural das palavras ou equaliza os termos de acordo com a raiz gramatical.

Uma forma de equalizar a tokenização do texto é através do *word count*. A frequência das palavras pode ser usada para avaliar aquelas mais frequentes, modelando seu uso ou a sua exclusão do contexto. Inicialmente, no nosso estudo, usamos o *word count* para efetuar o agrupamento das palavras, o que ajudou a identificar no texto as palavras com erros de escrita. Foi feita uma correção no texto de origem antes de aplicar os algoritmos de classificação.

3.8.3 Classificação do texto com TF-IDF

Frequência vem do termo em inglês *term frequency* (TF). Uma medida amplamente utilizada na recuperação da informação é o TF-IDF (WITTEN *et al*. 2016). A medida depende de quão comum uma determinada palavra aparece em um determinado documento, ou seja, verifica-se o quanto o texto é frequente de acordo com o resto do

documento. A medida do TF-IDF é tipicamente representada na Fórmula 5, em que o TF conta o número de vezes que uma palavra aparece em um documento; na Fórmula 5, o TF é multiplicado pelo IDF, retornando um termo t.

$$idf(termo) = log\left(\frac{Nº\ de\ documentos}{Nº\ de\ documentos\ com\ o\ termo}\right)$$

Fórmula 5 – Frequência do termo – Frequência inversa do termo

A Tabela 8 mostra um exemplo de como a frequência de cada classe aparece no texto. Os dados são de três processos da classe "autorizar cursar menos de 6 créditos", cada coluna representa um vetor e nela aparece a frequência da palavra usada em cada linha do texto para um determinado processo.

Tabela 8 – Frequência e termos

CLASSE e ID	requerimento	de	inscricao	em	menos	seis	créditos
1:1	1	2	1	2	1	1	1
1:2	0	1	1	0	1	1	1
1:3	2	1	0	0	1	1	1

O TF-IDF é um algoritmo eficiente para encontrar palavras que estão em uma coleção de documentos, mas há limitações para trabalhar com os sinônimos ou plurais de palavras (RAMOS, 2003). Em experimento efetuado, identificou-se que as palavras "drug" e "drugs" foram inseridas em duas categorias.

3.8.4 Weka data mining

Weka (*Waikato Environment for Knowledge Analysis)* ou (Waikato Ambiente para Análise de Conhecimento) é uma ferramenta de mineração de dados desenvolvida na linguagem Java mantida por um projeto de software livre na universidade de Waikato. O Weka incorpora uma coleção de algoritmos de aprendizado de máquina dentre outros.

A coleção de algoritmos permite pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização.

Com a coleção de algoritmos que a ferramenta Weka possui, há muitas possiblidades de tratamento de dados. Witten *et al.* (2016) destacam:

- Uso de modelos de aprendizado para descobrir mais sobre os dados.
- Possibilidade de prever novas instâncias com a aplicação de modelos de aprendizado.
- Aplicação de comparações entre modelos diferentes, comparação de desempenho e uso das métricas para gerar novos modelos de previsão.

O kit de ferramentas do Weka é parametrizável, o que facilita a escolha e a configuração de algoritmos; a suíte do Weka possibilita, ainda, a implementação de algoritmos próprios integrando com os algoritmos existentes na ferramenta. A Figura 15 mostra a tela de *input* principal da ferramenta ao utilizar um conjunto de dados. Após o *input*, muitas opções parametrizáveis são disponibilizadas.

As configurações e customizações da ferramenta têm por objetivo ajudar a descobrir mais sobre os dados tratados, o que é um problema do KDD vastamente discutido na área de mineração de dados. A ferramenta Weka possui algoritmos de aprendizado supervisionado e não supervisionado. Nos próximos tópicos, falaremos mais sobre alguns algoritmos utilizados neste trabalho e suas respectivas configurações, como o KNN (K-Nearest Neighbor Algorithm), a árvore de decisão e o Naive Bayes.

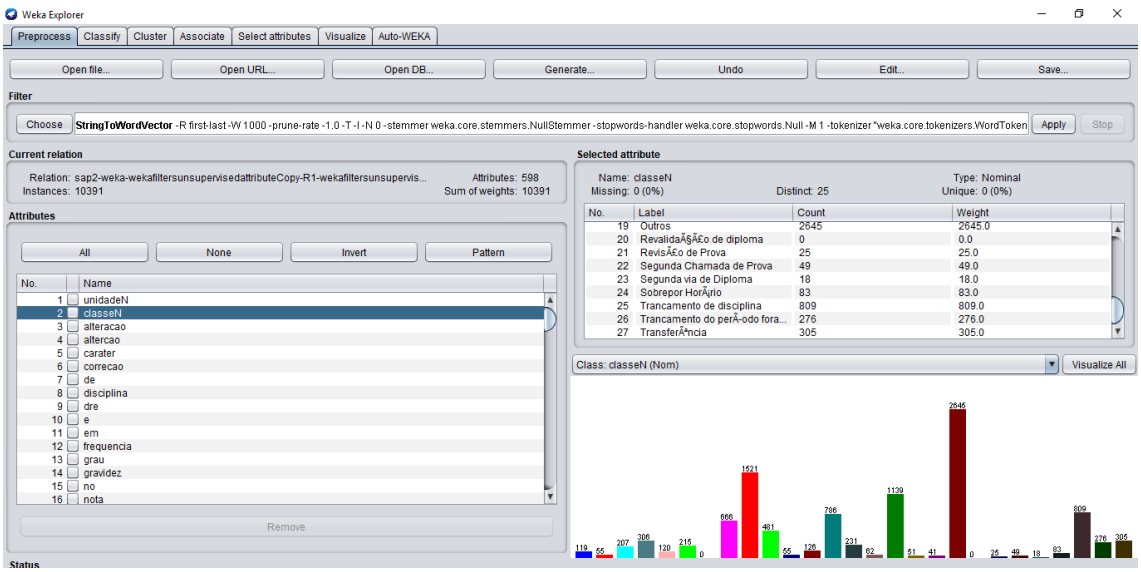


Figura 15 – Tela inicial do Weka para *input* de dados

Há muitas possibilidades de parametrizações na suíte do Weka no que diz respeito aos parâmetros de configuração, às características de importação e transformação dos dados e às possibilidades de filtragem. Mais informações podem ser obtidas no link[10] do projeto, que recebe contribuição de diversas áreas de estudo por ser uma ferramenta de código aberto.

Ainda sobre as possibilidades de visualização dos dados, a Tabela 9 apresenta um exemplo de como de saída dos termos e das frequências de uma determinada classe. No exemplo tem-se o agrupamento da classe por unidade e a frequência dos termos que nela ocorre.

Tabela 9 – Exemplo de planilha com TF-IDF do Weka

unidadeN	classeN	alteracao	altercao	carater	correcao	de
36000000	Outros	0.0	1.0	0.0	1.0	1.0
36000000	Transferência	0.0	0.0	0.0	0.0	1.0
36020000	Inscrição em disciplina fora do prazo	0.0	0.0	0.0	0.0	0.0
36020000	Estágio fora do prazo	1.0	0.0	0.0	1.0	1.0
36000000	Transferência	0.0	0.0	0.0	0.0	1.0
36010000	Dispensa de Disciplina	0.0	0.0	0.0	0.0	1.0

3.8.4.1 Configuração de parâmetros

A configuração dos parâmetros é um processo importante. As opções escolhidas para o tratamento do texto, como configuração de lista de palavras, *stop word*, entre outras, são etapas a serem definidas. A Tabela 11 e a Tabela 12 apresentam as configurações utilizadas no Weka e no *scikit-learn*, nas etapas de pré-processamento e classificação do texto.

3.8.5 Rapidminer

[10] Disponível em: <https://www.cs.waikato.ac.nz/ml/Weka/index.html>.

O Rapidminer é uma ferramenta de mineração de dados que possui uma suíte de diversos algoritmos de aprendizado de máquina e mineração de dados. A plataforma, apesar de ser comercializada, possui uma plataforma gratuita. Recentemente, o *plugin* rapidProM[11] foi inserido em sua suíte.

O rapidProM possibilita a análise de *workflows* científicos para mineração de processos (BOLT *et al.* 2016) com a construção do processo efetuado em blocos, em que são necessários:

- *Input* de eventos em formatos .csv, ou .xes.
- Construção do modelo (EPC, BPMN, Petri Net).
- Análise da conformidade do modelo construído.
- Métricas são demonstradas.

Além de o Rapidminer trabalhar com modelos complexos de mineração de processos, sua suíte permite, ainda, que eles sejam usados e reusados. Isso beneficia a área de mineração de processos e possibilita treinar os modelos para, a partir destes, criar outros novos (VAN DER AALST *et al.* 2017). A Figura 16 ilustra a arquitetura da ferramenta. Nem todos os *plugins* do ProM encontram-se disponíveis, mas os de modelagem mais usados e o de avaliação da conformidade (*conformance checking*) estão acoplados e disponíveis.

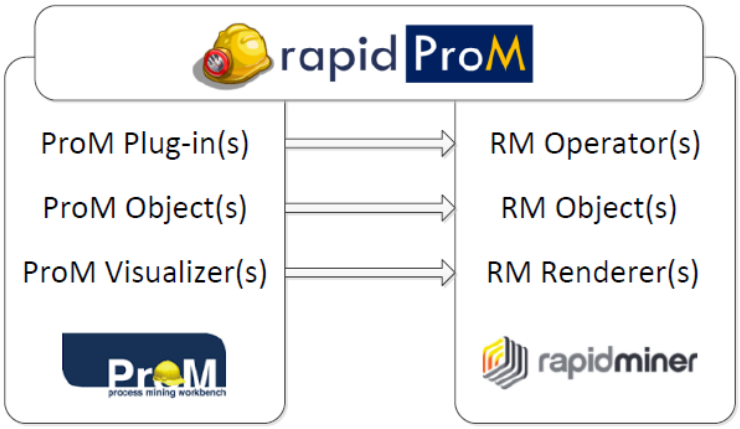


Figura 16 – Arquitetura do rapidProM

[11] http://www.rapidprom.org/

3.8.6 Python scikit-learn

Biblioteca de aprendizado de máquina da linguagem Python, o *scikit-learn* possui diversos algoritmos de classificação regressão e agrupamento. Os algoritmos de seleção de modelos oferecem esquemas de validação cruzada, entre eles o k-folder padrão, o k-folder estratificado e o *leave-one-out* (BUITINCK *et al.* 2013).

A biblioteca permite ainda efetuar o *split* de dados com combinação de treino e teste gerados a partir da ótica da validação cruzada. Com as funcionalidades GridSearchCV e RandomizedSearchCV, ajusta-se o conjunto de dados de treinamento e o desempenho é avaliado no conjunto de validação.

O conjunto de técnicas da biblioteca possui configurações de pré-processamento, incluindo o TF-IDF, que extrai *features* de um texto. A etapa de pré-processamento também inclui a formatação e normalização dos dados para vetores numéricos.

3.8.7 Algoritmos para avaliação do texto

Para avaliar a classificação do texto, três algoritmos foram testados e as métricas foram usadas para melhorar a adequação das classes com o uso dos algoritmos de classificação, definidos como classificadores. Aplicando o modelo ou uma função aprendida, a classificação rotula automaticamente novas instâncias identificadas em um conjunto de dados. O modelo identifica e baseia a saída no modelo de treinamento. Diversos algoritmos classificadores são apresentados na literatura e podem ser definidos como árvores de decisão (MONARD; BARANAUSKAS, 2003), baseados em regras (BLUM; MITCHELL, 1998), redes neurais artificiais ou modelos probabilísticos (bayesianos). Na classificação do texto e categorização das classes dos documentos utilizados a partir de um conjunto de documentos que definem os tipos de processos, incluindo unidades às quais eles pertencem, três modelos populares de aprendizado de máquina foram testadas. Os classificadores são:

- **K-Nearest Neighbors (KNN):** algoritmo classificador que trabalha com o objetivo de rotular a classificação de uma amostra a partir do vizinho mais próximo. Para a sua implementação, requerem-se o valor do *K* e a distância.

$$D = \sqrt{(x_1 + y_1)^2 + \cdots + (x_n + y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Fórmula 6 – Distância euclidiana

Na Fórmula 6, a distância de X = (x1, ..., xn) e Y = (y1, ... , yn) são dois pontos n-dimensionais representados na Figura 17, que mostra uma nova entrada a ser classificada definida na nuvem em cor preta. A distância a ser calculada possui o valor em x, e y considera os pontos verde e azul.

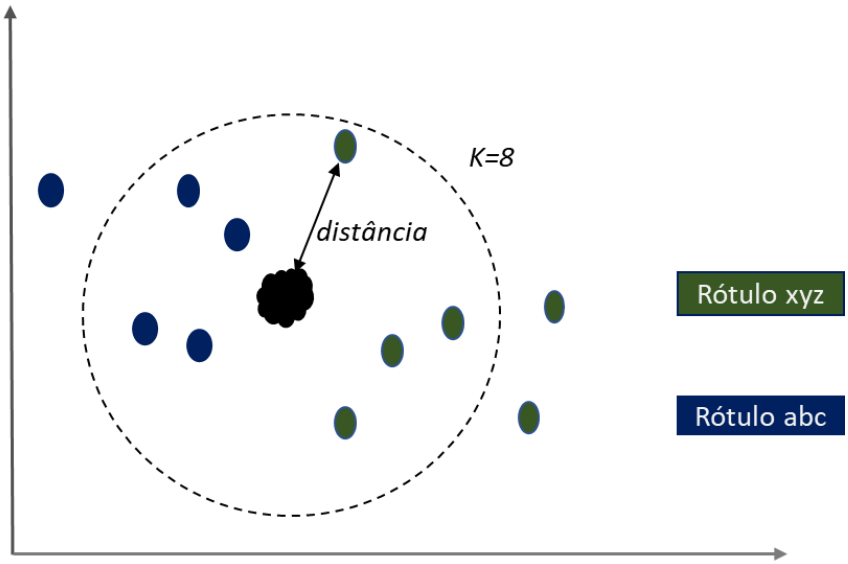


Figura 17 – Exemplo de classificação com dois rótulos de classe e k=8

Para casos em que o conjunto de dados é grande, o KNN procura os padrões de K mais próximos em todo o espaço (KRAMER, 2013). É necessário definir a medida de similaridade.

- **Naive Bayes**: frequentemente usado em tarefas que resolvem problemas de classificação de texto, o algoritmo Naive Bayes é um classificador bayesiano capaz de prever estatisticamente a probabilidade de um registro pertencer a uma determinada classe. O principal foco do algoritmo Naive Bayes é fornecer a base para algoritmos que se baseiam em probabilidades em relação a um conjunto de dados, mas sua aplicação também se estende para algoritmos que não trabalham explicitamente com probabilidade. Uma das características é o fato de possuir

dependência naive em relação a outras *features*, ou seja, o classificador parte do principio que os atributos são independentes.

A probabilidade processada pelo Naive Bayes é dada por $P(c|d)$ para que um documento seja pertencente a uma determinada classe a partir da probabilidade a priori da classe dada por $P(c)$. A probabilidade a priori de $P(c)$ define se o documento é da classe a partir das probabilidades condicionais dadas por $P(t_k|c)$ se o termo t_k ocorrer em um documento da mesma classe. A probabilidade a priori representa a probabilidade em D para a classificação de uma determinada classe da base de dados possuir os termos "reprovação" e "exclusão". O vocabulário é extraído a partir da função de treinamento, obtendo-se o grupo de documentos D, que representa a probabilidade total. A partir disso, um vetor de probabilidades a priori divide o número de documentos de cada classe; então, o Naive Bayes faz o trabalho de classificação, estimando o $P(c|d)$ para todas as classes existentes na base de dados e rotulando-as na categoria mais provável. Ao ganho computacional do Naive Bayes, é atribuída a função $P(D|C)$.

$$P(C|D) = \frac{P(D|C)\ P(C)}{P(D)}$$

Fórmula 7 – Teorema de Bayes

Na Fórmula 7, o Teorema de Bayes considera que $P(C\ e\ D = P(\ D\ e\ C)$, em que $P(D)$ determina a probabilidade total. A probabilidade a priori é dada por $P(C)$, e essa probabilidade é modificada pelo experimento. A probabilidade a posteriori é representada por $P(C|D)$, também chamada de nível de crença definida após a realização do experimento.

- **Árvores de decisão (AD)**: árvore de decisão é um tipo de classificador com desempenho geralmente bom que possui grande apatia de uso na literatura pelo fato de seu funcionamento interno ser fácil de entender, já que apresenta decisões relacionadas aos atributos dos itens analisados. É um classificador que utiliza do paradigma *bottom-up*, cujas características são: 1) os termos pertencentes a um objeto são representados por uma coleção fixa e 2) o número de classes pode ser definida a priori. A classificação a priori possibilita a modelagem da classificação,

tanto em treinamento supervisionado quanto em treinamento não supervisionado. O treinamento não supervisionado ocorre quando há a definição do modelo de treinamento de forma automática, usando método de indução. Os métodos de indução trabalham com a associação de uma hipótese baseada em outra a partir de bias indutivo.

A Figura 18 mostra uma árvore de decisão com estrutura de:

- Folha(s) que indicam uma classe;
- nó(s) de decisão, que representam a(s) estrutura(s) que define(m) algum tipo de teste sobre atributo, valor específico ou subconjunto de atributos, que podem ser determinados como ramos e subárvores para cada um dos valores possíveis de teste.

O nó de decisão inicializa no processo raiz. Na Figura 18, o nó de decisão pode ser representado por “aplicações” ou “saldo em conta corrente”; na sequência, cada resultado do teste de decisão é contemplado iniciando-se pela raiz da subárvore. Várias árvores distintas podem ser geradas a partir de um conjunto de dados. Dependendo do nó escolhido, o caminho pode ficar mais extenso.

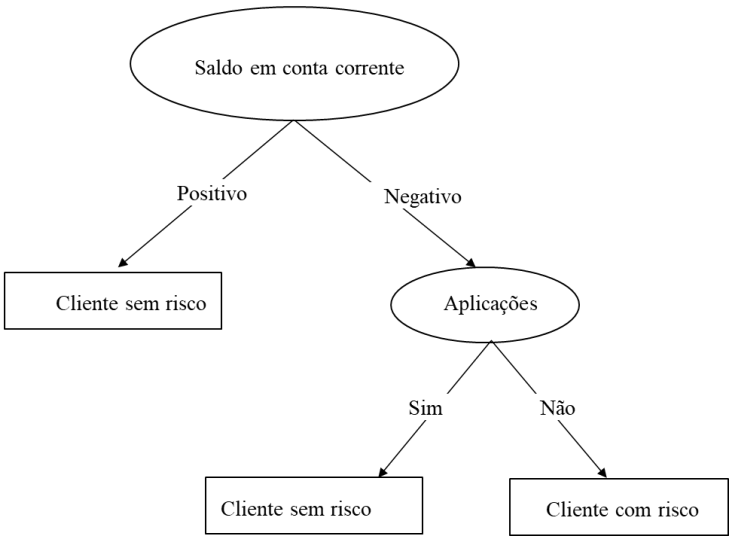


Figura 18 – Exemplo de uma árvore de decisão

Algoritmos como CART e C4.5 são exemplos de algoritmos de indução baseados em árvore de decisão (DRAZIN; MONTAG, 2012).

Quando o conjunto de dados é muito grande, a construção da árvore percorrendo todos os nós da mesma classe pode ser um processo complexo e, assim, gerar um conjunto preditivo ruim. Nesse caso, o resultado pode ser obtido através de uso de *outliers* e ruído pelo fato de o algoritmo memorizar uma determinada classe, e, com isso, pode gerar um resultado preditivo ruim. O ideal é que o modelo possibilite a generalização das amostras de treinamento para uma função subjacente que possibilite respostas razoáveis para novas entradas (REED, 1993).

Para o contexto da utilização de *outliers* e ruído, as árvores podem ser podadas. E nesse contexto, pode-se usar a abordagem de pré-poda (*prepruning*) que possibilita a interrupção da construção da árvore. Essa possibilidade requer a atribuição de limites quanto as características de profundidade da árvore. Outra técnica definida é a pós-remoção (*postpruning*), que substitui as subárvores por ramificações, trabalhando de baixo para cima da árvore e calculando a probabilidade dos nós folhas. Os nós folhas são comparados entre os nós irmãos e a diferença é podada. O erro estimado de cada nó filho é calculado e usado para derivar o erro total do nó pai; o nó pai é, então, removido de acordo com a frequência relativa dos nós filhos. A Fórmula 8 mostra a equação do cálculo da estimativa de erro, em que E representa o erro estimado, e são os exemplos classificados erradamente dado um determinado nó, N são os exemplos que alcançam o nó determinado e m representa todos os exemplos do conjunto de treinamento.

$$E = \frac{e + 1}{N + m}$$

Fórmula 8 – Cálculo da estimativa de erros

Conforme citado, as árvores de decisão possuem grande utilização nos métodos de classificação, portanto, sua implementação está disponível em diversas ferramentas, como Weka (DRAZIN; MONTAG, 2012), ProM, Rapidminer (BHARGAVA *et al.* 2013) e na biblioteca do *scikit-learn* do Python (BUITINCK *et al.* 2013).

3.8.8 Matrix de confusão

A matriz de confusão é produzida pelo *output* do classificador, e demonstra os valores reais e preditos de um algoritmo de classificação.

O detalhamento da matriz de confusão permite uma visualização detalhada de como o classificador se comportou na classificação dos dados (HAN *et al*. 2011). O *output* mostra com detalhes o reconhecimento de diferentes tuplas. A Tabela 10 demonstra um exemplo do resultado da matriz de confusão para três classes. Para cada uma das classes, temos o valor da classe atual e o valor da predição da classe. Para a classe "Alteração de Grau", existem 10 tuplas classificadas como "Alteração de Grau", as quais são também classificadas na predição, representando, portanto, 100% da classificação. Para a classe "Cancelamento de Matricula", o classificador indicou na predição uma classe como pertencente à de "Alteração de Grau". Na terceira classe do exemplo, representada por "Inclusão de Grau", o classificador indicou na predição que quatro classes pertencem à "Alteração de Grau".

Tabela 10 – Exemplo de matriz de confusão

Classe Atual	Predição da Classe: Alteração de grau	Cancelamento de Matrícula	Inclusão de Grau
Alteração de Grau	**10**	0	0
Cancelamento de Matrícula	1	**12**	0
Inclusão de Grau	4	0	**40**

Na diagonal da matriz de confusão da Tabela 10, são destacados em negrito os valores classificados corretamente para cada classe.

4 PROPOSTA DE AVALIAÇÃO DA CONFORMIDADE

Nossa abordagem para avaliação da conformidade do *log* de eventos em mineração de processos engloba dois momentos: o primeiro é o alinhamento do *log* não estruturado para adequação do *input* necessário; o segundo é a avaliação dos modelos produzidos por esse *log* que foi tratado. O nosso objetivo principal é mensurar a saída dos modelos gerados, uma vez que usamos técnicas de mineração e adequação de dados para tratar as entradas.

