

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Matheus Friedhein Flores

**PROPOSTA DE UMA ARQUITETURA PARA DETECÇÃO DE
SITUAÇÕES DE INTERESSE E AÇÕES PROATIVAS
EM PROCESSOS DE NEGÓCIOS**

Santa Maria, RS
2021

Matheus Friedhein Flores

**PROPOSTA DE UMA ARQUITETURA PARA DETECÇÃO DE
SITUAÇÕES DE INTERESSE E AÇÕES PROATIVAS
EM PROCESSOS DE NEGÓCIO**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

Orientador: Prof. Dr. Alencar Machado
Coorientador: Prof. Dr. Jonas Bulegon Gassen

Santa Maria, RS
2021

Flores, Matheus Friedhein

Proposta de Uma Arquitetura para Detecção de Situações de Interesse e Ações Proativas em Processos de Negócio/ Matheus Friedhein Flores. - 2021.

76p.; 30 cm

Orientador: Dr. Alencar Machado

Coorientador: Dr. Jonas Bulegon Gassen

Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-graduação em Ciência da Computação, Santa Maria, RS, 2021.

1. Apache Kafka 2. Apache Spark 3. Logs de Eventos 4. Processos de Negócio. I. Machado, Alencar II. Flores, Matheus Friedhein III. Gassen, Jonas Bulegon IV. Proposta de Uma Arquitetura para Detecção de Situações de Interesse e Ações Proativas em Processos de Negócio.

© 2021

Todos os direitos autorais reservados a Matheus Friedhein Flores. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

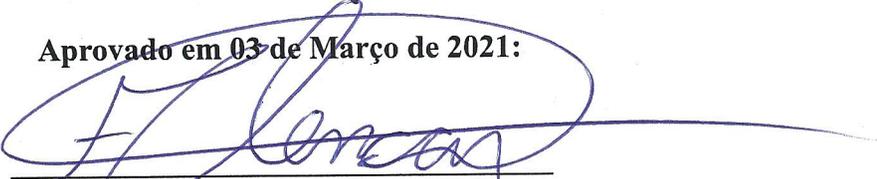
E-mail: matheusfriedhein@gmail.com

Matheus Friedhein Flores

**PROPOSTA DE UMA ARQUITETURA PARA DETECÇÃO DE
SITUAÇÕES DE INTERESSE E AÇÕES PROATIVAS
EM PROCESSOS DE NEGÓCIO**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Aprovado em 03 de Março de 2021:



Alencar Machado, Dr. (UFSM)
(Presidente/Orientador)



Jonas Bulegon Gassen, Dr. (AMF)
(Coorientador)



Ana Marilza Pernas Fleischmann, Dr^a. (UFPEL)



João Carlos Damasceno Lima, Dr. (UFSM)

Santa Maria, RS
2021

DEDICATÓRIA

À minha família

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois sem ele nada disso seria possível, a minha família, minha mãe Vera Regina Friedhein Flores, meu pai Flademir de Moraes Flores e meus irmãos Thomaz Friedhein Flores e Flaviane Friedhein Flores, os quais sempre me apoiaram e me incentivaram ao longo desta etapa. Desde o início, onde em muitas situações pensei em desistir de tudo, mas eles nunca me abandonaram e sempre incentivaram a persistir. Agradeço pelos ensinamentos, e pela educação que tive que fizeram me tornar a pessoa que sou hoje.

Agradeço a minha namorada Victoria Declerque, pela paciência, companhia e conversas, estando ao meu lado ao longo desta etapa, sempre disposta a ouvir meus medos e me acalmando para que chegasse ao fim. Obrigado pelo apoio e compreensão.

Agradeço aos meus orientadores, Alencar Machado e Jonas Bulegon Gassen, por sempre me orientar e ensinar da melhor maneira possível, pela oportunidade de aprendizado, pelos questionamentos e discussões ao longo do desenvolvimento deste trabalho. Agradeço também ao Vinícius Maran, por ter feito parte desta etapa, sempre disposto a ajudar, fazendo questionamentos e elencando itens importantes que me fizeram refletir inúmeras vezes ao longo deste trabalho.

Agradeço ao Programa de Pós Graduação em Ciência da Computação da Universidade Federal de Santa Maria, incluindo coordenação, secretaria, professores e colegas pela oportunidade, momentos, e conhecimento adquirido ao longo desses anos, que com certeza contribuíram muito para o meu crescimento pessoal e profissional.

Agradeço a Animati netPACS, amigos e colegas de trabalho, pela paciência e horários flexíveis em minhas ausências nas atividades da empresa, em especial para Lia Rosa, Cátia da Silva, Jean Berni, Cesar Souza, Cesar Frantz, Joedeson Junior e William Floriano.

Grato a todos que de alguma forma me apoiaram e incentivaram para chegar até aqui!

Obrigado!

RESUMO

PROPOSTA DE UMA ARQUITETURA PARA DETECÇÃO DE SITUAÇÕES DE INTERESSE E AÇÕES PROATIVAS EM PROCESSOS DE NEGÓCIO

AUTOR: Matheus Friedhein Flores
ORIENTADOR: Alencar Machado
COORIENTADOR: Jonas Bulegon Gassen

Com o avanço da tecnologia e a busca constante das organizações em melhorar seus processos organizacionais, visando maximizar o uso de recursos, diversas organizações estão investindo em melhorias no fluxo de seus processos internos. De forma geral, todas as organizações possuem tarefas que são realizadas por diferentes colaboradores, dada sua função. Um conjunto de tarefas, condições e eventos caracteriza um processo. As tarefas são tramitadas por diferentes setores e colaboradores, com ajuda ou não de softwares de gestão, caracterizando atividades do processo. Este trabalho propõe uma arquitetura que busca detectar situações de interesse por meio de atividades de processo (eventos) no ambiente de negócio e iniciar ações proativas frente a essas situações. Para isso, são utilizadas ferramentas para o processamento de *stream* de eventos, como: Apache Kafka e Apache Spark. Além das ferramentas citadas, para a detecção de situações de interesse em processos de negócio, o modelo proposto por Machado (MACHADO et al., 2017), foi estendido para o contexto de negócio. Visando avaliar o trabalho, foram desenvolvidos 02 (dois) estudos de caso baseado em cenários reais trabalhados ao longo desta dissertação. Sendo um no projeto de pesquisa e desenvolvimento intitulado “Plataforma para Integração de Sistemas baseada em mapeamento e automação de processos” e o outro no projeto intitulado “Plataforma de Defesa Sanitária Animal do Estado do Rio Grande do Sul (PDSA-RS)”. Foi desenvolvida uma aplicação protótipo com base em um dos cenários.

Palavras-chave: Apache Kafka. Apache Spark. Log de eventos. Processos de Negócio.

ABSTRACT

PROPOSAL OF A ARCHITECTURE FOR DETECTION OF SITUATIONS INTEREST AND PROACTIVE ACTIONS IN BUSINESS PROCESS

AUTHOR: Matheus Friedhein Flores

ADVISOR: Alencar Machado

CO ADVISOR: Jonas Bulegon Gassen

With the advancement of technology and the constant search of organizations to improve their organizational processes, aiming to maximize the use of resources, several organizations are investing in improvements in the flow of their internal processes. In general, all organizations have tasks that are performed by different employees, given their role. A set of tasks, conditions and events characterizes a process. Tasks are handled by different sectors and employees, with or without the help of management software, characterizing process activities. This work proposes an architecture that seeks to detect situations of interest through process activities (events) in the business environment and initiate proactive actions in the face of these situations. For this, tools are used for processing event streams, such as: Apache Kafka and Apache Spark. In addition to the tools mentioned, for detecting situations of interest in business processes, the model proposed by Machado (MACHADO et al., 2017), has been extended to the business context. In order to evaluate the work, 02 (two) case studies were developed based on real scenarios worked throughout this dissertation. One in the research and development project entitled "Platform for Systems Integration based on mapping and process automation" and the other in the project entitled "Platform for Animal Health Defense of the State of Rio Grande do Sul (PDSA-RS)". A prototype application was developed based on one of the scenarios.

Keywords: Apache Kafka. Apache Spark. Log event. Business Process.

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Estrutura Apache Kafka.....	19
Figura 2.2 - Extração de conhecimento com técnicas de mineração de processos..	22
Figura 2.3 - Estado do ambiente físico.....	27
Figura 2.4 - Plano de ações genérico.....	29
Figura 3.1 - Arquitetura.....	32
Figura 3.2 - Processamento de fluxo de eventos.....	34
Figura 3.3 - Modelo adaptado de Machado (MACHADO et al., 2017).....	35
Figura 3.4 - Módulo de gerenciamento de ações (MGA).....	38
Figura 3.5 - Módulo de gerenciamento de regras (MGR).....	39
Figura 3.6 - Módulo de gerenciamento de serviços (MGS).....	40
Figura 5.1 - Processo de solicitação de crédito.....	47
Figura 5.2 - Diagrama de sequência interação, cliente com a organização.....	50
Figura 5.3 - Visão geral organização bancária.....	51
Figura 5.4 - Exemplo de carta de controle. (a) sob controle estatístico (b) fora de controle estatístico.....	52
Figura 5.5 - Processamento de fluxo de eventos.....	53
Figura 5.6 - Processo emitir laudo.....	58
Figura 5.7 - Visão geral estudo de caso PDSA-RS.....	61
Figura 6.1 - Interface gráfica de monitoramento e controle de processos.....	67
Figura 6.2 - Gráfico de tempos das atividades de um processo.....	68
Figura 6.3 - Método produtor de eventos.	70
Figura 6.4 - Classe serviço de log.	70
Figura 6.5 - Classe serviço para detectar fraude.....	71

LISTA DE QUADROS

Quadro 4.1 - Características dos trabalhos relacionados.	42
Quadro 5.1 - Regras processo solicitar crédito.....	47
Quadro 5.2 - Situação atual: <(insatisfação do cliente), (ea1), (ea2)>.....	54
Quadro 5.3 - Evento ea2.....	54
Quadro 5.4 - Ações para situação indesejada.....	55
Quadro 5.5 - Regras sanitárias para certificação avícola - Processo Emitir Laudo...57	
Quadro 5.6 - Situação atual: <(Amostras inválidas), (ev1), (ev2)>.....	62
Quadro 5.7 - Evento ev2.....	63
Quadro 5.8 - Situação atual: <(Exame incorreto), (ev3), (ev4)>.....	63
Quadro 5.9 - Evento ev4.....	64
Quadro 5.10 - Ações para as situações de interesse em processos de negócio.....	65

LISTA DE ABREVIATURAS E SIGLAS

AA	Ação Automatizada
E	Evento
EO	Entidade Objeto
ES	Entidade Sujeito
FE	Evento de Fim
I	Entidades de Entrada
IE	Evento de Início
O	Entidades de Saída
P	Padrão
R	Relações Semânticas
SA	Situação Atual
SE	Serviço
SI	Situação de Interesse
REG	Regras
MGA	Módulo de Gerenciamento de Ações
MGR	Módulo de Gerenciamento de Regras
MGS	Módulo de Gerenciamento de Serviços

LISTA DE SÍMBOLOS

μ	Média de Valores Individuais
R	Amplitude
L	Distância
\bar{x}	Média
Δ	Delta
\wedge	Operador Lógico 'E'

SUMÁRIO

1. INTRODUÇÃO.....	13
1.1 QUESTÃO DE PESQUISA E OBJETIVOS	15
1.2 ORGANIZAÇÃO DO TEXTO	16
2. REVISÃO BIBLIOGRÁFICA	17
2.1 PROCESSAMENTO DE FLUXO DE DADOS	17
2.1.1 Apache Kafka	18
2.1.2 Apache spark.....	20
2.2 LOG	21
2.3 SENSIBILIDADE AO CONTEXTO	24
2.4 MODELO PROATIVO.....	26
3. ARQUITETURA PARA DETECÇÃO DE SITUAÇÕES DE INTERESSE EM PROCESSOS DE NEGÓCIO COM AÇÕES PROATIVAS	31
4. TRABALHOS RELACIONADOS.....	42
5. APLICAÇÃO DA ARQUITETURA - ESTUDO DE CASO	46
5.1 CENÁRIO PARA ORGANIZAÇÃO BANCÁRIA	46
5.1.1 Comportamento Para Situação De Crédito Negado	50
5.2 CENÁRIO PARA PLATAFORMA DE DEFESA SANITÁRIA ANIMAL DO RIO GRANDE DO SUL (PDSA-RS)	56
5.2.1 Comportamento Para As Situações De Interesse Em Processos De Negócio - Amostras Inválidas E Exame Incorreto.....	60
6. RESULTADOS	66
7. CONCLUSÃO	72

1. INTRODUÇÃO

Em organizações contemporâneas existem consideráveis desperdícios de recursos financeiro e humano. A falta de planejamento e acompanhamento dos processos existentes nas organizações trazem grande impacto nesses desperdícios. Esses fatores influenciam em produtos ou serviços com baixa qualidade e perda de clientes, que impactam de forma direta a organização. Processos de negócio são os meios pelos quais os produtos e serviços são desenvolvidos e disponibilizados para os clientes, fazendo com que seu gerenciamento seja fundamental para o sucesso da organização (ABPMP BPM CBOK, 2013).

Para tratar esses problemas, as organizações estão investindo cada vez mais em desenvolver, ampliar e melhorar os processos internos (SAYLAM; SAHINGOZ, 2013). De um modo geral as organizações são estruturadas em forma de processos, que indicam como o trabalho é realizado, visando agregar valor aos seus clientes (ABPMP BPM CBOK, 2013).

A modelagem e aplicação de processos de negócios é adotada nas organizações. Especialistas do negócio modelam e os traduzem em fluxos de trabalho, executáveis por sistemas de gestão (FRISCHBIER et al., 2012). Tais fluxos de trabalho podem ser mapeados como processamento de *stream de* eventos, que possibilitam obter resultados em tempo real. Por exemplo, uma plataforma digital para crédito pessoal, onde são manipulados inúmeros dados de clientes, com diversas funcionalidades disponíveis, como abertura de conta, atualização de cadastros, solicitações de crédito entre outros. Nesse contexto, existem atividades que um usuário precisa executar para liberar um crédito pessoal, por exemplo, as quais podem ser interpretadas por softwares de gestão como fluxo de eventos (KREPS et al., 2011).

O processamento de dados de *streaming* vem ganhando espaço e se destacando devido aos seus impactos fundamentais em uma vasta gama de casos de uso e cenários, como análise de negociações em tempo real (CHINTAPALLI et al, 2016).

Com a enorme quantidade de dados decorrente das atividades dos processos das organizações, são necessárias ferramentas que auxiliem no gerenciamento dos processos. Com isso, plataformas modulares vêm se popularizando, como é o caso

do Apache Kafka e Apache Spark, utilizados neste trabalho, para o processamento do fluxo de eventos. Ambas plataformas provêm um serviço contínuo, estável, equilibrado e escalável, mantendo o serviço ativo independentemente da quantidade de dados processados (CHINTAPALLI et al, 2016).

Visto que todos esses dados são processados e gerados a partir do fluxo de trabalho, é importante manter esses dados salvos como informações históricas, representados pelos logs de eventos. Todos os logs de evento refletem os eventos reais que ocorrem no cotidiano da organização, o que possibilita manter um histórico sobre o que aconteceu e acontece nos processos.

Tendo como base processos, é possível extrair e identificar parâmetros desejados do comportamento dos mesmos. Por exemplo, o parâmetro tempo da execução de atividades em um processo, que pode ser utilizado para identificar se o processo está executando dentro dos limites estabelecidos de qualidade da organização na prestação de algum serviço. Os limites estabelecidos, podem ser caracterizados como regras que a atividade do processo deve seguir, caso contrário, a atividade está sendo realizada de forma diferente do que foi definida. O não cumprimento das regras estabelecidas no processo podem acarretar para a organização resultados diferentes do esperado pela mesma, representando situações de interesse.

O modelo proposto por Machado (MACHADO et al., 2017), tem como objetivo gerenciar o ambiente onde situações ocorrem. Tais situações geram eventos e a detecção de padrões desses eventos caracterizam situações de interesse, podendo ser indesejada ou não. Além da detecção de situações de interesse, o modelo proposto por Machado (MACHADO et al, 2017) prevê iniciar ações proativas frente a essas situações detectadas. As ações são iniciadas com o intuito de minimizar os impactos no ambiente em questão, sendo utilizado um plano de ações a serem disparadas, visando alterar os eventos que estão acontecendo e que podem ocasionar uma situação de interesse.

O presente trabalho tem como objetivo propor uma arquitetura baseada em eventos, visando possibilitar o monitoramento do fluxo de eventos em tempo real, bem como detectar situações de interesse em processos de negócio, disparando ações proativas frente às situações detectadas.

Neste trabalho a arquitetura proposta utiliza o modelo de Machado (MACHADO et al., 2017) estendido. Foram acrescentadas ao modelo características

para utilização no ambiente de negócio, para detecção de situações de interesse. Para o processamento em tempo real, as ferramentas Apache Kafka e Apache Spark são utilizadas. Além do processamento em tempo real, a arquitetura utiliza logs de eventos que são persistidos a partir do processamento de fluxo de eventos, relacionados às execuções das atividades dos processos.

A arquitetura é composta por módulos que compreendem: as ações a serem iniciadas, regras definidas por especialistas em relação aos processos de negócio e serviços de terceiros que podem utilizar os dados processados pela arquitetura, os quais são armazenados em banco de dados (logs). Visando avaliar o trabalho, foram desenvolvidos 02 (dois) estudos de caso, ambos baseados em cenários reais de utilização.

1.1 QUESTÃO DE PESQUISA E OBJETIVOS

Em diversas organizações existe a busca constante em melhorar e aprimorar os processos de negócio, visando fornecer produtos e serviços que suprem as necessidades de seus clientes. Sendo assim, é importante que a organização tenha capacidade de monitorar e analisar os seus processos, para que seja possível identificar falhas na execução das atividades de processos (ABPMP BPM CBOOK, 2013). **Desse modo, a questão que motiva a presente pesquisa é:** uma arquitetura baseada em eventos pode permitir a detecção de situações de interesse em processos de negócios a fim de realizar ações proativas?

Neste trabalho, o **objetivo** é propor uma arquitetura para o processamento de eventos a partir de processos de negócio, visando a detecção e manipulação das situações de interesse (indesejada ou desejada), bem como iniciar ações proativas decorrente das situações de interesse detectadas.

Os **objetivos específicos** são:

- Modelar uma arquitetura para comportar o processamento de fluxo de eventos a fim de detectar situações de interesse.
- Estender o modelo de Machado (MACHADO et al., 2017), para identificar situações de interesse em processos de negócio e iniciar ações proativas frente às situações detectadas, visando minimizar impactos no ambiente organizacional;
- Testar por meio de estudos de caso (cenários) a utilização da arquitetura,

bem como o modelo de Machado (MACHADO et al., 2017) estendido em um ambiente real de negócio;

Vale ressaltar que a questão que motivou este trabalho foi elaborada no decorrer do projeto intitulado “Plataforma para Integração de Sistemas baseada em mapeamento e automação de processos” (número do projeto: 049141), que constitui a parceria entre a UFSM e o Sicredi - Instituição financeira cooperativa, para gestão de processos.