Neste capítulo, descrevemos a abordagem utilizada para analisar os dados e, posteriormente, criar os modelos e avaliar a conformidade do *log* com o modelo gerado.

4.1 TRATAMENTO DOS DADOS/ADEQUAÇÃO DE ENTRADA

Como visto no item 3.4, a estrutura requerida no ProM para a mineração de processos é um *log* de eventos estruturado. Outras ferramentas, como Disco (GÜNTHER; ROZINAT, 2012), Celonis e Minit (VAN DER AALST, 2016) também possuem esses requisitos de *log* estruturado, seguindo o padrão de uma instância e os traços (*traces*) definidos. De fato, nem todas as aplicações apresentam o padrão requerido, e, quando os sistemas permitem o *input* de texto de forma livre pelo usuário, esse problema se torna mais abrangente, principalmente se há necessidade de identificar instâncias e classes não normalizadas.

Considerando essa necessidade, para tratar o texto e classificá-lo, usamos os três algoritmos definidos no item 3.8.8. A abordagem demonstrou que existiam mais classes na estrutura de dados do que as levantadas e pré-classificadas no levantamento da necessidade. O modelo proposto para o tratamento e adequação do *log*, possui as etapas de tratamento de dados apresentadas na figura 19. A representação mostra a perspectiva a partir do tratamento dos dados e, considera a adequação do *log* de eventos para o formato de *input* requerido para a mineração de processos. Inicialmente, o tratamento dos

dados considera um *input* de classes abstraídas a partir da visão do usuário, visão obtida a partir da efetivação de um diagnóstico. Após a fase de diagnóstico; atuamos na classificação dos dados baseados nas classes. A geração do *output* foi usada para assimilar a necessidade de melhoria na classificação que atua sobre o texto que classifica a classe.

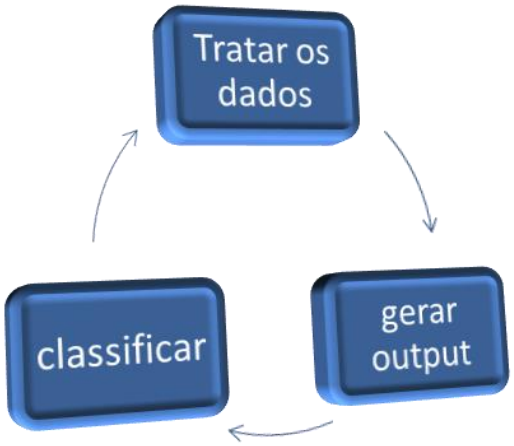


Figura 19 – Etapas do tratamento do texto

Os trabalhos de mineração de processos definem amplamente a aproximação da área de mineração de processos com a mineração de dados, mas, para Bolt *et al.* (2016), os resultados da mineração de processos tendem a ser muito diferentes dos resultados das técnicas clássicas de mineração de dados. Sua justificativa gira em torno das perspectivas do processo: a assimilação de classes, por exemplo, revisita a necessidade de junção de técnicas para adequação dos parâmetros. Para van der Aalst *et al*. (2017), o grande *gap* entre as técnicas de mineração de dados e de mineração de processos é o fato de a mineração de dados não contemplar processos de ponta a ponta. A área de mineração de processos também é vista como uma forma de agregar os modelos de mineração de dados com o aprendizado de máquina (VAN DER AALST *et al*. 2016).

4.2 AVALIAÇÃO DA CONFORMIDADE APLICADA SOBRE A VALIDAÇÃO CRUZADA

Neste item, discutiremos o modelo proposto a fim de avaliar a conformidade para um *log* de eventos não estruturado. A necessidade da formatação do *log* de eventos foi

tratada com os métodos descritos no capítulo 3 deste trabalho; nosso objetivo é fazer a avaliação dos modelos gerados a partir de então.

Propomos a utilização do método estatístico de validação cruzada (*cross validation*) para medir a aptidão do modelo gerado com o *log* de eventos. Nossa abordagem considera um *log* de eventos não estruturado, logo, esta técnica poderá mostrar ganhos. Como o processo de tratamento de dados passou por etapas aprofundadas de tratamento, a avaliação da conformidade do modelo gerado com o *log* poderá trazer ganho e *insights* relevantes.

4.2.1 Validação cruzada

Diversos trabalhos descrevem as características da validação cruzada na análise de dados. Em Refaeilzadeh *et al.* (2008), os autores descrevem que na mineração de dados, assim como na aprendizagem de máquina, a validação cruzada mais usada é o k=10. Já os autores Brink *et al.* (2016) apontam que as escolhas mais comuns para o k são os valores 5,10 e 20.

O *dataset* de treino T é dividido em k subconjuntos $T^{(1)},\ldots,T^{(K)}$ com tamanhos iguais. Cada um dos conjuntos de $T^{(i)}$. O pedaço do conjunto de dados que não é usado para treinamento será o conjunto de dados usado para teste, ou seja, cada $T^{(i)}$ é usado como conjunto de testes. Por fim, será calculado o erro do modelo, o qual considera cada iteração realizada e calcula quantas classificações incorretas foram feitas no conjunto de casos de teste.

Por fim, serão calculados o erro do modelo e a estimativa total do erro. A estimativa do erro do modelo considera cada iteração realizada e calcula quantas classificações incorretas foram feitas no conjunto de casos de teste; já a estimativa total do erro do modelo é dada pela soma das estimativas obtidas em cada iteração dividida por K.

Dada a característica do método estatístico, usamos a abordagem para auxiliar na escolha da melhor opção do modelo para a generalização de cada classe. Como insumos para o modelo, foram usados os dados classificados no $dataset^{++}$ composto pelos campos {processo + classe + data}. O "processo" é a *feature* que representa a chave principal na tabela original. A "classe" é a *feature* classificada a partir do resumo descrito

do item. A *feature* “data” identifica a data de abertura do processo. A ordem em que a solicitação foi feita é identificada no campo “data” e é importante para identificar a sequência das atividades.

A Figura 20 demonstra a característica da validação cruzada com os k subconjuntos do $dataset^{++}$. O subconjunto T gera o modelo dado pelo algoritmo do ProM definido em 3.7; o modelo representa o teste.

A característica da validação cruzada usa, ao final da avaliação, a média dos resultados obtidos em cada etapa, o que, no modelo proposto, é representado na saída da Figura 6. As métricas apontadas irão propor a estimativa do melhor modelo.

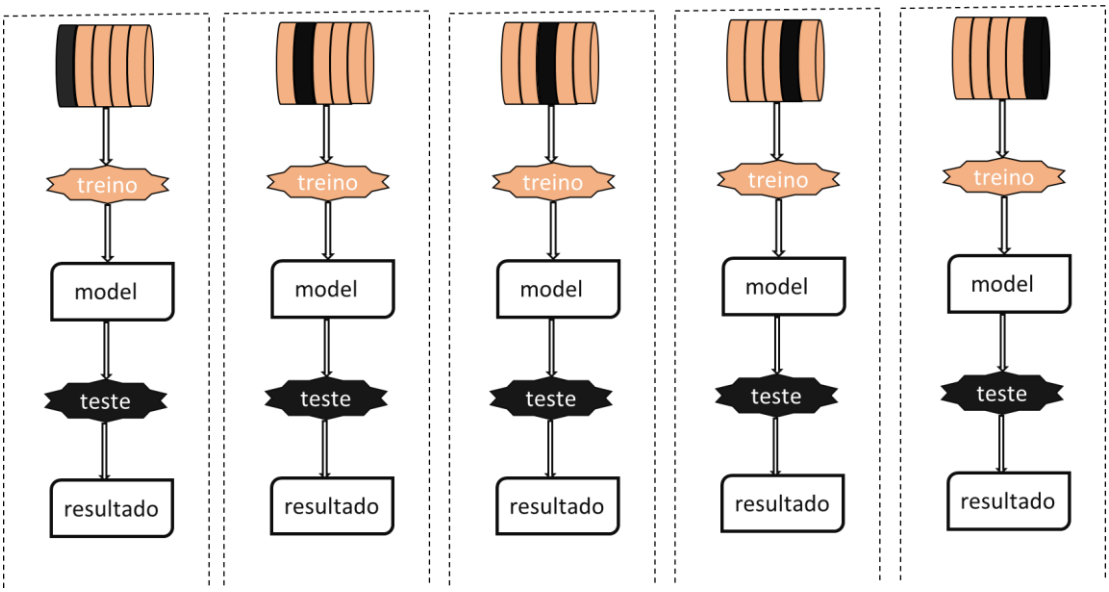


Figura 20 – Processo de 5 vezes a validação cruzada

Supondo a avaliação do primeiro item da Figura 20, temos o resultado no experimento demonstrado na Figura 21, em que o *p1* é a parte dos dados usados para gerar o modelo em Petri Net e representa o conjunto de dados de teste com 20% do *dataset* total, e os subconjuntos *p2, p3, p4* e *p5* representam a parte do conjunto de dados que será usada para treino e equivale a 80% do *dataset* total.

O processo será repetido até que $\pi - 1$ seja utilizado como treinamento e a outra parte como teste. Esse processo será repetido π vezes até que cada parte seja usada uma vez como conjunto de teste. O *dataset* de treino e teste será composto pelos os seguintes conjuntos:

- **C1** = Avaliação do modelo p1 com o resto do *log* = {p2, p3, p4, p5} → modelo Petri Net de p1= {p1};
- **C2** = Avaliação do modelo p2 com o resto do *log* = {p1, p3, p4, p5} → modelo Petri Net de p2 = {p2};
- **C3** = Avaliação do modelo p3 com o resto do *log* = {p2, p1, p4, p5} → modelo Petri Net de p3 = {p3};
- **C4** = Avaliação do modelo p4 com o resto do *log* = {p2, p3, p1, p5} → modelo Petri Net de p4 = {p4};
- **C5** = Avaliação do modelo p5 com o resto do *log* = {p2, p3, p4, p1} → modelo Petri Net de p5 = {p5}.

No final, para cada classe, serão obtidos 5 modelos que disponibilizarão um grupo de métricas para cada modelo, as quais serão usadas para avaliar o melhor modelo.

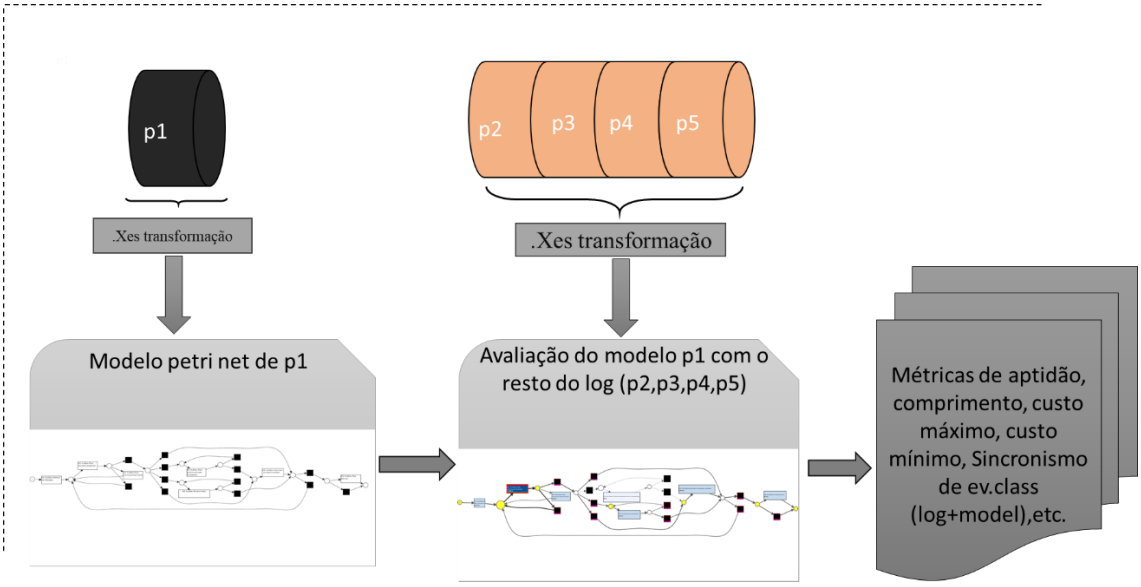


Figura 21 – Geração do modelo de cada subconjunto

4.2.2 Aplicação da validação cruzada no modelo

Utiliza-se a abordagem do método estatístico de validação cruzada para avaliar as medidas de precisão e aptidão (*fitness* e *precision*) do modelo em relação ao *log* de

eventos. Aplica-se a abordagem proposta em cada classe cujo modelo é inicialmente criado e, em seguida, faz-se a validação em relação ao restante do *log* de eventos. Para identificar medidas de cada parte, usamos os seguintes *plugins* implementados no ProM:

- Mine Petri net With Inductive Miner;
- Replay a Log on Petri Net for Conformance Analysis.

As métricas do ProM citadas são as métricas de saída apontadas no modelo da Figura 21. As características da implementação das métricas foram definidas no item 3.7.

4.2.3 Métricas

Para criar o modelo com a parte do *log* de cada *dataset* do k-folder, usamos o *plugin* do ProM *Mine Petri net with Inductive Miner*. A mineração indutiva usa internamente a estrutura de árvores – sua definição foi introduzida no item 3.6. Os *plugins* utilizados do ProM produzem as métricas:

- *NumOfCases*: agrupa o número de *cases* de cada *trace*, ou seja, aqueles que possuem o mesmo caminho;
- *Trace length*: contabiliza o caminho a ser percorrido pelo *log* (conta total de atividades);
- *Max move log cost*: tamanho do *trace*;
- *Max fitness cost*: tamanho do *trace* +1;
- *Move model fitness*: calcula a aptidão do *log* em relação aos movimentos do *trace*;
- *Trace* fitness: contabiliza a aptidão do *log*.

Extraindo essas métricas de cada modelo *k-1* com os 20% do *log* e aplicando à validação ao restante de 80% do *log*, obtêm-se as métricas para cada modelo.

4.3 AVALIAÇÃO EXPERIMENTAL DO MODELO

Neste subcapitulo, detalhamos como foi aplicada a implementação sugerida sobre a avaliação da conformidade usando a abordagem de mineração de processos. A evolução do experimento neste trabalho está dividida em duas partes. Na primeira parte, tratamos a classificação do texto e a obtenção das classes. O objetivo da classificação das classes contempla também a tratativa e a classificação do texto. Na segunda parte, com as classes definidas, usamos a abordagem da validação cruzada para cada classe, separando-as em cinco *folders* que, em seguida, geram o modelo para avaliação das métricas da conformidade através de *plugin* do ProM. Antes de entrarmos no experimento, discutimos o formato geral dos dados e as métricas de avaliação que foram usadas.

4.4 *DATASET* – TRATAMENTO DOS DADOS

Antes de aplicar o modelo proposto de validação cruzada em modelos de processos e avaliar as métricas de aptidão para mensurar a conformidade, consideramos o cenário de utilização do modelo para casos em que os dados não sejam estruturados, o modelo pode ser aplicado para dados estruturados.

Para a classificação e tratamento do texto, foram usadas as ferramentas Weka e Rapidminer. O objetivo foi a equalização de palavras do texto antes de usar o texto no classificador.

Foram efetuadas diversas rodadas de extração do texto com *split* em palavras para efetuar a análise do texto em formato de uma lista de palavras. Para cada palavra descrita de forma errada ou com abreviação, fez-se uma extração de todos os documentos em que tal palavra constava e uma leitura pré-análise, antes de fazer o *update* do texto. O processo se deu com o uso das ferramentas apresentadas na Figura 22. Em primeiro lugar, foi feita uma análise inicial dos documentos em relação à avaliação do texto; em seguida, o *dataset* foi atualizado; depois, uma nova extração foi efetuada e, a partir de então, os dados foram novamente inputados para avaliar as palavras e fazer a adequação ao texto.

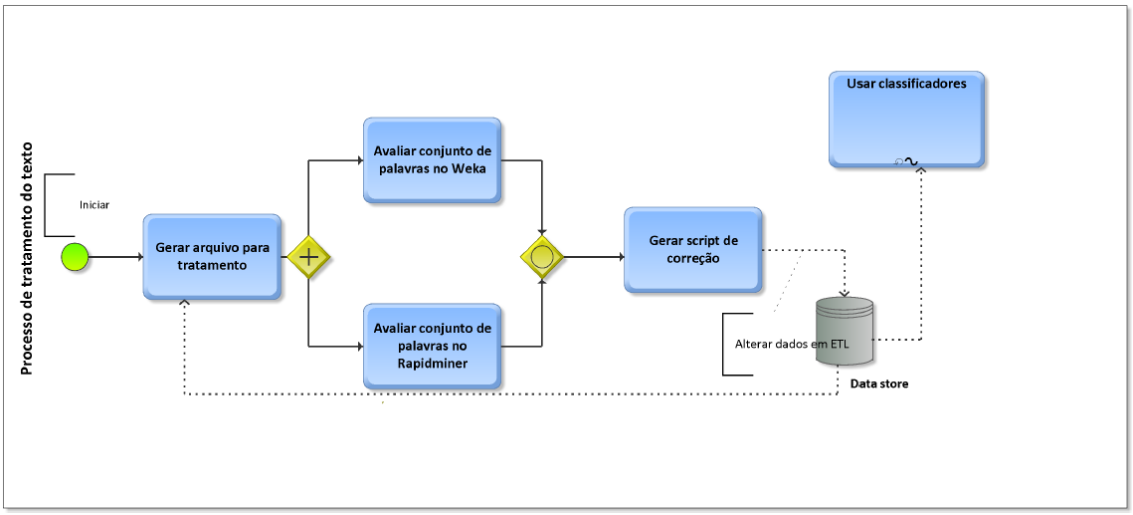


Figura 22 – Processo de seleção e tratamento do texto

Os agrupamentos das palavras para tratamento foi feito pelas ferramentas Weka, Rapidminer e biblioteca do Python. É importante ressaltar os seguintes pontos sobre essas ferramentas:

- Weka:

Há necessidade de transformar o texto para o padrãoreconhecido .arff. Com isso, o agrupamento é feito por meio da importação para o *dashboard* do Weka através do menu. A opção do processamento do *StringToWordVector* provém da configuração do filtro de algoritmo não supervisionado aplicada à lista de palavras. O filtro *StringToWordVector* converte um conjunto de atributo do tipo *string* para o tipo numérico, os quais representam as palavras no texto (BOUCKAERT *et al.* 2008).

O retorno do texto gera uma lista de atributos do tipo texto e numérico. Visualmente, isso facilita a identificação das palavras escritas de foma errada, com acentuação ou pontuação. A transformação do texto para o formato .arrf retorna diversas críticas de acentuação que foram tratadas até que a importação fosse realizada com sucesso.

Outras configurações, como *lowerCaseTokens*, e outros parâmetros apresentados na Tabela 11 foram escolhidos na configuração da primeira etapa de pré-processamento dos dados.

Em uma etapa posterior ao tratamento dos documentos, o texto foi utilizado no classificador dos algoritmos definidos no item 3.8.7 para avaliar as classes definidas que na Figura 22 estão representadas pelo sub-processo definido como “Usar classificadores”.

Tabela 11 – Lista de valores configurados no Weka

Parâmetro	**Valor**
Parâmetro	Valor
IDFTransform	*True*
Transforme	*True*
lowerCaseTokens	*True*
minTermFreq	1
normalizeDocLength	*Normalize all data*
outputWordCounts	*True*

- Rapidminer:

Na ferramenta Rapidminer, a sugestão é de uso do operador Process Documents From File (GUPTA; MALHOTRA, 2015), que gera um vetor de palavras a partir de uma coleção de texto. A primeira opção utilizada foi a tokenização, esta abordagem foi utilizada com o filtro de “*non letters*”, que remove os caracteres numéricos, formatando uma lista de palavras que utilizamos para analisar e processar as correções no texto. A lista de palavras foi utilizada em paralelo junto com a ferramenta Weka. A configuração utilizada no fluxo da tokenização considerou os parâmetros definidos na Tabela 12. A saída do fluxo utilizando esta configuração gera uma listagem de palavras como mostrada no Apêndice A, Figura 44, que mostra a saída do arquivo tokenizado. Na listagem, 12 tipos distintos de escrita para a palavra “transferência” são identificados no conjunto de 1.962 ocorrências que compõe todos os documentos, sendo 1.014 ocorrências com a escrita correta. Em média, 52% das ocorrências no texto estavam preenchidas de forma completa e correta de acordo com a palavra transferência.

Tabela 12 – Lista de valores configurados no Rapidminer

Operador	Valor
Process Documents from Files Text Processing	*Input* da lista com os documentos
Tokenize Text Processing	*non letters*
Tokenize Text Processing	*linguistic tokens*
Filter Stopwords (English) Text Processing	-
Filter Stopwords (French) Text Processing	-
Stem (Porter) Text Proce	-
Transform Cases Text Processing	-
outputWordCounts	*True*

- *Scikit-learn* Python:

O texto com as classes pré-definidas com a unidade e o campo texto tratado foi utilizado na biblioteca do *scikit-learn* com os classificadores definidos no item 3.8.6 com a configuração de:

  - TF-IDF;
  - *cross validation*;
  - *dataframe*;
  - *split* do *dataset* de treino em 30%;
  - *split* do *dataset* de teste em 70%;
  - parâmetro de *cross validation* = 5.

4.4.1 *DATASET* – Escolha das features e seleção inicial das classes

O *dataset* utilizado neste experimento foi obtido de uma base de dados semiestruturada de abertura de processo e encaminhamento de solicitação de aprovação. Os processos são provenientes de solicitações feitas para requerer determinadas ações referentes a áreas específicas – no caso do *dataset* utilizado, à área acadêmica, em que

solicitações de notas, aprovação, exclusão de disciplina, entre outras, são feitas por alunos e os pedidos são encaminhados para as devidas centrais acadêmicas responsáveis.

Para o experimento, foi escolhido um período de corte de data, visto que o *dataset* possui armazenamento de muitos anos. Além do corte de data, consideramos somente os processos em que a tramitação possuía pelo menos uma de duas atividades **"A19 Para Arquivar" OU "A64 Arquivado na própria Unidade"**, as quais indicam o encerramento do processo.

A coleção de *features* dos processos para a abertura é composta pelos campos "Número", "interessado", "unidade", "assunto", "data", "resumo", "autuador" e "órgão". A coleção de *features* dos processos para a tramitação é composta pelos campos "processo", "origem", "destino", "despacho" e "data".

As *features* da tramitação possuem todo o histórico dos processos abertos, e o texto a ser classificado encontra-se na *feature* "resumo da abertura do processo", na qual se descrevem a necessidade, o motivo e a solicitação realizada. O ponto de atenção desse item está relacionado à classificação da *feature* "resumo", que tem o mesmo código (0671-8) para todo o assunto definido como "Assuntos acadêmicos".