1.2 ORGANIZAÇÃO DO TEXTO

O texto desta dissertação está organizado da seguinte maneira: no capítulo 2 é descrito o referencial teórico, contextualizando a visão de vários autores em uma abordagem conceitual sobre logs de eventos, processamento de fluxo de dados de *stream*, as tecnologias Apache Kafka e Apache Spark, sensibilidade ao contexto e o modelo proposto por Machado (MACHADO et al, 2017), o capítulo 3 apresenta a arquitetura baseada em eventos para detecção de situação interesse em processos de negócio, o capítulo 4 apresenta os trabalhos relacionados, o capítulo 5 descreve o estudo de caso que caracteriza 02 (dois) cenários para aplicabilidade da arquitetura proposta, com o intuito de verificar a viabilidade de sua utilização prática, o capítulo 6 descreve os resultados obtidos e por fim o capítulo 7 apresenta as considerações finais sobre o trabalho, descrevendo as contribuições da pesquisa e propondo trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são descritos a fundamentação para o desenvolvimento do trabalho, contextualizando a visão de vários autores em uma abordagem conceitual sobre logs de eventos, processamento de fluxo de dados de *stream*, as tecnologias Apache Kafka e Apache Spark e o modelo proposto por Machado (MACHADO et al, 2017).

2.1 PROCESSAMENTO DE FLUXO DE DADOS

Segundo os autores Cugola e Margara (2012), sistemas de processamento de fluxo tem o dever de fornecer serviços constantemente, para estarem a todo momento disponíveis. Sendo que de maneira alguma o serviço seja interrompido e fique indisponível, exceto por causas naturais ou de terceiros (GIUFFRIDA; IORGULESCU; TANENBAUM, 2014).

Processamento de fluxo de dados é um paradigma computacional que tem o foco em análise, agregação e transformação de um amplo volume de fluxos de dados que são continuamente atualizados (GEDIK et al., 2014) (CUGOLA; MARGARA, 2012). A sequência resultante dos eventos criados, pode ser definida como um fluxo de dados de eventos transmitidos por produtores. As informações são geradas por diversos produtores, como aplicativos móveis e aplicações *web*. O processamento de fluxo de dados é indispensável a fim de extrair informações relevantes, para diversas funcionalidades (LUCKHAM, 2001) (ETZION; NIBLETT, 2011).

Neste trabalho, o fluxo de dados é resultado da execução de atividades de processos de negócio, o que resulta em diversas informações disponíveis a serem processadas. Para o processamento dessas informações disponíveis a todo instante, algumas ferramentas se destacam por fornecer segurança e confiança no processamento. As próximas seções discutem as ferramentas de processamento de *stream* de eventos.

2.1.1 Apache Kafka

O Apache Kafka¹ (a seguir denominado Kafka) é uma plataforma de processamento para ambientes em tempo real, utilizando o sistema de mensagens distribuídas que pode lidar com grande volume de dados. Os dados são gerados em grandes volumes por meio de várias atividades, por exemplo, softwares de gerenciamento, onde diversas informações são geradas e manipuladas no decorrer da execução das atividades de diversos processos (HIRAMAN et al, 2018).

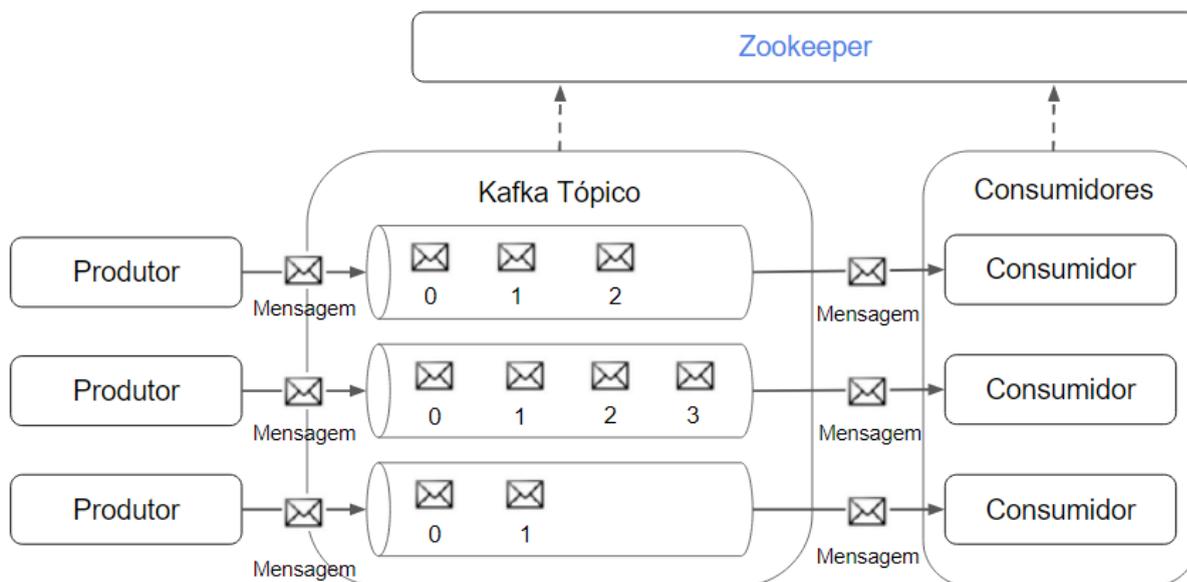
De acordo com Kreps (KREPS et al., 2011) esses tipos de dados podem ser considerados como fluxo de dados. O processamento de *stream* se tornou um popular paradigma que permite obter resultados em tempo real e continuamente para grandes volumes de dados. Abaixo são descritos alguns pontos que caracterizam o Kafka, como:

- **Escalabilidade:** é uma estrutura de fácil dimensionamento sem tempo de inatividade.
- **Alto volume:** é projetada para trabalhar com alto volume de dados, sem que ocorram problemas devido a essa alta demanda.
- **Confiabilidade:** foi projetado de forma particionada, replicado e distribuído com tolerância a falhas.
- **Transformação de dados:** trabalho ao fornecer novo fluxo de dados do produtor.
- **Baixa latência:** visando as mensagens tradicionais, requer baixa latência o que evita atrasos (HIRAMAN et al, 2018).

O Kafka foi projetado para ser confiável, escalável, rápido e durável. Para isso o mesmo dispõe de uma estrutura, como ilustrado no fluxograma na Figura 2.1.

¹ <https://kafka.apache.org/>

Figura 2.1 - Estrutura Apache Kafka.



Fonte: (HIRAMAN et al, 2018).

Com base na Figura 2.1, Kafka é composto por (HIRAMAN et al., 2018):

- **Tópicos:** onde mensagens são publicadas e armazenadas. Todas as mensagens que o Kafka gerencia são organizadas em tópicos, sendo que a mensagem contém uma carga útil em unidade bytes. Aplicações produtoras gravam mensagens nos tópicos, aplicações consumidoras leem as mensagens dos tópicos, podendo um tópico ser particionado, o que leva ao equilíbrio no gerenciamento das mensagens.
- **Produtores:** são editores das mensagens destinadas a um ou mais tópicos. Os produtores enviam dados para os brokers. Toda vez que um produtor publica uma mensagem para um broker, ele pode escolher a partição que deseja ou a que esteja livre no momento.
- **Consumidores:** lê dados dos brokers, assinam um ou mais tópicos e consomem mensagens publicadas, retirando dados dos intermediários.
- **Conectores:** são responsáveis por extrair dados da *stream* fornecidos pelos produtores e entregar aos consumidores ou processadores de *Stream*.
- **Processador de Stream:** são aplicativos que transformam *Stream* em dados de tópicos em outros *Stream* de dados de tópicos no Kafka Cluster.
- **Broker:** o cluster Kafka geralmente consiste em vários brokers, visando o

equilíbrio de carga. Os agentes Kafka não têm estado, portanto, usam o Zookeeper para manter seu estado de cluster.

- **Zookeeper:** usado para gerenciar e coordenar os *brokers* existentes no Kafka. É utilizado principalmente para notificar produtores e consumidores sobre a presença de qualquer novo broker no sistema Kafka ou falha em algum broker.

O Kafka é uma plataforma de *streaming* que permite aos usuários publicar dados e também se inscrever em diferentes fluxos de dados. Kafka armazena os fluxos de maneira tolerante a falhas. É usado para construir componentes confiáveis de dados de *streaming* em tempo real. Possui velocidade de processamento excelente, durabilidade e é utilizado em amplas áreas no conceito de processamento de dados em tempo real, bem como aplicativos de *streaming* (ICHINOSE et. al, 2017).

Com isso, neste trabalho o Kafka é utilizado para o processamento de *streams* de evento, a fim de processar todas as informações disponíveis e geradas pelos produtores (clientes) e repassando para os consumidores (serviços). Visando alcançar o objetivo que é a detecção de situações de interesse baseado em eventos em processos de negócio. O processamento é importante para que todas as informações analisadas sejam coerentes e completas, sem que ocorra perda de informações. Na próxima seção, são descritos conceitos sobre Apache Spark.

2.1.2 Apache Spark

Segundo Ichinose et.al (2017) o Apache Spark (a seguir denominado Spark) é uma estrutura de processamento distribuída que armazena e processa dados em larga escala. O Spark é provido de diversos componentes, dentre eles o Spark *streaming*, utilizado para processar dados de *stream*.

O Spark² é uma estrutura de processamento de grande volume de dados executada em *clusters*, melhorando a confiabilidade e tolerância a falhas, juntamente com os recursos de processamento em memória. O Spark³ suporta bibliotecas para criar e redimensionar aplicativos de *streaming*. Algumas

² <https://spark.apache.org/>

³ <https://spark.apache.org/streaming/>

características e razões importantes pelas quais o Spark se distingue de outras ferramentas disponíveis, sendo (i) um mecanismo mais rápido, de uso geral com processamento em lote para fins gerais, e (ii) com capacidade de associar conjuntos de dados em várias fontes de dados diferentes (SHORO; SOOMRO, 2015).

Além disso o Spark também suporta as informações providas por sistemas de transmissão de mensagens de dados, como o Apache Kafka (seção 2.1.1), suportando comunicação assíncrona, e tornando mais acessível o desenvolvimento de aplicativos adequados para cada fluxo de dados (ICHINOSE et al., 2017).

Neste trabalho o modelo de detecção de situações de interesse é inserido na estrutura do Spark, o qual contém todos os conceitos necessários para a análise dos eventos que estão ocorrendo no ambiente de negócio. Visando manter o histórico sobre todos os eventos processados, surgem os logs de eventos, conceituado na próxima seção.

2.2 LOG

Esta dissertação tem o interesse em dados armazenados, com o intuito de analisar dados sobre a realização das atividades de processos no formato de eventos, assim produzindo Logs destes processos. Um log de evento consiste no registro de séries de eventos ocorridos durante o processo de funcionamento de uma organização. Sendo que o armazenamento desse registro é realizado, principalmente, por sistemas de informação (VAN DER AALST et al., 2011) (GLAVAN, 2011). Eventos são considerados como uma *tupla* contendo os seguintes campos (VAN DER AALST et al., 2011):

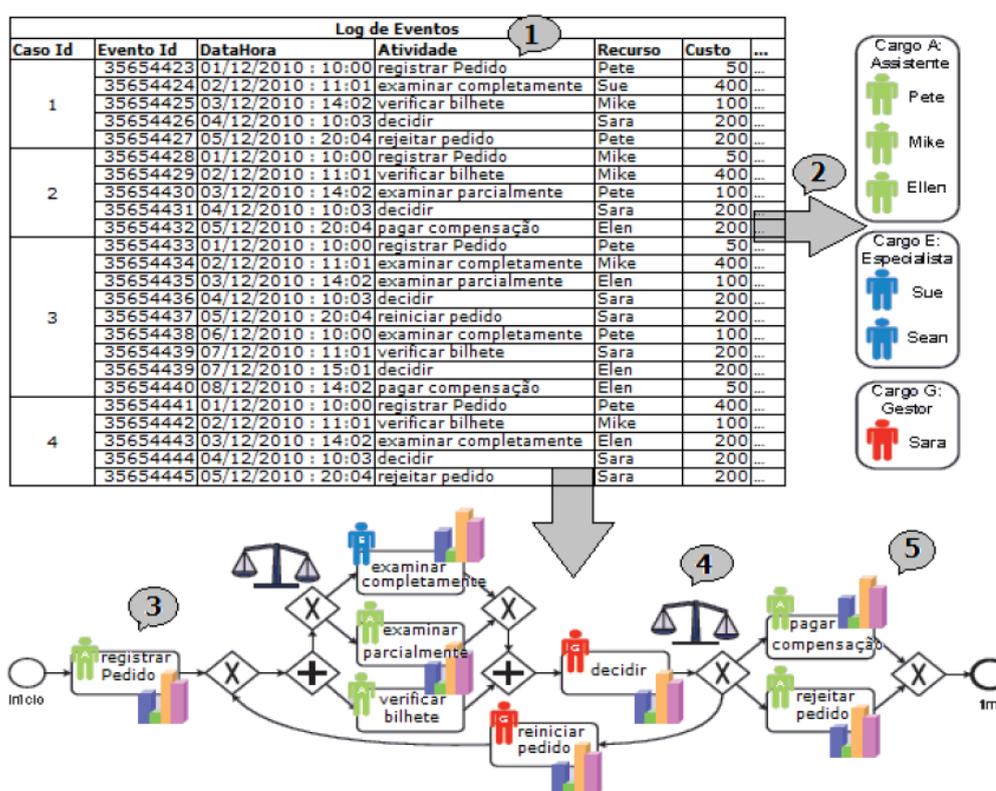
- **Id:** identificação;
- **Timestamp:** data e hora de início da atividade;
- **Atividade:** descrição da Atividade;
- **Recurso:** responsável pela atividade;
- **Custo:** custo da atividade;
- **Dados extras:** informações adicionais sobre uma atividade.

Em alguns casos nem todos os dados estão disponíveis, quando isso acontece, é necessário manter no mínimo, o Id da Atividade e alguma ordem de

execução (VAN DER AALST, 2011). Em alguns casos específicos, também existem os registros de início e fim da atividade, sendo assim, além dos dados acima uma descrição extra é acrescentada, com o objetivo de diferenciar as atividades.

Eventos são agrupados pela ordem em que ocorrem e passam a ser referenciados como *traces* ou *cases*. Cada case possui um ID (identificador único) e agrupa uma quantidade finita de eventos. Com isso, um log de eventos é constituído por uma sequência de *case*. A Figura 2.2 apresenta um exemplo de log de eventos.

Figura 2.2 - Extração de conhecimento com técnicas de mineração de processos.



Fonte: (VAN DER AALST et al., 2011).

Os pontos principais exibidos na Figura 2.2, podem ser apresentados da seguinte forma:

1. Conjunto de eventos com informações sobre a instância do processo e da atividade a que se referem. Em alguns casos aparecem informações temporais e recursos utilizados na atividade;

2. Os registros de log permitem descobrir os cargos que constituem a organização, mediante a identificação dos grupos de pessoas com atividades similares;
3. Permite descobrir o fluxo de trabalho, observando a análise das sequências das atividades nos eventos do processo;
4. Possibilita identificar regras de decisão utilizadas nos processos;
5. Permite a extração das informações de desempenho do processo. (VAN DER AALST, 2012).

A qualidade dos registros é de extrema importância, pelo fato que os dados dos eventos são o ponto de partida para qualquer atividade, sendo que a qualidade dos resultados depende muito da qualidade das informações de entrada utilizadas (VAN DER AALST, 2012). Portanto, os eventos devem ser tratados e verificados adequadamente nos sistemas de informação que suportam os processos.

Os dados podem estar incompletos e incoerentes com a realidade, possuir ruído, bem como atividades silenciosas que são realizadas na empresa (atividades realizadas por terceiros) que não são registradas no sistema interno da organização (VAN DER AALST, 2012).

Para maximizar os resultados que podem ser extraídos dos logs, as organizações devem concentrar esforços em garantir a maior qualidade possível nos registros de eventos. Para isso, podem ser avaliados sob vários critérios, como: fidedignidade, completude, clareza e segurança. De acordo com van der Aalst (2011) esses critérios são conceituados da seguinte forma:

- **Fidedignidade:** deve garantir que os eventos registrados foram realmente executados e que, além disso, a informação neles contidas sejam verdadeiras.
- **Completude:** deve garantir que os registros de eventos sejam completos, ou seja, para uma determinada aplicação não existem eventos ausentes.
- **Clareza:** deve garantir que qualquer evento registrado tenha uma semântica bem definida.
- **Segurança:** no registro dos eventos deve ser considerada a privacidade e segurança, ou seja, os envolvidos devem ter conhecimento dos eventos que estão sendo registrados e a forma como poderão ser utilizados.

Os registros de logs de eventos podem ser encontrados em uma grande variedade de formas, visto que cada sistema tem seu mecanismo próprio de registro de eventos. Desenvolvido com base nos princípios da (i) simplicidade, utilizar a maneira mais simples possível para representar informações; (ii) flexibilidade, ser possível capturar logs de eventos a partir de qualquer tipo de domínio de aplicação; (iii) extensibilidade, facilidade de ser ajustado podendo adicionar novos padrões mantendo compatibilidade; e por fim (iv) expressividade, possibilidade de anexar semântica humano-interpretáveis para logs de eventos (GÜNTHER; VERBEEK, 2014).

Além dos logs de eventos, o modelo utilizado para detecção de situações de interesse proposto por Machado (MACHADO et al, 2017), foi estendido para processos de negócios, descrito com mais detalhes na seção seguinte.

2.3 SENSIBILIDADE AO CONTEXTO

Computação sensível ao contexto é umas das áreas que possibilita desenvolver aplicativos flexíveis, adaptáveis e que podem ser executados automaticamente, visando beneficiar os usuários. Sendo que contexto está relacionado a qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade, do qual entidade é uma pessoa, um lugar ou até mesmo um objeto relevante para a interação entre o usuário e a aplicação (MACHADO, 2015) (DEY; ABOWD, 1999).

Além da computação sensível ao contexto, a Computação Ubíqua proposta por Weiser (1991), considera que o ambiente computacional não deve impor restrições ao usuário para ser utilizado. Com isso, é necessário que o ambiente computacional seja invisível e transparente, com soluções de interações intuitivas para que o usuário possa interagir com a computação sem precisar utilizar de conhecimentos específicos da área.

Tendo como foco sistemas que auxiliem os usuários em tarefas diárias e corriqueiras, surgem os sistemas sensíveis à situação para ambientes inteligentes. Esses sistemas utilizam informações providas por sensores (taxa de batimentos cardíacos ou pressão sanguínea do cidadão) ou por sistemas de informação à medida que o usuário interage com o mesmo. Para que essas informações sejam utilizadas em prol dos usuários elas devem ser interpretadas em alto nível

conceitual, como distinguir quando as interações do usuário podem lhe ocasionar situações que não são esperadas. Por exemplo, se em uma determinada atividade que o usuário realiza em uma empresa que contém regras e essas não são cumpridas, resultaria em algo que não é esperado para o usuário e a empresa.

Esses conceitos de alto nível são chamados de situação, sendo um estado de abstração do contexto de interesse das aplicações. Uma situação pode ser definida como um conjunto de características de contexto que são instáveis em um determinado intervalo de tempo (MACHADO et al, 2017).

Sendo assim, o objetivo é evitar que situações indesejadas aconteçam, como, por exemplo, em uma certa atividade do cotidiano de uma empresa é necessário que algumas regras sejam seguidas. Se caso alguma dessas regras não seja cumprida, um comportamento não esperado é detectado, ocasionando a situação 'regra não seguida'. Nesse exemplo, um comportamento proativo será o sistema gerar a probabilidade para determinar se a regra é cumprida. Dessa forma, na próxima vez que o usuário realizar essa atividade, serão executadas específicas ações, caracterizando um comportamento proativo de se antecipar a essa situação. Ter comportamento proativo significa ser consciente de uma situação para aprender através dela, assim adquirindo experiência para que, quando existirem evidências de que esta situação ocorrerá novamente, seja possível agir antecipadamente para eliminar ou suavizar seu impacto no ambiente (MACHADO et al., 2017).

Sendo assim é possível detectar situações no ambiente, por meio do contexto de interesse. O contexto é subconjunto ou fatia de uma parte do contexto, onde aplicações teriam interesse e são utilizadas para tomar a decisão de qual ação deve ser executada, frente a situação detectada (MACHADO et al., 2017).

O contexto é representado por um conjunto de entidades e suas relações semânticas. Visto que entidades são conceitos abstratos utilizados para raciocinar sobre um domínio de interesse, por exemplo, pessoa, lugar, tempo, dispositivos eletrônicos, entre outros (CHEN, 1976) (MACHADO et al., 2017).

As relações semânticas {R} formam o contexto e são caracterizadas por triplas $\langle Es, p, Eo \rangle$, onde a Entidade que age como sujeito (Es) e a Entidade de age como objeto (Eo) representam instâncias de entidades do ambiente (MACHADO et al, 2017). Por exemplo, na relação $\langle Empresa, temCliente, Clientes \rangle$, Empresa e Clientes representam entidades ligadas pelo predicado temCliente. Sujeitos e objetos podem ser representados por variáveis em regras de inferência. Por

exemplo, $\langle a, \text{temCliente}, b \rangle$, onde a representa qualquer entidade instanciada no domínio da empresa e b representa qualquer entidade instanciada no domínio do cliente/usuário (MACHADO et al, 2017). As relações semânticas constituem os eventos.

Segundo McCarthy (2002), evento é algo que aconteceu ou tem acontecido, sendo detectado como algo instantâneo no tempo. Por exemplo, a utilização de um serviço provido por sistemas de informação, ou seja, uma funcionalidade para realizar alguma atividade/tarefa, produziria um evento. Com base nos eventos é possível detectar um padrão e com isso uma situação de interesse. Na próxima seção está conceituado o modelo para detecção de situação de interesse por meio de eventos, que neste trabalho são gerados a partir de sistemas de informação.

2.4 MODELO PROATIVO

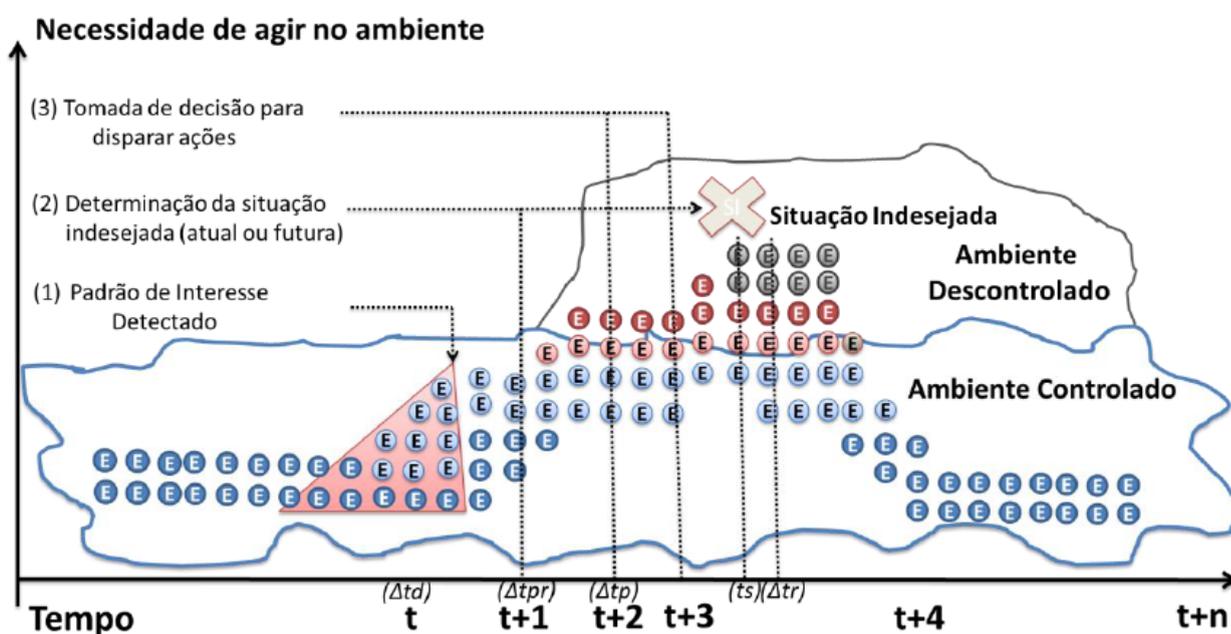
O trabalho proposto por Machado (MACHADO et al., 2017) aborda a identificação e o gerenciamento de situações de interesse, em Ambientes Inteligentes, buscando executar ações frente a essas situações. O trabalho de Machado foi desenvolvido para ambientes de vivência assistida, onde o autor propõe uma abordagem para detecção e manipulação reativa e proativa para sistemas agirem frente às situações detectadas. Tendo em vista, eliminar ou suavizar o impacto no ambiente, bem como expandir sua capacidade para gerenciar novas situações indesejadas, que surjam no ambiente de vivência do usuário ao longo do tempo. O presente trabalho considerou estender o modelo proposto por Machado (MACHADO et al., 2017), para o ambiente de negócio pelo fato de comportar eventos e situações em tempo real.

Machado (MACHADO et al., 2017) define que situações são compostas por eventos, esses eventos ocorrem em uma janela de tempo válida denominada de janela deslizante (*Sliding Window*). As correlações desses eventos podem evidenciar uma situação. As ocorrências seguidas desses eventos podem significar um padrão de eventos resultando em uma situação de interesse, como mostra a Figura 2.3.

Caso ocorra uma situação de interesse, sendo ela indesejada ou não, ações podem ser executadas para controlar o ambiente, podendo ser ações reativas ou proativas. Ação Reativa é definida como a execução de uma ação, podendo ela alterar eventos que estão acontecendo e determinando a situação corrente (atual),

pois essa é de interesse do sistema que aconteça. Ações Proativas devem agir frente a eventos que estão acontecendo no ambiente e podem gerar uma situação de interesse futura. Neste sentido, uma Ação Proativa é definida como uma ação que tem o objetivo de alterar eventos que estão acontecendo no momento atual, além de caracterizar a probabilidade significativa de uma situação interesse acontecer no futuro, buscando suavizar ou eliminar essa situação (MACHADO et al., 2017).

Figura 2.3 - Estado do ambiente físico.



Fonte: (MACHADO et al., 2017).

Na Figura 2.3, de acordo com Machado (MACHADO et al., 2017) são apresentadas as fases, com os tempos utilizados para o sistema processar os eventos. Para determinar se uma situação interesse aconteceu ou irá acontecer no ambiente, sendo que este pode estar em dois estados: controlado ou descontrolado. Vale ressaltar que a situação de interesse pode ou não ser desejada. Neste trabalho são levados em consideração a detecção de situações que ocorram em processos de negócio.

Um fluxo de eventos que não é de interesse caracteriza um ambiente controlado, não sendo preciso agir. Porém, se um padrão de eventos de interesse é detectado (Δtd), isso pode caracterizar que o ambiente se tornará incontrolado

portanto, aumentando a intervenção de ações proativas. Avaliando o fluxo de eventos da Figura 2.3, em um tempo t , o ambiente está em um estado controlado e, ao final desta janela, um padrão de eventos (padrão provido pela aplicação) é detectado (1), sendo necessário realizar a determinação (inferência ou predição) de que uma situação envolve ou envolverá o usuário (MACHADO et al, 2017).

Desse modo, dando continuidade a análise da Figura 2.3, os middlewares podem detectar um padrão de interesse e têm de $t+1$ até $t+2$ para fazer a predição (Δt_{pr}) da situação detectada, sobrando tempo suficiente para executar ações proativas (Δt_p) em relação à situação. Caso em $t+1$ se identifique que a situação não está em curso, é possível identificar a probabilidade da ocorrência dessa situação (2), sendo que algum tipo de critério deve ser analisado para determinar se a taxa de probabilidade é relevante até $t+2$. Se esta taxa é positiva, deve-se escolher a política mais apropriada de ações para disparar ações proativas em até $t+3$ (3) (MACHADO et al, 2017).

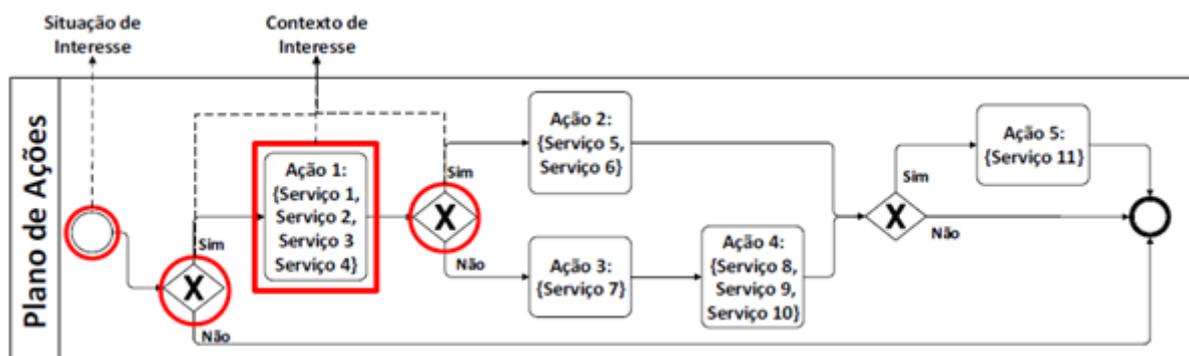
De acordo com Machado (2015) ações podem ser categorizadas em dois tipos: (i) Ações Humanas e (ii) Ações Automatizadas, sendo neste trabalho levado em consideração as Ações Automatizadas. Ação Automatizada pode ser classificada considerando o sujeito sobre o qual esta ação deseja agir: (a) *Em relação à Usuário*: refere-se a ações que são realizadas com o intuito de comunicar algo com a pessoa; (b) *Em relação a Dispositivo*: refere-se a ações que mudam o estado de um dispositivo; (c) *Em relação à Organização*: refere-se a uma ação para informar ou chamar uma organização. Todas as Ações Automatizadas são executadas pelas funcionalidades de um dispositivo. Neste trabalho, um conjunto de serviços $\{Se\}$ compõem uma Ação Automatizada. Os serviços devem ser organizados por meio de uma ordem lógica de execução, que corresponde a ações no ambiente. As ações automatizadas são representadas pela seguinte declaração (1):

Aa: (nome, $\{Sp\}$, $\{Se\}$, $\{E\}$) (1) (MACHADO, 2015).

Uma ação é constituída por um nome, um $\{Sp\}$ que consiste em uma ou mais situações que são condições para a correta execução da Ação Automatizada; um $\{Se\}$, que é um conjunto de serviços (funcionalidades ativadas em dispositivos) que devem ser utilizados para essa ação; e um $\{E\}$, que é um conjunto de eventos internos gerados como Efeito da ação executada (MACHADO, 2015).

Por exemplo, uma Ação Automatizada pode conter uma situação “X” como precondições para ser executada, essa ação pode ser o envio de um SMS para o responsável do processo, ilustrada pela Figura 2.4.

Figura 2.4 – Plano de ações genérico.



Fonte: (MACHADO et al., 2017).

Analisando a Figura 2.4, segundo Machado (MACHADO et al., 2017), é o plano de ações que deixa explícito quais ações devem ser disparadas em relação à situação de interesse detectada. Sendo que cada ação é composta por serviços. Os serviços são funcionalidades providas por softwares, que são executados à medida que a situação é detectada.

No modelo de Machado (MACHADO et al., 2017), Evento é algo que aconteceu ou tem acontecido, sendo detectado como algo instantâneo no tempo. Por exemplo, o consumo de um serviço (ativação de uma funcionalidade) produziria um Evento (o instante que a execução do serviço inicia) ou mesmo o início ou término de uma situação. Dessa forma, um Evento é o registro de algo que aconteceu, juntamente com o instante no tempo desse fato. Eventos conceituados por Machado (MACHADO et al., 2017), podem ser de dois tipos, (i) evento externo gerado externamente ao sistema, sendo gerado por Ações Humanas e (ii) evento interno, são gerados por uma Ação Automatizada.

Evento é composto por um nome, sendo caracterizado por um tipo (interno ou externo), um tempo dentro de Δt_d (fluxo de eventos monitorados como tempo de dados), e um conjunto de relações semânticas do contexto $\{R\}$. Quando um evento não é produzido por uma entidade simples (dado bruto de um sensor ou registro

simples do consumo de um serviço), este deve também conter um padrão (p) para sua detecção (MACHADO, 2015) (2).

Evento: (nome, tipo, tempo, {R}, p) (2).

No trabalho de Machado (MACHADO et al., 2017) os dados do ambiente utilizados pelos sistemas são obtidos por meio dos sensores e agregados com entidades do ambiente (contexto), para a geração de informação útil. Porém, as ações feitas no ambiente pelos sistemas são alcançadas pelas funcionalidades implementadas como serviços *web*, que estão embarcados em objetos como: smartphones, computadores entre outros no ambiente. Com isso utilizar o trabalho proposto por Machado (MACHADO et al., 2017) é de grande valia para esta dissertação, pois estendendo-o a outro contexto, espera-se obter sucesso nos resultados deste trabalho. Para mais detalhes sobre os conceitos semânticos abordados, procurar em Machado (2015).

Outros trabalhos, como de Forkan (FORKAN et al., 2015), Coronato, De Pietro e Paragliola (2014), definem que para entender as mudanças futuras no comportamento do usuário, um sistema deve considerar ações proativas e eventos atuais. Assim, uma abordagem pode compreender o desvio comportamental do usuário, utilizando modelos de reconhecimento de padrões. Com base na revisão bibliográfica apresentada neste capítulo, no próximo capítulo segue a arquitetura proposta neste trabalho.

3 ARQUITETURA PARA DETECÇÃO DE SITUAÇÕES DE INTERESSE EM PROCESSOS DE NEGÓCIO COM AÇÕES PROATIVAS

No trabalho de Machado (MACHADO et al., 2017) o domínio da aplicação da pesquisa foi ambientes de vivência assistida, gerenciando as situações ocorridas no ambiente. Para isso, o autor faz uso de diversos sensores para captar informações de contexto de usuários com problemas cognitivos, com o intuito de evitar que esses usuários se coloquem em situações de risco. O modelo proposto por Machado (MACHADO et al., 2017), possibilita ser estendido para outros ambientes, podendo ser utilizado em qualquer contexto, seja ele na área da saúde, segurança ou de negócios.

Em vista disso, visando utilizar o modelo proposto por Machado (MACHADO et al., 2017), em processos de negócios, o mesmo foi estendido para possibilitar detectar situações de interesse, porém sem dados de sensores e sim informações de softwares de gestão.

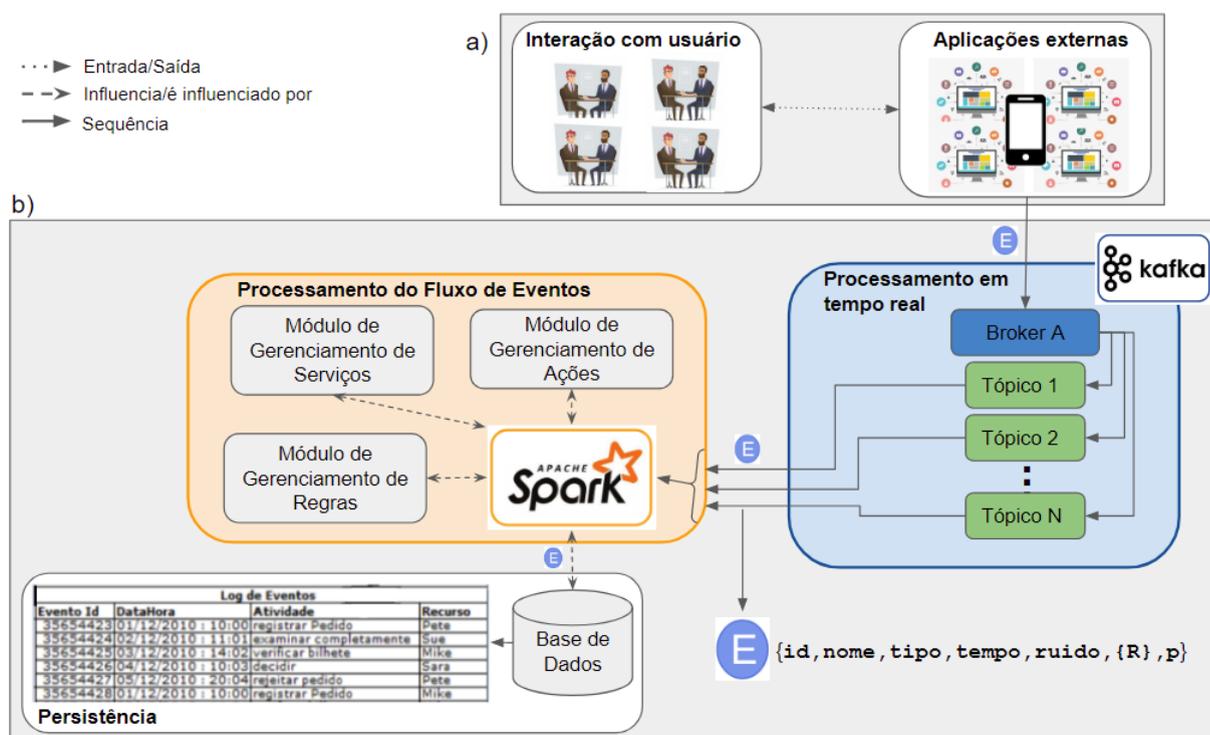
Vale ressaltar que no contexto de negócios, existem diversos aspectos que podem ser gerenciados, visto que é gerada uma vasta gama de informações, com base na utilização de softwares internos de gestão organizacional. Em todas as organizações existem processos de negócio, porém nem todas colocam importância a ponto de melhorar e expandir seus processos.

Softwares para gestão de processos são empregados visando maximizar fatores como, por exemplo, a qualidade dos produtos ou serviços comercializados. Entretanto, mesmo implantando e desenvolvendo melhorias, ainda assim ocorrem diversos fatores que resultam em perdas para as organizações, seja humano ou financeiro. Nesse sentido, este trabalho propõe uma arquitetura baseada nas informações geradas durante a realização das atividades (eventos), que compõem os processos de negócio. Com isso, vale lembrar a questão que motivou este trabalho. É possível detectar situações de interesse nos processos de negócio, bem como iniciar ações proativas frente a essas situações, visando minimizar impactos no ambiente de negócio?

A arquitetura tem como objetivo ser uma ferramenta para monitoramento de processos, tendo em vista manter o ambiente de negócio sob controle em relação às políticas e diretrizes definidas por especialistas do negócio. Dado as informações

geradas por softwares de terceiros. Ou seja, a realização das atividades que compõem os processos gera eventos e esses são interceptados pela arquitetura (processamento em tempo real), possibilitando realizar análises que resultam na detecção ou não de uma situação de interesse em processos de negócio. Além do processamento em tempo real do fluxo de eventos, a arquitetura propõe armazenar todas as informações processadas em logs de eventos (seção 2.2), possibilitando que aplicações externas utilizem essas informações para outros contextos. A Figura 3.1 ilustra de forma geral a arquitetura.

Figura 3.1 - Arquitetura.



Fonte: Autoria própria.

A partir da Figura 3.1, é possível observar como a arquitetura está organizada, com seus respectivos módulos, bem como ela é inserida no ambiente de negócio, visando a detecção de situações de interesse em processos de negócio.

Neste contexto, este trabalho propõe uma arquitetura baseada em eventos para a detecção de situações de interesse em processos de negócio e ações

proativas frente a essas situações, visando minimizar impactos que podem ser gerados por essas situações.