Para a escolha das *features*, utilizamos como base todos os processos abertos na unidade 36 CT – Centro de Tecnologia. A escolha foi feita com base no levantamento do caso de estudo feito junto com o gestor da área, dado que os processos levantados foram feitos a partir do entendimento desse centro. Em seguida, as três principais *features* escolhidas para classificação das classes foram "Processo", "resumo" e "unidade de abertura". Como o campo "resumo" é um campo de texto livre, diversas características são cadastradas, como número do processo (para alguns casos), palavras abreviadas, texto em inglês, em português, em francês e acentuação. Para equalizar os dados, antes da transformação do texto e outros procedimentos serem executados, seguimos os seguintes passos para tratamento do texto:

- Remoção de números;
- remoção de pontuação;
- padronização dos termos para letras minúsculas;
- remoção dos nomes das pessoas;
- remoção de palavras em inglês;
- correção de algumas palavras escritas de forma incorreta.

Para três classes específicas foi necessário um tratamento antes da remoção dos números. São elas:

- Autorizar Mais de 32 créditos;
- Autorizar Menos de 6 créditos;
- Cursar 1/3 de disciplinas fora do curso.

Para as três classes, inicialmente foram removidos os números do campo "resumo", mas, para esses casos específicos, os números definem a classe. Logo, uma segunda tratativa para ajustar foi adequá-los e mantê-los.

Ainda a partir da análise dos dados, algumas subunidades foram retiradas da base por apresentarem quantidade de dados insignificante para o modelo. O corte da base de dado foi executado após se avaliarem os processos abertos para toda a base de dados referente à unidade de estudo.

A classificação inicial do texto deu-se pela extração da categorização das classes com base no conhecimento do levantamento do processo, ou seja, para uma classe definida como "alteração de grau", inicialmente usamos esse entendimento para classificar os processos pertencentes a ela. Para todas as classes levantadas, conseguimos extrair classe para os processos. A extração da classificação inicial foi efetuada com o *script* da Fórmula 9.

```
UPDATE processo set classe = 'Alteração de Grau'
WHERE assunto = '0671-8' AND unidade like '%36%'
                        AND
        resumo like '%Alteração de Grau%';
```

Fórmula 9 – Classificação inicial das classes

4.4.2 Resultado dos classificadores

Com abordagem de correção das palavras, remoção de *stop words* e aplicação do classificador KNN no Weka com a configuração demonstrada na Tabela 11 com *split* em dois momentos – o primeiro de 66% para treino e o segundo de 70% –, obtivemos

resultado superior nos dois casos quando utilizamos o *dataset* com as palavras com o pré-processamento. Nos dois experimentos, foi utilizada a validação cruzada com k = 5.

Para ambos os momentos de avaliação, as métricas estão bem próximas, tanto em relação à porcentagem de *split* para a avaliação quanto em relação ao ganho. Ou seja, a diferença entre o uso de 66% ou 70% na aplicação do classificador sem o pré-processamento é de 0,04%; já na utilização do texto com pré-processamento, a diferença é de 0,06%. Os valores apresentados na Figura 23 representam o número de instâncias classificadas corretamente de acordo com o uso do classificador KNN.

Nos resultados apresentados no Apêndice A, comparando-se a medida do KNN nas ferramentas Weka e scikit-learn, os valores mais otimistas foram os obtidos na classificação feita no Weka. Porém, no que diz respeito à medida de precisão que representa o cálculo dos Verdadeiros Positivos (TP) / (Verdadeiros Positivos (TP) + Falsos positivos (FP)), após o pré-processamento do texto ela se manteve a mesma no algoritmo do Weka, enquanto, no scikit-learn, a medida demostrou-se mais equalizada em relação ao *recall* – que representa a medida que classifica os valores encontrados nas classes e também possui um valor menor. O *recall* representa o cálculo dos Verdadeiros positivos (TP) / verdadeiros positivos (TP) + Falsos Negativos (FN). Já a medida *F1-Score* ficou próxima do *recall* em ambas as plataformas, o *F1-score* que combina as medidas de precisão e *recall* como dissemos possuem valores distintos se olharmos as plataformas separadamente (weka e scikit-learn) mas na própria plataforma os valores são bastante próximos do *recall*. O *F1* tem como princípio retornar uma métrica única que possibilite uma visão da qualidade do modelo, sendo calculada por 2* (precisão * *recall*) / precisão + *recall*. As três medidas estão representadas na Figura 37 do Apêndice A, representando a classificação do *dataset* final para geração do *log* de eventos para a mineração dos modelos de acordo com as classes.

As métricas para o *Naive Bayes* são apresentadas na Figura 38 do Apêndice A. Para esse classificador, após o pré-processamento no campo “resumo”, apenas a medida de precisão ficou menor na plataforma Weka, já as métricas *recall* e *F1* ficaram mais otimistas, seguindo o mesmo padrão do classificador KNN.

O terceiro e último classificador, o de árvore de decisão (ou J48, como é definido no Weka), apresentou para ambas as plataformas métricas superiores a 96% de precision para o texto final, conforme mostra a Figura 38 do Apêndice A.

Usando da mesma abordagem definida em mineração de processos sobre a definição de que uma métrica boa para a aptidão é uma métrica de 80% para o *fitness*, nosso objetivo foi obter pelos classificadores métricas pelo menos a partir deste valor para o *recall*. Por isso, diversas rodas de avalição e revisão do texto foram executadas no experimento, avaliando-se o texto da classe recomendada e revendo a extração das *features* "resumo" e "classe".

O item 4.5.1 apresenta mais detalhes das métricas da classificação do texto, as medidas de cada classe e sua evolução. As classes aqui definidas serão utilizadas para gerar o modelo do processo aplicando as técnicas de mineração de processos.

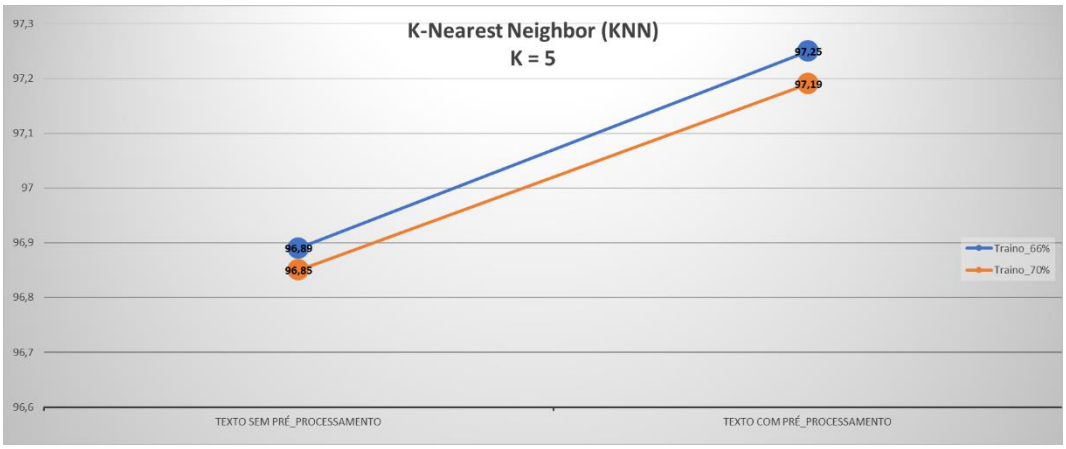


Figura 23 – Classificador KNN – Métricas

4.4.3 Experimento com os classificadores

O Apêndice C mostra a evolução dos experimentos sobre os dados. A classificação começou com 17 classes, de acordo com o escopo do estudo de caso levantado; em seguida, evoluiu para 31 e finalizou com 27 classes no total.

A evolução do experimento deu-se pela seguinte sequência:

- **Experimento I** – Weka KNN: As medidas de Precisão, *Recall* e *F-Measure* de cada classe estão demonstradas no Apêndice C, Tabela 24. O classificador foi o KNN na ferramenta Weka, e utilizou-se a validação cruzada com cinco *folders* em todo o *dataset*, com um total de 17 classes mapeadas de acordo com o escopo especificado e o modelo mapeado. Nesta análise, para seis classes não foi possível

obter pelo menos uma das métricas de avaliação. Em uma análise mais detalhada, observamos que as classes “Revalidação de diploma” e “Segunda via de Diploma” possuíam apenas dois processos em todo o *dataset* com a classificação, enquanto para a classe “Intercâmbio” não havia processo mapeado.

Experimento II – Weka KNN: Na evolução do experimento, após avaliação do resultado da matrix de confusão (*confusion matrix*) do Experimento I, foi feita a reavaliação do texto e a adequação das classes. Nova consulta foi efetuada com base na Fórmula 9 para incluir novos termos. Após a avaliação, novas classes foram mapeadas. A nova classificação resultou na identificação de 31 classes que estão representadas na

Tabela 25 do Apêndice V. Nesse segundo momento, foi possível verificar que apenas para duas classes não foi possível obter uma das métricas de avaliação, diferentemente do Experimento I, em que seis classes ficaram sem pelo menos uma das métricas.

Para as 31 classes, o algoritmo de classificação no Weka validou a escrita do agrupamento da classe e considerou a característica de *case sensitive*, classificando em classes distintas caso a classe estivesse escrita com letras maiúsculas e minúsculas. A escrita com número em vez de texto também foi considerada como classe distinta. A escrita errada na definição da classe “dem” no lugar de “de” foi outro ponto que duplicou a classe, e as métricas de classificação foram geradas de forma separada. Nesta validação, o agrupamento considerou classes distintas quando na verdade deveria ter sido considerado uma, para:

- “Trancamento de Disciplina” e “Trancamento de disciplina”;
- “Cursar 1/3 de disciplinas fora do curso” e “Cursar um terço de disciplinas fora do curso”;
- “Inscrição dem disciplina fora do prazo” e “Inscrição em disciplina fora do prazo”

- **Experimento III** – Classificação Weka e Python: Após diversas iterações seguindo o descrito nos experimentos I e II para a adequação das classes, finalmente as classes foram estabelecidas como 27 no total.

  As 27 classes foram utilizadas nos três classificadores propostos neste trabalho, e as métricas demonstram um bom resultado para a maioria delas. A Tabela 26, a Tabela 27 e a Tabela 28 do Apêndice C demonstram as métricas dos três algoritmos para cada classe, tanto no pacote *scikit-learn* do python quanto no Weka datamining tool kit.

  Para os três classificadores, a classe "Revalidação de diploma" não apresentou métricas. Isso ocorreu devido à falta de balanceamento dos *datasets* de treino e teste. Para essa classe, somente dois processos foram identificados. Como conclusão, foi identificado que, na base de dados, a classe pertence ao assunto de código '0069-8', definido como "Revalidação/Reconhecimento", diferentemente do assunto deste estudo, de código "0671-8", definido como "Assuntos acadêmicos".

  No que diz respeito às métricas de cada classe, podemos ainda observar, nas tabelas do Apêndice C, que:

  - O classificador Naive Bayes não classificou cinco classes no python, enquanto, no Weka, essas classes possuem métricas, ainda que baixas. Apenas a classe "Revalidação de diploma" não foi classificada em ambos (Scikit e Weka).
  - No classificador *Decison Tree*, apenas a classe "Revalidação de diploma" não foi classificada em ambos (Scikit e Weka).
  - O classificador KNN não classificou "Disciplina avulsa" e "Revalidação de diploma" em nenhum dos dois ambientes (Scikit e Weka).
  - A classe "inscrição em disciplina" possui a menor métrica para todos os classificadores.

Considerando o F1 acima de 90% nos dois ambientes (Scikit e Weka), foram mapeadas 21 classes no classificador KNN, 22 no *Decision Tree* e 18 no *Naive Bayes.* Essas classes serão utilizadas para gerar os modelos dos processos que serão apresentados no item Capítulo 44.5.

O gráfico da Figura 24 mostra em detalhes a métrica apresentada para a classe com o texto original sem o pré-processamento.

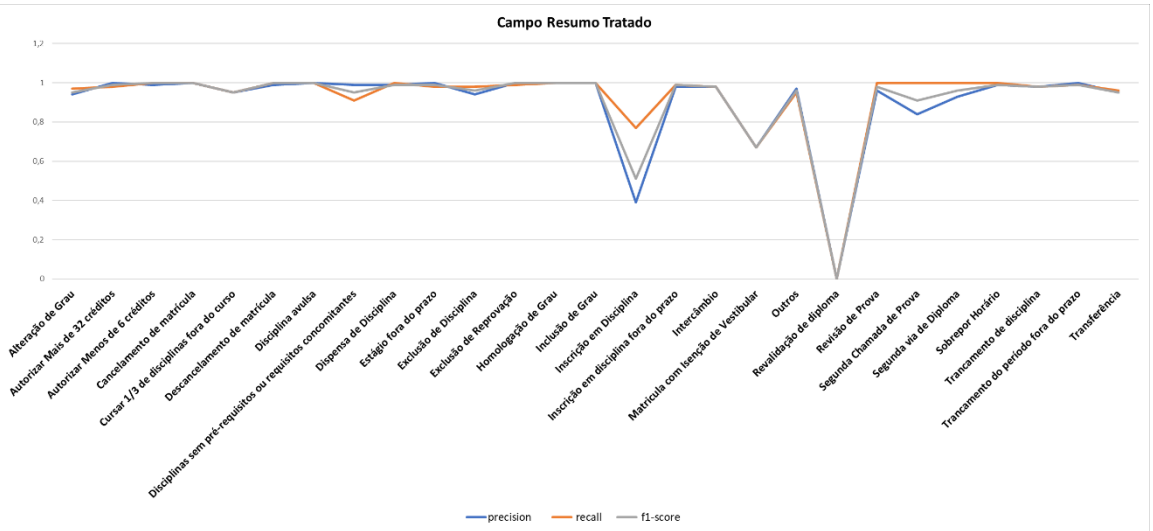


Figura 24– KNN *scikit-learn* – Aplicação sobre o resumo tratado

A Figura 25 mostra mais detalhes sobre a precisão calculada pelo KNN no scikit-learn, considerando todas as classes, o gráfico mostra as métricas sobrepondo o texto do resumo original com o texto do resumo tratado.

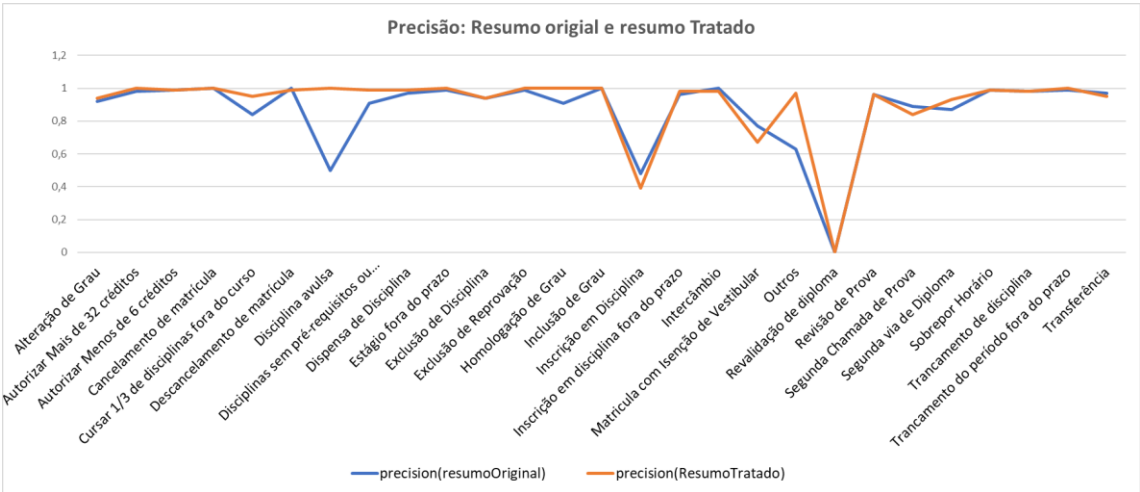


Figura 25 – Precisão das classes em relação ao resumo original e ao tratado

4.5 CLASSIFICAÇÃO DOS MODELOS

Neste item, descreveremos o processo de classificação dos modelos de processos gerados a partir dos dados que foram tratados sob a perspectiva do tópico 5.1 e com a aplicação da abordagem discutida no capítulo 4.

4.5.1 Análise inicial – Estudo de avaliação experimental

O direcionamento do uso da abordagem utilizando a validação cruzada sobre os modelos e sua separação ocorreu após diversos experimentos serem desenvolvidos na tentativa de mineração do *log* de forma geral e uso de técnicas que possibilitassem a verificação da conformidade entre o modelo entendido e o *log* de eventos.

Como havíamos detalhado o modelo existente, havia um direcionamento sobre a expectativa do modelo a ser gerado. Algumas questões essenciais sobre as unidades e os *times* dos trâmites que apresentavam no modelo foram algumas das primeiras iniciativas a serem olhadas no modelo gerado pelo *log.*

Ainda a fim de definir a melhor estratégia foram feitos experimentos utilizando o *plugin* do ProM, "*multi-perspective process explorer*", definido por Mannhardt *et al.* (2015), no entanto, dois pontos de visão impeditiva fizeram com que procurássemos outro direcionamento: a questão da leitura do *log* de eventos que, o *plugin* não suportou grande montante de eventos para processamento, gerando quebra de interface e também por não possuir a opção de exportação da métricas para avaliação. Alguns modelos gerados com parte do *log* da classe "sobrepor Horário" são apresentados na Figura 39 e na Figura 40. O *plugin petri net* também não demonstrou ganhos na avaliação do modelo entendido em relação ao modelo gerado, mesmo se utilizando apenas uma classe, como o mostrado na Figura 41. De uma forma mais abrangente, mesmo após a classificação do texto e da criação das classes com a utilização de todo o *log* de eventos, sempre obtivemos resultados mais voltados para modelos *spaghetti*, conforme mostra Figura 7.

Outro experimento feito foi em relação à concatenação da atividade com sua unidade de destino. Esse experimento foi feito utilizando o *plugin* net *With Inductive Miner*, da família do método definido no item 3.6.3. O experimento deu-se em razão de haver várias atividades que se repetem para a *feature* que descreve a tramitação. No

entanto, o modelo gerado apresentou-se de forma mais abrangente, tornando-o complexo e aproximando-o de um modelo do tipo *spaghetti* (a Figura 42 mostra o modelo gerado concatenando o número da unidade e a atividade para apenas um processo). Outros experimentos empreendidos na tentativa de identificar, dentro dos algoritmos propostos, a identificação da conformidade em relação ao modelo que o *log* gera *versus* o modelo mapeado utilizaram os seguintes *plugins*:

- Mine Petri net With Inductive Miner;

Mine Petri net With Inductive Miner – Mudando a opção para visualizar PETRI NET (DOT);

- *Convert Petri Net to* BPMN *diagram*;
- *Mine for Heuristic Net using Heuristic Miner / Convert Heuristic net into Flexible model*;
- *Convert c_ne*t to BPMN;
- *Replay a Log on Petri Net for conformance Analysis*.

Visando melhorar o entendimento dos modelos gerados pelo *log*, usamos a abordagem dos *plugins* que geram modelos em BPMN, apresentada por Kalenkova *et al*. (2014). Para melhorar o entendimento do que a ferramenta ProM proporciona, utilizamos um id de um protocolo (uma instância) para analisar os *plugins* que o ProM possui para a análise do modelo BPMN. Inicialmente, criamos o modelo BPMN de forma manual, representando todos os departamentos pelos quais ele tramita. A Figura 24 mostra o modelo criado de forma manual, utilizando instância de um processo de solicitação acadêmica. O protocolo possui uma instância completa de [aberto – arquivado] com 11 *traces*. O processo pertence à classe "Exclusão de reprovação". No modelo manual, desenhamos 6 atores que participam do processo:

- Escola politécnica;
- Secretaria acadêmica;
- Dep. de engenharia metalúrgica;
- Seção de Pessoal;
- Divisão de registro de estudantes (DRE);
- Centro de Tecnologia (CT).

Dos seis atores identificados, temos pelo menos um (“Seção de Pessoal”) que claramente não mapeamos no processo. A atividade de providenciar/atender também se repete seis vezes, inclusive na mesma unidade que abre o processo. Consistências no processo mapeado também são avaliadas em relação ao local de abertura e trâmite entre departamentos, exceto a seção de pessoal. No entanto, os modelos gerados na ferramenta ProM com *plugins* BPMN não demonstram com *insights* que possam ajudar a avaliar os dois modelos, um dos pontos que não conseguimos identificar no modelo são os atores envolvidos no processo, uma perspectiva que pode gerar *insights* de forma abrangente. Um exemplo é o *trace* mapeado que possibilitou identificar unidades que a tramitação passa e que não estava mapeada no processo. Na visão do processo gerado com o *plugin*, com o mesmo *trace*, não conseguimos obter a mesma visão de modelo conforme exemplificado na Figura 39.

Outras ferramentas, como Disco (GÜNTHER; ROZINAT, 2012), Celonis e Minit (VAN DER AALST, 2016), foram utilizadas sob liberação de licença acadêmica, por serem ferramentas não *open source*, no início da pesquisa, para analisar a questão da conformidade em relação à melhor perspectiva e identificação.

Após toda análise e testes feitos e, principalmente, após avaliar e verificar a perspectiva de modelos direcionados a serem complexos e tendenciosos a obtermos modelos *spaghetti*, houve a decisão de utilizar a abordagem de validação cruzada. Os experimentos serão apresentados no item Capítulo 44.6.

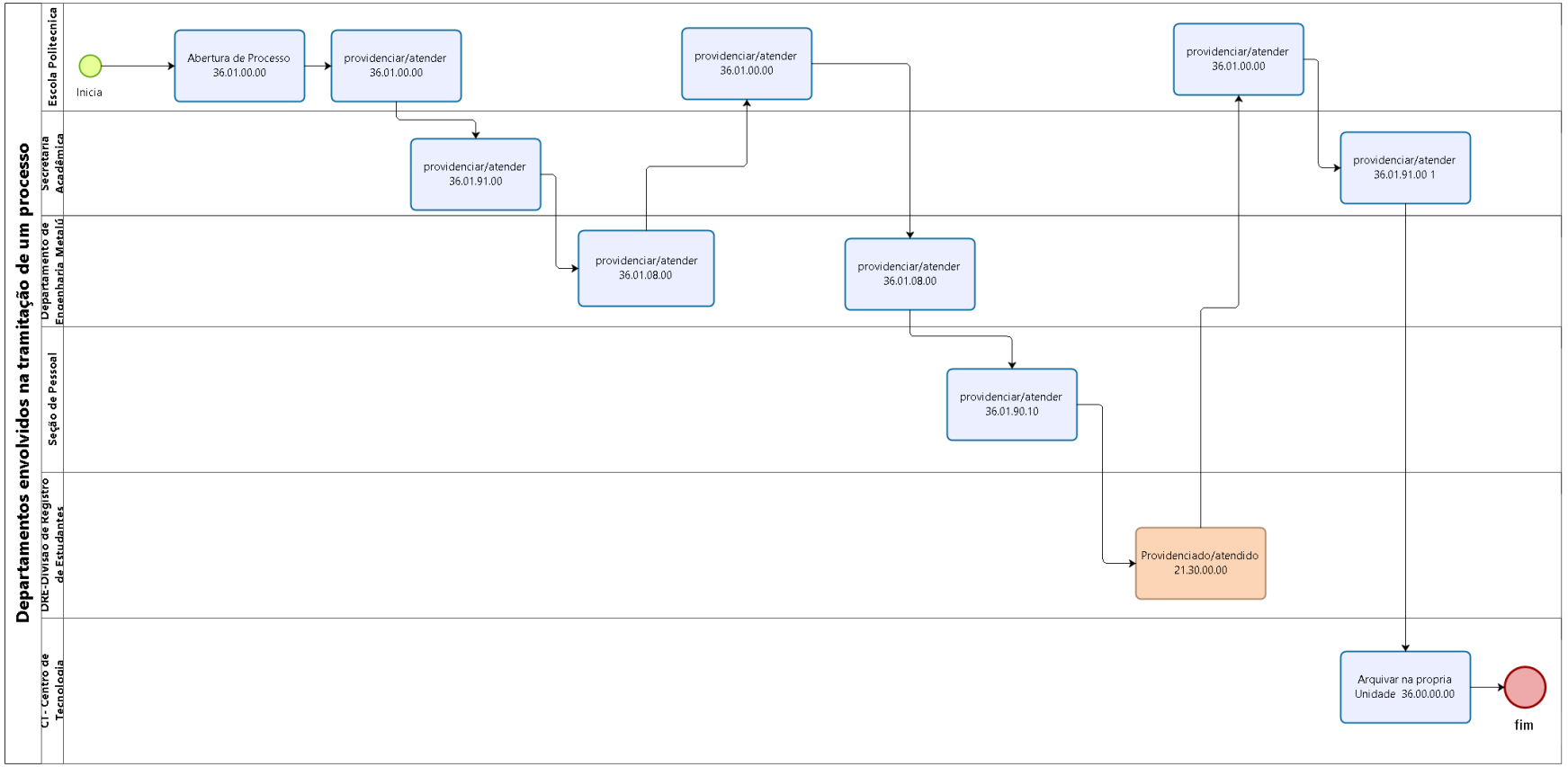


Figura 26 – Modelo BPMN de um processo

4.6 AVALIAÇÃO CRUZADA SOBRE OS MODELOS

O modelo da validação cruzada sobre o modelo proposto no item Capítulo 44.2 foi efetuada para cada classe do processo. A validação foi feita de forma que, para cada classe, foi gerado o particionamento dos dados da *feature* raiz definida pela tabela de processo e, em seguida, para cada *folder*, foi efetuada a busca do histórico dos processos. Nessa abordagem, garantimos que no caso de uma classe possuir 100 processos, estes foram divididos por uma função randômica do python em cinco *folders*. Nessa abordagem, ao final da montagem do *log* de cada *folder*, o quantitativo de linhas do histórico muda, uma vez que o número de tramitação do processo não é equalizado, podendo variar de 8 a mais de 50 tramitações, em casos mais raros. A tramitação representa o histórico do processo. Cada processo é uma instância e cada tramitação do histórico representa o traço (*trace*) do *log* de eventos.

Nesse particionamento das classes, o número de instâncias em cada *folder* é feito em dois passos:

- Aplicando-se uma função randômica nos dados;
- Salvando-se os arquivos em cinco *folders*;
- Gerenciando-se o retorno do histórico dos processos para agrupar o *log* de eventos.

Após a aplicação da função randômica na *feature* que retorna o número do processo, o agrupamento de cada particionamento foi gerenciado. Como exemplo do particionamento, na Tabela 13, o *folder* 1 tem 805 processos para a classe “Dispensa de disciplina” com 7.299 trâmites. Ainda na Tabela 13, as medidas de aptidão de cada *folder* foram processadas no primeiro experimento que se deu em relação a gerar o modelo com todo o *log* do *folder* um e, em seguida, avaliar o modelo gerado com os dados dos demais quatro *folders*. A aptidão mínima e máxima para cada *folder* é apresentada após a execução da abordagem de uso dos *plugins* de geração do modelo *Miner petri net with inductive miner* e com o *plugin mine petri net for conformance Analysis*, que trabalham juntos de forma a gerar o modelo e, em seguida, aplicar a avaliação do modelo sobre o *log*.

Para avaliar o modelo, utilizamos quatro *folders* para criação do modelo e, em seguida, o avaliamos com os dados do *folder* 1. O experimento resultou nas métricas demonstradas na Tabela 14. Demonstrou-se que, com a abordagem de criar o modelo com quatro *folders* e avaliar sobre os dados de *log* de um *folder*, as métricas sempre apresentam melhor resultado do que os obtidos com a abordagem de criar o modelo com os dados de um *folder* e avaliar com os dados do *log* dos outros quatro. Isso ocorre porque o modelo é mais abrangente do que os dados, o que acaba criando uma visão de contemplar todo o modelo. Essa estratégia não se mostrou eficiente para trabalhar com o objetivo de avaliação da conformidade que um modelo exibe em relação ao *log* de eventos; logo, nossa abordagem será usar um *folder* para criar o modelo e os demais dados serão utilizados para avaliação do modelo.

No primeiro experimento, utilizamos cinco classes, gerando 25 modelos de dados de um *folder* e 25 modelos com dados de 4 *folders*. O Apêndice B mostra a Tabela 19 com as métricas de aptidão para três classe, em que a aptidão entre o modelo gerado por um *folder* foi avaliada no restante do *log*. A classe "Alteração de Grau" possui 72 *traces* com aptidão 1 e os demais com métrica de 0,09, 0,9 e .083.

Tabela 13 – Classe "Dispensa de disciplinas" – Modelo com 1 *folder*

Folder	Minimum	Maximum	Std. Deviation
FOLDER1	0,75	1	0,01
FOLDER2	0,86	1	0,01
FOLDER3	0,78	1	0,01
FOLDER4	0,75	1	0,01
FOLDER5	0,86	1	0

Tabela 14 – Classe "Dispensa de disciplinas" – Modelo com 4 *folders*

Folder	Minimum	Maximum	Std. Deviation
FOLDER1	0,89	1	0
FOLDER2	1	1	0
FOLDER3	0,94	1	0
FOLDER4	1	1	0
FOLDER5	0,96	1	0

4.7 AVALIAÇÃO CRUZADA: GENERALIZAÇÃO DOS MODELOS POR CLASSE

Neste item, descreveremos o experimento efetuado em relação a avaliar os modelos de cada classe, com objetivo de verificar a similaridade entre os modelos gerados. Nesse sentido, alguns pontos foram identificados sob a perspectiva de comportamento dos dados – por exemplo, um processo com atividade de arquivamento possuindo várias tramitações posteriores e finalizando na atividade de abertura de processo, ou seja, voltando para a atividade de iniciar. Os modelos por classes são apresentados a seguir para três classes. Foram gerados 15 modelos que foram alterados e reavaliados. O *plugin* utilizado para gerá-los foi o mesmo aplicado para avaliar as métricas de conformidade.

4.7.1 Classe: “Alteração de Grau”

A Figura 27 mostra o modelo do *folder* 1 da classe “Alteração de Grau” antes e depois de sofrer alteração.

A atividade inicial de *start* do processo é mantida como “abertura do processo” ou “AG_AutGrau-Abertura de Processo” na classe atual. Para cada modelo gerado, observou-se que existe a repetição das atividades após a atividade inicial, não havendo um padrão de atividade fixa na segunda posição ou segunda tarefa do processo. Nos modelos gerados a partir dos dados dos *folders* 4 e 5 após atividade de finalização do processo, ela volta para a abertura do processo.

Outras características são apresentadas nos modelos, como: os modelos do *folder* 1 e *folder* 4, após a última atividade, voltam para a atividade inicial de “Abertura de Processo”. Apesar de a atividade “Para comentar/opinar/analisar” só estar presente no modelo do *folder* 1, exemplificado na Figura 27, ela não modifica o modelo em sua estrutura, visto que se encontra com o mesmo comportamento do agrupamento da maioria das atividades, ou seja, logo após a atividade inicial.

Apenas essas duas atividades são singulares, porém, nos modelos que se seguem para todos os *folders*, notam-se características distintas em sua forma visual e estrutural. Exemplo: atividade inicial de “Abertura de processo”, que, no *folder* 1, não indica de forma clara ser a atividade inicial do processo.

Por outro lado, no modelo do *folder* 4, apesar de a atividade inicial estar claramente descrita como “Abertura de processo”, podemos notar que existe no modelo uma atividade retornando para ela após o fim. Foram feitos ajustes nos modelos dos *folders* 1 e 4.

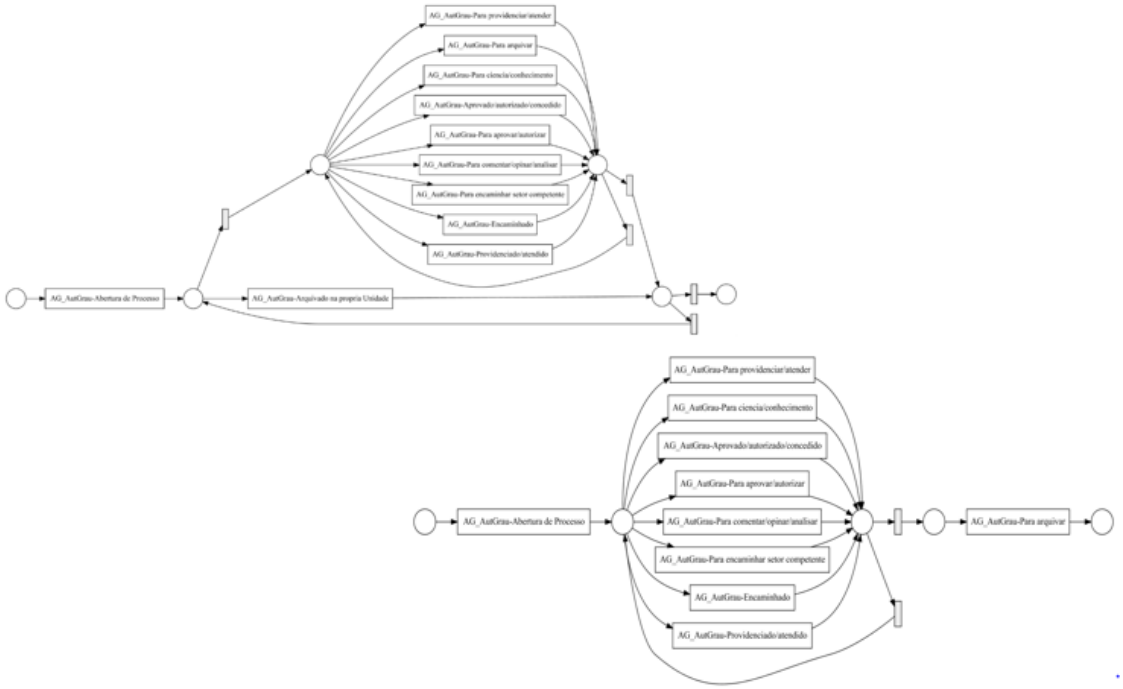
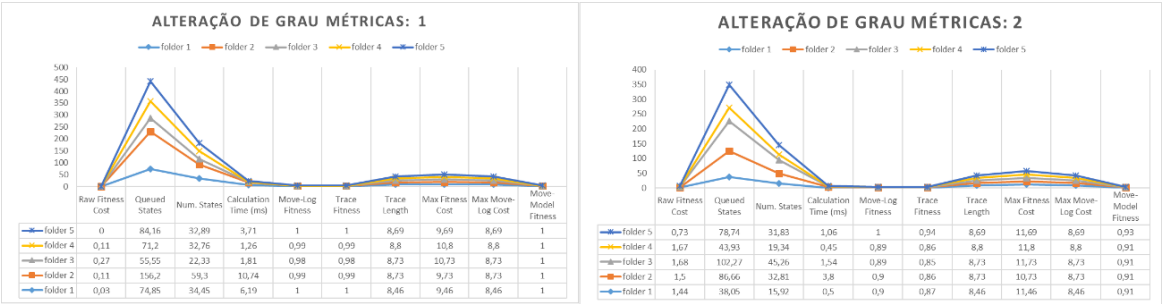


Figura 27 – Modelo da classe “Alteração de Grau” – *folder* 1

As métricas dos modelos inicial e adequado de cada modelo da classe “Alteração de Grau” foram geradas no ProM e estão representadas nos gráficos da Figura 27. No primeiro gráfico da imagem, é possível verificar que todas as métricas sofreram mudanças, exceto aqueles referentes ao tamanho do *trace* e ao custo máximo do movimento do *log*. Isso se justifica pelos seguintes fatores:

- *Trace length* → Não houve deleção nas atividades que compõem o *trace* do *log*, houve apenas ajuste no conteúdo da descrição.
- *Max move-log cost* → O *folder* do *log* a ser comparado com o modelo não sofreu nenhum tipo de alteração.
- *Move model fitness* → A aptidão do movimento do modelo mudou de 1 para ± 0,91 para todos os *folders.*
- *Move log fitness* → A aptidão do movimento do *log* diminuiu para quase todas as métricas dos *folders* após a alteração.

- *Fitness trace* → A métrica de aptidão dos *traces* diminuiu para quase todas as métricas dos *folders* após a alteração.
- *Max fitness cost* → O custo máximo da aptidão aumentou para todos os *folders* após a alteração.



	Raw Fitness Cost	Queued States	Num. States	Calculation Time (ms)	Move-Log Fitness	Trace Fitness	Trace Length	Max Fitness Cost	Max Move-Log Cost	Move-Model Fitness
folder 5	0	84,16	32,89	3,71	1	1	8,69	9,69	8,69	1
folder 4	0,11	71,2	32,76	1,26	0,99	0,99	8,8	10,8	8,8	1
folder 3	0,27	55,55	22,33	1,81	0,98	0,98	8,73	10,73	8,73	1
folder 2	0,11	156,2	59,3	10,74	0,99	0,99	8,73	9,73	8,73	1
folder 1	0,03	74,85	34,45	6,19	1	1	8,46	9,46	8,46	1

	Raw Fitness Cost	Queued States	Num. States	Calculation Time (ms)	Move-Log Fitness	Trace Fitness	Trace Length	Max Fitness Cost	Max Move-Log Cost	Move-Model Fitness
folder 5	0,73	78,74	31,83	1,06	1	0,94	8,69	11,69	8,69	0,93
folder 4	1,67	43,93	19,34	0,45	0,89	0,86	8,8	11,8	8,8	0,91
folder 3	1,68	102,27	45,26	1,54	0,89	0,85	8,73	11,73	8,73	0,91
folder 2	1,5	86,66	32,81	3,8	0,9	0,86	8,73	10,73	8,73	0,91
folder 1	1,44	38,05	15,92	0,5	0,9	0,87	8,46	11,46	8,46	0,91

Figura 28 – Métricas dos modelos da classe “Alteração de Grau” antes e depois da alteração

Com as adequações, todos os *folders* sofreram alteração para a métrica *trace fitness*, que inicialmente apresentou melhores resultados, e após a alteração o pior resultado se manteve acima de 0,84 para a métrica. O mesmo comportamento foi observado para a métrica *Move model fitness*, que apresentou alteração para todos os *folders*. As métricas da análise feita após alteração mostraram resultados inferiores assim como ocorreu com a métrica do *trace* fitness.

Sobre a *feature* que impacta diretamente na criação dos modelos no Apêndice A, as tabelas Tabela 20, Tabela 21 e Tabela 22 apresentam mais detalhes sobre a aderência da atividade para cada *folder* em três classes distintas. Para a maioria, há aderência de 100%, no entanto, para alguns casos, há aderência menor que 50%. Para a classe “Alteração de Grau”, tem-se:

- Comentado/opinado → inaderência de 40% para *folders* 1, 3 e 4.
- Para aprovar/autorizar → inaderência de 40% para *folders* 2, 3 e 4.
- Para comentar/opinar/analisar → inaderência de 20% para *folders* 2, 3, 4 e 5.

A inaderência avaliada nas atividades reforça as métricas de aptidão definidas como *trace fitness* apresentadas na Figura 28, que apresenta melhores resultados para os *folders* 1 e 5 da classe “Alteração de Grau”.

4.7.1.1 Análise dos modelos

Na análise dos modelos obtidos antes de fazer a adequação, alguns *insights* importantes foram observados. Nesse cenário, foi identificado o desejo de se verificarem modelos equalizados, ou seja, modelos simples, principalmente que representassem uma proximidade estrutural em relação aos K modelos gerados. Os pontos observados para cada *folder* foram:

Folder 1:

- Processo com tramitação da unidade origem e destino iguais.
- Processos com duas atividades definidas como “Para arquivar” em sequência.
- Processos com atividade de “Para arquivar” e, em seguida, com várias tramitações sem finalizar com uma atividade de “Para arquivar”.
- 39 processos com atividade “Arquivado na propria Unidade” que foram alterados para a atividade “Para arquivar”.

Folder 2:

- 37 processos com atividade “Arquivado na própria Unidade” que foram alterados para a atividade “Para arquivar”.
- Processo com 10 tramitações em que a sexta possui atividade “Arquivado na própria Unidade”.
- Processo com atividade de “Abertura” e, em seguida, de “Arquivado na própria Unidade”.

Folder 3:

- 37 processos com atividade “Arquivado na própria Unidade” que foram alterados para a atividade “Para arquivar”.
- Processo com 13 tramitações e, na nona tarefa, a atividade “Arquivado na propria Unidade” foi substituída por "Para providenciar/atender”.
- Processo com 20 tramitações e, na quarta tarefa, a atividade “Arquivado na própria Unidade” foi substituída por “Para providenciar/atender”.
- Processo com 16 tramitações e, na décima tarefa, a atividade “Arquivado na propria Unidade” foi substituída por “Para providenciar/atender”.

Folder 4:

- 39 processos com atividade “Arquivado na própria Unidade” que foram alterados para a atividade “Para arquivar”.
- Processo com apenas 3 atividades, uma de abertura e duas “Para arquivar”.
- Processo com duas atividades de abertura.
- Processo com mais de uma atividade “Arquivado na própria Unidade”, sendo uma delas substituída por “Para arquivar”.

Folder 5:

- 41 processos com atividade “Arquivado na propria Unidade” que foram alterados para a atividade “Para arquivar”;
- Processo com três atividades de “abertura do processo”, sendo duas substituídas por “Para atender/providenciar”;
- Processo com última atividade “Para providenciar/atender”, que foi substituída por “Para arquivar”.

Os itens dos cinco *folders* foram identificados com os itens que impactaram os modelos causando a falta de equalização dos modelos. No entanto, podemos verificar que os pontos são similares, ainda que existam casos particulares, como, por exemplo, processo com mais de uma atividade de “Abertura”.

4.7.2 Classe “Dispensa de Disciplina”

A classe “Dispensa de Disciplina” é composta por 16 atividades. As métricas dos modelos inicial e adequado de cada modelo da classe “Dispensa de Disciplina” foram geradas no ProM e estão representadas nos gráficos da Figura 29. No primeiro gráfico da imagem, é possível verificar que todas as métricas sofreram mudanças, assim como na classe “Alteração de Grau”, exceto as referentes ao tamanho do *trace* e o custo máximo do movimento do *log*. Isso se justifica pelos seguintes fatores:

1. *Trace length* → Não houve deleção nas atividades que compõem o *trace* do *log*, apenas ajuste no conteúdo da descrição.
2. *Max move-log cost* → O *folder* do *log* a ser comparado com o modelo não sofreu nenhum tipo de alteração.

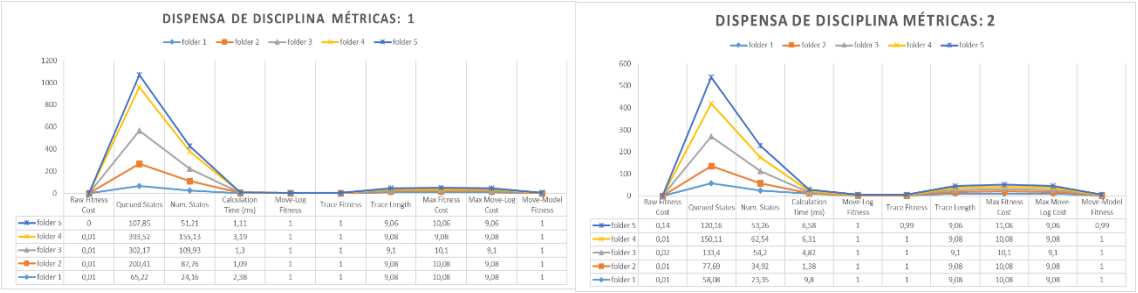


Figura 29 – Métricas dos modelos inicial e adequado da classe “Dispensa de Disciplina”

Apesar das adequações, o único *folder* que sofreu alteração para a métrica *trace* fitness foi o *folder* 5, que inicialmente apresentou o valor 1, e após a alteração mudou para 0,99 a medida de *fitness*. O mesmo comportamento foi observado para a métrica *Move model fitness*, que apresentou a mesma métrica que o *trace* fitness. Das 16 atividades que compõem toda a base da classe, são usadas 12 na geração do modelo representado no *folder* 1 da Figura 30. A atividade “DD_DispDis-Providenciado/atendido” aparece em dois processos como a última. Um processo que se iniciou com a primeira atividade igual a “Para encaminhar setor competente”. Para o modelo da Figura 31, os dois itens foram adaptados para a última atividade igual a “Arquivado na propria Unidade” e a atividade inicial padrão “Abertura de processo”.

No modelo da Figura 30, a atividade que inicia o processo afetou o caminho inicial. Podemos verificar que, na Figura 31, gerada após a alteração da atividade inicial de “Para encaminhar setor competente” para “Arquivado na propria Unidade”, o modelo contempla a separação da atividade de início.

Além disso, para todos os *folders* na geração do modelo adequado, houve alteração na atividade final para “Para arquivar” ou “Arquivado na própria unidade”, definindo-se pelo maior percentual existente no *folder* da classe como atividade final. Por exemplo: No *folder* 1, sete processos foram setados da atividade final “Providenciado/atendido” para “Para arquivar”; no *folder* 5, 87 processos foram setados da atividade final “Para providenciar/atender” para “Arquivado na propria Unidade”, uma

vez que 90% dos processos dessa classe finalizam com essa atividade. Vale ressaltar que os processos foram filtrados inicialmente com a premissa de haver pelo menos uma atividade que descrevesse a justificativa de que o processo foi finalizado.