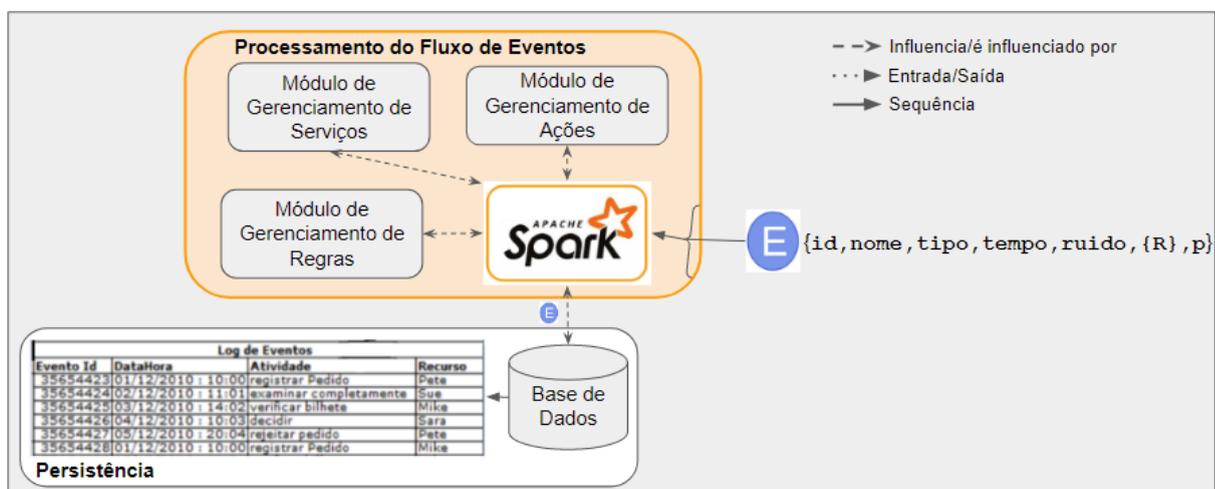
Ainda sobre a Figura 3.1, é possível observar que em a) é ilustrado o contexto de negócio, ambiente onde as informações são geradas, baseado na realização das atividades que compõem um processo. Cada atividade/tarefa é executada por algum colaborador da organização, por meio de softwares de terceiros que realizam a gestão das informações dos processos organizacionais. À medida que as atividades dos processos vão sendo realizadas, por meio de softwares de gestão (na Figura 3.1 identificado como aplicações externas), a arquitetura processa os fluxos dos eventos gerados. Em b) na Figura 3.1, é ilustrada a organização da arquitetura com seus respectivos módulos e ferramentas utilizadas, sendo eles:

Processamento em tempo real: módulo onde o Kafka, plataforma que auxilia o processamento do fluxo de eventos, por meio das mensagens, tópicos e *brokers*, possibilita o processamento contínuo e ininterrupto de dados (descrito na seção 2.1.1), à medida que as atividades que compõem os processos vão sendo realizadas.

Tendo em vista que diversas atividades são executadas por usuários diferentes ao mesmo tempo. O Kafka atua no processamento e orquestração para os *brokers* e tópicos que estão disponíveis, repassando para o Spark, caracterizado na arquitetura como um consumidor de eventos, descrito com mais detalhe no item abaixo.

Processamento de fluxo de eventos: módulo onde o Spark (descrito na seção 2.1.1), consome e processa o fluxo de eventos, utilizando os módulos de gerenciamento de regras (MGR), módulo de gerenciamento de ações (MGA) e módulo de gerenciamento de serviços (MGS). A Figura 3.2, ilustra especificamente o módulo de processamento de fluxo de eventos.

Figura 3.2 – Processamento de fluxo de eventos.



Fonte: Autoria própria.

Conforme a Figura 3.2 ilustra, o processamento de fluxo de eventos é composto por 04 (quatro) módulos, são eles:

Módulo de Gerenciamento de Ações (MGA): módulo responsável por detectar e iniciar ações proativas de acordo com a situação detectada. De acordo com o modelo proposto por Machado (MACHADO et al, 2017), descrito na seção 2.4, o autor descreve plano de ações à medida que as situações vão sendo detectadas. O plano de ações é utilizado para pré-definir ações a serem iniciadas frente à situação detectada nos processos de negócio.

Com isso o Spark é utilizado como uma estrutura responsável por interpretar o modelo estendido de Machado (MACHADO et al., 2017). O Spark é um mecanismo computacional em *cluster*, de uso geral rápido e confiável, que fornece interfaces de programação de aplicativos em várias linguagens de programação, como Java e Python (SHORO; SOOMRO, 2015). A especialidade dessa ferramenta é tornar a análise de dados mais rápida, com os programas em execução e na gravação de dados. É compatível com o Kafka em questões de protocolos de comunicação, o que facilita a integração de ambas tecnologias.

Nesse sentido, o Spark é utilizado para suportar o modelo proposto por Machado (MACHADO et al., 2017), buscando entendê-lo para ser aplicado a

Neste trabalho é relacionado a sistemas de terceiros, sendo uma especialização do conceito *em relação a organização*, definido por (MACHADO et al, 2017), utilizado para o atendimento ao cliente e organização dos processos do ambiente de negócio.

Softwares com objetivo de automatizar rotinas e tramitações burocráticas, com serviços informatizados, que proveem os dados a serem manipulados e gerenciados. Na Figura 3.1 em a) Sistemas legados é caracterizado por aplicações externas, simbolizando os softwares utilizados pelas organizações para o gerenciamento dos processos e atividades. Vale ressaltar que o escopo do trabalho está relacionado em processar os eventos em tempo providas por um software legado.

b) Ruído (seção 2.2), neste trabalho é algum dado/fluxo inesperado no fluxo normal do processo, caracterizando informações equivocadas. Os motivos não previstos que resultam em informações equivocadas podem ser, por exemplo, o responsável pela tramitação do processo em um certo setor entrou em férias. Porém, houve um descuido e o processo não foi repassado para nenhum outro colaborador, ficando ocioso e conseqüentemente aumentando o tempo de execução (duração) do mesmo. Pelo fato da arquitetura consumir dados de outros softwares, é possível que esses dados sejam incoerentes com a realidade, por diversos motivos. Por exemplo, se o software que produz os eventos tiver algum problema em relação a falta de energia, conexão com a internet, atraso na entrega das informações, etc. Por meio dos tempos definidos é possível identificar esses dados, utilizando como base as informações contidas nos logs de eventos. Ruído é uma propriedade de dados opcional, pode ser inserido ou não mediante o interesse do usuário/organização que faz uso da arquitetura.

Os eventos armazenados em banco de dados (logs), podem apresentar incoerências em relação a outros eventos, por motivos externos que interferem na duração das atividades, levando em conta a data e a hora de cada atividade. Como descrito no exemplo anterior, onde o colaborador tirou férias e o processo ficou ocioso, motivo externo que ocasionou a demora do processo e conseqüentemente incoerência com outros processos. Diante disso, cada dado antes de ser inserido no log de eventos é submetido a um cálculo base para ser categorizado como ruído ou não. O cálculo base equivale a um valor de tempo, aceito acima do tempo médio definido pelos dados históricos do processo que está sendo analisado. Em T , é representado o tempo em que o processo levou para ser executado, ou seja, desde

o início até ser concluído. A média representa o tempo médio de execução de um processo, com base nos logs de eventos. X é um parâmetro informado por um especialista da área de onde o processo é executado. Em (3), a equação que representa um ruído.

$$\text{ruído} = T > (x * \text{media}) \quad (3).$$

Com isso, se o processo Y que tem relação com uma situação de interesse Z, demora em média 10 minutos (média = 10) para ser executado (de acordo com os logs de eventos). Onde, X (parâmetro informado por especialistas), é $x = 2$, ou seja, 2 (dois) vezes a média (10), resulta no tempo aceito. Sendo que, os especialistas estimaram que o tempo aceito deve ser inferior a 20 minutos. Então logo no processo Y, T é menor que 20 (vinte), caracterizando um dado válido, não considerado ruído (4).

$$\text{ruído} = 10 > (2 * 10) \quad (4).$$

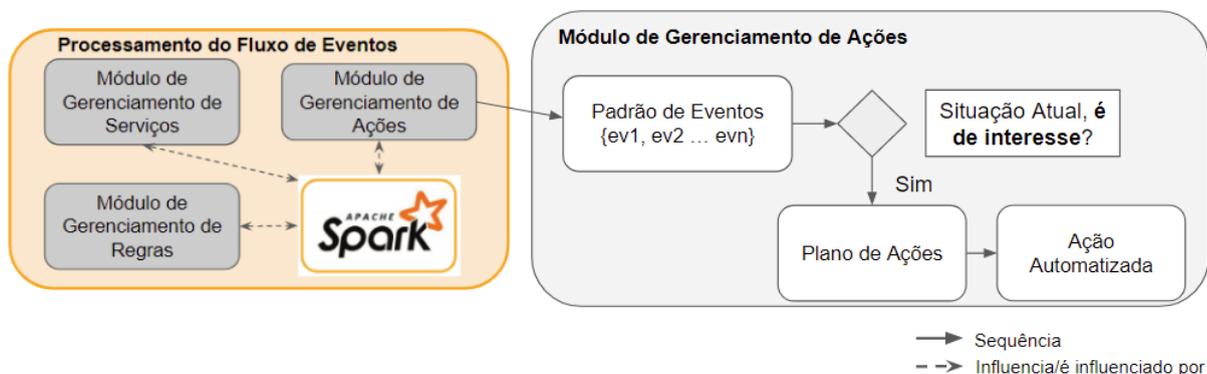
Para diferenciar os eventos com e sem ruído, a propriedade de dado ruído foi adicionado ao conceito evento (5). Com isso, é possível distinguir entre eventos úteis a serem utilizados por sistemas externos (serviços) e eventos que devem ser descartados, referente ao resultado da equação em (3). Uma vez que, ruído é caracterizado por ser um parâmetro do tipo booleano.

$$\text{Evento:}(\text{nome}, \text{tipo}, \text{tempo}, \text{ruído}, \{R\}, p) \quad (5).$$

A Figura 3.4 ilustra de forma detalhada, como são iniciadas ações frente às situações de interesse detectadas em processos de negócio.

O MGA busca suportar conceitos e relações para disparar ações proativas com base no modelo de Machado (MACHADO et al, 2017), sendo que um padrão de eventos pode ou não significar uma situação de interesse. Neste trabalho, para que uma situação de interesse seja detectada, à medida que as atividades dos processos são realizadas, eventos são gerados. Diante disso, os eventos são consumidos e processados pelo Spark, onde o MGR (descrito no item seguinte) é requisitado, permitindo analisar se uma atividade não está de acordo com o que foi definido (regras).

Figura 3.4 - Módulo de gerenciamento de ações (MGA).



Fonte: Autoria própria.

Após a detecção das situações em processos de negócio, o plano de ações é ativado, com o intuito de iniciar a ação para minimizar os impactos na organização.

Módulo de Gerenciamento de Regras (MGR): módulo responsável por gerenciar as regras anexadas aos processos. Em um processo de negócio existem diversas políticas e requisitos que restringem e especificam o que deve ser feito. Tais regras são determinadas por especialistas no processo, sendo estabelecidas na modelagem do processo. A arquitetura proposta leva em consideração essas regras, para que a partir delas possam ser detectadas situações com base no processamento de eventos, relacionados a atividades dos processos. No MGR Figura 3.5, as regras (6), são constituídas por nome, o processo que faz parte, a atividade do processo que está relacionada e as relações semânticas {R}, definida em (MACHADO et al., 2017).

Reg: [{Nome, Processo, Atividade, {R}, Servico}] (6).

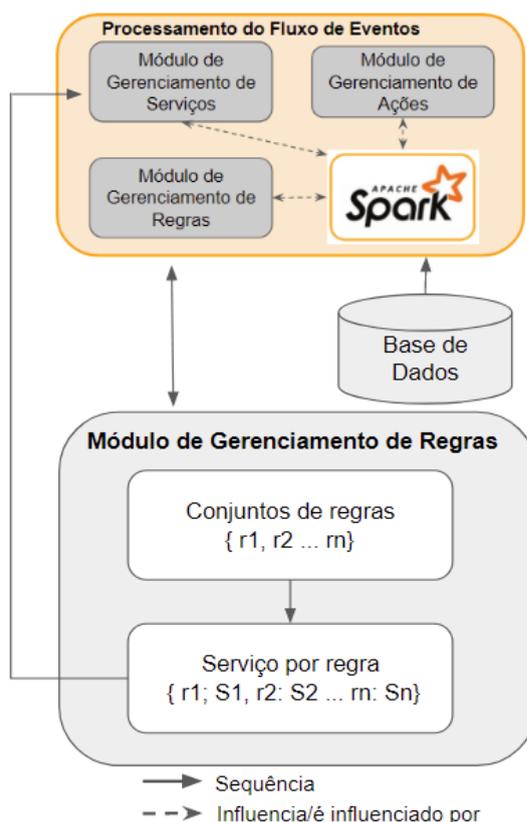
As relações semânticas {R} devem corresponder as atividades do processo de negócio, como por exemplo:

<AtividadeA, temRestricao, RestricaoA>,

<AtividadeB, temRestricao, RestricaoB>.

Sendo às regras inseridas como restrições, interpretadas como parâmetros definidos por especialistas, utilizado para verificar se os eventos estão de acordo ou não com o que foi definido por eles.

Figura 3.5 - Módulo de gerenciamento de regras (MGR).



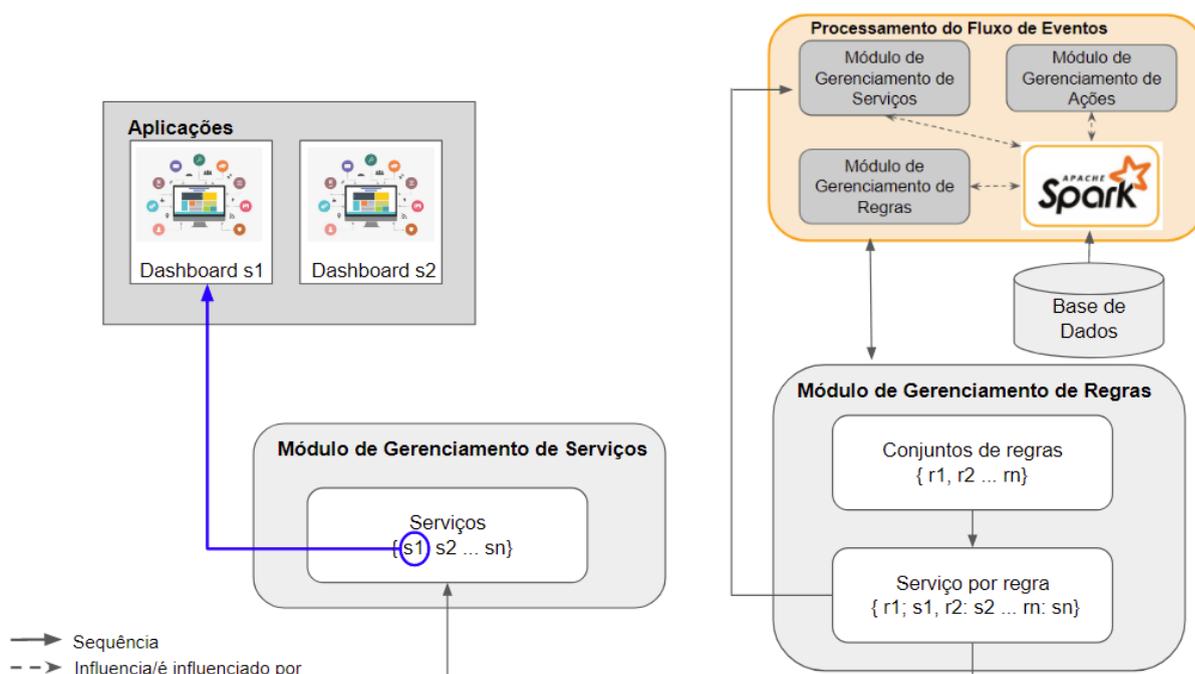
Fonte: Autoria própria.

Cada regra definida pode ou não conter um serviço externo relacionado. Por exemplo, se uma regra é inserida com o intuito de especificar o tempo de duração do início ao fim de uma atividade no processo. Essa regra pode conter um serviço externo (módulo MGS, descrito no item abaixo), que gerencia os dados processados pela arquitetura, contidos nos logs de eventos. Tais serviços externos podem ser, por exemplo, um painel *web* informativo sobre a duração das atividades, um software para análise de fluxo de atividades dos processos, entre outros serviços.

Módulo de Gerenciamento de Serviços (MGS): módulo responsável por gerenciar os serviços externos disponíveis e que interagem de forma direta/ indireta com a

arquitetura, ilustrado na Figura 3.6. O MGS permite que os serviços possam manipular os dados armazenados na base de dados (logs de eventos), referente ao processamento das atividades dos processos de negócio (processadas como eventos). Um serviço que tenha como objetivo a análise de desempenho ou auditoria de fluxo de eventos (verificar se as atividades dos processos estão de acordo com o que foi definido pelos especialistas no negócio), podem utilizar das informações processadas pela arquitetura para alcançar seus objetivos. Visto que todos os eventos processados em tempo real pela arquitetura, são persistidos na base de dados.

Figura 3.6 - Módulo gerenciamento de Serviços (MGS).



Fonte: Autoria própria.

Na Figura 3.6, é possível observar que além da regra definida, um serviço pode estar ou não ligado a ela. Se no MGR uma regra foi definida relacionada a atividade 'Cadastrar usuário' e nessa regra foi estipulado que o tempo estimado é de 10 minutos para realizar o cadastro. Além do tempo estimado para realização da atividade, um serviço externo também foi definido para a mesma regra, caracterizado por s1. O serviço s1, por exemplo, consiste em uma aplicação

dashboard de análise de tempo. Todas as informações disponíveis que o dashboard (serviço s1) utiliza, foram processadas pela arquitetura, estando em log de eventos na base de dados (descrito no próximo item).

De acordo com essas informações, o serviço pode apresentar ao usuário interessado, os dados de outras formas de acordo com o objetivo do serviço. Sendo assim, o serviço pode ter diversos objetivos, manipulando os eventos processadas referente a realização das atividades dos processos.

Persistência: módulo responsável por armazenar todos os eventos processados, relacionados a execução das atividades dos processos de negócio por usuários em softwares de gestão, em uma base de dados. Cada atividade realizada e processada pela arquitetura é inserida na base de dados, caracterizando os logs de eventos (descrito na seção 2.2). Com isso, possibilita ter informações históricas do que aconteceu e tem acontecido.

Em vista disso, é possível ter parâmetros históricos para mensurar diversos aspectos, por exemplo, tempo de realização das atividades, tempo ocioso em que o processo ficou dependendo de uma determinada atividade, auditoria do fluxo das atividades, entre outros aspectos gerenciais.

Na arquitetura foram utilizadas algumas ferramentas que contém características com o potencial para suprir a demanda necessitada, sendo o processamento de tempo real e o processamento de fluxo de eventos. O Kafka e Spark (descritos na seção 2.1.1 e 2.1.2 respectivamente), foram utilizados por serem ferramentas que provêm escalabilidade e confiança no processamento das informações dos processos de negócio. Porém, outras ferramentas que tenham o mesmo objetivo podem ser utilizadas.

Como forma de avaliar o objetivo do trabalho, bem como comparar com trabalhos já existentes na literatura, o próximo capítulo descreve os trabalhos relacionados a este.

4 TRABALHOS RELACIONADOS

Os trabalhos relacionados considerados foram aqueles que abrangem logs de eventos, processamento de fluxo de eventos com ênfase em detecção de situações de interesse em processos de negócio.