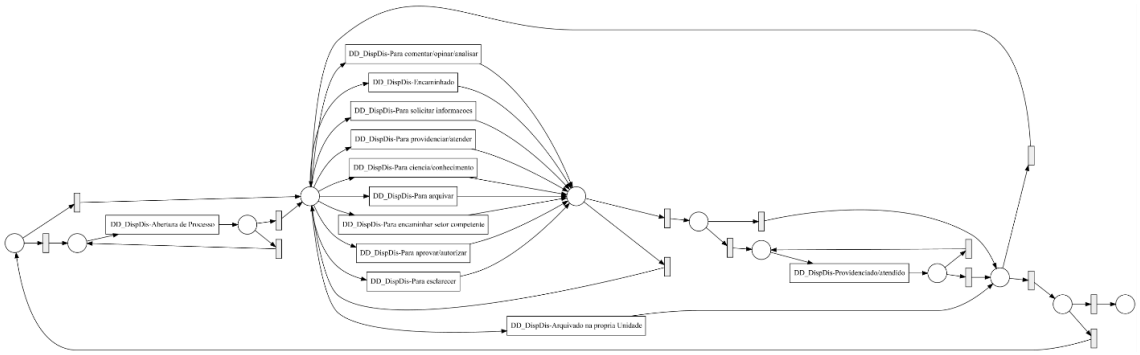


Figura 30 – Classe “Dispensa de Disciplina” – Modelo do *folder* 1 (inicial)

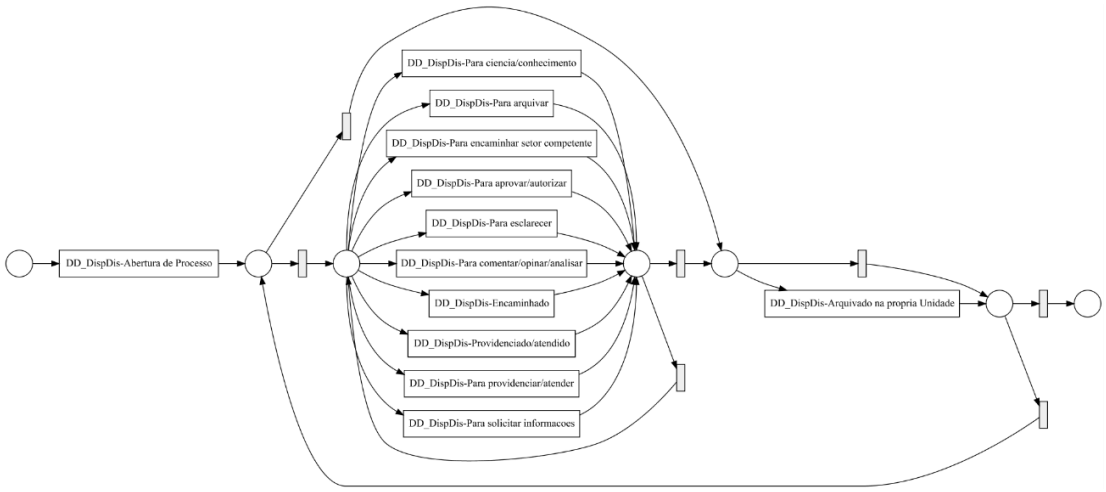


Figura 31 – Classe “Dispensa de Disciplina” – Modelo do *folder* 1 (adequado)

4.7.3 Classe “Exclusão de Reprovação”

Para a classe “Exclusão de Reprovação”, o modelo do *folder* 1, que possui 62 processos distintos e 767 linhas de tramitação, manteve, para todos os processos, a atividade inicial “Abertura de processo” e a atividade final “Arquivado na propria Unidade”. Para essa classe, não houve intervenção na atividade para ajustar o modelo.

Mesmo apresentando duas atividades distintas como última atividade do processo, há uma que se destaca, que é a “Arquivado na propria Unidade” com 93,54% dos processos, enquanto a atividade de “Para arquivar” finaliza 6,46% dos processos.

Já o modelo do *folder* 2 tem 57 processos distintos e 636 linhas de tramitação. Também possui as mesmas características do *folder* 1, tanto para atividade inicial quanto para a atividade final, desta vez com 96,49% dos processos finalizando na atividade de "Arquivado na propria Unidade".

O *folder* 3 apresenta 63 processos distintos com 891 linhas de tramitação. 96,29% dos processos finalizam na atividade "Arquivado na propria Unidade". Nesse *folder,* foi identificado um processo com a atividade final "Para providenciar/atender", que foi alterada para "Arquivado na propria Unidade". As figuras Figura 47 e Figura 48 representam os modelos antes e depois da alteração. Nos dois modelos, as atividades se mantêm.

O penúltimo *folder* dessa classe, o *folder* 4, possui 60 processos distintos com 743 linhas de tramitação, e 96,29% dos processos finalizam na atividade "Arquivado na propria Unidade".

O *folder* 5, por sua vez, tem 61 processos distintos com 754 linhas de tramitação, e 97,45% dos processos finalizam na atividade "Arquivado na propria Unidade".

Considerando os cinco *folders* da classe "Exclusão de Reprovação", somente no *folder* 3 foi necessário fazer ajuste. Havia um processo que estava com a última atividade diferente das atividades "Arquivado na propria Unidade" e "Para arquivar" as quais definem a tarefa final do processo. Vale ressaltar que essa última atividade é apenas um parâmetro de término do processo. Como já citado, verificamos também nesta classe após ter uma das duas atividades como final, a característica dos dados possibilita voltar o processo para um nível de tramitação qualquer.

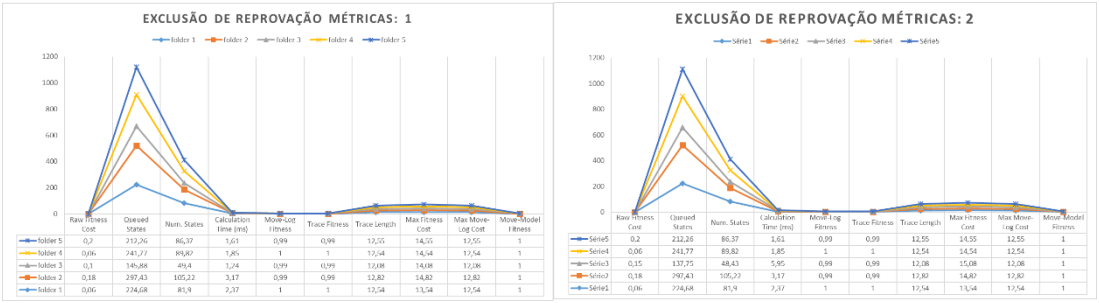


Figura 32 – Métricas dos modelos inicial e adequado da classe "Exclusão de Reprovação"

No gráfico da Figura 32, as métricas dos modelos inicial e adequado de cada *folder* da classe "Exclusão de Reprovação" foram geradas no ProM. Antes de aplicar o *plugin*

para gerar o modelo, foi efetuado o ajuste em um processo que teve a atividade final alterada, mudando de “Para providenciar/atender” para “Arquivado na propria Unidade”.

Mesmo com alteração em apenas um *folder*, podemos observar, na Figura 32, que não houve alteração em nenhum *folder* para as medidas trace fitness e *Move model fitness*. Isso se justifica pelos seguintes fatores:

1. *Trace length* → Não houve deleção nas atividades que compõem o *trace* do *log*, apenas ajuste no conteúdo da descrição.
2. *Max move-log cost* → O *folder* do *log* a ser comparado com o modelo não sofreu nenhum tipo de alteração.

4.8 ESTUDO DE CASO – DEFINIÇÃO

Neste item, exporemos o levantamento e a análise do estudo de caso da POLI/UFRJ (Escola Politécnica da Universidade Federal do Rio de Janeiro) como estudo do conceito do modelo proposto neste trabalho.

O estudo de caso considera um processo detalhado a partir do levantamento efetuado por alunos juntamente com o coordenador da área, que definiu o processo e solicitação de processos acadêmicos da POLI/UFRJ. O mapeamento foi feito através de entrevistas, modelagem do processo e, posteriormente, validação e melhoria do modelo desenhado.

Os modelos foram levantados a partir de documento mapeado com o entendimento do processo. Basicamente, o processo definido e documentado possui na sua composição três momentos:

- Tratamento de Irregularidade via Memorando
- Tratamento de Irregularidade via Processo
- Processos Internos

Esses momentos são partes dos diversos processos de trabalho que cada unidade acadêmica possui e são focados na vida acadêmica dos alunos. Dos três tipos de processos

levantados, o “Tratamento de Irregularidade via Processo” é aquele que detém sistemicamente o controle na base de dados SAP (solicitação de processos acadêmicos). Como o modelo proposto requer *log* de sistemas para teste, usaremos o processo “Tratamento de Irregularidade via Processo” na avaliação. Vale ressaltar que esse tipo de processo possui diversos subprocessos dentro dele, sendo cada tipo de irregularidade dentro desta tratada.

4.8.1 Levantamento do processo - escopo

Para o mapeamento do processo, foram feitas entrevistas e reuniões presenciais com a coordenadora da POLI/UFRJ, que detalhou todo o processo acadêmico da unidade. O mapeamento foi feito por uma equipe de estudantes da COPPE/UFRJ (mestrandos e doutorandos) que, após documentarem as atividades realizadas dentro do escopo referente ao tratamento de irregularidades da escola Politécnica (POLI), produziram o modelo dos processos em BPMN.

4.8.1.1 Andamento via processo

As irregularidades tratadas via processo são iniciadas no setor de protocolo da escola Politécnica (POLI) ou do Centro de Tecnologia (CT), dependendo do assunto.

A opção de andamento via processo acontece para aqueles que não podem ser regularizados via memorando. Em casos assim, o trâmite da tratativa é a solicitação do aluno para que seja instaurado um processo administrativo.

Havendo a necessidade de instauração do processo, o aluno procura o setor da POLI ou o CT e solicita a abertura do processo. No setor de protocolo, a solicitação protocolada é encaminhada para o coordenador responsável, que a analisa e a leva para a assembleia de coordenadores.

As decisões são registradas no sistema SAP pela Daec. Se o parecer for favorável, o processo seguirá até o departamento responsável e o registro da decisão ficará no histórico do aluno, que não será notificado. No caso de parecer negativo, o aluno é comunicado e pode recorrer em duas ocasiões:

- Se houver fato novo: recorre levando os fatos novos para o coordenador, que novamente levará para a decisão em assembleia
- Se não houver fato novo: Em caso de discordar da decisão, há possibilidade de recorrer à congregação.

Para o envio do recurso, existe um prazo de 30 dias; depois, o processo é arquivado. Um processo arquivado pode ser reaberto.

No mapeamento do andamento via processo, foi modelado o BPMN demonstrado na Figura 33. Para esse tipo de tratamento, foram levantados sete envolvidos que representam os autores do processo:

- Aluno: solicitante do processo para regularizar alguma situação de sua vida acadêmica.
- Setor de protocolo da POLI/UFRJ: departamento responsável por receber o aluno e efetivar a solicitação da irregularidade, atividade que requer análise de documento e abertura do processo no sistema SAP.
- Coordenador/departamento responsável: unidade acadêmica responsável por algum parecer para o processo ou encaminhamento para outras unidades.
- Comissão de coordenadores: corpo responsável por avaliar as solicitações.
- Comissão de ensino (Daec): diretoria de ensino e cultura, responsável por solicitar aos coordenadores dos cursos a listagem de alunos irregulares. A listagem é solicitada depois do período estipulado após o prazo de trancamento de cada período.
- Congregação: responsável por receber as solicitações de excepcionalidades, que são votadas pela comissão de ensino.

A Figura 33 apresenta os departamentos que são definidos no modelo do processo. Os responsáveis por avaliar as irregularidades estão representados em cada raia do processo da Figura 33. No documento de levantamento do processo, 19 tipos de irregularidades foram identificadas e são apresentadas na Tabela 15. Dos 19 tipos de irregularidades, cinco seguem o tratamento de excepcionalidade.

Tabela 15 – Tipos de irregularidades

Irregularidade	Tratamento de excepcionalidade	Tratamento via Processo	Local de tratamento
Menos de 6 créditos		Sim	
Mais de 32 créditos		Sim	
1/3 de disciplinas fora do curso		Sim	
Disciplinas sem pré-requisitos ou requisitos concomitantes		Sim	
Homologação de Grau		Sim	
Exclusão de Reprovação		Sim	
Dispensa de Disciplina			
Revalidação de diploma			CT
Transferência			CT
Disciplina avulsa		Sim	CT
Colação de grau fora da época	Sim		
Estágio fora do prazo	Sim		
Intercâmbio	Sim		
Cancelamento de matrícula			
Trancamento do período fora do prazo			CT
Destrancamento do período fora do prazo			CT
Diploma especial	Sim		
Cancelamento de Diploma	Sim		

Na definição do gestor, tratamento de excepcionalidades é quando os casos excepcionais são votados pela comissão de ensino e, em seguida, encaminhados para a congregação. Quando a excepcionalidade pode ser tratada via coordenador, ele avalia, elabora o memorando e leva à Daec. Uma comissão avalia e depois leva à congregação. Essas situações não são tratadas via *autSec* nem via processo, mas, ao final, a regularização é gravada no sistema e, estando correta, a situação é regularizada.

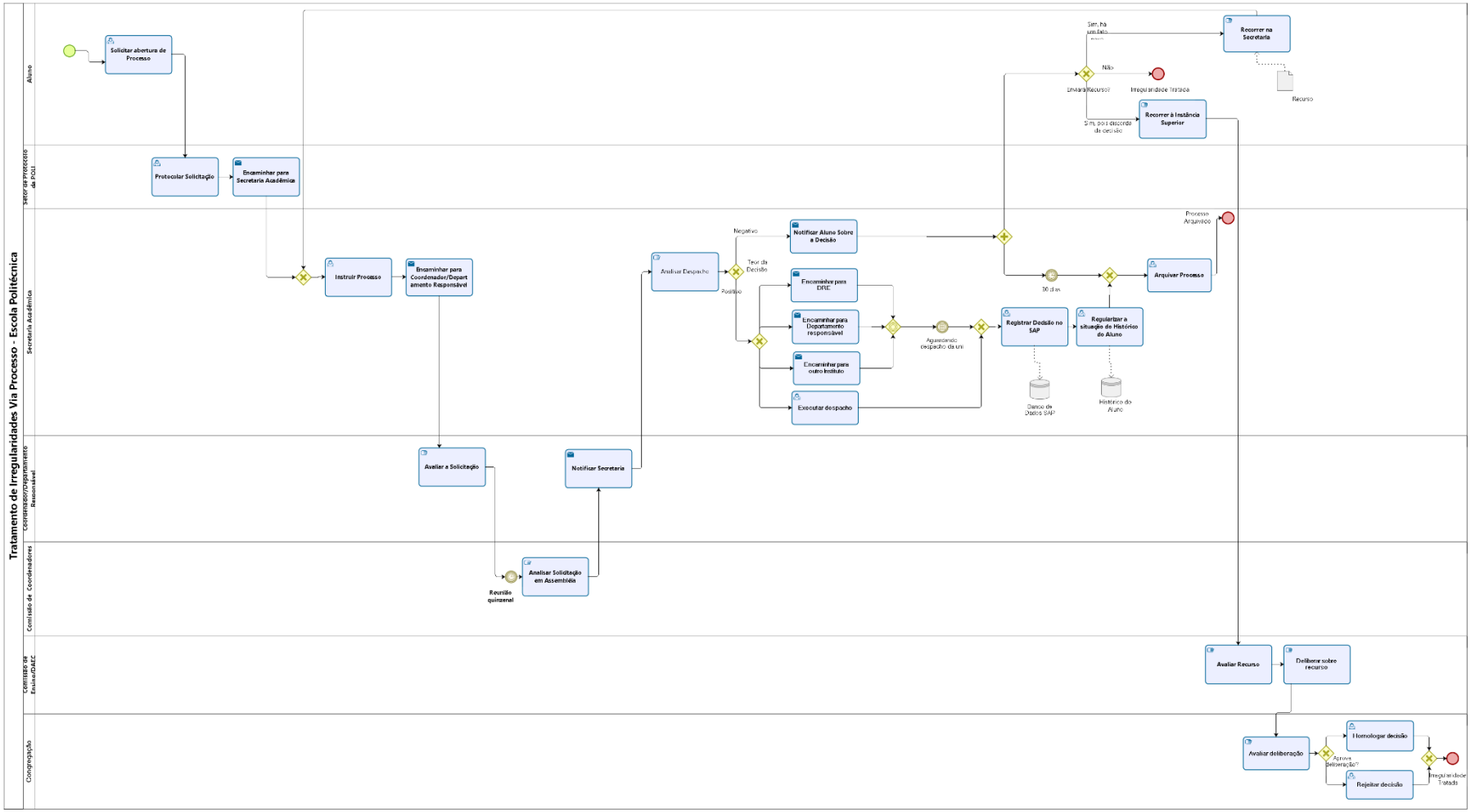


Figura 33 – Modelo do processo de regularização de assuntos acadêmicos

4.8.1.2 Processos internos

No detalhamento dos processos, foi efetuado o levantamento dos processos internos atrelados ao tratamento de irregularidades, o qual também foi mapeado – um artefato à parte, mas que pode influenciar a decisão, uma vez que gera impacto no trâmite. O modelo do processo mapeado na Figura 45 identifica dois envolvidos nos cenários, o aluno e a COA – Comissão de Orientação e Acompanhamento Acadêmico.

A COA é acionado nas seguintes situações:

- Quando ocorre o cancelamento de matrícula (jubilamento);
- quando o aluno está ativo no curso há mais de 14 períodos;
- quando o coeficiente de rendimento (CR) é menor que 3 em períodos consecutivos;
- quando acontece a reprovação da mesma disciplina quatro vezes.

Nessas situações, um processo interno é aberto e direcionado para a COA, que o acompanha até a conclusão. O aluno assina um termo de compromisso e, se não cumpri-lo em três períodos (que é o prazo de acompanhamento da COA), o processo é levado para a Congregação. Se o aluno cumprir com a exigência, sua situação é regularizada.

4.8.1.3 Andamento via memorando

O andamento via memorando ocorre nas situações em que a Diretoria Adjunta de Ensino e Cultura (Daec) faz o acompanhamento de situações dos alunos. O levantamento é feito aos coordenadores, que devem enviar listagem por e-mail das situações dos alunos. As situações de irregularidade apontadas para tratativa neste molde são as seguintes:

- O aluno solicita autorização para cursar mais de 1/3 de disciplinas fora do curso;
- o aluno solicita autorização para cursar menos de seis créditos no período;
- o aluno abre solicitação para cursar disciplinas sem pré-requisitos ou requisitos concomitantes (que, em alguns casos, devem ser autorizadas);
- o aluno solicita autorização para cursar mais de 32 créditos.

Outras situações, como inscrição e trancamento de disciplinas fora do prazo, são tratadas diretamente com o coordenador, para elas logo, não existe processo e a solicitação é feita via memorando. O modelo em BPMN desse processo está demonstrado na Figura 46 e mantém, na sua definição, a Daec e a coordenação como as duas áreas envolvidas. O modelo mostra que o parecer da regularização é inserido no sistema SAP.

3.8.1 Base de dados

A base de dados SAP armazena 276 assuntos distintos que são representados por um código e uma descrição. Dentre os assuntos, o código 0671-8, por exemplo, representa os assuntos acadêmicos. Outros assuntos, como afastamento, aposentadoria, promoção etc. estão na mesma base.

O esquema da Figura 34 mostra todas as tabelas da base de dados do SAP, no entanto, o histórico das tramitações encontra-se em 5 tabelas: (1 - Tramitação, 2 - Despacho, 3 - Assunto, 4 - Processo, 5 - Unidade).

Em análise da base total, temos os seguintes *status*:

1. Menor data de processo cadastrado: 1923-01-01.
2. Maior data de processo cadastrado: 2016-09-05.
3. Total de processos: 2.131.570, sendo 45.690 provenientes da unidade Escola Politécnica.
4. Total de processos abertos na unidade Escola Politécnica, cujo assunto é “assuntos acadêmicos”: 1.408.
5. Total de processos abertos na unidade Centro de Tecnologia, cujo assunto é “assuntos acadêmicos”: 16.631.

Filtrando os processos para as unidades CT e POLI verificamos que, no CT, o ano com maior número de abertura de processos foi 2015, enquanto na POLI foi 2016. Na Tabela 23, podemos ver que, em anos anteriores a 2015, os processos acadêmicos da Poli apresentam um baixo volume de dados na base, enquanto no CT o número é mais equalizado.

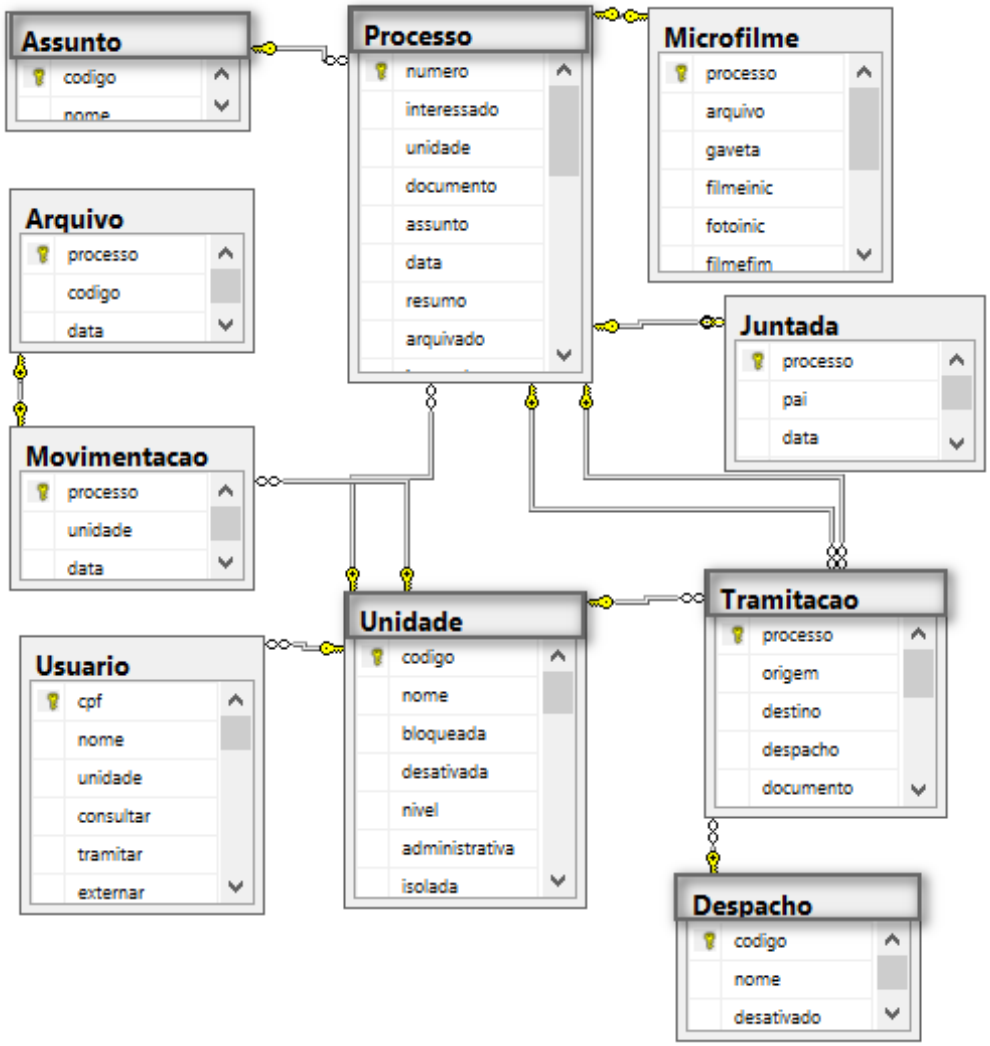


Figura 34 – Diagrama do banco de dados SAP

4.8.2.1 Desafios/características identificadas na base de dados

Alguns desafios sobre os dados foram mapeados analisando-se o processo "Tratamento de Irregularidades via Processo". São eles:

- O campo **"Resumo"** da tabela de processo é livre para digitação do usuário.
- O campo de data da tabela de processo guarda a data e a hora como [00:00:00.000].
- A tabela de tramitação apresenta a coluna de despacho como "*null*".

- As colunas de data e ordem possuem, às vezes, datas distintas para a primeira linha da tramitação. O campo de data é igual ao da tabela de processo.

Na tabela de tramitação para processo, o valor *null* nas colunas despacho e a data com valor zero foram considerados da seguinte forma:

- Despacho → considerar como atividade inicial.
- Data → considerar a data informada na tabela de processo cuja chave identificadora é a mesma da tabela tramitação.

Tabela 16 – Descrição do campo "Resumo" – Cursar menos de seis créditos

Resumo
Autorizacao para Cursar **M**enos de Seis Creditos
Autorizacao para Cursar **m**enos de Seis Creditos
AUTORIZAÇAO **P**/CURSAR MENOS SEIS CREDITOS
AUTORIZAÇAO **PARA** CURSAR MENOS 6 CREDITOS
Autorizacao **para** Cursar Menos de 6 Creditos **Fora do Prazo. DRE 011111111**
AUTORIZAÇAO PARA CURSAR MENOS DE 6 CREDITOS.
menos de 6 créditos
Menos de 6 créditos
INSCRICAO EM UM MENOS DE **SEIS** CREDITOS EM UM MESMO PERIODO
REQUERIMENTO DE INSCRICAO EM MENOS DE **6** CREDITOS EM UM MESMO PERIODO
MENOS DE **06 (SEIS)** CREDITOS

Efetuando a consulta da Fórmula 10 sem o parâmetro do assunto, conseguimos o retorno de 360 documentos, e, com o parâmetro, 356 documentos. Logo, podemos concluir que, mesmo para toda a base, tem-se um acoplamento sobre a descrição (pelo menos o tipo de assunto *versus* a descrição), conforme mostra o resultado da Tabela 16, cujo resultado provém da Fórmula 10:

```
SELECT distinct resumo, 'Menos de 6 créditos' AS tipo_processo from processo
where resumo like '%credito%' and (
resumo like '%seis%' or
resumo like '% 6 %')
and assunto = ‘0671-8'
```

Fórmula 10 – Consulta de filtro

No entanto, a cláusula resumo *like* '% 6 %' pode ter retornado uma linha que possui uma matrícula e nada tem relacionado à classe "cursar 6 créditos". A Tabela 17, na linha 1, demonstra o efeito da busca, utilizando a cláusula *like* '% 6 %'. Para casos assim, o classificador poderá indicar a classificação errada da classe.

Tabela 17 – Exemplo do retorno da consulta executada

resumo	Tipo_processo	Classificação: Ok, não-ok
EQUIVALÊNCIA DE 4 PARA 6 CRÉDITOS NA DISC. LÍNGUA INSTRUMENTAL(INGLÊS)I E II(LEG)	Menos de 6 créditos	Não ok
Equivalência de disciplinas e permissão p/ cursar menos de 6 créditos	Menos de 6 créditos	Ok
INCRICAO EM MENOS DE SEIS CREDITOS EM UM MESMO PERIODO	Menos de 6 créditos	Ok
Inscrição em Disciplinas com menos de 6 créditos - Coord. Licenciatura	Menos de 6 créditos	Ok

A partir das informações levantadas, principalmente sobre o escopo do processo, foi possível iniciar uma avaliação tanto da base de dados, em relação ao que foi levantado, quanto do processo, em relação ao que estava sendo identificado no banco de dados.

4.8.4 Avaliação de dados relacionada ao modelo

O escopo do projeto e o entendimento do processo direcionaram o entendimento inicial da análise da base de dados no que se refere à necessidade de preparar o *log* para mapeamento do modelo e aplicação de métodos de mineração de processos.

A Tabela 18 mostra a análise sobre os dados em relação aos itens mapeados no modelo do processo de "Tratamento de irregularidade via processo", com o objetivo de identificar nos dados os atores envolvidos.

A Figura 33, identifica os sete atores do processo, mas dois deles (Coordenador / Departamento Responsável e Congregação) possuem informações de quantidade irrelevante na perspectiva de avaliação das unidades como mostrado na Tabela 18.

Tabela 18 – Abstração de atores – Visão dos dados

Autor	Observação
Aluno	Tem o nome do requerente do processo.
Setor de Protocolo da POLI	Na base de dados, a Unidade Escola Politécnica (36.01.00.00) aparece como 3 lugar de quantidade de processos abertos; apesar do processo ter sido mapeado neste centro. 1º (36.00.00.00→CT- Centro de Tecnologia): 16.573 2º (36.02.00.00→Escola de Quimica): 16.369 3º (36.01.00.00→Escola Politecnica): 1.209
Secretaria acadêmica	→Escola Politécnica (14.933) Tramitaram pela secretaria acadêmica unidade 36.01.00.00 →Escola de Quimica (6.237) Tramitaram pela secretaria acadêmica 36.02.00.00
Coordenador / Departamento Responsável	Em toda a base só existe um processo "23079.007080/2015-19" inerente a unidade (31.01.07.00 -> Congregacao)
Comissão de coordenadores	36.01.50.00→ Coordenação de Relações Internacionais (39 processos) 36.02.06.00→ Coordenação de Estágio (741 processos) 36.03.00.00→ COPPE-Coord.de Prog.de Pos Grad. em Engenharia (285 processos) * Não representa todos, apenas 4% dos quase 23 mil processos finalizados.
Comissão de Ensino / DAEC	36.01.01.01 DAEC → Diretoria Adjunta de Ensino e Cultura (336 processos)
Congregação	31.01.07.00→ Congregação (Nenhum processo)

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÃO

A análise da conformidade sob o *log* de eventos e o processo mapeado é um cenário dentro da área de mineração de processos que necessita de esforços para evoluir. Fazer validação de regras definidas ainda é uma tarefa mais fácil do que abranger a avaliação da conformidade com o foco de um processo como um todo.

No entanto, um processo mapeado e definido pode ser de grande ajuda na abstração de artefatos de regras e validação das mesmas. Para cenários em que a proveniência do tratamento do *log* passou por avaliação e classificação por algoritmos autônomos, a avaliação da conformidade sob o modelo mapeado tem um ganho singular, visto que a própria abordagem de avaliar a conformidade entre o modelo e o *log* faz uma avaliação entrelaçada com a instância que fez a classificação das *features*, ou seja, os classificadores.

Por outro lado, a abrangência do estudo mostrou que, dependendo da abordagem adotada e da necessidade de uso da validação com vários modelos, ainda há necessidade de evolução no contexto de automatizar a possibilidade de avaliação de diversos modelos simultaneamente. Além disso, o processo inicial de avaliação de ferramentas e algoritmos que possibilitem a validação de modelo com o *log*, mesmo que de forma particionada, não identificou variedades de ferramentas com esta possibilidade. Na realidade, apenas na ferramenta ProM foi identificada a viabilidade de exportação do *log* processado com as métricas geradas.

A análise do item 5.4.1.1, relativamente ao desejo de visualizar a similaridade estrutural entre os modelos, direcionou para a análise dos pontos divergentes. Ao avaliar cada diferença de *folder*, verificou-se que as divergências compartilham, em sua grande maioria, a mesma estrutura para cada *folder*, algumas inclusive quantitativamente.

Em relação à estrutura de cada *folder*, neste estudo, o modelo da validação cruzada proposto para avaliar a conformidade demonstrou que a abordagem de classificação e geração dos modelos apresentou métricas positivas. Em relação ao *log* de eventos, e

considerando as medidas de aptidão com foco em 80% que a literatura recomenda (VAN DER AALST, 2016), estas foram representadas de forma positiva, já que a aptidão apresentada no item 5.4.1 para os cinco *folders* da classe "Alteração de grau" possuem métricas superior a 80%. A abordagem também demonstrou grande aderência em relação a três dimensões de qualidade (Aptidão, Simplicidade e Precisão) discutidas no item 3.7 que mensuram a qualidade e são discutidas em termos de auxiliar em abordagem de auditoria e qualidade.

Vale ressaltar que em caso de métrica inferior a 80%, a abordagem pode ser entendida como o caso a não ser escolhido para a avaliação da conformidade do modelo com o *log*. Ou seja, o fato da métrica apresentar valor inferior a 80% de aptidão, também mensura a conformidade do processo em relação ao *log* de eventos.

Nosso estudo não foi baseado nem teve foco em nenhuma abordagem de auditoria, no entanto, de forma abstrata, o levantamento do processo foi um manual de qualidade cujas lacunas buscamos avaliar e preencher. Com essa visão, identificamos que o modelo desenhado possui pontos que divergem em relação aos modelos gerados pelo *log* de eventos. Outro fator que impacta de forma negativa as análises está relacionado às ferramentas de mineração de processos. Para as abordagens existentes, o maior gargalo é em relação à capacidade de validar um modelo proposto de forma automática, sem que haja intervenção crítica no modelo.

5.2 TRABALHOS FUTUROS

A avaliação da conformidade relacionada à aptidão detalhada no capítulo 4 mostrou-se eficaz, no entanto, a abordagem de criação dos modelos usando a validação cruzada é um processo oneroso, uma vez que é necessário criar vários modelos e na sequência fazer a comparação com o *log*. Como proposta de trabalho futuro, uma automatização para avaliar as conformidades dos cinco *folders versus* os modelos se faz necessária.

Dentro do modelo proposto de validação cruzada, é preciso utilizar outras abordagens de classificação com a perspectiva de identificar outras *features* como aluno, quantidade de processos que ele já solicitou, quais os processos com mais tramitações,

quais os usuários mais impactados com o efeito *ping pong* nas tramitações identificadas no sistema etc. Assim, será possível extrair outras visões relacionadas às perspectivas dos dados que podem auxiliar na validação mais acurada da conformidade.

Sob a visão da perspectiva dos dados, é necessário avaliar a qualidade do *input* do texto digitado e a incidência na adequação da classificação, utilizando abordagem de classificador com algoritmos de redes neurais.

Considerando a perspectiva do sistema, é indispensável avaliar a generalização do modelo uma das métricas de qualidade.

Avaliar a abordagem utilizando dados estruturados também é fundamental, pois, como definimos no item 4.4, o trabalho atual avaliou o modelo utilizando dados não estruturados. Para a abordagem com o uso de dados estruturados, entendemos que as técnicas utilizadas para classificação do texto não precisam ser aplicadas.

Ainda, executar o experimento usando K=10 em vez de k=5 para gerar os modelos e além disto, utilizar uma série temporal de dados abrangendo o dobro de anos do experimento atual mostram-se ações cruciais para trabalhos futuros.

REFERÊNCIAS

ABPMP, BRASIL. BPM CBOK V3. 0: Guia para o Gerenciamento de Processos de Negócio-Corpo Comum de Conhecimento. *2º edição*, 2013.

BHARGAVA, Neeraj *et al.* Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, v. 3, n. 6, 2013.

BLUM, Avrim; MITCHELL, Tom. Combining labeled and unlabeled data with co-training. 1998, [S.l.]: ACM, 1998. p. 92–100.

BOLT, Alfredo; DE LEONI, Massimiliano; VAN DER AALST, Wil M. P. Scientific workflows for process mining: building blocks, scenarios, and implementation. *International Journal on Software Tools for Technology Transfer*, v. 18, n. 6, p. 607–628, 2016.

BOLT, Alfredo; LEONI, Massimiliano De; AALST, Wil M. P. Van der. Scientific workflows for process mining: building blocks, scenarios, and implementation. *International Journal on Software Tools for Technology Transfer*, v. 18, n. 6, p. 607–628, 1 nov. 2016.

BOUCKAERT, Remco R *et al.* Weka manual for version 3-6-0. *University of Waikato, Hamilton, New Zealand*, 2008.

BRINK, Henrik; RICHARDS, Joseph; FETHEROLF, Mark. *Real-world machine learning*. [S.l.]: Manning Publications Co., 2016.

BUIJS, JCAM. Mapping data sources to xes in a generic way. *Maters Thesis*, 2010. Disponível em: <https://pure.tue.nl/ws/portalfiles/portal/46977079>. Acesso em: 1 jan. 2018.

BUITINCK, Lars *et al.* API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

ÇELA, Ornela; RIEU, Dominique; OTHERS. Model Consolidation: A Process Modelling Method Combining Process Mining and Business Process Modelling. *Enterprise, Business-Process and Information Systems Modeling*. [S.l.]: Springer, 2018. p. 117–130.

CHEN, Min; MAO, Shiwen; LIU, Yunhao. Big data: A survey. *Mobile networks and applications*, v. 19, n. 2, p. 171–209, 2014.

DANIEL, Florian; DUSTDAR, S; BARKAOUI, K. Process mining manifesto. 2011, [S.l: s.n.], 2011. p. 169–194.

DAVENPORT, Thomas H. *Big data no trabalho: derrubando mitos e descobrindo oportunidades*. [S.l.]: São Paulo: Campus, 2014.

DE LEONI, Massimiliano; VAN DER AALST, Wil MP; DEES, Marcus. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, v. 56, p. 235–257, 2016.

DOS SANTOS FERREIRA, Camila; GEROLAMO, Mateus Cecílio. Análise da relação entre normas de sistema de gestão (ISO 9001, ISO 14001, NBR 16001 e OHSAS 18001) e a sustentabilidade empresarial. n. 4, p. 689–703, 2016.

DRAZIN, Sam; MONTAG, Matt. Decision tree analysis using weka. *Machine Learning-Project II, University of Miami*, p. 1–3, 2012.

ESPOSITO, Pedro M *et al.* Uma abordagem para a mineração dos processos de uma universidade. *In: V Workshop on Business Process Management,* p. 485–492, 2011.

ESPOSITO, Pedro. M. MANA: identificação, mineração, análise e reengenharia de processos de Negócio. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

FEELEY, Shannon; JACOWAY, Ian; RIOS, Gina. Process Mining the Credit Suisse Advisory Process. 2017, [S.l.]: WORCESTER POLYTECHNIC INSTITUTE, 2017.

GHAWI, Raji. Process Discovery using Inductive Miner and Decomposition. *arXiv preprint arXiv:1610.07989*, 2016.

GODOY, Arilda Schmidt. Pesquisa qualitativa: tipos fundamentais. *Revista de Administração de empresas*, v. 35, n. 3, p. 20–29, 1995.

GÜNTHER, Christian W; ROZINAT, Anne. Disco: Discover Your Processes. *BPM (Demos)*, v. 940, p. 40–44, 2012.

GUPTA, Gaurav; MALHOTRA, Sumit. Text documents tokenization for word frequency count using rapid miner (taking resume as an example). *Int. J. Comput. Appl*, p. 0975–8887, 2015.

HAN, Jiawei; PEI, Jian; KAMBER, Micheline. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011.

HORNIX, Peter TG. Performance analysis of business processes through process mining. *Master's Thesis, Eindhoven University of Technology*, 2007.

ILYASOVA, N *et al.* Particular Use of BIG DATA in Medical Diagnostic Tasks. *Pattern Recognition and Image Analysis*, v. 28, n. 1, p. 114–121, 2018.

KALENKOVA, A.A.a *et al.* Process mining using BPMN: relating event logs and process models. *Software and Systems Modeling*, cited By 1; Article in Press, 2015. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84944916586&partnerID=40&md5=be23d25182d9cff3bb2885a599febe4d>.

KALENKOVA, A.A.a; DE LEONI, M.b; VAN DER AALST, W.M.P.a B. Discovering, analyzing and enhancing BPMN models using ProM. 2014, [S.l.]: CEUR-WS, 2014. p. 36–40. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84915766041&partnerID=40&md5=3cb028b66dc3bc6951284c616e0325b5>.

KRAMER, Oliver. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*. [S.l.]: Springer, 2013. p. 13–23.

KUMAR, MV; THOMAS, Likewin; ANNAPPA, B. Distilling Lasagna from Spaghetti Processes. 2017, [S.l.]: ACM, 2017. p. 157–161.

KUMAR, Vineet; REINARTZ, Werner. *Customer relationship management: Concept, strategy, and tools*. [S.l.]: Springer, 2012.

LAN, Man *et al.* Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, v. 31, n. 4, p. 721–735, 2009.

LEEMANS, Sander JJ; FAHLAND, Dirk; VAN DER AALST, Wil MP. Discovering block-structured process models from event logs-a constructive approach. 2013, [S.l.]: Springer, 2013. p. 311–329.

LEEMANS, Sander JJ; FAHLAND, Dirk; VAN DER AALST, Wil MP. Exploring processes and deviations. 2014, [S.l.]: Springer, 2014. p. 304–316.

LI, Jiexun; WANG, Harry Jiannan; BAI, Xue. An intelligent approach to data extraction and task identification for process mining. *Information Systems Frontiers*, v. 17, n. 6, p. 1195–1208, 2015.

MANNHARDT, Felix; DE LEONI, Massimiliano; REIJERS, Hajo A. The Multi-perspective Process Explorer. 2015, [S.l: s.n.], 2015. p. 130–134.

MANNHARDT, Felix; TAX, Niek. Unsupervised event abstraction using pattern abstraction and local process models. *arXiv preprint arXiv:1704.03520*, 2017.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. *Técnicas de pesquisa*. [S.l.]: São Paulo: Atlas, 2015. v. 7.

MARRANGHELLO, Norian. Redes de petri: Conceitos e aplicaçoes. *São Paulo: DCCE/IBILCE/UNESP*, 2005.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Indução de regras e árvores de decisão. *Sistemas Inteligentes. Rezende, SO Editora Manole Ltda*, p. 115–140, 2003.

OUTMAZGIN, Nesi; SOFFER, Pnina. A process mining-based analysis of business process work-arounds. *Software & Systems Modeling*, v. 15, n. 2, p. 309–323, 2016.

PÉREZ-CASTILLO, Ricardo *et al.* Assessing event correlation in non-process-aware information systems. *Software & Systems Modeling*, v. 13, n. 3, p. 1117–1139, 2014.

PERSSON, Sara. *Qualitative Methods in Software Engineering*. 2004. Department of Communication Systems, 2004. Disponível em: <http://fileadmin.cs.lth.se/serg/old-serg-dok/docs-masterthesis/59_Rep.5520.Persson.pdf>. Acesso em: 5 ago. 2018.

PREMCHAISWADI, Wichian; POROUHAN, Parham. Process modeling and bottleneck mining in online peer-review systems. *SpringerPlus*, v. 4, n. 1, p. 1, 2015.

RAMOS, Juan. Using tf-idf to determine word relevance in document queries. 2003, [S.l: s.n.], 2003. p. 133–142.

REED, Russell. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, v. 4, n. 5, p. 740–747, 1993.

REFAEILZADEH, Payam; TANG, Lei; LIU, Huan. Cross-validation. *Encyclopedia of database systems*, p. 1–7, 2008.

ROZINAT, Anne; VAN DER AALST, Wil MP. Conformance checking of processes based on monitoring real behavior. *Information Systems*, v. 33, n. 1, p. 64–95, 2008a.

ROZINAT, Anne; VAN DER AALST, Wil MP. Conformance checking of processes based on monitoring real behavior. *Information Systems*, v. 33, n. 1, p. 64–95, 2008b.

SINUR, J. The Business Rule Engine 2003 Magic Quadrant. *Gartner Group Research Note*, 2003.

SOUSA, Marco *et al.* Evaluation of BPM tools open source/freeware. 2018, [S.l.]: IEEE, 2018.

TAX, Niek *et al.* Event abstraction for process mining using supervised learning techniques. 2016, [S.l.]: Springer, 2016. p. 251–269.

VAN DER AALST, Wil. Business process management as the “Killer App” for Petri nets. *Software & Systems Modeling*, v. 14, n. 2, p. 685–691, 2015.

VAN DER AALST, Wil *et al.* Process mining manifesto. 2011, [S.l.]: Springer, 2011. p. 169–194.

VAN DER AALST, Wil MP. Decomposing Petri nets for process mining: A generic approach. *Distributed and Parallel Databases*, v. 31, n. 4, p. 471–507, 2013a.

VAN DER AALST, Wil MP. Decomposing Petri nets for process mining: A generic approach. *Distributed and Parallel Databases*, v. 31, n. 4, p. 471–507, 2013b.

VAN DER AALST, Wil MP *et al.* Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*, v. 9, n. 1, p. 87, 2010.

VAN DER AALST, Wil MP. *Process Mining: Data Science in Action*. 2. ed. [S.l.]: Springer, 2016.

VAN DER AALST, Wil MP. Process mining: discovering and improving Spaghetti and Lasagna processes. 2011, [S.l.]: IEEE, 2011. p. 1–7.

VAN DER AALST, Wil MP. Process mining in the large: a tutorial. *Business Intelligence*. [S.l.]: Springer, 2014. p. 33–76.

VAN DER AALST, Wil MP. Using Process Mining to Bridge the Gap between BI and BPM. *IEEE Computer*, v. 44, n. 12, p. 77–80, 2011.

VAN DER AALST, Wil MP; ADRIANSYAH, Arya; VAN DONGEN, Boudewijn. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 2, n. 2, p. 182–192, 2012.

VAN DER AALST, Wil MP; BOLT, Alfredo; VAN ZELST, Sebastiaan J. RapidProM: mine your processes and not just your data. *arXiv preprint arXiv:1703.03740*, 2017.

VAN DER AALST, Wil MP; REIJERS, Hajo A; SONG, Minseok. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, v. 14, n. 6, p. 549–593, 2005.

VAN DER AALST, Wil MP; VAN DONGEN, Boudewijn F. Discovering petri nets from event logs. *Transactions on Petri Nets and Other Models of Concurrency VII*. [S.l.]: Springer, 2013. p. 372–422.

VAN DER AALST, Wil MP; VERBEEK, HMW. Process discovery and conformance checking using passages. *Fundamenta Informaticae*, v. 131, n. 1, p. 103–138, 2014.

VAN DER AALST, Wil; WEIJTERS, Ton; MARUSTER, Laura. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge & Data Engineering*, n. 9, p. 1128–1142, 2004.

VAN LOOY, Amy. *Business process maturity: A comparative study on a sample of business process maturity models*. [S.l.]: Springer Science & Business Media, 2014.

VEIGA, Gabriel M; FERREIRA, Diogo R. Understanding spaghetti models with sequence clustering for ProM. 2009, [S.l.]: Springer, 2009. p. 92–103.

WASILEWSKI, Adam. Business process management suite (BPMS) market changes 2009- 2015. *Information Systems in Management*, v. 5, 2016. Disponível em: <file:///C:/Users/rosan/Downloads/Wasilewski_2016%20vol5(4).pdf>. Acesso em: 7 jul. 2017.