Desta busca resultaram cinco (05) trabalhos (Quadro 4.1), seja pela forma de uso das áreas, metodologias ou objetivos. A busca teve como palavras chaves utilizadas, logs de eventos, processamento de eventos em tempo real e contexto, como situação e ação. Esses conceitos foram escolhidos e definidos como base na busca por estar diretamente ligado com o objetivo do trabalho, as ferramentas e conceitos utilizados.

Até o momento não foram encontrados trabalhos que utilizam juntos ambos conceitos abordados nesta dissertação.

Quadro 4.1 - Características dos trabalhos relacionados.

Características/ Trabalho		1	2	3	4	5	TP
Ano		2015	2018	2018	2019	2019	-
Logs de Eventos		√	√	√	√	√	√
Processamento de Fluxo de Eventos		-	-	√	-	-	√
Contexto	Situação	√	√	-	√	√	√
	Ação	√	-	-	-	-	√

Fonte: Autoria Própria.

1: (LAKSHMANAN et al, 2015); **2:** (POLATO et al, 2018); **3:** (BATYUK; VOITYSHYN; VERHUN, 2018); **4:** (MANNHARDT; LANDMARK, 2019); **5:** (TAX; SIDOROVA; VAN DER AALST, 2019); **TP:** Trabalho Proposto.

Os logs de eventos (seção 2.2), estão relacionados a dados registrados em base de dados referente a execução de atividade/tarefa, que possibilitam extrair

informações relevantes sobre o fluxo de trabalho e realização das atividades dos processos, análise de tempo ou custos, por exemplo. Além dos trabalhos que utilizam os logs como base, que estão diretamente relacionados a este trabalho, existe também o processamento de fluxo de eventos. Sendo assim, até o momento da realização desta pesquisa, não foram encontrados trabalhos que relacionassem logs de eventos e processamento de fluxo de eventos com objetivos semelhantes a este trabalho. Procurou-se encontrar trabalhos onde os autores utilizam das mesmas ferramentas descritas e utilizadas nesta dissertação, sendo descritos abaixo.

O trabalho proposto por Lakshmanan (2015), relata uma abordagem com base em modelos probabilísticos de processos, sendo capaz de prever a probabilidade de atividades futuras. Com isso, são geradas informações relevantes para os gerentes de negócios sobre o andamento do processo. Porém, nas informações não constam previsões relacionadas a algo que aconteceu e não esteja de acordo com o que foi definido. Para testar e validar seu trabalho, os autores utilizaram o cenário de simulações de sinistros de seguros de automóveis que lidam com processos de negócios. As previsões fornecidas pela técnica desenvolvida por Lakshmanan, podem gerar alertas para os responsáveis pelo caso sobre a probabilidade de resultados importantes ou indesejados em uma atividade em execução.

Polato (2018), propõe em seu trabalho um modelo para prever os tempos e lidar com comportamentos inesperados, com base nos dados adicionais presente nos logs de eventos. Para isso, os autores utilizam técnicas de aprendizagem de máquina para treinar algoritmos e detectar comportamentos não esperados.

O trabalho dos autores Batyuk, Voityshyn e Verhun (2018), tem o objetivo de compreender os rastros dos processos de negócio, utilizando os logs de eventos. Os autores utilizam arquitetura lambda combinada com algoritmos de mineração de processos online e offline com análise avançada e aprendizado de máquina. Para validar seu trabalho, os autores implementaram uma plataforma que foi integrada a um sistema de gestão para monitorar as atividades realizadas dos processos.

O trabalho proposto por Mannhardt e Landmark (2019), tem como objetivo aplicar técnicas da mineração de processos em logs de eventos do controle de tráfego ferroviário (estudo de caso aplicado). Os autores utilizaram indicadores de desempenho (tempos), derivados dos dados que possibilitam investigar a qualidade das decisões tomadas pelos responsáveis da ferrovia. Para validar o trabalho, os

autores desenvolveram um aplicativo web a fim de explorar o tráfego de trens e analisar as decisões dos responsáveis, para ordem de qual trem tem prioridade nos cruzamentos, visando minimizar os atrasos. Como conclusão, os autores relataram, que alguns casos os responsáveis pelas linhas davam prioridade nos cruzamentos a trens que conseqüentemente atrasariam outros trens, ocasionando uma situação de interesse que não era desejada.

O trabalho proposto por Tax, Sidorova e Van Der Aalst (2019), conceitua técnicas para filtrar atividades caóticas nos logs de eventos. O estudo de caso utilizado pelos autores é constituído por uma coleção de dezessete (17) logs de eventos derivados da execução de processos de negócios. Com base nos resultados dos autores, eles argumentam que atividades caóticas impactam na qualidade dos modelos de processos e no ambiente organizacional.

Com base nos trabalhos conceituados, é possível identificar que todos utilizam algumas das tecnologias propostas nesta dissertação, visando manipular grandes volumes de dados. Além de possibilitar tratar algum fato que ocorreu, como uma situação caótica, similar a situação de interesse (definida neste trabalho), podendo ser infrequente ou frequente dependendo das informações. Apesar dos trabalhos utilizarem os conceitos de forma separada, mesmo assim, ambos consideram a detecção de uma situação de interesse, sendo ela indesejada, caótica ou outra característica similar.

Com base no Quadro 4.1, os trabalhos de Mannhardt e Landmark (2019) e Tax, Sidorova e Van Der Aalst (2019) e Polato (POLATO et al, 2018), tem o foco nos logs de eventos, onde os autores utilizaram conceitos de mineração de logs para extrair informações relevantes sobre o contexto analisado e identificar situações de interesse/caóticas. Mas não realizam o processamento em tempo real dos dados, somente utilizam os logs.

Analisando ainda o Quadro 4.1, o trabalho dos autores Batyuk, Voityshyn e Verhun (2018), descrevem utilizar o processamento em tempo real de fluxo de eventos, visando a detecção de alguma situação de interesse no contexto. Porém em nenhum deles uma ação frente a situação é iniciada, e sim apenas apontam a situação. No trabalho do autor Lakshmanan (2015), o mesmo utiliza dos logs de eventos para detectar uma situação e iniciar ações frente a situação detectada, porém não utiliza do processamento em tempo real, a detecção acontece com base nos logs de evento.

Vale ressaltar, que apesar dos trabalhos terem similaridade com este pelo uso dos mesmos conceitos, nenhum dos trabalhos citados contempla a utilização do processamento de fluxo de eventos e os logs de eventos em conjunto em uma estrutura única, visando detectar situações de interesse em processos de negócio e ações frente às situações. Sendo que o objetivo do presente trabalho é propor uma arquitetura baseada em eventos para a detecção de situações de interesse em processos de negócio, bem como iniciar ações proativas frente a essas situações.

A fim de validar a arquitetura e o objetivo deste trabalho, no próximo capítulo estão descritos dois estudos de caso, com cenários elaborados baseado na experiência de projetos de pesquisa e desenvolvimento, sendo um deles o que motivou este trabalho.

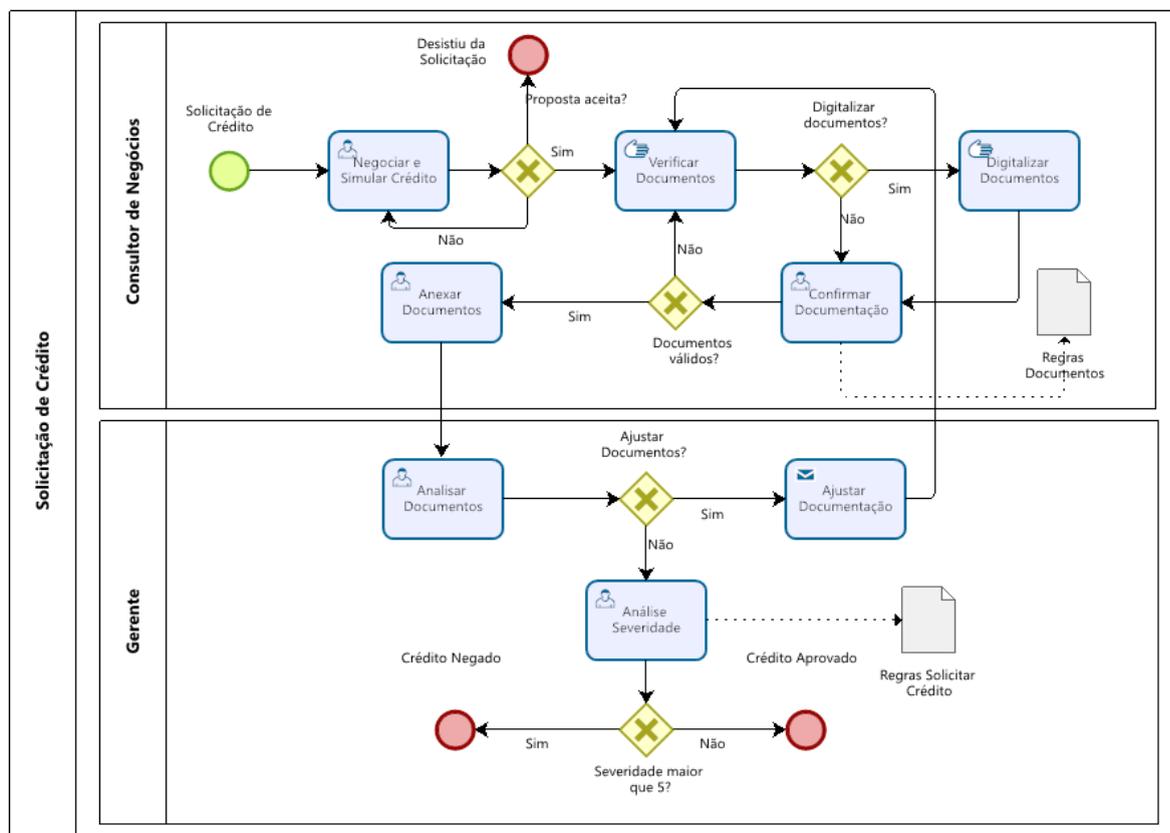
5 APLICAÇÃO DA ARQUITETURA - ESTUDO DE CASO

Para avaliar o trabalho proposto foram desenvolvidos 02 (dois) cenários visando abranger diferentes contextos em que processos de negócios são utilizados. Vale ressaltar que o objetivo deste trabalho está ligado diretamente ao processamento de fluxo de eventos decorrente da realização de atividades presentes em processos de negócio. Com isso, o objetivo deste trabalho é propor uma arquitetura baseada em eventos, para a detecção de situações de interesse em processos de negócio, bem como iniciar ações proativas frente a estas situações. Os cenários estão descritos nas próximas seções 5.1 e 5.2.

5.1 CENÁRIO PARA ORGANIZAÇÃO BANCÁRIA

O cenário a seguir foi desenvolvido baseado na experiência adquirida durante o projeto entre a UFSM e o Sicredi – Sistema de Crédito Cooperativo. Porém, por questões de sigilo da informação sobre as execuções dos processos internos da cooperativa, o cenário foi adaptado não refletindo o contexto real da organização. Considerando a questão de pesquisa deste trabalho: é possível por meio de uma arquitetura baseada em eventos a detecção de situações de interesse em processos de negócio, bem como iniciar ações proativas diante dessas situações? Buscando responder essa questão, exemplificando a utilização da arquitetura proposta, o processo descrito abaixo é relacionado, ficticiamente, a solicitação de crédito ilustrada na Figura 5.1.

Figura 5.1 - Processo de solicitação de crédito.



Powered by
bizagi
Modeler

Fonte: Autoria próprio.

Na Figura 5.1, as atividades 'Análise severidade' e 'Confirmar documentação' contém regras definidas por especialistas da organização, que determinam o que pode ou não ser aceito na atividade que faz parte do processo 'Solicitar crédito'. No Quadro 5.1, são representadas as regras de cada atividade.

Quadro 5.1 – Regras processo solicitar crédito.

(continua)

Regras	Processo	Atividade	Características	
			Tempo estimado	Política de crédito
R1	Solicitar crédito	Analisar severidade	30 minutos	Analisar histórico pessoal física

(continuação)

R2	Solicitar crédito	Analisar Documentos	1 hora	Checar documentos obrigatórios (validade 10 anos pessoa física)
-----------	-------------------	---------------------	--------	---

Fonte: Autoria própria.

Com isso, o cenário leva em consideração a interação de um cliente que solicita um produto e a organização que comercializa esse produto. Neste cenário, será produzida a situação chamada 'insatisfação do cliente' iniciada pelo evento 'crédito negado'. Visando controlar o ambiente de negócio, ações proativas são iniciadas de acordo com a situação detectada.

Considerando 'João' de 26 anos, analista de desenvolvimento e trabalha em uma empresa privada de tecnologia. Certo dia 'João' decidiu comprar um veículo, porém não possuía toda a quantia necessária para a compra, diante deste fato, optou por ir em um banco e solicitar um crédito para efetuar sua compra.

'João', antes da decisão de comprar o veículo, já havia adquirido produtos anteriormente da organização, pois solicitou um crédito para construção civil (linha de crédito casa) para finalizar os acabamentos de seu apartamento, e teve seu crédito aprovado. Com isso, não ocorreu a detecção do evento 'crédito negado', consequentemente não causando a situação 'insatisfação do cliente' relacionado ao produto oferecido pela organização.

Satisfeito, 'João' retorna à organização para então solicitar o crédito para a compra do seu veículo. Com pressa para fechar o negócio, o mesmo solicita o crédito da linha veículos. Esta linha de crédito tramita mais rápido em relação às outras na organização. Entretanto, na solicitação efetuada por 'João' ocorreu erros no processo especificamente, na tarefa 'analisar documentos', onde é feita a checagem de documentos obrigatórios (política interna).

A demora não prevista na tramitação do processo excedeu o tempo definido como regra em R2 (Quadro 5.1), definido em 1 hora, tempo aceitável para finalizar, sendo assim, 'João' optou por cancelar o processo de solicitação. Pois estava com pressa em receber o crédito porque já tinha negócio fechado com o vendedor do veículo. Com isso, decide alterar a linha de crédito, para a linha de crédito pessoal simples, solicitando um novo pedido.

Porém, foi diferente do seu primeiro contato com a organização, onde todos os processos anteriores tinham sido aprovados. Quando 'João' solicitou o crédito relacionado a linha de crédito pessoal simples, teve sua solicitação negada, pelo fato de não contemplar os requisitos definidos pela organização. 'João' continha severidade maior que 05 (cinco). Severidade é um requisito imposto para identificar que a parcela do financiamento é alta para a renda declarada de 'João' (limite aceito é 4 - definido por especialistas). Ocasionalmente para a organização a situação indesejada 'insatisfação do cliente'. Para que a situação indesejada fosse detectada, o evento 'analisar documentos' inicia a situação indesejada 'insatisfação do cliente', porém essa situação só vai se concretizar quando outro evento ocorrer, sendo ele, 'crédito negado'.

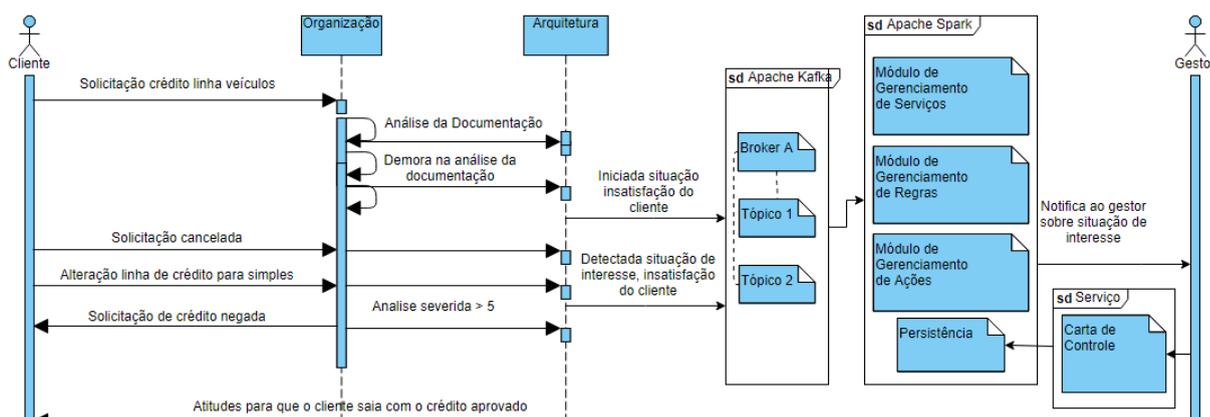
A ocorrência desses eventos tem efeito negativo sobre a organização, pois é o inverso do objetivo estratégico definido e sempre buscado pela mesma, que visa sempre a satisfação do cliente em relação aos seus produtos ou serviços prestados. A ocorrência desse evento gera para a organização uma situação indesejada, definida como 'insatisfação do cliente'. Diante desta situação, a organização definiu algumas medidas para gerenciar todos seus processos. Tendo em vista prevenir que situações como essas não ocorram, e se caso virem a ocorrer, iniciar medidas para que o objetivo definido pela organização seja alcançado, visando a satisfação de seus clientes com o serviço fornecido pela organização. Vale ressaltar que todos os eventos foram criados a partir da realização das atividades, para que 'João' tivesse seu crédito aprovado, sendo processado em tempo real.

Visando minimizar as ocorrências de situações indesejadas nos processos de negócio, uma das medidas foi implantar a arquitetura que provê ferramentas focadas no processamento de eventos, para monitorar e gerenciar o ambiente de negócios. Todas as informações resultantes das interações dos clientes com a organização, são gerenciadas pelas ferramentas que compõem a arquitetura, como; Kafka, Spark e os módulos de ações, regras e o serviço externo, para analisar graficamente os dados. Neste cenário a carta de controle é utilizada como serviço externo, pois têm a capacidade de auxiliar graficamente a organização na análise da execução de seus processos a partir dos logs de eventos.

Referente a solicitação de 'João' ao ser detectada uma situação de interesse 'crédito negado', que não estão de acordo com as regras impostas ao MGR. A aplicação (Spark), imediatamente envia uma notificação por algum canal de

comunicação para o gestor da organização, de acordo com MGA, como mostra a Figura 5.2. A ação definida é iniciada decorrente da situação detectada, com o intuito de minimizar os danos em ambos lados, tanto da organização quanto do cliente.

5.2 - Diagrama de sequência interação, cliente com a organização.



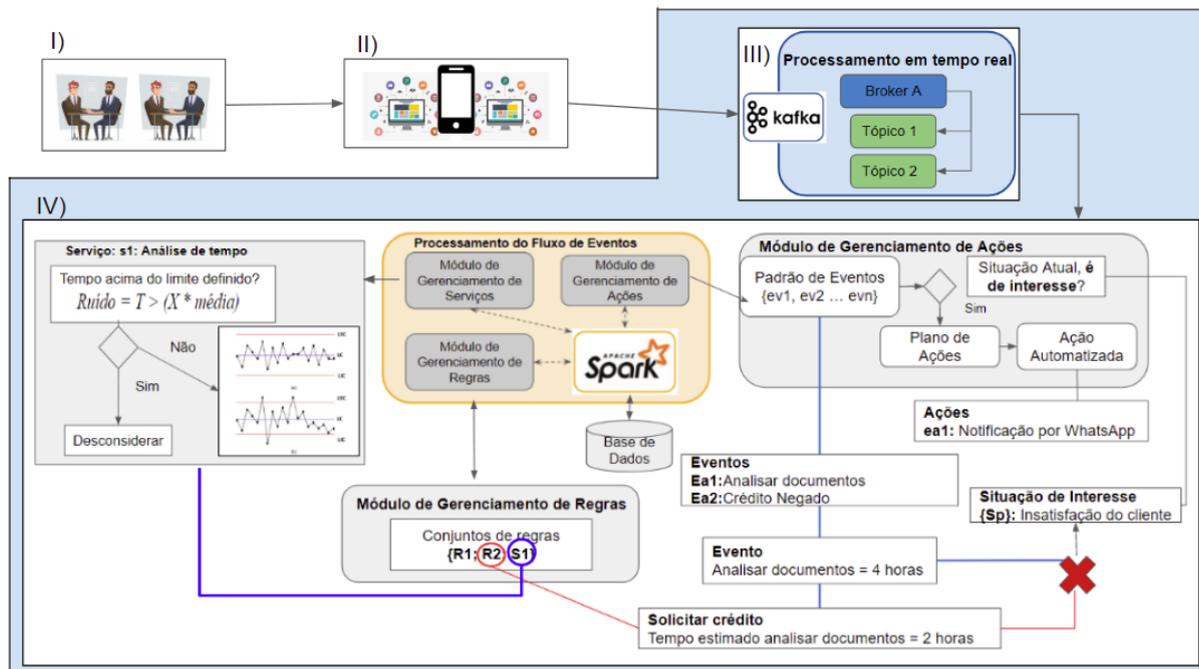
Fonte: Autoria própria.