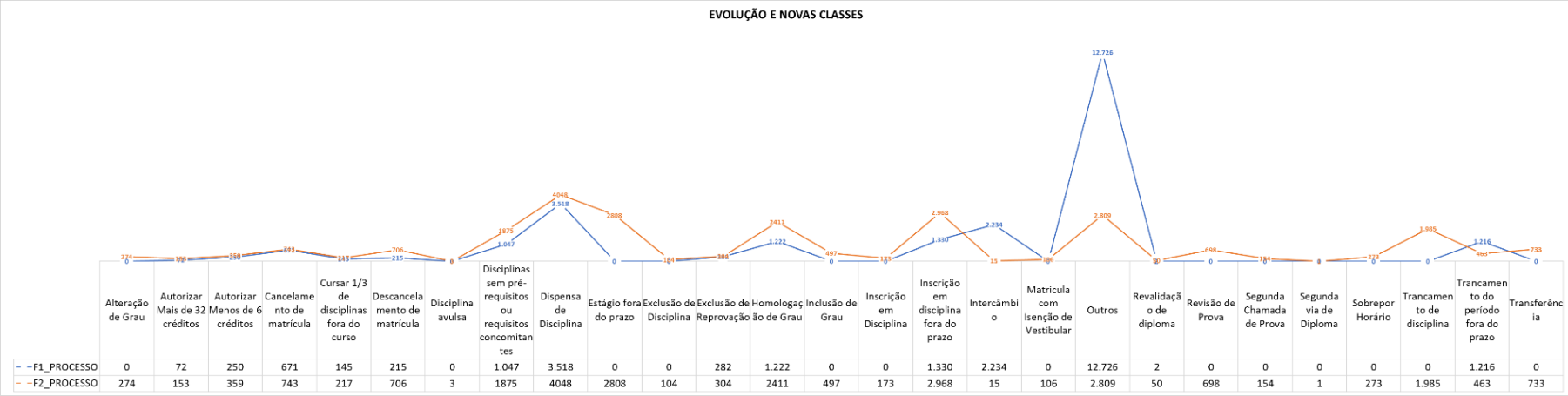
WEIJTERS, AJMM; VAN DER AALST, Wil MP; DE MEDEIROS, AK Alves. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, v. 166, p. 1–34, 2006.

WITTEN, Ian H *et al. Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2016.

YANG, Wan-Shiou; HWANG, San-Yih. A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, v. 31, n. 1, p. 56–68, 2006.

Apêndice A – Experimento

A Figura 35 – Evolução das classes mostra mais detalhes sobre a evolução do número de documentos de cada classe.



	Alteração de Grau	Autorizar Mais de 32 créditos	Autorizar Menos de 6 créditos	Cancelamento de matrícula	Cursar 1/3 de disciplinas fora do curso	Descancelamento de matrícula	Disciplina avulsa	Disciplinas sem pré-requisitos ou requisitos concomitantes	Dispensa de Disciplina	Estágio fora do prazo	Exclusão de Disciplina	Exclusão de Reprovação	Homologação de Grau	Inclusão de Grau	Inscrição em Disciplina	Inscrição em disciplina fora do prazo	Intercâmbio	Matricula com Isenção de Vestibular	Outros	Revalidação de diploma	Revisão de Prova	Segunda Chamada de Prova	Segunda via de Diploma	Sobrepor Horário	Trancamento de disciplina	Trancamento do período fora do prazo	Transferência
F1_PROCESSO	0	72	250	671	145	215	0	1.047	3.518	0	0	282	1.222	0	0	1.330	2.234	0	12.726	2	0	0	0	0	0	1.216	0
F2_PROCESSO	274	153	359	743	217	706	3	1875	4048	2808	104	304	2411	497	173	2.968	15	106	2.809	50	698	154	1	273	1.985	463	733

Figura 35 – Evolução das classes

O gráfico da Figura 36 mostra em detalhes a métrica apresentada para a classe com o texto original sem o pré-processamento.

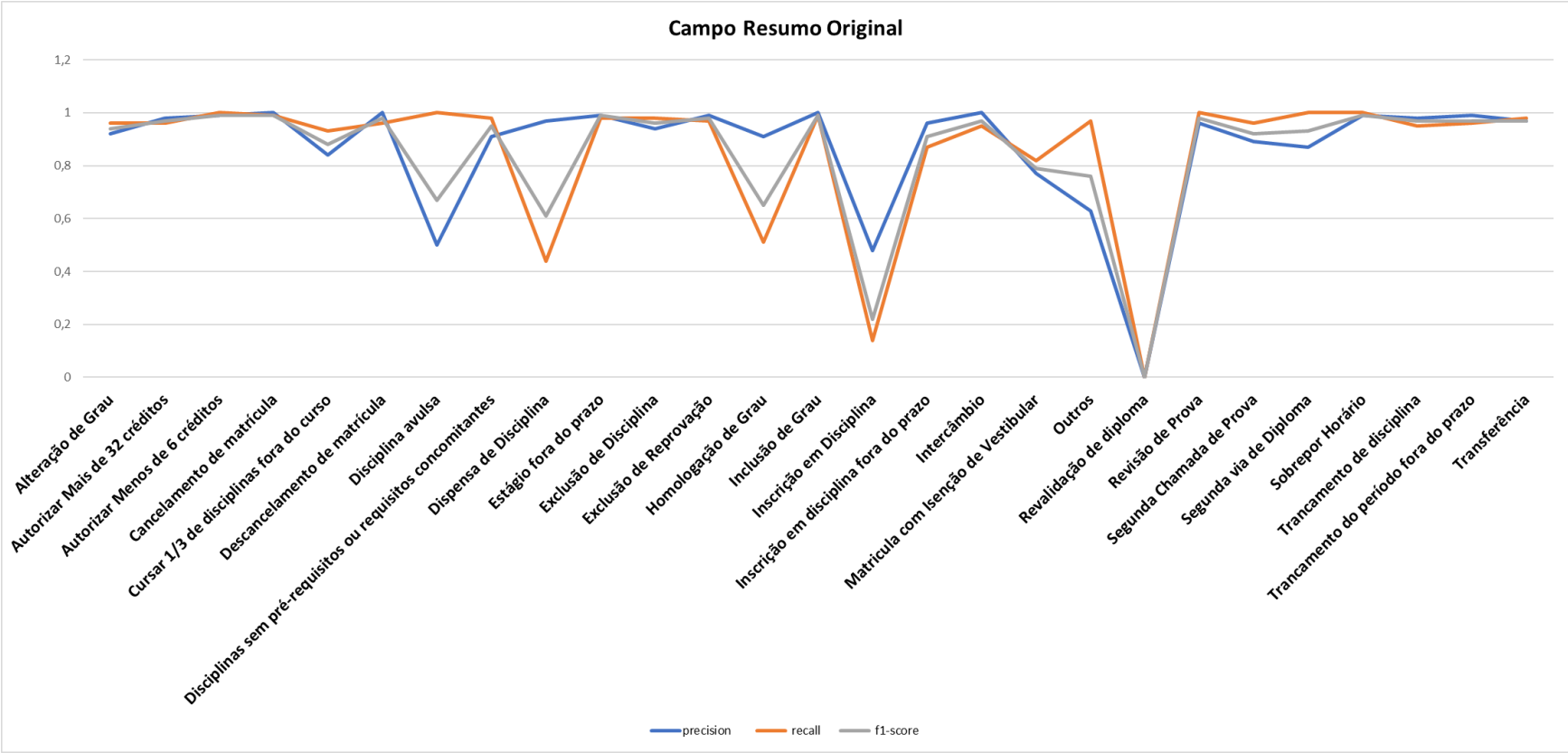


Figura 36 – KNN *scikit-learn* – Aplicação sobre o resumo original

As figuras Figura 35 apresenta no gráfico as métricas processadas no classificadores KNN no weka e no scilit-learn apresentado as metricas para o resumo tratado e o resumo original.

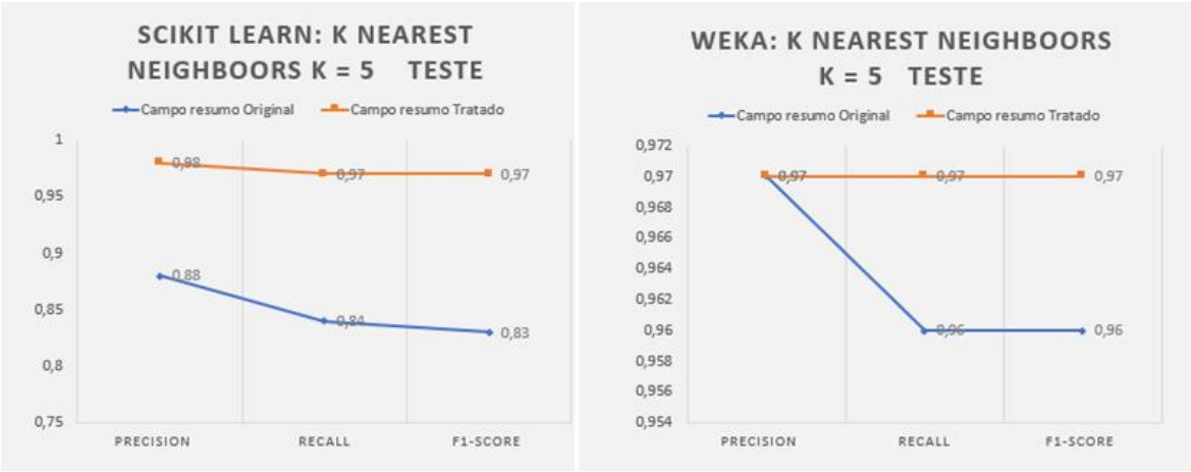


Figura 37 – Métricas do classificador KNN no Weka e no Scikit-learn

As figura Figura 37 apresenta no gráfico as métricas processadas no classificadores naive Bayes no weka e no scilit-learn apresentado as métricas para o resumo tratado e o resumo original.

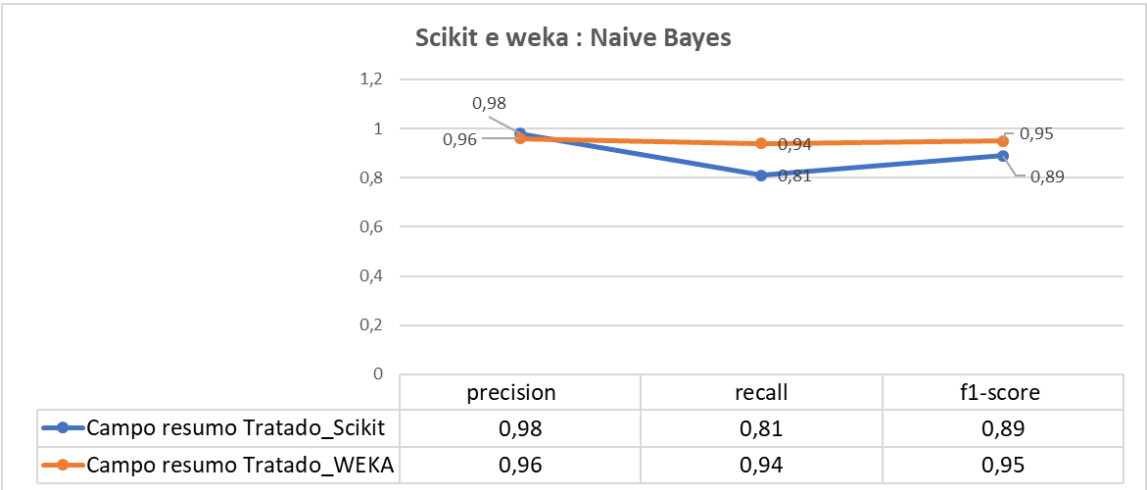


Figura 38 – Métricas do classificador Naive Bayes no Weka e no Scikit-learn

As Figura 39 e Figura 40 mostram o processo da classe “sobrepor horário” com o uso do *plugin* de Multi-perspective, que possui por objetivo avaliar as dimensões de qualidade.

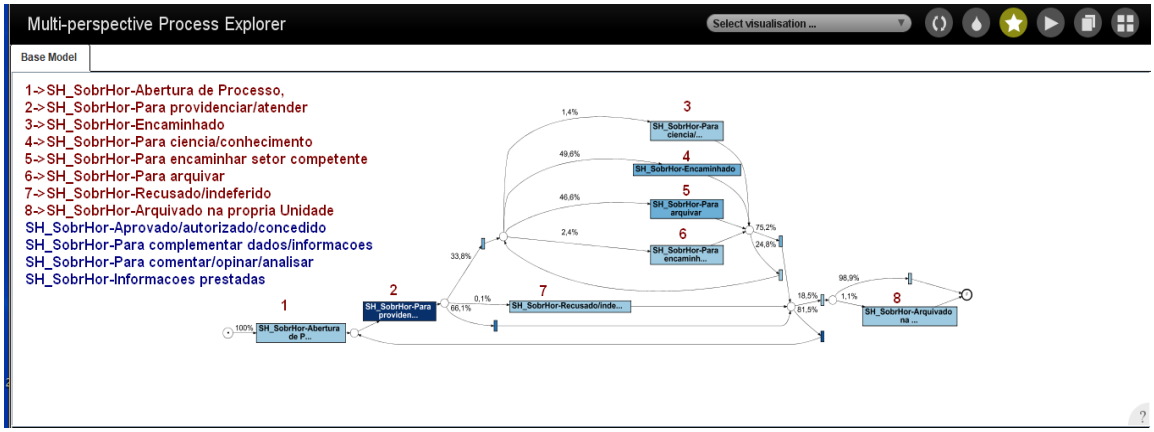


Figura 39 – Quatro dimensões usando o *plugin* DataAwareExplorer – gerado no ProM

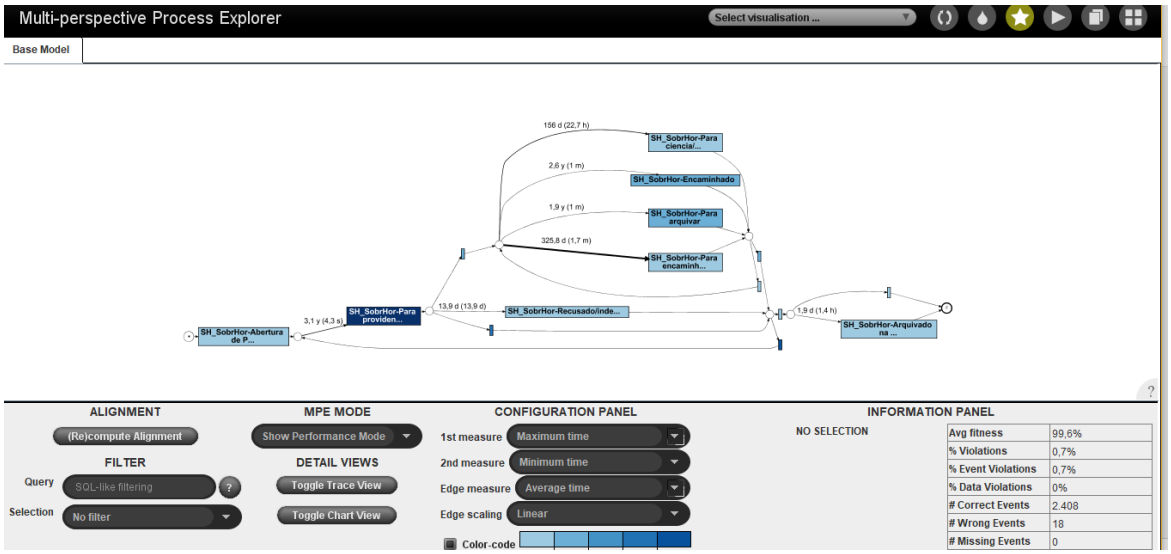


Figura 40 – Processo da classe sobrepor horário – gerado no ProM

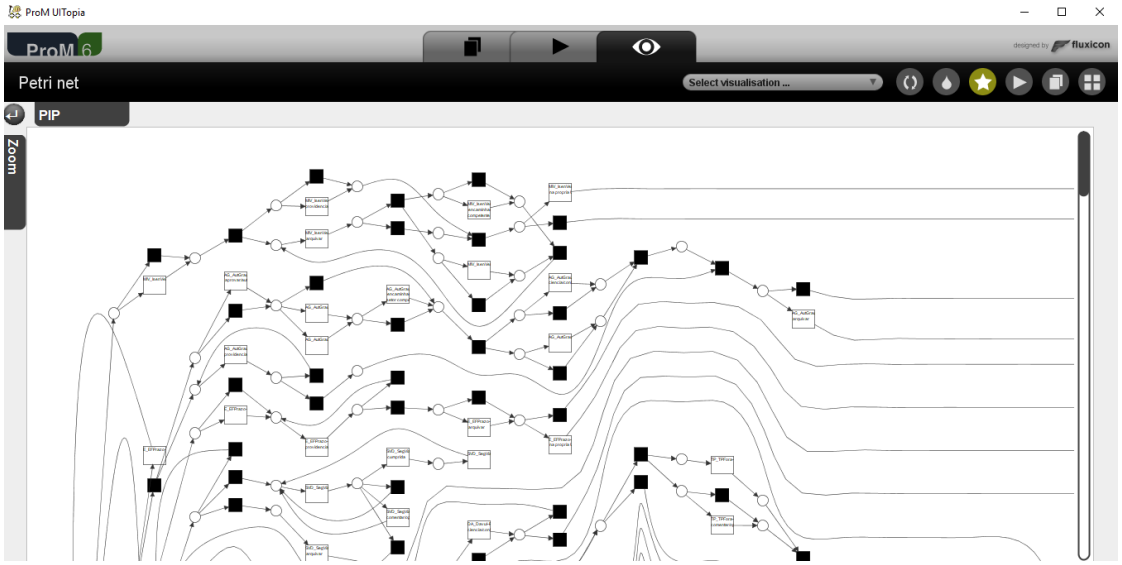


Figura 41 – Modelo gerado no *plugin* Petri Net – gerado no ProM

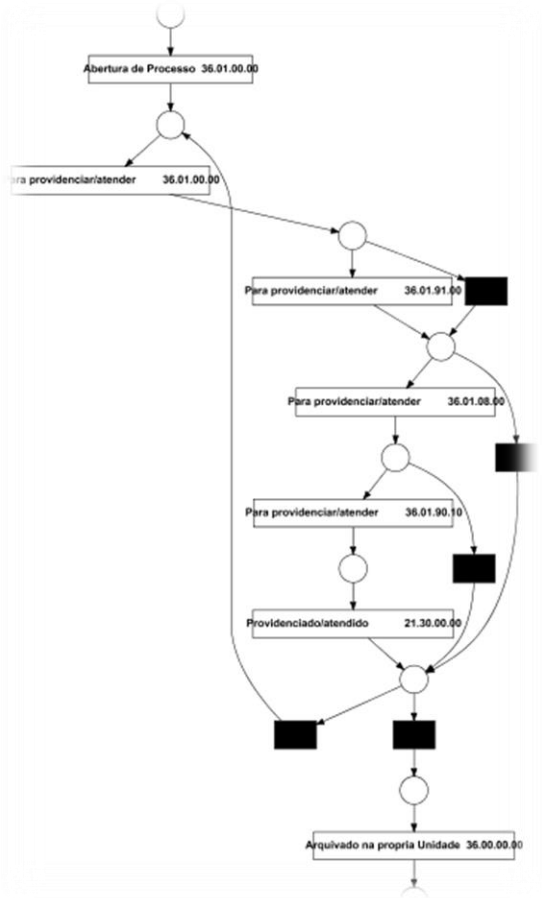


Figura 42 – Mine Petri net With Inductive Miner com Visualize Petri NET (dot) – gerado no ProM

A Figura 43, mostra o modelo de processo BPMN com transformação de Petri Net para BPMN.

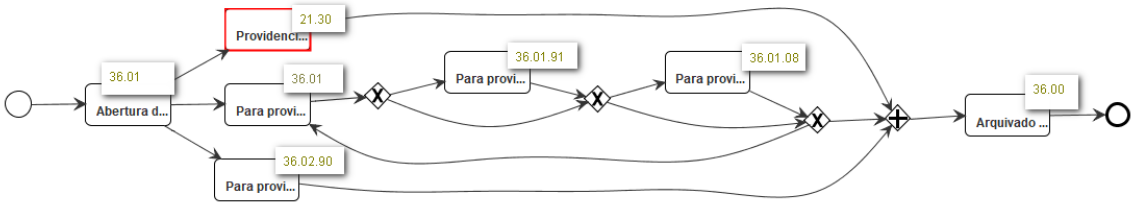


Figura 43 – Modelo BPMN – *Plugin* ProM

Tabela 19 – Agrupamento de aptidão

CLASSE: Dispensa de disciplinas	**Trace Fitness**															
FOLDER1	0,8	0,83	0,88	0,91	0,92	0,94	0,95	0,96	0,97	1	***	***	***	***	***	***
	2	1	2	3	3	1	1	1	1	479	***	***	***	***	***	***
FOLDER2	0,86	0,87	0,88	0,89	0,91	0,92	0,93	0,94	0,95	0,97	1	***	***	***	***	***
	1	1	2	2	4	3	1	2	2	1	489	***	***	***	***	***
FOLDER3	0,78	0,86	0,87	0,88	0,89	0,91	0,92	0,93	0,94	0,95	0,96	0,97	1	***	***	***
	1	1	1	1	2	3	3	1	1	2	1	1	501	***	***	***
FOLDER4	0,75	0,83	0,86	0,88	0,9	0,91	0,92	0,94	0,96	1	***	***	***	***	***	***
	1	1	1	2	3	2	1	2	2	490	***	***	***	***	***	***
FOLDER5	0,86	0,89	0,91	0,92	1	***	***	***	***	***	***	***	***	***	***	***
	1	2	1	1	491	***	***	***	***	***	***	***	***	***	***	***
Autorizar Mais de 32 créditos	**Trace Fitness**															
FOLDER1	1	2	1	1	1	2	3	1	3	2	1	2	1	31	***	***
	0,79	0,81	0,83	0,86	0,87	0,88	0,89	0,9	0,91	0,92	0,93	0,94	0,96	1	***	***
FOLDER2	1	2	1	1	1	2	3	1	3	2	1	2	1	31	***	***
	0,79	0,81	0,83	0,86	0,87	0,88	0,89	0,9	0,91	0,92	0,93	0,94	0,96	1	***	***
FOLDER3	1	1	3	2	2	1	1	1	1	36	***	***	***	***	***	***
	0,83	0,87	0,88	0,89	0,9	0,91	0,92	0,93	0,94	1	***	***	***	***	***	***
FOLDER4	1	2	1	2	2	2	1	2	1	36	***	***	***	***	***	***
	0,79	0,81	0,86	0,88	0,89	0,9	0,91	0,94	0,96	1	***	***	***	***	***	***
FOLDER5	1	1	1	2	3	1	1	2	41	***	***	***	***	***	***	***
	0,83	0,84	0,87	0,88	0,89	0,9	0,93	0,94	1	***	***	***	***	***	***	***
Alteração de Grau	**Trace Fitness**															
FOLDER1	1	1	1	72	***	***	***	***	***	***	***	***	***	***	***	***
	0,83	0,9	0,93	1	***	***	***	***	***	***	***	***	***	***	***	***
FOLDER2	1	2	2	1	3	2	2	3	1	61	***	***	***	***	***	***
	0,75	0,78	0,82	0,88	0,89	0,9	0,92	0,93	0,94	1	***	***	***	***	***	***
FOLDER3	1	1	4	1	2	2	2	3	4	4	3	4	3	3	3	43
	0,6	0,75	0,8	0,81	0,83	0,87	0,88	0,89	0,9	0,91	0,92	0,93	0,94	1	1	1
FOLDER4	1	1	1	1	2	1	1	2	2	3	1	69	***	***	***	***
	0,8	0,81	0,83	0,87	0,88	0,89	0,9	0,91	0,92	0,93	0,95	1	***	***	***	***
FOLDER5	1	75	***	***	***	***	***	***	***	***	***	***	***	***	***	***
	0,95	1	***	***	***	***	***	***	***	***	***	***	***	***	***	***

As tabelas Tabela 20, Tabela 21 e Tabela 22 mostram a avaliação dos cinco *folders* em relação às atividades que contemplam. Podemos verificar que a maioria das atividades está contemplada nos cinco *folders*, no entanto, para poucos casos, há aderência inferior a 50%.