O cenário descrito acima demonstra que uma situação de interesse no processo de negócio foi detectada. Diante disso, o módulo de gerenciamento de ações (MGA) e a aplicação (Spark), ficam responsáveis por disparar ações (descrito na próxima seção), para alertar o responsável pelos processos de negócio, que algo ocorreu diferente do esperado.

5.1.1 Comportamento Para Situação De Crédito Negado

Aplicações pervasivas, segundo Machado (MACHADO et al., 2017) são pequenas aplicações que manipulam dados de contexto e tomam decisões referentes a situações de interesse, que neste trabalho é vinculado a situações em processos de negócio. Nesta dissertação, uma aplicação pervasiva é levada em consideração, visando processar fluxo de eventos e a partir disso, identificar a situação de interesse em processos de negócio, bem como iniciar ações proativas frente a situações detectadas. A Figura 5.3 ilustra a visão geral do cenário descrito na seção 5.1, desde a criação dos eventos a partir da realização das tarefas, até a ação iniciada de acordo com a situação detectada.

Figura 5.3 - Visão geral organização bancária.



Fonte: Autoria própria.

Na Figura 5.3, é possível observar que em, (I) ilustra-se a negociação entre cliente e gerente de negócio (colaborador da organização), (II) a partir da negociação entre ambos, é gerada diversas informações nos sistemas de gestão utilizadas pela organização como, por exemplo, cadastros de dados pessoais, atualização de dados, simulações de créditos entre outras. À medida que as atividades do processo vão sendo realizadas, em (III) os eventos são gerados e consumidos pelo Apache Kafka com seus brokers e tópicos disponíveis, o que viabiliza processar um fluxo de eventos simultaneamente.

Após o processamento feito pelo Kafka, o fluxo de eventos é repassado para o Spark, que na arquitetura proposta nesta dissertação é utilizado em conjunto com 04 (quatro) módulos, ilustrado na Figura 5.5. Neste cenário a regra R2, contém o serviço, ilustrado na Figura 5.3 (IV), que utiliza a Carta de Controle para mensurar graficamente informações sobre os tempos decorrente das realizações das atividades dos processos. Vale ressaltar que uma regra pode ter ou não um serviço correspondente.

Regra R2 (7) pode ser representado por:

Reg: [{
 Nome: R2,
 Processo: 'Solicitar crédito',
 Atividade: 'Analisar documentos',
 {R},
 Serviço: 'Dashboard Carta de Controle'
 }] (7).

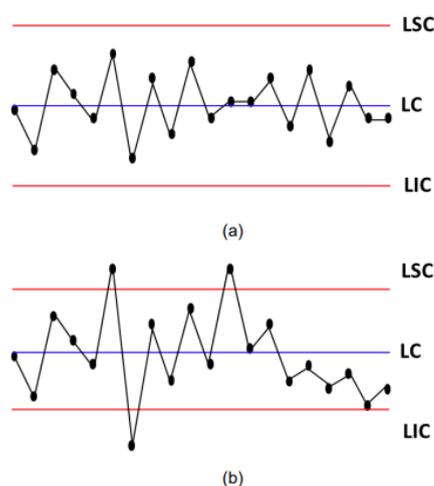
Sendo {R} (descrito na seção 2.3), definido como:

<AnalisarDocumentos, temRestricao, TempoEstimado>

Ou seja, o contexto de interesse é representado pela atividade do processo em questão, restrição que faz parte dessa atividade e o valor definido por especialistas ou políticas da empresa. Com isso, neste cenário o fator tempo em que os processos são iniciados e finalizados são levados em consideração, sendo estabelecidos limites aceitos com base no tempo de duração das atividades.

Esses limites são conhecidos como (i) linha superior (limite superior de controle - LSC) e (ii) linha inferior (limite inferior de controle - LIC), além de (iii) linha central (limite central - LC). Neste trabalho é utilizada a carta de controle para representar os limites, ilustrada na Figura 5.4.

Figura 5.4 - Exemplo de carta de controle. (a) sob controle estatístico (b) fora de controle estatístico.

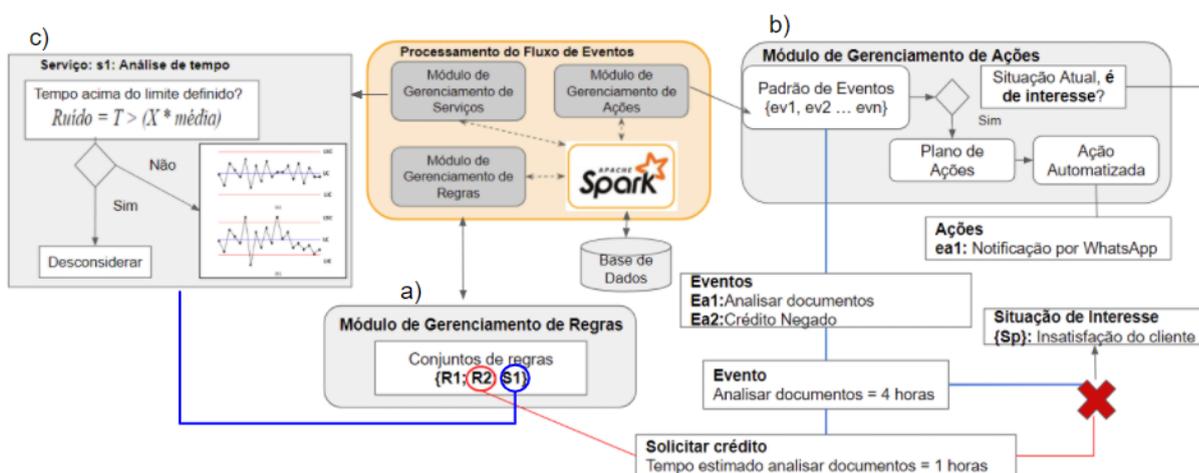


Vale ressaltar que a carta de controle é uma ferramenta utilizada para Controle Estatístico de Processos, além de aplicar o controle de qualidade nas informações analisadas (OLIVEIRA et al., 2013). Neste cenário, a carta de controle foi utilizada para verificar e explorar as informações dos tempos de duração contidos nos logs de eventos, resultado das execuções das atividades dos processos. O tempo aceito foi definido por especialistas. Para que não seja considerado ruído deve ser inferior a 1 hora, para a realização da atividade.

A carta de controle é utilizada para representar graficamente com base em amostras temporais (logs de eventos) os limites inferiores e superiores, bem como a média de tempo que uma atividade pode levar para ser realizada. Vale lembrar que um dado só pode ser analisado se não for considerado ruído, ou seja, tempos que não excedem muito além do que foi definido por um especialista no negócio. A aplicação que provê o serviço de análise e visualização aos gestores, faz uso dos dados armazenados em log, que foram processados pela arquitetura.

Na Figura 5.5, se um processo que está em tramitação detectar que os documentos de João (usuário), estão sendo analisados (evento ea1 foi detectado) de acordo com as regras (5.5.a), representado no Quadro 5.2, a situação 'insatisfação do cliente' é então iniciada pelo sistema.

Figura 5.5 - Processamento de fluxo de eventos.



Fonte: Autoria própria.

Com isso, o sistema escolhe uma ação para ser disparada (5.5.b), sendo ação a1, que envia uma notificação por algum canal de comunicação para acionar o

responsável pelo processo. Com o intuito de alertar ao gestor para que o mesmo tome alguma atitude, referente a demora na análise dos documentos.

Quadro 5.2 - Situação atual: <(insatisfação do cliente), (ea1), (ea2)>

<i>Nome do Evento:</i>	ea1
<i>Descrição:</i>	Analisar documentação
<i>Tipo:</i>	Interno
<i>Ruído:</i>	false
<i>{R}</i>	<Usuario, solicitarCredito, Crédito Pessoal> ^ <RegrasProcesso, verificarDocumentacao, Documentos> ⇒ <Documentos, ModuloGerenciamentoDeRegras, RegraExcedida>
<i>Padrão:</i>	SELECT e FROM PATTERN[every e=Event (name= "ea1")]

Fonte: Autoria própria.

Com o monitoramento contínuo do ambiente de negócio em relação ao fluxo de eventos que estão ocorrendo, após a notificação ter sido enviada, o sistema reconhece que a situação 'insatisfação do cliente' tenha sido encerrada, representada no Quadro 5.3.

Quadro 5.3 - Evento ea2.

<i>Nome do Evento:</i>	ea2
<i>Descrição:</i>	Crédito negado
<i>Tipo:</i>	Interno
<i>Ruído:</i>	false
<i>{R}</i>	<Usuario, solicitarCredito, Crédito Pessoal> ^ <RegrasProcesso, verificarDocumentacao, Documentos> ^ <Colaborador, verificarDocumentacao, Documentos> ⇒<Documentos, ModuloGerenciamentoDeRegras, RegraObedecida>
<i>Padrão:</i>	SELECT e FROM PATTERN[every e=Event (name= "ea1")]

Fonte: Autoria própria.

Como apresentado acima, se o evento ea2 é detectado, João tem seu crédito aprovado, não ocorrendo a situação de ‘insatisfação do cliente’. De acordo com o modelo proposto por Machado (MACHADO et al., 2017) (seção 2.4), as ações correspondentes aos eventos gerados são executadas, como descrito no cenário (seção 5.1), a ação a1 gera o evento e1, ação a2 gera o evento e2 e assim por diante. As ações para a situação de ‘insatisfação do cliente’ são representadas a seguir no Quadro 5.4, sendo declaradas como Ação (nome, {Sp}, {Se}, {E}):

Quadro 5.4 - Ações para situação indesejada.

<i>Nome:</i>	Notificação via whatsapp após ea1;
<i>{Sp}</i>	crédito negado;
<i>Serviços utilizados {Se}</i>	
<i>Se:</i>	notificaWpp;
<i>{I}</i> :	Aplicacao pervasiva, “verificarDocumentação”;
<i>{O}</i> :	Serviço Interno, notificacaoWhatsapp
<i>Eventos resultantes de ação {E}</i>	
<i>Nome:</i>	a1
<i>Tipo:</i>	Interno
<i>Ruído:</i>	false
<i>{R}</i>	⟨notificaoApp_AcaoAutomatizada, temServico, notificacaoService⟩ ∧ notificacaoService, eUmTipoDe, Interacao_Usuario⟩
<i>Padrão:</i>	SELECT e FROM PATTERN[every e=Evento(nome ='a1')-> and Evento(nome ='ea1')]

Fonte: Autoria própria.

Cada ação iniciada gera um evento correspondente. Neste cenário a ação iniciada é ‘Notificação via *whatsapp* após ea1’ e o evento correspondente é a1. Tal ação é disparada, visando alertar ao responsável que foi detectada uma situação de interesse no processo ‘Solicitar crédito’, na atividade ‘Analisar documentos’. A ação iniciada está relacionada a uma funcionalidade disponível, que contém como entrada

(*inputs* - {I}), aplicação pervasiva e como saída (*outputs* - {O}), o serviço de notificação por um canal de comunicação, sendo neste caso, mensagem por whatsapp (MACHADO et al., 2017). A ação iniciada tem o intuito de minimizar as perdas e impactos que a situação detectada possa ocasionar para a organização. A próxima seção descreve outro cenário em um contexto diferente da organização bancária.

5.2 CENÁRIO PARA PLATAFORMA DE DEFESA SANITÁRIA ANIMAL DO RIO GRANDE DO SUL (PDSA-RS)

Além do cenário descrito anteriormente, tendo em vista abranger contextos distintos para simular o uso da arquitetura, foi desenvolvido um segundo cenário baseado na Plataforma de Defesa Sanitária Animal do estado do Rio Grande do Sul (PDSA-RS). O cenário compõe a certificação sanitária de granjas avícolas de reprodução, para controle das doenças Salmonella e Mycoplasma.

A PDSA-RS foi desenvolvida no projeto entre a Universidade Federal de Santa Maria (UFSM) e o Fundo de Desenvolvimento e Defesa Sanitária Animal do Rio Grande do Sul (Fundesa/RS), a Superintendência Federal de Agricultura do MAPA no Rio Grande do Sul (SFA-RS) e a Secretaria Estadual da Agricultura, Pecuária e Desenvolvimento Rural (SEAPDR).

A plataforma tem como objetivo informatizar todas as etapas para a certificação avícola de granjas matrizes para reprodução. A mesma encontra-se em um ambiente real de utilização, com informações reais sobre animais confinados de todo o estado. Para a realização da certificação avícola é necessário o cumprimento de regras estabelecidas pelo ministério da agricultura do estado do Rio Grande do Sul. Por exemplo, para a certificação sanitária das doenças Salmonella E Mycoplasma, as seguintes instruções normativas, [44 de 23/08/2001](#)⁴ e [78 de 03/11/2003](#)⁵, devem ser seguidas. Tais regras (Quadro 5.5) definem exames, quantidades de amostras e o período que devem ocorrer as colheitas de cada granja avícola (coleta de materiais). Os materiais devem ser coletados seguindo as regras, para que sejam válidos e possam ser analisados, visando a emissão do laudo por

⁴<https://www.defesa.agricultura.sp.gov.br/legislacoes/instrucao-normativa-sda-44-de-23-08-2001,734.html>

⁵http://www.adepara.pa.gov.br/sites/default/files/INSTRU%C3%87%C3%83O%20NORMATIVA%20N%C2%BA%2078%2C%20SALMONELLA_0.pdf

um laboratório agropecuário e posterior emissão de certificação pelo Ministério da Agricultura.

Para este cenário 02 (dois) casos são levados em consideração, são eles, (i) caso relacionado a informações de coletas inválidas, caracterizado pela situação ‘amostras inválidas’, iniciada pelo evento ‘preencher formulário de colheita’, (ii) e no outro caso, a situação de interesse é ‘exame incorreto’, iniciada pelo evento ‘digitar resultados’.

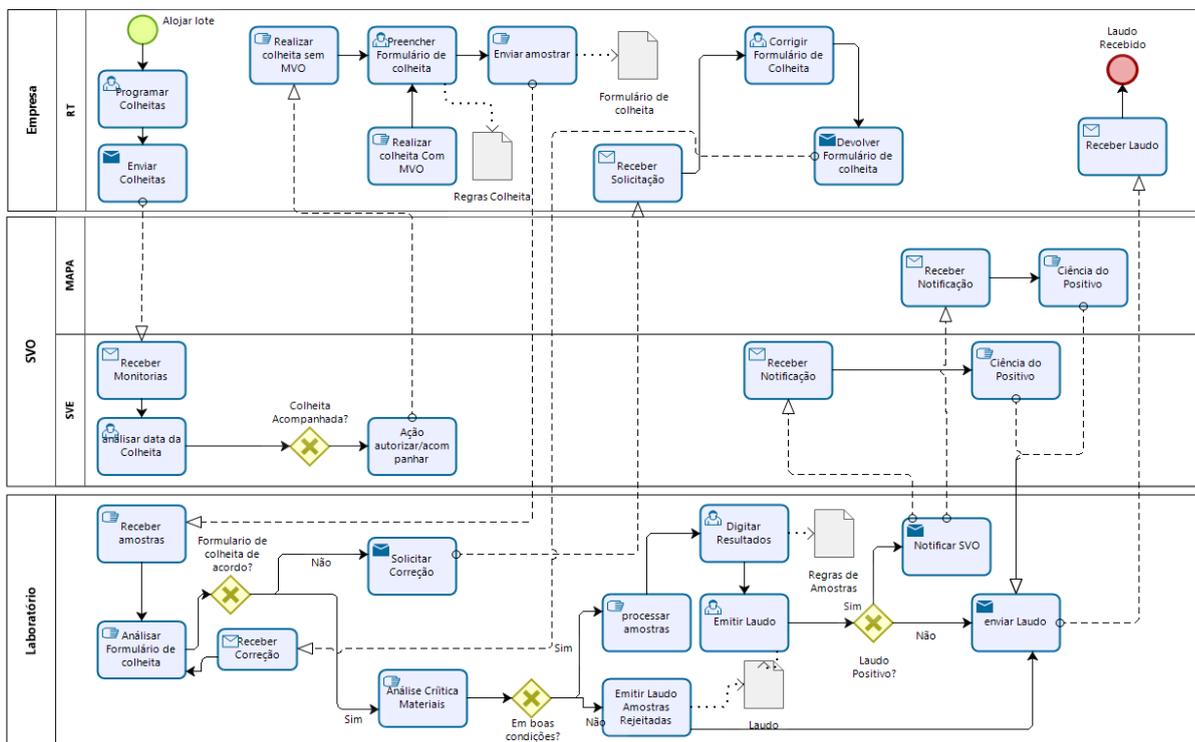
Quadro 5.5 - Regras sanitárias para certificação avícola - Processo Emitir Laudo.

Nome	Atividade	Período	Mycoplasma	Salmonella
			Exame/ Tipo material/ Amostras	Exame/ Tipo material/ Amostras
R1	Preencher formulários de colheita	De 01 a 05 dias		Bacteriológico ou Molecular/ Aves mortas/50 aves/núcleo
R2	Preencher formulários de colheita	De 11 a 23 dias	Sorológico/ Soros/ MG-MM: 300 amostras/núcleo	Bacteriológico ou Molecular/Fezes frescas/ 01 pool de 100 amostras/núcleo
				Sorológico/Soros/ 100 amostras/núcleo
R3	Preencher formulários de colheita	De 24 a 26 semanas	Sorológico/ Soros/ MG-MM: 150 amostras/núcleo	Bacteriológico/ Fezes frescas/ 01 pool de 100 amostras/núcleo
R4	Digitar resultados		Sorológico/ Soros/ MG-MM: 150 amostras/núcleo	Bacteriológico/ Fezes frescas/ 01 pool de 100 amostras/núcleo

Fonte: Autoria própria.

As etapas que compõem a certificação avícola de granjas são organizadas em processos de negócio. Tais processos seguem as regras estabelecidas, ou seja, normativas impostas por órgãos regulamentadores, para a certificação. Neste cenário o processo apresentado é o ‘Emitir Laudo’, ilustrado na Figura 5.6. É um dos principais processos da PDSA-RS, sendo que permite às empresas que contém granjas aviárias comercializar os itens de sua produção dentro ou fora do país.

Figura 5.6 - Processo emitir laudo.



Fonte: (DESCOVI; EBLING, 2021).

As atividades, 'Preencher formulário de colheita' e 'Digitar resultados', conforme a interação do usuário com a plataforma, produz eventos internos captados pela arquitetura. Vale ressaltar, que essas atividades são consideradas as mais críticas de todo o processo, portanto foram escolhidas para serem descritas neste cenário.

Para a detecção da situação de interesse por meio do processamento de fluxo de eventos, as regras são levadas em consideração. À medida que as atividades são realizadas por usuários, eventos são gerados. Tais eventos estão ligados diretamente a um conjunto de regras. Com isso, foi atribuído às atividades 'Preencher formulários de colheita' e 'Digitar resultados' regras a serem seguidas, representadas no Quadro 5.5.

As regras definem o que deve ser levado em consideração ao coletar amostras de materiais para análise das doenças, com o objetivo de emitir o certificado.

Maria é representante técnica (veterinária) da empresa Galide, que contém uma granja aviária, que produz ovos e carnes. Na infraestrutura da granja contém 4 galpões separados em 2 núcleos, sendo esta separação uma forma também de impedir que uma doença se prolifere para as outras granjas. Cada núcleo tem aproximadamente 300 metros de distância de um para outro, com capacidade de 10 mil aves por galpão.

Fazendo uso da PDSA-RS, Maria digitou coletas de material (fezes, ovos, sangue, etc ...) das aves. Para que os materiais colhidos sejam utilizados na análise laboratorial e o certificado seja emitido, as colheitas devem seguir os períodos definidos nas regras, representadas no Quadro 5.5 (R1; R2; R3 e R4).

Maria coletou informações relacionadas à regra R1 no dia 10/01/2021, caracterizando período de 01 a 05 dias de vida das aves, com material do tipo aves mortas e quantidade de 30 aves mortas por núcleo. Com base nessas informações é importante seguir exatamente o que as regras definem, pois caso não sejam seguidas, o certificado não pode ser emitido, o que ocasiona perdas para a empresa e suas granjas. Após a realização da regra R2, Maria como representante técnico da empresa, insere as informações coletadas sobre os animais na PDSA-RS, preenchendo os formulários com os dados relacionados às coletas realizadas na granja da empresa Galide. Com a utilização da arquitetura para o processamento do fluxo de eventos, ao inserir as informações de campo na PDSA-RS, Maria infelizmente se confundiu com as quantidades amostrais dos soros e acabou invertendo as quantidades destinada para Salmonella e Mycoplasma, resultando na detecção de uma situação de interesse.

De acordo com as regras, o evento 'formulário de colheita preenchido', ocasionou a detecção da situação de interesse 'amostras inválidas', pelo fato de as informações não serem iguais às que foram definidas na regra R2. A ocorrência desse evento poderia resultar em prejuízos financeiros, resultando nos piores casos em sacrificar todos os animais da produção, por apresentar informações equivocadas sobre os animais em confinamento.

Porém, com a utilização da arquitetura, ao ser detectada a situação 'amostras inválidas', imediatamente são iniciadas ações automatizadas, visando diminuir os impactos na empresa. É enviada uma notificação à Maria, alertando-a sobre os dados incorretos, dando a possibilidade da mesma fazer as correções necessárias.

Após a correção das informações, as amostras coletadas são enviadas ao laboratório, órgão que analisa detalhadamente cada amostra e os classifica. Após essa análise é emitido o laudo, sendo positivo ou negativo. Se caso foi encontrada alguma doença nos animais confinados na granja, o resultado é positivo, ou negativo, se nenhuma doença foi encontrada, o que possibilita manter os animais em confinamento.

Visto que somente o laboratório é o responsável por emitir o laudo, o mesmo é requisitado por inúmeras empresas avícolas, o que torna o processo mais lento e passível a erros. Com isso, no evento 'Digital Resultado', é da responsabilidade de Carlos, usuário responsável por liberar o laudo. Ao analisar as amostras lançadas por Maria, Carlos fez somente análise dos soros contendo MG e MM para Mycoplasma, (Quadro 5.5) R4, não levando em consideração as amostras para Salmonella. Diante desse fato, resultou na detecção da situação de interesse, 'exame incorreto'.

Entretanto, fazendo uso da arquitetura, uma ação automatizada foi iniciada, ocasionando o alerta para Carlos, visando informá-lo que os exames que o mesmo realizou são incompletos para certificar as granjas como livres para as doenças Mycoplasma e Salmonella.

Em outras ocorrências, sem o uso da arquitetura com a emissão dos alertas como ações para minimizar os impactos (no momento que aconteceu), seria necessário a realização de outros exames para comprovar que não existem doenças nos animais confinados nas granjas. Tais exames resultariam em um custo elevado acima do normal para a empresa, além de comprometer todos os períodos do processo de confinamento e produção, bem como nos piores casos a perda total da produção.

5.1.2 Comportamento Para As Situações De Interesse Em Processos De Negócio - Amostras Inválidas E Exame Incorreto.

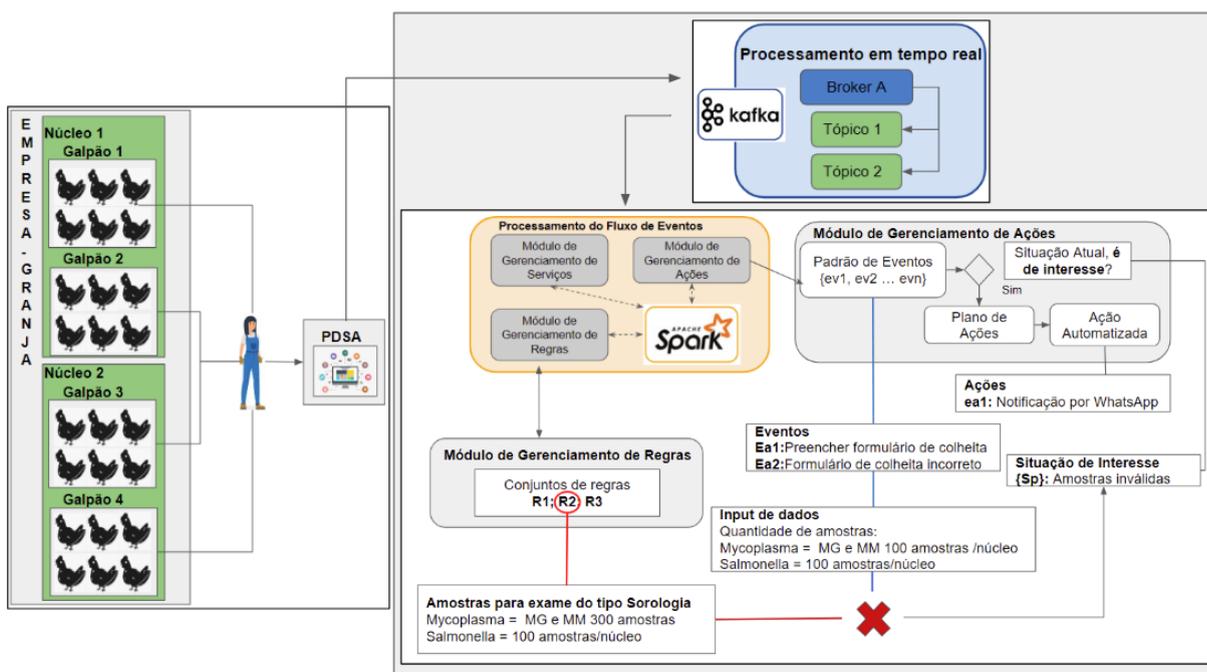
Na arquitetura apresentada neste trabalho, o módulo de gerenciamento de ações (MGA) e o módulo de gerenciamento de regras (MGR), são utilizados neste cenário. Cada atividade no processo equivale a um evento interno no software, definido por Machado (Machado et al, 2017) na seção 2.4. Cada processo tem suas regras que representam restrições, em outros casos servem para regulamentar a

tramitação dos processos nas organizações. A Figura 5.7, ilustra de forma geral a arquitetura inserida em conjunto com a PDSA-RS. Neste cenário os eventos que são considerados ruídos, estão ligados diretamente a demoras inesperadas, sendo que o tempo aceito para a interceptação do evento é 60 segundos (definido pelo especialista no processo). A demora pode estar ligada diretamente a falhas de internet, problemas com servidores e infraestrutura, como falta de energia elétrica, por exemplo.

Visto que o processo ‘Emitir Laudo’, contém as regras representadas no Quadro 5.5, são inseridas em MGR. A regra R2 (8) é representada como:

```
Reg: [{
  Nome: R2,
  Processo: ‘Emitir Laudo’,
  Atividade: ‘Preencher formulário de colheita’,
  {R},
  Serviço: ‘Não se aplica’
}] (8).
```

Figura 5.7 – Visão geral estudo de caso PDSA-RS.



Fonte: Autoria própria.

Com base na realização das atividades de Maria, um fluxo de eventos foi gerado e consumidos pelos tópicos contidos nos brokers disponíveis no Kafka. Após o processamento do fluxo de eventos pelo Kafka, os mesmos são consumidos pelo Spark, onde é analisado se estão de acordo com as regras definidas para a atividade no processo em {R}.

As regras são definidas com base no contexto de interesse definido por Machado (MACHADO et al, 2017). Neste cenário a relação semântica que identificam o contexto de interesse é:

<Salmonella, deveConter, Indicadores>

<Mycoplasma, deveConter, Indicadores>

Sendo que indicadores são caracterizados por exame, tipo de material e quantidades de amostras, informações que definem o que deve ser levado em consideração para cada doença. Com isso é levado em consideração um determinado período no tempo, desde o evento inicial até o final, estando relacionado a janela deslizante proposta por Machado (Machado et al, 2017). Após a detecção da situação de interesse baseada nas regras, o MGA é ativado, para iniciar ações proativas, visando minimizar os impactos que a situação de interesse pode ocasionar para a organização. No processo de 'Emitir Laudo' ao ser tramitado, estando em andamento e for detectado 'preencher formulário de colheita' (evento ev1 foi detectado), a situação de interesse 'amostras inválidas' é iniciada por MGA, como mostra a Figura 5.7, e representada no Quadro 5.6.

Quadro 5.6 - Situação atual: <(Amostras inválidas), (ev1), (ev2)>

<i>Nome do Evento:</i>	<i>ev1</i>
<i>Descrição:</i>	<i>Preencher formulário de colheita</i>
<i>Tipo:</i>	<i>Interno</i>
<i>Ruído:</i>	<i>false</i>
<i>{R}</i>	<Usuario, PreencherFormularioColheita, SolicitarCertificacao> ^ <MGR, verificarAmostras, DadosColheita> ⇒ <Usuario, confirmarPreenchimento, preenchimentoAceito>
<i>Padrão:</i>	<i>SELECT e FROM PATTERN[every e=Event (name= "ev1")</i>

Porém, para a situação de interesse se concretizar, o evento 'formulário de colheita incorreto' (evento ev2), representado no Quadro 5.7, deve ocorrer resultando na detecção da situação de interesse 'amostras inválidas'.

Quadro 5.7 – Evento ev2.

<i>Nome do Evento:</i>	ev2
<i>Descrição:</i>	<i>Formulário de colheita incorreto</i>
<i>Tipo:</i>	<i>Interno</i>
<i>Ruído:</i>	false
<i>{R}</i>	<Usuario, PreencherFormularioColheita, SolicitarCertificacao> ^ <MGR, verificarAmostras, DadosColheita>^ <DadosColheita, verificarAmostraRegras, Regras> ⇒ <MGR, confirmarPreenchimento, amostrasInválidas>
<i>Padrão:</i>	<i>SELECT e FROM PATTERN[every e=Event (name= "ev1")</i>

Fonte: Autoria própria.

Além da situação de interesse 'amostras inválidas' no processo 'Emitir laudo', a situação 'tipo de exame incorreto', também é detectada, visto que por ser processamento em tempo real de fluxo de eventos, conforme as atividades do processo são realizadas, os eventos são criados e processados simultaneamente.

Para a situação de interesse 'exame incorreto' ser detectada, o evento 'digitar resultados' (evento ev3) é analisado em relação às regras (número de amostras inválidas) estando diferente é detectado, como mostra o Quadro 5.8.

Quadro 5.8 - Situação atual: <(Exame incorreto), (ev3), (ev4)>.

(Continua)

<i>Nome do Evento:</i>	ev3
<i>Descrição:</i>	<i>Digitar resultados</i>
<i>Tipo:</i>	<i>Interno</i>
<i>Ruído:</i>	false

(Continuação)

{R}	<Usuario, DigitarResultados, EmitirLaudo> ^ <MGR, verificarAmostrasExames, DadosColheita> ⇒ <Usuario, confirmarResultados, amostrasValidadas>
Padrão:	SELECT e FROM PATTERN[every e=Event (name= "ev3")]

Fonte: Autoria própria.

Porém, para a situação de interesse no processo se concretizar, o evento 'tipo de exame incorreto' (evento ev4) é consumido, ocasionado pelo número de amostras inválidas, definidas na regra R2, representado no Quadro 5.9.

Quadro 5.9 - Evento ev4.

Nome do Evento:	ev4
Descrição:	Tipo de exame incorreto
Tipo:	Interno
Ruído:	false
{R}	<Usuario, DigitarResultados, EmitirLaudo > ^ <MGR, verificarAmostrasExames, DadosColheita>^ <DadosColheita, verificarRegrasAmostras, Regras> ⇒ <MGR, confirmarResultados, amostrasInválidas>
Padrão:	SELECT e FROM PATTERN[every e=Event (name= "ev4")]

Fonte: Autoria própria.

Após a detecção das situações de interesse 'amostras inválidas' e/ou 'exame incorreto' no processo de negócio 'Emitir laudo', o MGA inicia ações automatizadas para reduzir os impactos no ambiente, representada no Quadro 5.10. De acordo com o plano de ação definido, a ação iniciada é o envio de um alerta informativo para o responsável sobre tal situação e os danos que possam vir a ocorrer. Neste cenário foi levado em consideração a mesma ação para ambas situações.

Quadro 5.10 - Ações para as situações de interesse em processos de negócio.

<i>Nome:</i>	<i>Notificação via whatsapp após ea1;</i>
<i>{Sp}</i>	<i>Situação de Interesse; (Amostras inválidas e/ou Exame incorreto)</i>
<i>Serviços utilizados {Se}</i>	
<i>Se:</i>	<i>notificaWpp;</i>
<i>{I}</i> :	<i>Aplicação pervasiva, “Preencher Formulário de Colheita ou Digitar Resultados”</i>
<i>{O}</i> :	<i>Serviço Interno, notificacaoWhatsapp</i>
<i>Eventos resultantes de ação {E}</i>	
<i>Nome:</i>	<i>ea1</i>
<i>Tipo:</i>	<i>Interno</i>
<i>Ruído:</i>	<i>false</i>
<i>{R}</i>	<i>⟨notificacaoApp_AcaoAutomatizada, temServico, notificacaoService⟩ ∧ notificacaoService, eUmTipoDe, Interacao_Usuario⟩</i>
<i>Padrão:</i>	<i>SELECT e FROM PATTERN[every e=Evento(nome ='ea1')->and Evento(nome ='ea1')]</i>

Fonte: Autoria própria.

O módulo de gerenciamento de serviço (MGS), não foi utilizado neste cenário por não ser de interesse dos gestores visualizar ou gerenciar as informações. Pois a PDSA-RS gerencia de forma eficiente as informações sobre os processos tramitados por seus usuários. O próximo capítulo descreve resultados obtidos a partir do desenvolvimento de um protótipo no contexto desta dissertação.

6 RESULTADOS

Um protótipo de uma aplicação pervasiva foi elaborado com base no cenário 'Organização bancária' (seção 5.1), a fim de auxiliar o monitoramento das execuções das atividades dos processos, utilizando dados temporais como parâmetro. Os dados são providos por um software interno utilizado pela organização para gerenciar suas tarefas diárias. O software de gestão armazena todas as tarefas diárias em log de eventos e a aplicação consome essas informações.

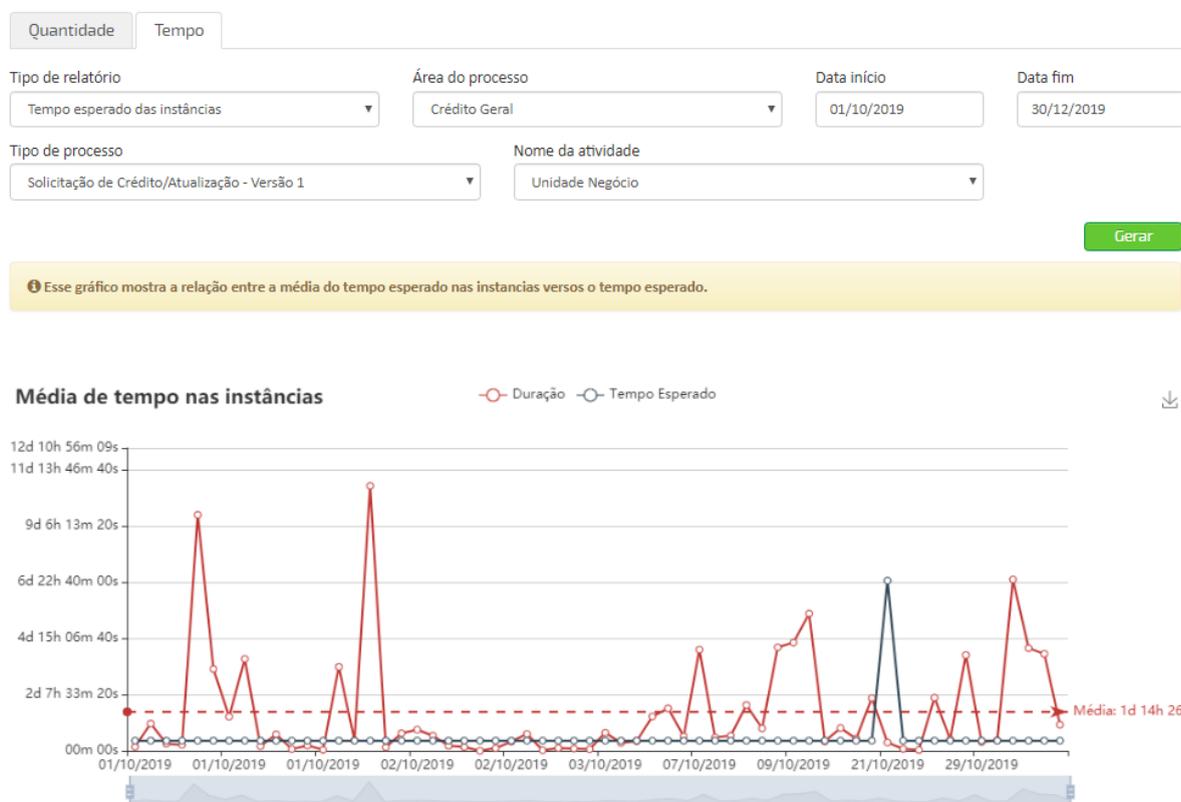
Os tempos esperados que um processo pode levar desde o seu início até ser finalizado, são definidos por especialistas da área de processos (regras). Com base nas regras é possível analisar se o tempo definido está sendo respeitado, caso não esteja, uma situação de interesse pode ser detectada. Além dos tempos das execuções dos processos, outras verificações poderiam ser realizadas, por exemplo, auditoria no fluxo de execução dos processos. Definindo a ordem de execução do processo como regras, seria possível analisar se o processo está sendo executado como foi modelado.

A aplicação é constituída por algumas interfaces, sendo uma delas ilustrada pela Figura 6.1, relacionada à linha do tempo da execução de apenas um processo. Sendo o processo do tipo 'Crédito Geral', com os tempos e suas respectivas datas conforme ocorreu a realização da atividade por um usuário da organização, armazenadas no log de eventos.

Cada ponto no gráfico ilustrado na Figura 6.1, são atividades realizadas no processo, sendo representado na figura o tempo em que a atividade levou para ser realizada. No cenário descrito, foi mencionado sobre a linha de crédito simples e de veículos, na Figura 6.1, a linha de crédito em questão é a área (linha) de Crédito Geral.