Tabela 20 – Aderência de atividades por *folder*: classe "Alteração de Grau"

Atividades distintas	***Folder1***	***Folder2***	***Folder3***	***Folder4***	***Folder5***	**Aderência por *folder***
Abertura de Processo	✓	✓	✓	✓	✓	100%
Aprovado/autorizado/concedido	√	√			√	60%
Arquivado na propria Unidade	✓	✓	✓	✓	✓	100%
Comentado/opinado		√			√	40%
Encaminhado	✓	✓	✓	✓	✓	100%
Para aprovar/autorizar	√				√	40%
Para arquivar	✓	✓	✓	✓	✓	100%
Para ciencia/conhecimento	✓	✓	✓	✓	✓	100%
Para comentar/opinar/analisar	√					20%
Para encaminhar setor competente	√	√	√		√	80%
Para providenciar/atender	✓	✓	✓	✓	✓	100%
Providenciado/atendido	√	√		√	√	80%

Tabela 21 – Aderência de atividades por *folder*: classe “Dispensa de disciplina”

Atividades distintas	*Folder1*	*Folder2*	*Folder3*	*Folder4*	*Folder5*	Aderência por *folder*
Abertura de Processo	✓	✓	✓	✓	✓	100%
Aprovado/autorizado/concedido			√	√	√	60%
Arquivado na propria Unidade	✓	✓	✓	✓	✓	100%
Encaminhado	✓	✓	✓	✓	✓	100%
Exigência cumprida					√	10%
Para aprovar/autorizar	√	√				40%
Para arquivar	✓	✓	✓	✓	✓	100%
Para ciencia/conhecimento	✓	✓	✓	✓	✓	100%
Para comentar/opinar/analisar	√			√	√	60%
Para encaminhar setor competente	✓	✓	✓	✓	✓	100%
Para esclarecer	√					10%
Para providenciar/atender	✓	✓	✓	✓	✓	100%
Para solicitar informações	✓	✓	✓		✓	80%
Providenciado/atendido	✓	✓	✓	✓	✓	100%

Tabela 22 – Aderência de atividades por *folder*: classe “Exclusão de reprovação”

Atividades distintas	*Folder1*	*Folder2*	*Folder3*	*Folder4*	*Folder5*	Aderência por *folder*
Abertura de Processo	✓	✓	✓	✓	✓	100%
Aprovado/autorizado/concedido		√	√	√		60%
Arquivado na propria Unidade	✓	✓	✓	✓	✓	100%
Encaminhado	√		√	√	√	80%
Exigência cumprida	√				√	20%
Para arquivar	✓	✓	✓	✓	✓	100%
Para ciencia/conhecimento	✓	✓	✓	✓	✓	100%
Para comentar/opinar/analisar	✓	✓	✓	✓	✓	100%
Para complementar dados/informações			√	√		20%
Para cumprir exigência	√	√				20%
Para encaminhar setor competente	√		√	√	√	80%
Para providenciar/atender	✓	✓	✓	✓	✓	100%
Providenciado/atendido	√	√	√		√	80%

A utilização da tokenização do texto com a ferramenta Rapidminer gerou uma lista de palavras conforme Figura 44.

Result History | WordList (Process Documents from Files)

Data

Word	Attribute Name	Total Occurences
TRANSF	TRANSF	7
TRANSFENCIA	TRANSFENCIA	1
TRANSFERECIA	TRANSFERECIA	5
TRANSFERENCA	TRANSFERENCA	1
TRANSFERENCIA	TRANSFERENCIA	885
TRANSFERENCIAP	TRANSFERENCIAP	2
TRANSFERENÇIA	TRANSFERENÇIA	36
TRANSFERNCIA	TRANSFERNCIA	5
TRANSFERÊNCIA	TRANSFERÊNCIA	1014
TRANSFERÊNÇIA	TRANSFERÊNÇIA	2
TRANSFETRENCIA	TRANSFETRENCIA	1
TRANSFRENCIA	TRANSFRENCIA	1
TRANSFÊRENCIA	TRANSFÊRENCIA	2
TRANSMISSAO	TRANSMISSAO	1

Figura 44 – Tokenização do campo “resumo” no Rapidminer

Apêndice B – Estudo de caso: detalhes

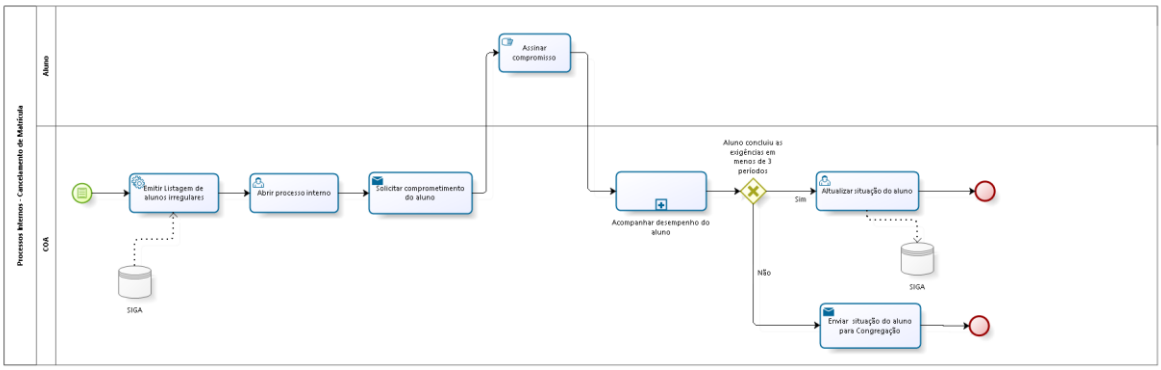


Figura 45 – Tratamento de Irregularidade via Processo

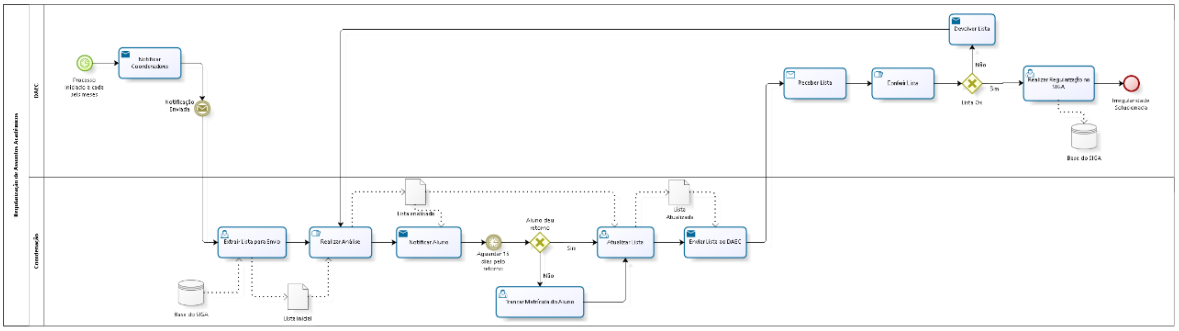


Figura 46 – Tratamento de Irregularidade via Memorando

Tabela 23 – Total de processos por ano entre 2005 e 2016 (Centro de Tecnologia e Escola Politécnica)

Total da unidade 36.00.00.00 desde 2005		Total da unidade 36.01.00.00 desde 2005	
Quantidade	Ano	Quantidade	Ano
1360	2016	593	2016
2022	2015	522	2015
1217	2014	2	2014
1480	2013	1	2013
1278	2012	1	2012
1245	2011	6	2011
1486	2010	2	2010
1284	2009	5	2009
1006	2008	1	2006
532	2007	10	2005
513	2006	11	2004
335	2005	9	2003

Apêndice C – Classificadores

A Tabela 24 mostra as medidas de precisão, *recall* e F-Measure ao utilizar o *dataset* no primeiro experimento.

A Tabela 25 demonstra a evolução das classes com as métricas de classificação. Neste experimento, foram utilizados os mesmos parâmetros dos da Tabela 24.

A Tabela 26, a Tabela 27 e a Tabela 28 mostram as métricas dos experimentos utilizando os três classificados. As medidas foram extraídas dos classificadores utilizando o pacote Python *Scikit-learn* e Weka Datamining Tool Kit, utilizando 70% do dataset como treino e 30% como.

Tabela 24 – Experimento II – Weka Classificador KNN

Classes	Precision	Recall	F-Measure
Autorizar Mais de 32 créditos	0,574	0,874	0,673
Autorizar Menos de 6 créditos	0,35	-	0,311
Cancelamento de matrícula	0,995	-	0,997
Cursar 1/3 de disciplinas fora do curso	0,904	0,996	0,949
Descancelamento de matrícula	0,951	1	0,975
Disciplina avulsa	0	1	0
Disciplinas sem pré-requisitos ou requisitos concomitantes	0,953	1	0,974
Dispensa de Disciplina	0,985	1	0,992
Estágio fora do prazo	0	1	0
Exclusão de Reprovação	0,993	1	0,997
Homologação de Grau	0,982	-	0,991
Inscrição em disciplina fora do prazo	0,982	0,814	0,992
Intercâmbio	-	1	-
Outros	0,996	1	0,931
Revalidação de diploma	-	0,28	-
Trancamento do período fora do prazo	0,994	0	0,997
Transferência	-	0	-

Tabela 25 – Experimento I – Weka Classificador KNN

Classes	Precision	Recall	F-Measure
Alteração de Grau	0,957	0,986	0,972
Autorizar Mais de 32 créditos	0,994	1	0,997
Autorizar Menos de 6 créditos	0,991	0,999	0,995
Cancelamento de diploma	-	0	-
Cancelamento de matrícula	0,993	0,998	0,995
Cursar 1/3 de disciplinas fora do curso	0,953	0,98	0,967
Cursar um terço de disciplinas fora do curso	0,796	0,723	0,758
Descancelamento de matrícula	0,992	0,999	0,996
Disciplina avulsa	1	0,667	0,8
Disciplinas sem pré-requisitos ou requisitos concomitantes	0,971	0,997	0,984
Dispensa de Disciplina	0,983	0,999	0,991
Estágio fora do prazo	0,989	0,998	0,993
Exclusão de Disciplina	0,906	0,994	0,948
Exclusão de Reprovação	0,984	0,997	0,991
Homologação de Grau	0,992	1	0,996
Inclusão de Grau	0,975	0,996	0,985
Inscrição dem disciplina fora do prazo	0,972	0,996	0,984
Inscrição em Disciplina	0,676	0,63	0,652
Inscrição em disciplina fora do prazo	0,714	0,444	0,548
Intercâmbio	0,97	0,956	0,963
Matricula com Isenção de Vestibular	0,707	0,642	0,673
Outros	0,987	0,944	0,965
Revalidação de diploma	-	0	-
Revisão de Prova	0,831	1	0,908
Segunda Chamada de Prova	0,88	0,968	0,922
Segunda via de Diploma	0,857	0,894	0,875
Sobrepor Horário	0,99	1	0,995
Trancamento de Disciplina	0,727	0,364	0,485
Trancamento de disciplina	0,982	0,991	0,987
Trancamento do período fora do prazo	0,983	0,997	0,99
Transferência	0,961	0,985	0,973

Tabela 26 – Classificador KNN no Weka e Python: 70% de Treino e 30 % e Teste

Classes	Python			Weka		
	Precisão	Recall	F1	Precisão	Recall	F1
Alteração de Grau	0.95	0.98	0.96	0,94	0,93	0,94
Autorizar Mais de 32 créditos	0.99	0.99	0.99	1,00	1,00	1,00
Autorizar Menos de 6 créditos	0.99	1.00	0.99	0,98	1,00	0,99
Cancelamento de matrícula	0.99	1.00	0.99	0,99	1,00	1,00
Cursar 1/3 de disciplinas fora do curso	0.90	0.95	0.92	0,96	0,88	0,91
Descancelamento de matrícula	1.00	0.99	0.99	0,99	1,00	0,99
Disciplina avulsa	0.00	0.00	0.00	-	-	-
Disciplinas sem pré-requisitos ou requisitos concomitantes	0.96	0.97	0.97	0,97	1,00	0,98
Dispensa de Disciplina	0.99	0.99	0.99	0,98	1,00	0,99
Estágio fora do prazo	1.00	0.99	0.99	0,96	1,00	0,98
Exclusão de Disciplina	0.90	1.00	0.95	0,78	1,00	0,88
Exclusão de Reprovação	1.00	1.00	1.00	0,97	1,00	0,98
Homologação de Grau	1.00	1.00	1.00	0,98	1,00	0,99
Inclusão de Grau	1.00	0.99	0.99	0,95	1,00	0,97
Inscrição em Disciplina	0.44	0.45	0.44	0,61	0,68	0,64
Inscrição em disciplina fora do prazo	0.98	0.99	0.99	0,98	1,00	0,99
Intercâmbio	0.97	0.97	0.97	0,97	1,00	0,99
Matricula com Isenção de Vestibular	0.75	0.70	0.72	0,79	0,69	0,73
Outros	0.97	0.95	0.96	0,99	0,92	0,95
Revalidação de diploma	0.00	0.00	0.00	-	-	-
Revisão de Prova	0.94	1.00	0.97	0,77	1,00	0,87
Segunda Chamada de Prova	0.86	0.97	0.91	0,78	1,00	0,87
Segunda via de Diploma	0.85	0.93	0.89	0,88	0,93	0,90
Sobrepor Horário	0.99	1.00	1.00	0,97	1,00	0,98
Trancamento de disciplina	0.99	0.99	0.99	0,99	0,99	0,99
Trancamento do período fora do prazo	1.00	0.98	0.99	0,97	0,98	0,98
Transferência	0.96	0.97	0.97	0,95	1,00	0,97

Tabela 27 – Classificador *Decision tree* no Weka e Python: 70% de Treino e 30 % e Teste

Classes	Python			Weka		
	Precisão	Recall	F1	Precisão	Recall	F1
Alteração de Grau	0.97	0.98	0.98	1,00	0,93	0,96
Autorizar Mais de 32 créditos	0.99	1.00	1.00	1,00	1,00	1,00
Autorizar Menos de 6 créditos	0.99	1.00	1.00	0,99	1,00	1,00
Cancelamento de matrícula	0.99	1.00	0.99	1,00	1,00	1,00
Cursar 1/3 de disciplinas fora do curso	0.91	0.94	0.93	0,98	0,89	0,93
Descancelamento de matrícula	1.00	1.00	1.00	1,00	1,00	1,00
Disciplina avulsa	0.67	1.00	0.80	1,00	1,00	1,00
Disciplinas sem pré-requisitos ou requisitos concomitantes	0.95	1.00	0.97	0,97	1,00	0,98
Dispensa de Disciplina	0.99	1.00	0.99	1,00	1,00	1,00
Estágio fora do prazo	1.00	1.00	1.00	1,00	1,00	1,00
Exclusão de Disciplina	0.97	0.98	0.98	0,88	1,00	0,93
Exclusão de Reprovação	1.00	1.00	1.00	0,97	1,00	0,99
Homologação de Grau	1.00	1.00	1.00	0,99	1,00	1,00
Inclusão de Grau	1.00	0.99	0.99	1,00	1,00	1,00
Inscrição em Disciplina	0.61	0.50	0.55	0,64	0,68	0,66
Inscrição em disciplina fora do prazo	0.98	0.99	0.99	0,99	0,99	0,99
Intercâmbio	0.98	0.99	0.98	1,00	1,00	1,00
Matricula com Isenção de Vestibular	0.68	0.72	0.70	0,81	0,91	0,85
Outros	0.98	0.95	0.97	0,99	0,96	0,97
Revalidação de diploma	0.00	0.00	0.00	-	-	-
Revisão de Prova	0.89	0.96	0.92	1,00	0,83	0,91
Segunda Chamada de Prova	0.87	0.93	0.90	0,79	0,98	0,88
Segunda via de Diploma	0.85	0.97	0.91	0,93	0,93	0,93
Sobrepor Horário	0.99	1.00	0.99	0,99	1,00	0,99
Trancamento de disciplina	0.99	1.00	0.99	0,98	1,00	0,99
Trancamento do período fora do prazo	1.00	1.00	1.00	1,00	0,99	0,99
Transferência	0.97	0.99	0.98	0,96	1,00	0,98

Tabela 28 – Classificador Naive Bayes no Weka e Python (Treino e Teste com 70% e 30%)

Classes	Python			Weka		
	Precisão	Recall	F1	Precisão	Recall	F1
Alteração de Grau	0.98	0.94	0.96	0,87	0,95	0,90
Autorizar Mais de 32 créditos	0.99	0.92	0.95	0,98	1,00	0,99
Autorizar Menos de 6 créditos	0.97	1.00	0.99	0,99	1,00	0,99
Cancelamento de matrícula	0.96	0.99	0.98	0,97	1,00	0,99
Cursar 1/3 de disciplinas fora do curso	0.85	0.99	0.92	0,91	0,97	0,94
Descancelamento de matrícula	1.00	1.00	1.00	0,98	1,00	0,99
Disciplina avulsa	0.00	0.00	0.00	1,00	1,00	1,00
Disciplinas sem pré-requisitos ou requisitos concomitantes	0.92	1.00	0.96	0,98	0,83	0,90
Dispensa de Disciplina	0.98	1.00	0.99	0,98	1,00	0,99
Estágio fora do prazo	0.97	1.00	0.98	0,98	1,00	0,99
Exclusão de Disciplina	0.00	0.00	0.00	0,66	1,00	0,79
Exclusão de Reprovação	0.99	1.00	0.99	0,99	1,00	1,00
Homologação de Grau	0.98	1.00	0.99	0,99	1,00	1,00
Inclusão de Grau	0.98	0.99	0.98	0,97	1,00	0,99
Inscrição em Disciplina	0.00	0.00	0.00	0,25	0,81	0,38
Inscrição em disciplina fora do prazo	0.94	0.99	0.97	0,98	0,99	0,98
Intercâmbio	1.00	0.85	0.92	1,00	1,00	1,00
Matricula com Isenção de Vestibular	0.00	0.00	0.00	0,47	1,00	0,64
Outros	0.94	0.92	0.93	0,99	0,85	0,92
Revalidação de diploma	0.00	0.00	0.00	-	-	-
Revisão de Prova	0.96	0.50	0.66	0,91	0,83	0,86
Segunda Chamada de Prova	0.86	0.86	0.86	0,78	1,00	0,87
Segunda via de Diploma	0.00	0.00	0.00	0,83	1,00	0,91
Sobrepor Horário	0.99	1.00	1.00	0,98	1,00	0,99
Trancamento de disciplina	0.98	1.00	0.99	0,96	0,96	0,96
Trancamento do período fora do prazo	0.99	0.99	0.99	0,75	0,99	0,85
Transferência	0.94	0.97	0.95	0,93	0,98	0,95

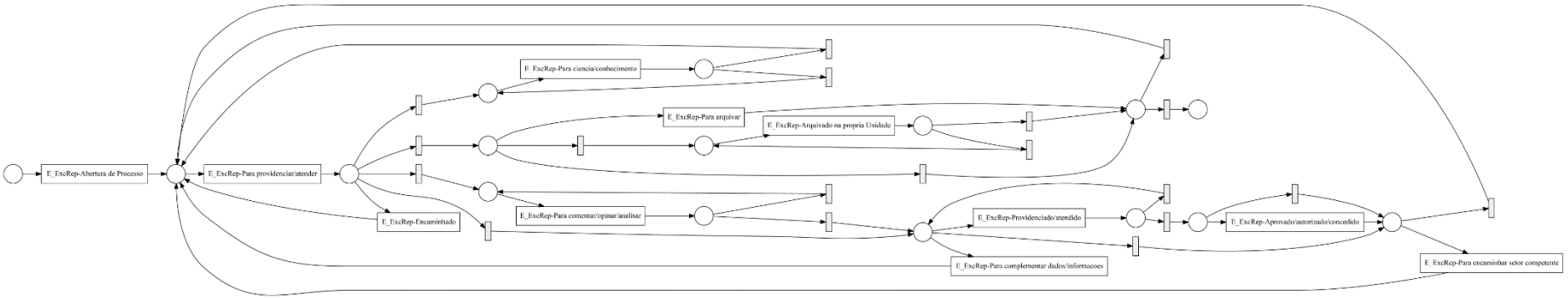


Figura 47 – Classe Exclusão de Reprovação Modelo do folder 3 (inicial)

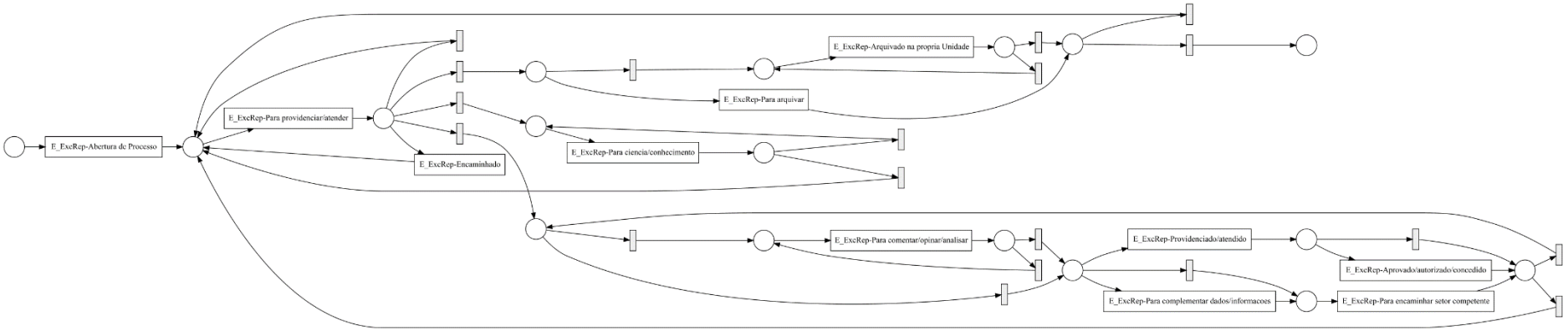


Figura 48 – Classe Exclusão de Reprovação Modelo do folder 3 (Adequado)