A aplicação dispõe de uma interface para realizar buscas e gerar relatórios relacionados aos tempos de execução dos processos, com base nos logs de eventos. Na Figura 6.1 estão ilustrados alguns campos que viabilizam especificar o período de data de início e fim, o tipo de processo, além da área e a atividade em que o mesmo se encontra. São campos que facilitam e auxiliam as buscas por dados mais precisos sobre o processo e suas atividades.

Figura 6.1 - Interface gráfica de monitoramento e controle de processos.

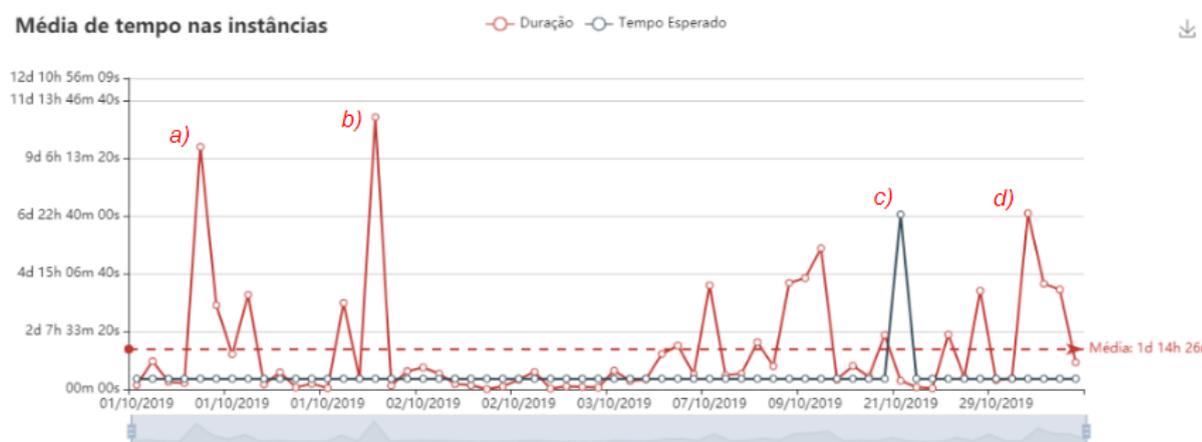


Fonte: Autoria própria.

Quando inserido os campos desejáveis para a busca, um gráfico relacionado a carta de controle (serviço externo, utilizado no cenário) é exibido. No gráfico, especificamente no eixo X os tempos estimados e no Y a data que ocorreu o fato. As linhas e pontos, ilustrado no gráfico contigo na Figura 6.1, caracterizam a média de tempo (linha pontilhada vermelha), os picos de tempo de alguns processos que ocorreram demoras além do esperado (linha contínua vermelha) e o tempo esperado (linha pontilhada azul). A fim de melhorar a visualização, a Figura 6.2 ilustra de forma detalhada as informações relacionadas à busca representada na Figura 6.1.

Na Figura 6.2 existem linhas que representam o tempo que levou para todas as atividades do processo serem executadas, desde quando foi iniciado até ser finalizado, o tempo esperado e tempo médio da execução de todas as atividades do processo no período indicado.

Figura 6.2 - Gráfico de tempos das atividades de um processo.



Fonte: Autoria própria.

A partir do ponto de vista do gestor responsável pelos processos, analisando a Figura 6.2, é possível identificar outros pontos, além do A e B demarcados na Figura, como pontos fora dos tempos esperados e aceitos, definidos nas regras (MGR). Se alguns deles estiverem no intervalo de tempo que equivale a um padrão de atraso, a aplicação pode vir a detectar uma situação de interesse e com isso disparar ações com base no MGA.

É possível observar ainda na Figura 6.2, que a maior parte dos pontos que simbolizam os eventos estão acima da linha de tempo esperado, pelo fato da busca estar sendo realizada dentro do período de um ano. Nesse caso, como a busca foi selecionada, visando apresentar os tempos contidos nos logs de eventos, as análises de casos especiais que possivelmente caracterizariam ruído, não são levadas em consideração.

Para que fossem feitas análises sobre informações que apresentam ruído, o período informado na busca deve ser de apenas um dia. Na Figura 6.2, por ser um protótipo baseado em um aplicação desenvolvida no projeto que motivou esta pesquisa, as interfaces foram modificadas a ponto de não representarem como o processo realmente ocorreu na organização.

Porém, se levado em consideração que na Figura 6.2 foi feita a busca por tempos no período de um dia, os casos A e B, seriam categorizados como ruído,

pelo fato de excederem o tempo limite aceito para realização da atividade, o que equivale a 1 hora descrito na seção 5.1.1. Nesse sentido é possível observar que os tempos apresentados são maiores que todos os tempos da amostra, sendo retirado da análise para detecção de situação de interesse.

Mas em C e D, os tempos não são considerados ruídos, porém é quando a situação insatisfação do cliente é iniciada (descrita na seção 5.1). Com base na Figura 2.3 (ambiente físico) em C e D equivale a um padrão (Δt_d) de interesse detectado. Sendo que ainda existe $t+1$ e $t+2$ (períodos no tempo), com isso sobrando tempo para executar ações proativas (Δt_p) de acordo com a situação detectada.

Com isso vale ressaltar, que com o período de execução do processo inserido, com data de início e fim, é constatado que os tempos ocorridos ultrapassam o tempo esperado e definido em MGR. Sendo assim, é detectada a situação de interesse, iniciando ações proativas por parte da aplicação, podendo ser, como descrito no cenário (seção 5), um alerta por algum canal de comunicação ao responsável pelo processo.

Por motivos de sigilo de informações privadas, os códigos desenvolvidos ao longo do projeto que proporcionaram as Figuras 6.1 e 6.2, não foram descritos. Porém para exemplificar o processamento do fluxo de eventos foi desenvolvida uma aplicação em módulos na linguagem Java, com classes e métodos que simulam a criação de eventos do ambiente de negócio.

O contexto da aplicação é relacionado a um processo de compra de um produto (semelhante a um processo de solicitação de crédito), onde um usuário faz uma solicitação de compra, e a partir disso além de registrar em log, as informações são filtradas por um detector de fraude, que no contexto deste trabalho seria uma situação de interesse.

A Figura 6.3 representa a classe que consiste no produtor de eventos, relacionados a uma solicitação de compra (semelhante a solicitação de crédito). Para isso foi utilizado um identificador de usuário e compra randômico, para simular o ambiente de negócio na medida que os eventos são gerados com a realização das atividades dos processos.

Além da classe que simula a geração de eventos, todas as informações gerenciadas são registradas em logs de eventos, como mostra a Figura 6.4.

Figura 6.3 - Método produtor de eventos.

```

public class NewOrderMain {
    public static void main(String[] args) throws ExecutionException, InterruptedException, IOException {
        try(var orderDispatcher = new KafkaDispatcher<Order>()) {
            try(var emailDispatcher = new KafkaDispatcher<String>()) {
                for(var i=0 ; i< 10; i++) {
                    var userId = UUID.randomUUID().toString();
                    var orderId = UUID.randomUUID().toString();
                    var amount = new BigDecimal(Math.random() * 5000 + 1);

                    var order = new Order(userId,orderId,amount);

                    orderDispatcher.send(topic "ECOMMERCE_NEW_ORDER", userId, order);

                    var email = "Welcome! WE are processing your order! "+ UUID.randomUUID().toString();
                    emailDispatcher.send(topic "ECOMMERCE_SEND_EMAIL", userId, email);
                }
            }
        }
    }
}

```

Fonte: Autoria própria.

Pensando nisso, foi criado um serviço especificamente com essa função para que seja mantido um histórico de acontecimentos sobre a execução dos eventos, que equivalem às atividades do processo.

Figura 6.4 - Classe serviço de log.

```

public class LogService {
    public static void main(String[] args) throws IOException {
        var logService = new LogService();

        try(var service = new KafkaService(
            LogService.class.getSimpleName(),
            Pattern.compile("ECOMMERCE.*"),
            logService::parse,
            String.class,
            Map.of(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class))) {
            service.run();
        }
    }

    private void parse(ConsumerRecord<String, String> record) {
        System.out.println("-----");
        System.out.println("LOG: "+ record.topic());
        System.out.println(record.key());
        System.out.println(record.value());
        System.out.println(record.partition());
        System.out.println(record.offset());
    }

    private static Properties properties() {
        Properties properties = new Properties();
        properties.setProperty(ConsumerConfig.BootstrapServersConfig, "127.0.0.1:9092");
        properties.setProperty(ConsumerConfig.KeyDeserializerClassConfig, StringDeserializer.class.getName());
        properties.setProperty(ConsumerConfig.ValueDeserializerClassConfig, StringDeserializer.class.getName());
        properties.setProperty(ConsumerConfig.GroupIdConfig, LogService.class.getSimpleName());
        return properties;
    }
}

```

Fonte: Autoria própria.

No processo de compra, foi implementado um serviço que filtra eventos que podem ser considerados fraudes (semelhante a situação de insatisfação do cliente), sendo detectadas e descartadas. A classe permite também a visualização de qual partição do Kafka que foi detectada a situação, representada na Figura 6.5. A fraude nesse contexto equivale a uma situação de interesse.

Figura 6.5 - Classe serviço para detectar fraude.

```
public class FraudDetectorService {  
  
    public static void main(String[] args) throws IOException {  
        var fraudService = new FraudDetectorService();  
        try (var service = new KafkaService<>(  
            FraudDetectorService.class.getSimpleName(),  
            topic "ECOMMERCE_NEW_ORDER",  
            fraudService::parse,  
            Order.class,  
            Map.of())) {  
            service.run();  
        }  
    }  
  
    private void parse (ConsumerRecord<String, Order> record) {  
        System.out.println("-----");  
        System.out.println("Processing new order, checking for fraud");  
        System.out.println(record.key());  
        System.out.println(record.value());  
        System.out.println(record.partition());  
        System.out.println(record.offset());  
  
        try {  
            Thread.sleep(5000);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
        System.out.println("Order processed");  
    }  
}
```

Fonte: Autoria própria.

Vale ressaltar que o código fonte do protótipo está hospedado na plataforma de desenvolvimento para gerenciamento de projetos ⁶GitHub, disponível no endereço <https://github.com/MatheusFriedhein/EcommerceStream>. Com base em todas as informações apresentadas e a arquitetura proposta, no próximo capítulo está descrita a conclusão do trabalho.

⁶ <https://github.com/>

7 CONCLUSÃO

As contribuições do trabalho, podem ser descritas como: (i) é possível detectar uma situação de interesse em processos de negócio com base no fluxo de eventos processados em tempo real, decorrente da realização das atividades dos processos, (ii) elaboração de 02 (dois) cenários, que foram concebidos baseado na experiência de projetos de pesquisa e desenvolvimento, que envolvem ambientes reais de negócio, sendo levado em consideração todo o contexto que caracteriza o fluxo de trabalho organizacional em processos, e (iii) estender o modelo proposto por Machado (MACHADO et al., 2017) para o contexto diferente do elaborado pelo autor, possibilitando neste trabalho a detecção de situações de interesse em processos de negócio, bem como iniciar ações proativas frente às situações detectadas e (iv) apresentar às organizações uma arquitetura que seja imparcial a sua utilização, não dependendo de alterações em sistemas legados.

Vale ressaltar que o objetivo geral e específicos foram alcançados nesta dissertação, pelo fato de ter proposto uma arquitetura baseada em eventos, visando a detecção de situações de interesse em processos de negócio, com ações proativas. A arquitetura busca minimizar impactos no ambiente organizacional, agregando valor não somente para a organização, mas também para seus clientes.

Com isso conclui-se que a arquitetura baseada em eventos, composta neste trabalho pelas utilizando das ferramentas Kafka e Spark, e o modelo proativo de Machado (MACHADO et al., 2017), juntamente com os módulos apresentados, tem muito a acrescentar para as organizações que se preocupam em desenvolver e melhorar seus processos. Visando prestar um serviço ou fornecer um produto que supre a necessidade de seus clientes. Vale ressaltar que, para que seja possível realizar a detecção das situações de interesse em processos de negócios, foi necessário adaptar o modelo de Machado adicionando a propriedade de dados (Sistemas Legados) e conceito (Ruído), possibilitando a manipulação dos eventos processados pela arquitetura.

Como trabalho futuro, pretende-se aplicar em um ambiente real para testes mais específicos, visando realizar análises como: capacidade de processamento (mensurar quantidade de informações suportadas no processamento) e

escalabilidade. Realizar estudo apropriado focado em tratar lentidão nos sistemas legados, bem como a integração com softwares de terceiros.

Como contribuições, a primeira versão deste trabalho, foi publicada no evento ICEIS: *International Conference on Enterprise Information Systems, Online Streaming*, Maio 2020, com o título “Towards a Conceptual Model for Undesired Situation Detection through Process Mining” (Qualis B2).

REFERÊNCIAS BIBLIOGRÁFICAS

- ABPMP BPM CBOK. **Guia para o gerenciamento de processos de negócio - corpo comum de conhecimento**. Brasil, MA: Association of Business Process Management Professionals. v.3, 2013.
- BATYUK, Anatoliy; VOITYSHYN, Volodymyr; VERHUN, Volodymyr. Software architecture design of the real-time processes monitoring platform. In: **2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)**. IEEE, 2018. p. 98-101.
- CORONATO, Antonio; DE PIETRO, Giuseppe; PARAGLIOLA, Giovanni. A situation-aware system for the detection of motion disorders of patients with autism spectrum disorders. **Expert Systems with Applications**, v. 41, n. 17, p. 7868-7877, 2014.
- CUGOLA, Gianpaolo; MARGARA, Alessandro. Processing flows of information: From data stream to complex event processing. **ACM Computing Surveys (CSUR)**, v. 44, n. 3, p. 1-62, 2012.
- CHEN, Peter Pin-Shan. The entity-relationship model—toward a unified view of data. **ACM Transactions on Database Systems (TODS)**, v. 1, n. 1, p. 9-36, 1976.
- CHINTAPALLI, Sanket et al. Benchmarking streaming computation engines: Storm, flink and spark streaming. In: **2016 IEEE international parallel and distributed processing symposium workshops (IPDPSW)**. IEEE, 2016. p. 1789-1792.
- D'SILVA, Godson Michael et al. Real-time processing of iot events with historic data using apache kafka and apache spark with dashing framework. In: **2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)**. IEEE, 2017. p. 1804-1809.
- DEY, A.; ABOWD, G. The context toolkit: Aiding the development of context-enabled applications. **Proceedings of the SIGCHI conference on Human factors in computing systems, Pittsburgh, Pennsylvania, US**, p. 434-441, 1999.
- DESCOVI, Glênio; EBLING, Denilson. **Towards a Blockchain Architecture for Animal Sanitary Control**. 2021. No prelo.
- ETZION, Opher; NIBLETT, Peter. **Event processing in action**. Manning, 2011.
- FORKAN, Abdur Rahim Mohammad et al. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. **Pattern Recognition**, v. 48, n. 3, p. 628-641, 2015.
- FRISCHBIER, Sebastian et al. C.: Emergence as Competitive Advantage-Engineering Tomorrow's Enterprise Software Systems. In: **In: ICEIS 2012 (2012) A. Buchmann et al.** 2012.

GEDIK, Buğra et al. Elastic scaling for data stream processing. **IEEE Transactions on Parallel and Distributed Systems**, v. 25, n. 6, p. 1447-1463, 2013.

GIUFFRIDA, Cristiano; KUIJSTEN, Anton; TANENBAUM, Andrew S. Safe and automatic live update for operating systems. **ACM Sigplan Notices**, v. 48, n. 4, p. 279-292, 2013.

GUNTHER, Christian W.; VERBEEK, H. M. W. **Xes-standard definition**. 2014.

GLAVAN, Ljubica. Understanding process performance measurement systems. **Business Systems Research Journal**, v. 2, n. 2, p. 25-38, 2011.

HIRAMAN, Bhole Rahul et al. A study of Apache Kafka in big data stream processing. In: **2018 International Conference on Information, Communication, Engineering and Technology (ICICET)**. IEEE, 2018. p. 1-3.

ICHINOSE, Ayae et al. A study of a video analysis framework using Kafka and spark streaming. In: **2017 IEEE International Conference on Big Data (Big Data)**. IEEE, 2017. p. 2396-2401.

KREPS, JAY, NEHA NARKHEDE, AND JUN RAO. "Kafka: A distributed messaging system for log processing." **Proceedings of the NetDB**. Vol. 11. 2011.

LAKSHMANAN, Geetika T. et al. A markov prediction model for data-driven semi-structured business processes. **Knowledge and Information Systems**, v. 42, n. 1, p. 97-126, 2015.

LUCKHAM, D. The power of events: An introduction to complex event processing in distributed enterprise systems. additions-wesley. **Reading**, 2001.

MACHADO, A. Sensibilidade à Situação Em Ambientes De Vivência Assistida: Uma Abordagem Reativa, Proativa E Extensível, 2015.

MACHADO, Alencar et al. Reactive, proactive, and extensible situation-awareness in ambient assisted living. **Expert Systems with Applications**, v. 76, p. 21-35, 2017.

MANNHARDT, Felix; LANDMARK, Andreas D. Mining railway traffic control logs. **Transportation research procedia**, v. 37, p. 227-234, 2019.

MCCARTHY, J. Actions and Other Events in Situation Calculus. In. **Internacional Conference On Principles And Reasoning**. Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann Publishers, Toulouse, [s.n.], 2002. p. 615 - 628.

MHAND, Mouna Amrou; BOULMAKOUL, Azedine; BADIR, Hassan. Scalable and distributed architecture based on Apache Spark Streaming and PROM6 for processing RoRo terminals logs. In: **Proceedings of the New Challenges in Data Sciences: Acts of the Second Conference of the Moroccan Classification Society**. 2019. p. 1-4.

OLIVEIRA, Camila Cardoso de et al. Manual para elaboração de cartas de controle para monitoramento de processos de medição quantitativos em laboratórios de ensaio. **Instituto Adolfo Lutz. São Paulo**, p. 18, 2013.

POLATO, Mirko et al. Time and activity sequence prediction of business process instances. **Computing**, v. 100, n. 9, p. 1005-1031, 2018.

PRESSMAN, Roger S. **Software engineering: a practitioner's approach**. Palgrave macmillan, 2005.

SAYLAM, Rabia; SAHINGOZ, Ozgur Koray. Process mining in business process management: Concepts and challenges. In: **2013 International Conference on Electronics, Computer and Computation (ICECCO)**. IEEE, 2013. p. 131-134.

SANDHA, Sandeep Singh; KACHUEE, Mohammad; DARABI, Sajad. Complex event processing of health data in real-time to predict heart failure risk and stress. **arXiv preprint arXiv:1707.04364**, 2017.

SHORO, Abdul Ghaffar; SOOMRO, Tariq Rahim. Big data analysis: Apache spark perspective. **Global Journal of Computer Science and Technology**, 2015.

TAX, Niek; SIDOROVA, Natalia; VAN DER AALST, Wil MP. Discovering more precise process models from event logs by filtering out chaotic activities. **Journal of Intelligent Information Systems**, v. 52, n. 1, p. 107-139, 2019.

VAN DER AALST, Wil et al. Process mining manifesto. In: **International Conference on Business Process Management**. Springer, Berlin, Heidelberg, 2011. p. 169-194.

VAN DER AALST, Wil. Process mining: Overview and opportunities. **ACM Transactions on Management Information Systems (TMIS)**, v. 3, n. 2, p. 1-17, 2012.

VERBEEK, H. M. W. et al. Xes, xesame, and prom 6. In: **International Conference on Advanced Information Systems Engineering**. Springer, Berlin, Heidelberg, 2010. p. 60-75.